Retracting Graphs to Cycles

Samuel Haney
Duke University
shaney@cs.duke.edu

Mehraneh Liaee
Northeastern University
mehraneh@ccs.neu.edu

Bruce M. Maggs
Duke University & Akamai Technologies
bmm@cs.duke.edu

Debmalya Panigrahi
Duke University
debmalya@cs.duke.edu

Rajmohan Rajaraman
Northeastern University
rraj@ccs.neu.edu

Ravi Sundaram
Northeastern University
koods@ccs.neu.edu

Abstract

We initiate the algorithmic study of retracting a graph into a cycle in the graph, which seeks a mapping of the graph vertices to the cycle vertices so as to minimize the maximum stretch of any edge, subject to the constraint that the restriction of the mapping to the cycle is the identity map. This problem has its roots in the rich theory of retraction of topological spaces, and has strong ties to well-studied metric embedding problems such as minimum bandwidth and 0-extension. Our first result is an $O(\min\{k, \sqrt{n}\})$-approximation for retracting any graph on n nodes to a cycle with k nodes. We also show a surprising connection to Sperner's Lemma that rules out the possibility of improving this result using certain natural convex relaxations of the problem. Nevertheless, if the problem is restricted to planar graphs, we show that we can overcome these integrality gaps by giving an optimal combinatorial algorithm, which is the technical centerpiece of the paper. Building on our planar graph algorithm, we also obtain a constant-factor approximation algorithm for retraction of points in the Euclidean plane to a uniform cycle.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Graph algorithms, Graph embedding, Planar graphs, Approximation algorithms.

Digital Object Identifier 10.4230/LIPIcs.ICALP.2019.65

Category Regular Paper Track A

Related Version A full version of the paper is available at `https://arxiv.org/abs/1904.11946`

Funding All the authors were partially supported by NSF grant CCF 1535972. Additional funding information appears below.
Debmalya Panigrahi: NSF grant CCF 1527084, an NSF CAREER Award CCF 1750140, and the Joint Indo-US Networked Center for Algorithms under Uncertainty.
Ravi Sundaram: NSF grant CNS 1718286.

Acknowledgements We would like to thank Seffi Naor for helpful discussions on the problems considered in this paper.

46th International Colloquium on Automata, Languages, and Programming (ICALP 2019).
Editors: Christel Baier, Ioannis Chatzigiannakis, Paola Flocchini, and Stefano Leonardi;
Article No. 65; pp. 65:1–65:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Originally introduced in 1930 by K. Borsuk in his PhD thesis [5], *retraction* is a fundamental concept in topology describing continuous mappings of a topological space into a subspace that leaves the position of all points in the subspace fixed. Over the years, this has developed into a rich theory with deep connections to fundamental results in topology such as Brouwer's Fixed Point Theorem [22]. Inspired by this success, graph theorists have extensively studied a discrete version of the problem in graphs, where a *retraction* is a mapping from the vertices of a graph to a given subgraph that produces the identity map when restricted to the subgraph (i.e., it leaves the subgraph fixed). For a rich history of retraction in graph theory, we refer the reader to [21]. Define the *stretch* of a retraction to be the maximum distance between the images of the endpoints of any edge, as measured in the subgraph. We use stretch-k retraction to mean a retraction whose stretch is k; in particular, a stretch-1 retraction is a mapping where every edge of the graph is mapped to either an edge of the subgraph, or both its ends are mapped to the same vertex of the subgraph[1].

In this paper, we study the algorithmic problem of finding a *minimum stretch retraction* in a graph. This problem belongs to the rich area of metric embeddings, but somewhat surprisingly, has not received much attention in spite of the deep but non-constructive results in the graph theory literature. The graph retraction problem has a close resemblance to the well-studied 0-extension problem [6, 24, 25] (and its generalizations such as metric labeling [27, 8]), which is also an embedding of a graph G to a metric over a subset of terminals H with the constraint that each vertex in H maps to itself. The two problems differ in their objective: whereas 0-extension seeks to minimize the *average* stretch of edges, graph retraction minimizes the *maximum* stretch. The different objectives lead to significant technical differences. For instance, a well-studied linear program called the earthmover LP has a nearly logarithmic integrality gap for 0-extension. In contrast, we show that a corresponding earthmover LP for graph retraction has integrality gap $\Omega(\sqrt{n})$. A well-studied problem in the metric embedding literature that considers the maximum stretch objective is the *minimum bandwidth* problem, where one seeks an isomorphic embedding of a graph into a line (or cycle) that minimizes maximum stretch. In contrast, in graph retraction, we allow homomorphic maps[2] but additionally require a subset of vertices (called the *anchors*) to be mapped to themselves.

From an applications standpoint, our original motivation for studying minimum-stretch graph retraction comes from a distributed systems scenario where the aim is to map processes comprising a distributed computation to a network of servers where some processes are constrained to be mapped onto specific servers. The objective is to minimize the maximum communication latency between two communicating processes in the embedding. Such anchored embedding problems can be shown to be equivalent to graph retraction for general subgraphs, and arise in several other domains including VLSI layout, multi-processor placement, graph drawing, and visualization [20, 19, 31].

[1] In the literature, a stretch-1 retraction is often simply referred to as a retraction or a retract [21]. Also, in many studies, a (stretch-1) retraction requires that the two end-points of an edge in the graph are mapped to two end-points of an edge in the subgraph. These studies differentiate between the case where the subgraph being retracted to is reflexive (has self-loops) or irreflexive (no self-loops). In this sense, our notion of graph retraction corresponds to their notion of retraction to a reflexive subgraph.

[2] A *homomorphic* map is one where an image can have multiple pre-images, while an *isomorphic* map requires that every image has at most one pre-image.

1.1 Problem definition, techniques, and results

We begin with a formal definition of the minimum stretch retraction problem.

▶ **Definition 1.** *Given an unweighted* guest *graph $G = (V, E)$ and a* host *subgraph $H = (A, E')$ of G, a mapping $f : V \to A$ is a retraction of G to H if $f(v) = v$ for all $v \in A$. For a given retraction f of G to H, define the* stretch *of an edge $e = (u, v) \in E(G)$ to be $d_H(f(u), f(v))$, where d_H is the distance metric induced by H, and define the* stretch *of f to be the maximum stretch over all edges of graph G. The goal of the* minimum-stretch graph retraction *problem is to find a retraction of G to H with minimum stretch. We refer to the vertices of A as* anchors.

The graph retraction problem is easy if the subgraph H is acyclic (see, e.g., [29]); therefore, the first non-trivial problem is to retract a graph into a cycle. Indeed, this problem is NP-hard even when H is just a 4-cycle [13]. Given this intractability result, a natural goal is to obtain an algorithm for retracting graphs to cycles that *approximately* minimizes the stretch of the retraction. This problem is the focus of our work. While there has been considerable interest in identifying conditions under which retracting to a cycle with stretch 1 is tractable [17, 21, 37], there has been no work (to the best of our knowledge) on deriving approximations to the minimum stretch.[3]

We consider the following lower bound for the problem: if anchors u and v are distance ℓ in H, and there exists a path of p vertices in G between u and v, then every retraction has stretch at least ℓ/p. This lower bound turns out to be tight when H is acyclic, which is the reason retraction to acyclic graphs is an easy problem. However, this lower bound is no longer tight when H is a cycle. For example, consider a grid graph where H is the border of the grid. The lower bound given above says that any retraction has stretch at least $\Omega(1)$. However, using the well-known Sperner's lemma, we show that the optimal retraction has stretch at least $\Omega(\sqrt{n})$.

Using just the simple distance based lower bound, we show that the gap on the grid is in fact the worst possible by giving a $O(\min\{k, \sqrt{n}\})$-approximation for the problem, where k is the number of vertices of H. Our algorithm works by first mapping vertices of the graph into a grid, then projecting vertices outward to the border from the largest *hole* in the grid, which is the largest region containing no vertices.

▶ **Theorem 2.** *There is a deterministic, polynomial-time algorithm that computes a retraction of a graph to a cycle with stretch at most $\min\{k/2, O(\sqrt{n})\}$ times the optimal stretch, where n and k are respectively the number of vertices in the graph and the cycle.*

Our results for retracting a general graph to a cycle appear in Section 2. We also give evidence that the gap induced by Sperner's lemma on a grid graph is fundamental, showing an $\Omega(\min\{k, \sqrt{n}\})$ integrality gap for natural linear and semi-definite programming relaxations of the problem. To overcome this gap, we focus on the special case of planar graphs, of which the grid is an example. Retraction in planar graphs has been considered in the past, most notably in a beautiful paper of Quilliot [30] that uses homotopy techniques to characterize stretch-1 retractions of a planar graph to a cycle. Quillot's proof, however, does not yield an efficient algorithm. In Section 3, we provide an exact algorithm for retraction in planar graphs by developing the gap induced by Sperner's lemma on a grid into a general lower bound on the optimal stretch for planar graphs.

[3] One direct implication of the NP-hardness proof is that approximating the maximum stretch to a multiplicative factor better than 2 is also NP-hard.

▶ **Theorem 3.** *There is a deterministic, polynomial-time algorithm that computes a retraction of a planar graph to a cycle with optimal stretch.*

Unfortunately, our techniques rely heavily on the planarity of the graph, and do not appear to generalize to arbitrary graphs. While we leave the question of obtaining a better approximation for general graphs open, we provide a more sophisticated linear programming formulation that captures the Sperner lower bound on general graphs as a possible route to attack the problem.

We also study natural special cases and generalizations of the problem, all of which are presented in the full version of our paper [18]. First, we consider a geometric setting, where a set of points in the Euclidean plane has to be retracted to a uniform cycle of anchors. By a uniform cycle of anchors we mean a set of anchors which are distributed uniformly on a circle in the plane. We obtain a constant approximation algorithm for this problem, by building on our planar graph algorithm. We next consider retraction of a graph of bounded treewidth to an *arbitrary subgraph*, and obtain a polynomial-time exact algorithm. Finally, we apply the lower bound argument of [24] for 0-extension to show that a general variant of the problem that seeks a retraction of an arbitrary weighted graph G to a metric over a subset of the vertices of G is hard to approximate to within a factor of $\Omega(\log^{1/4-\epsilon} n)$ for any $\epsilon > 0$.

1.2 Related work

List homomorphisms and constraint satisfaction. The graph retraction problem is a special case of the *list homomorphism* problem introduced by Feder and Hell [13], who established conditions under which the problem is NP-complete. Given graphs G, H, and $L(v) \subset V(H)$ for each $v \in V(G)$, a list homomorphism of G to H with respect to L is a homomorphism $f : G \to H$ with $f(v) \in L(v)$ for each $v \in V(G)$.

Several special cases of graph retraction and variants of list homomorphism have been subsequently studied (e.g., [12, 21, 36, 37]). These studies have established and exploited the rich connections between list homomorphism and Constraint Satisfaction Problems (CSPs). Though approximation algorithms for CSPs and related problems such as Label Cover have been extensively studied, the objective pursued there is that of maximizing the number of constraints that are satisfied. For our graph retraction problem, this would correspond to maximizing the number of edges that have stretch below a certain threshold. Our notion of approximation in graph retraction, however, is the least factor by which the stretch constraints need to be relaxed so that all edges are satisfied.

0-extension, minimum bandwidth, and low-distortion embeddings. From an approximation algorithms standpoint, the graph retraction problem is closely related to the 0-extension and minimum bandwidth problems [14, 4, 15, 35, 9, 32]. In the 0-extension problem, one seeks to minimize the average stretch, which can be solved to an $O(\log k/\log\log k)$ approximation using a natural LP relaxation [6, 11]. In contrast, we give polynomial integrality gaps for the graph retraction problem. In the minimum bandwidth problem, the objective is to find an embedding to a line that minimizes maximum stretch, but the constraint is that the map must be isomorphic rather than that the anchor vertices must be fixed. In a seminal result [14], Feige designed the first polylogarithmic-approximation using a novel concept of volume-respecting embeddings. A slightly improved approximation was achieved in [10] by combining Feige's approach with another bandwidth algorithm based on semidefinite-programming [4]. Interestingly, the minimum bandwidth problem is NP-hard even for (guest) trees, while graph retraction to (host) trees is solvable in polynomial time. Conversely, the bandwidth problem is solvable in time $O(n^b)$ for bandwidth b graphs [16], while graph retraction to a cycle is NP-complete even when the host cycle has only four vertices. Nevertheless, it is conceivable that volume-respecting embeddings, in combination with random projection, could lead to effective approximation algorithms for graph retraction to a cycle in a manner similar to what was achieved for VLSI layout on the plane [35].

Also related are the well-studied variants of linear and circular arrangements, but their objective functions are average stretch, as opposed to maximum stretch. Finally, another related area is that of *low-distortion embeddings* (e.g., [23]), where recent work has considered embedding one specific n-point metric to another n-point metric [26, 28, 2] similar to the graph retraction problem. But low-distortion embeddings typically require *non-contracting* isomorphic maps, which distinguishes them significantly from the graph retraction problem.

A related recent work studies low-distortion *contractions* of graphs [3]. Specifically, the goal is to determine a maximum number of edge contractions of a given graph G such that for every pair of vertices, the distance between corresponding vertices in the contracted graph is at least a given affine function of the distance in G. Several upper bounds and hardness of approximations are presented in [3] for many special cases and problem variants. While graph retraction and contraction problems share the notion of mapping to a subgraph, the problems are considerably different; for instance, in the graph retraction problem the subgraph H is part of the input, and the objective is to minimize the maximum stretch.

2 Retracting an arbitrary graph to a cycle

In this section, we study the problem of retracting an arbitrary graph to a cycle over a subset of vertices of the graph. Let G denote the guest graph over a set V of n vertices, with shortest path distance function d_G. Let H denote the host cycle with shortest path distance function d_H over a subset $A \subseteq V$ of k anchors.

Arguably, the simplest lower bound on the optimal stretch is the *distance-based* bound $\ell(G, H) = \max_{u,v \in A} d_H(u, v)/d_G(u, v)$, since every retraction places a path of length $d_G(u, v)$ in G on a path of length at least $d_H(u, v)$ in H.

We now present our algorithm (Algorithm 1), which achieves a stretch of $\min\{k/2, \ell(G, H)\sqrt{n}\}$. Here, we give a high level overview of the algorithm. The first step of algorithm is to embed the input graph G into a grid of size $k/4 \times k/4$ subject to some constraints. The second step is to find the largest empty sub-grid D such that no point is mapped inside of D and center of D is within a desirable distance from center of grid M. And final step is to project the points in grid M to its boundary with respect to center of sub-grid D.

We now show how to implement the first step of Algorithm 1. Our goal is to embed each vertex $u \in G$ to some point $g(u)$ in a $k/4 \times k/4$ grid such that for every u, v, we have the following inequality, where $d_\infty(a, b)$ denotes the L_∞ distance between a and b. (That is, for two points (x_1, y_1) and (x_2, y_2), $d_\infty((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$.)

$$d_\infty(g(u), g(v)) \leq \ell(G, H) d_G(u, v) \tag{1}$$

Additionally, we require that H is embedded to the boundary of the grid, such that adjacent anchors lie on adjacent grid points.

▶ **Lemma 4.** *For every G, we can find an embedding g satisfying inequality 1.*

Proof. We incrementally construct the embedding g. Initially, we place the anchors on the boundary of the grid so that the boundary is isometric to d_H. (This can be done since H is a cycle.) Since $d_\infty(g(u), g(v)) \leq d_H(u, v)$ and $d_H(u, v) \leq \ell(G, H) d_G(u, v)$, inequality 1 holds for all anchors u and v in H.

Algorithm 1 Algorithm for retracting an arbitrary graph to a cycle

---

Input: Graph G, host cycle H

Output: Embedding function f

Embedding in a grid: Determine embedding g from G into a $k/4 \times k/4$ grid M such that H is embedded one-to-one to the boundary of M and for every $u, v \in V$, $d_\infty(g(u), g(v)) \le \ell(G, H) d_G(u, v)$.

Find largest hole: Find the largest square sub-grid D of M such that (a) its center c is at L_∞ distance at most $k/16$ from the center of M and (b) there is no vertex u in G for which $g(u)$ is in the interior of D.

Projection embedding: For all v in G:

1. $R(v) \leftarrow$ ray originating from the center of D and passing through $g(v)$.
2. $f(v) \leftarrow$ the anchor on the boundary of grid M nearest in the clockwise direction to the intersection of $R(v)$ with the boundary of M.

return f

We next inductively embed the remaining vertices of G. Suppose we need to embed vertex v_i, and vertices $U = v_1, \ldots, v_{i-1}$ have already been embedded. Assume inductively that the embedding of the vertices of U satisfies inequality 1 for the vertices in U.

Let $B_\infty(g(u), r)$ denote the L_∞ ball around $g(u)$ with radius r (note that these balls are axis-aligned squares). Let x be any point in $\bigcap_{u \in U} B_\infty(g(u), \ell(G, H) d_G(u, v_i))$. If we set $g(v_i) = x$, then inequality 1 holds for all points in $U \cup \{v_i\}$. We now show that this intersection is nonempty (it is straightforward to find an element in the intersection). The set of axis aligned squares has Helly number[4] 2; therefore it is enough to show that for every $u, u' \in U$, $B_\infty(g(u), \ell(G, H) d_G(u, v_i))$ and $B_\infty(g(u'), \ell(G, H) d_G(u', v_i))$ intersect. Otherwise,

$$d_\infty(g(u), g(u')) > \ell(G, H)(d_G(u, v_i) + d_G(u', v_i)) \ge \ell(G, H) d_G(u, u').$$

This contradicts our induction hypothesis that the set of vertices in U satisfies inequality 1, and completes the proof of the lemma. ◀

In the following lemma, we analyze the projection embedding step of the algorithm.

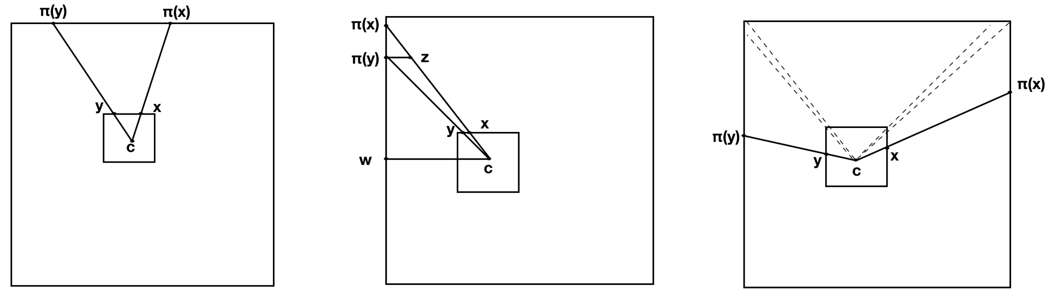
▶ **Lemma 5.** *Suppose r is the side length of the largest empty square D inside M. Then for any vertices u and v in G, $d_H(f(u), f(v))$ is at most $1 + (10\sqrt{2}k/r) d_\infty(g(u), g(v))$.*

Proof. For any point x, let $\pi(x)$ denote the intersection of the boundary of M and the ray from the center c of D passing through x. Note that for any vertex v in G, $f(v)$ is the anchor in H nearest in clockwise direction to $\pi(g(v))$. We show that for any $x, y \in M$, the distance between $\pi(x)$ and $\pi(y)$ along the boundary of M is at most $(10\sqrt{2}k/r) d_\infty(x, y)$.

We first argue that it is sufficient to establish the preceding claim for points on the boundary of D, at the loss of a factor of $\sqrt{2}$. Let x and y be two arbitrary points in M but not in the interior of D. Let x' (resp., y') denote the intersection of $R(x)$ (resp., $R(y)$) and the boundary of D. From elementary geometry, it follows that $d(x', y') \le d(x, y)$, where d is the Euclidean distance; since $d_\infty(x, y) \ge d(x, y)/\sqrt{2}$ and $d_\infty(x', y') \le d(x', y')$, we obtain $d_\infty(x', y') \le \sqrt{2} d_\infty(x, y)$. Since $\pi(x) = \pi(x')$ and $\pi(y) = \pi(y')$, establishing the above statement for x' and y' implies the same for x and y, up to a factor of $\sqrt{2}$.

---

[4] A family of sets has Helly number h if any minimal subfamily with an empty intersection has h or fewer sets in it.



(a) Points x and y are on the same side of square D, and points $\pi(x)$ and $\pi(y)$ are on one side of boundary of M parallel to segment $\overline{xy}$.

(b) Points x and y are on the same side of square D, and points $\pi(x)$ and $\pi(y)$ are on one side of boundary of M orthogonal to segment $\overline{xy}$.

(c) General case where Points x and y (resp. points $\pi(x)$ and $\pi(y)$) are anywhere on the boundary of D (resp. on the boundary of M)

Figure 1 Embedding of points inside the grid M to its boundary using an empty square D. Referred to in the proof of Lemma 5.

Consider points x and y on the boundary of D. We consider three cases. In the first two cases, x and y are on the same side of D. In the first case (Figure 1a), $\pi(x)$ and $\pi(y)$ are on the same side of the boundary of M and segment $\overline{\pi(x)\pi(y)}$ is parallel to segment $\overline{xy}$; then, by similarity of triangle formed by c, x, and y and the one formed by c, $\pi(x)$ and $\pi(y)$, we obtain that the distance between $\pi(x)$ and $\pi(y)$ is at most $3kd_\infty(x,y)/(16r)$. In the second case (Figure 1b), $\pi(x)$ and $\pi(y)$ are on same side of the boundary of M, and segment $\overline{\pi(x)\pi(y)}$ is orthogonal to segment $\overline{xy}$. In this case, w.l.o.g. assume that $\pi(y)$ is closer to center c than $\pi(x)$ with respect to d_∞ distance. Let point z be a point on segment $\overline{c\pi(x)}$ such that segments $\overline{xy}$ and $\overline{\pi(y)z}$ are parallel. From center c extend a line parallel to segment $\overline{xy}$ until it hits the side of M on which $\pi(x)$ and $\pi(y)$ are. Let w be the intersection. Using elementary geometry and similarity argument, we have the following:

$$\frac{|\overline{\pi(x)\pi(y)}|}{|\overline{z\pi(y)}|} = \frac{|\overline{\pi(x)w}|}{|\overline{cw}|} \leq \frac{k/4}{k/16} = 4 \quad \text{and} \quad \frac{\overline{z\pi(y)}}{\overline{xy}} = \frac{\overline{\pi(y)w}}{r} \leq \frac{k}{4r}$$

We thus obtain $\frac{|\overline{\pi(x)\pi(y)}|}{|\overline{xy}|} \leq k/r$. For the third case (Figure 1c), we observe that $d_\infty(x,y)$ is at least half the shortest path between x and y that lies within the boundary of D. This latter shortest path consists of at most five segments, each residing completely on one side of the boundary of D. We apply the argument of the first and second case to each of these segments to obtain that the distance between $\pi(x)$ and $\pi(y)$ is at most $10kd_\infty(x,y)/r$.

To complete the proof, we note that distance between anchor nearest (clockwise) to $\pi(x)$ and anchor nearest (clockwise) to $\pi(y)$ is at most one plus the distance between $\pi(x)$ and $\pi(y)$. Therefore, the $d_H(f(u), f(v))$ is at most $1 + 10\sqrt{2}kd_\infty(g(u), g(v))/r$. ◀

▶ **Theorem 6.** *Algorithm 1 computes a retraction of G to the cycle H with stretch at most the minimum of $k/2$ and $O(\sqrt{n})$ times the optimal stretch.*

Proof. By Lemma 4, the embedding g satisfies inequality 1 for every u and v in G. By a straightforward averaging argument, there exists a square of side length $k/(8\sqrt{n})$ whose center is at L_∞ distance at most $k/16$ from the center of M and which does not contain $g(u)$ for any u in V. By Lemma 5, the projection embedding ensures that for any u and v in V,

$d_H(f(u), f(v))$ is at most $1 + O(\sqrt{n})\ell(G, H)d_G(u, v)$. Since the distance in H cannot exceed $k/2$, the claim of the theorem follows. ◀

The Sperner bottleneck. Unfortunately, we cannot improve on the approximation ratio in Theorem 6 using only the distance-based lower bound. Consider the following instance: the guest graph G is the $\sqrt{n} \times \sqrt{n}$ grid, and the host H is the cycle of G formed by the $4\sqrt{n}$ vertices on the outer boundary of G. It is easy to see that the distance-based lower bound has a value of 2 on this instance. On the other hand, using Sperner's Lemma from topology, we show that a stretch of $o(\sqrt{n})$ is ruled out:

▶ **Lemma 7.** *The optimal stretch achievable for an n-vertex grid is at least $2\sqrt{n}/3$.*

Proof. Suppose we triangulate the grid by adding northwest-to-southeast diagonals in each cell of the grid. Consider the following coloring of the boundary H with 3 colors. Divide H into three segments, each consisting of a contiguous sequence of at least $\lfloor 4\sqrt{n}/3 \rfloor$ vertices; all vertices in the first, second, and third segment are colored red, green, and blue, respectively. Let f be any retraction from G to H. Let c_f denote the following coloring for $G \setminus H$: the color of u is the color of $f(u)$. By Sperner's Lemma [34], there exists a tri-chromatic triangle. This implies that there are two vertices within distance at most two in G that are at least $4\sqrt{n}/3$ apart in the retraction f, resulting in a stretch of at least $2\sqrt{n}/3$. ◀

Note that $k = \Theta(\sqrt{n})$ in this instance, so the above lemma also rules out an $o(k)$ approximation using the distance-based lower bound. A natural approach to improving the approximation factor is to use an LP or SDP relaxation for the problem. Indeed, the so-called *earthmover LP* used for the closely related 0-extension problem [24, 7] can be easily adapted to our minimum stretch retraction problem. Similarly, SDP relaxations previously used for minimum bandwidth and related problems [4, 33] can also be adapted to our problem. However, these convex relaxations also have an integrality gap of $\Omega(\sqrt{n})$ for precisely the same reason: they capture the distance-based lower bound but not the one from Sperner's lemma on the grid (see the full version of the paper [18] for a detailed discussion of these LP/SDP relaxations and integrality gaps).

In spite of these gaps, we show that the grid is not a particularly challenging instance of the problem. In fact, in the next section, we give an exact algorithm for retraction in planar graphs, of which the grid is an example. Retraction of planar graphs to cycles has been considered in the past, and non-constructive characterizations of stretch-1 embeddings were known [30]. Our constructive result, while using planarity extensively, suggests that there might be a general technique for addressing the Sperner bottleneck described above. Indeed, we give a candidate LP relaxation (in the full version of the paper [18]) that captures the Sperner bound on the grid. Rounding this LP to obtain a better approximation ratio, or showing an integrality gap for it, is an interesting open question.

3 Retracting a planar graph to a cycle

The main result of this section is the following theorem.

▶ **Theorem 8.** *Let G be a planar graph and H a cycle of G. Then there is a polynomial time algorithm that finds a retraction from G to H with optimal stretch.*

We begin by presenting some useful definitions and elementary claims in Section 3.1. We then present an overview of our algorithm in Section 3.2. Finally, we present the algorithm and its analysis in Section 3.3, leading to the proof of Theorem 8.

3.1 Preliminaries

We begin with a simple lemma that reduces the problem of finding a minimum-stretch retraction to the problem of finding a stretch-1 retraction, in polynomial time. Formally, suppose we have an algorithm $\mathcal{A}$ that, given graphs G and H either finds a stretch-1 retraction from G to H, or proves that no such retraction exists. Then, we can use this algorithm to find the minimum stretch embedding of G into H, using Lemma 9 below, whose straightforward proof is deferred to the full paper [18]. Let G_k be the graph where we replace each edge $e \in G, e \notin H$ with a path of k edges. Clearly, G_k can be computed in polynomial time.

▶ **Lemma 9.** *G can be retracted to H with stretch k if and only if G_k can be retracted in H with stretch-1.*

The following lemma, proved in [18], implies that degree-1 vertices can be eliminated.

▶ **Lemma 10.** *Without loss of generality, we can assume G is 2-vertex connected.*

Lemmas 9 and 10 apply to general graphs. In the rest of this subsection, we focus our attention on planar graphs. We note that all the transformations in Lemmas 9 and 10 preserve planarity of the graph. In 2-connected planar graph, every face of a plane embedding is bordered by a simple cycle. Finally, we can assume that there is a planar embedding of G with H bordering the outer face. If this is not the case, $G \setminus H$ contains at least two connected components, which can each be retracted independently.

Next, we give some definitions related to planar graphs. We call G *triangulated* if it is maximally planar, i.e., adding any edge results in a graph that is not planar. Equivalently, G is triangulated if every face of a plane embedding (including the outer face) of G has 3 edges. We will make use of the Jordan curve theorem, which says that any closed loop partitions the plane into an inner and outer region (see e.g. [1]). In particular, this implies that any curve crossing from the inner to the outer region intersects the loop. For some cycle C in G and a plane embedding of G, we denote the subset of $\mathbb{R}^2$ surrounded by C as R_C (including the intersection with C itself). We say that $R \subset \mathbb{R}^2$ is *inside* cycle C of G for a plane embedding if $R \subseteq R_C$. If R is inside C, we also say that C *surrounds* R. In a slight abuse of notation, we say C surrounds subgraph G' of G for some fixed plane embedding, if C surrounds the subset of $\mathbb{R}^2$ on which G' is drawn in the plane embedding.

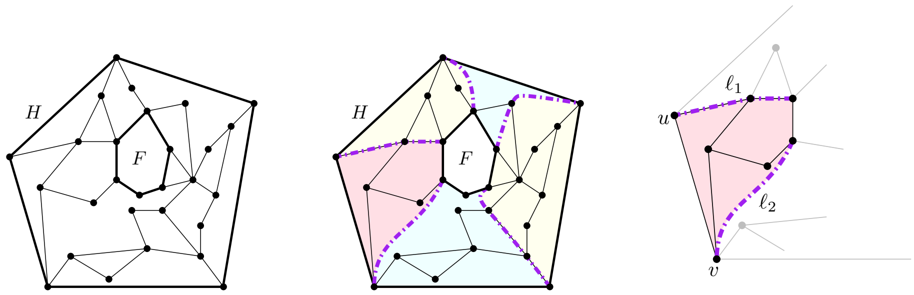
3.2 Overview of our algorithm

Consider some plane embedding of graph G such that H is the subgraph of G bordering G's outer face. We give a polynomial-time algorithm that finds a stretch-1 retraction from G to H or proves that none exists. Using Lemma 9, this immediately yields an algorithm that finds a minimum stretch retraction from G to H.

Fix a planar embedding of G, let H be defined as above, and let F be a bounded face of G. A key component of our algorithm is to find a suitable set of curves connecting F to H. Our aim is to find a set of $k = |V(H)|$ curves in $\mathbb{R}^2$ such that the following hold.

- Each curve begins at a distinct vertex of F and ends at a distinct vertex of H.
- The curves do not intersect each other.
- A curve that intersects an edge of G either contains the edge, or intersects the edge only at its vertices.
- Each curve lies totally in $R_H \setminus F$.

We call curves with these properties *valid* with respect to F. We argue that the curves partition $R_H \setminus F$ (up to their boundaries being duplicated) into a set of regions. Each of



(a) A graph G. The outer cycle H and the face F are shown in bold.

(b) Non-intersecting curves partition the region contained in H but not in F.

(c) Vertices on ℓ_1 are mapped to u, and vertices on ℓ_2 are mapped to v. All other vertices in the region are mapped arbitrarily to u or v.

Figure 2 Using non-intersecting curves to find an embedding from face F to H.

these regions is defined by the subset of $\mathbb{R}^2$ surrounded by the closed loop made up of two of the aforementioned curves, a single edge of H, and a path on the boundary of F.

Given a face F and a set of curves valid with respect to F, we can give a stretch-1 retraction from G to H. In essence, the curves partition the graph into regions such that all vertices in a particular region map to one of two end-points of a particular edge of H. See Figure 2 for an illustration.

Of course, it is not obvious that a valid set of curves exists for a given face, and, if it does, how to compute it. We show that if the graph has a stretch-1 retraction, then there is some face F with k valid curves, and that we can efficiently compute them. Our algorithm (Algorithm 2) iterates over all faces in the graph, in each case finding the maximum number of valid curves it can with respect to that face. The number of valid curves we can find is the length of the shortest cycle surrounding F. If the shortest cycle C surrounding F has length ℓ, then it is impossible to find more than ℓ valid curves with respect to F: By the Jordan curve theorem, each curve must intersect C, and by the definition, valid curves do not intersect each other and can intersect C only at its vertices. Our construction of the valid curves shows that this is tight (i.e. we can always find ℓ curves). We show that if a stretch-1 retraction exists, then there is some face for which $\ell = k$. Algorithm 2 gives an outline of the algorithm.

Algorithm 2 Outline for finding a stretch-1 retraction, or proving that none exists.

```
1: for inner face F in G do
2:     Compute maximum number of valid curves between F and H p_1, ..., p_ℓ
3:     if ℓ = k then
4:         Compute stretch-1 retraction from G to H using p_1, ..., p_k
5:     end if
6: end for
7: If no retraction was computed, report no stretch-1 retraction exists
```

3.3 Algorithm and analysis

This section gives the details of various components of Algorithm 2, and provides a proof of correctness. The following is an outline of the rest of the section:

1. Lemma 12 shows how to compute a stretch-1 retraction using the k valid curves in line 4 of Algorithm 2.
2. Next, Lemma 13 shows that if a stretch-1 retraction exists, there must be some face F in the graph such that the smallest cycle surrounding F has length k.
3. Finally, Lemma 15 gives a construction of largest set of valid curves for a given face F from line 2, and shows that the number of curves computed equals the length of the smallest cycle surrounding F.

We begin by showing in Lemma 11 a somewhat obvious fact: A set of valid curves partition $R_H \setminus F$, and each region of the partition contains a single edge of H. We then show in Lemma 12 that this partition can be used to produce a stretch-1 embedding. See Figure 2 for pictorial presentation of these two lemmas.

▶ **Lemma 11.** *Let $\{p_1, \cdots, p_k\}$ be a set of curves that are valid with respect to F. Let Z denote the set of faces of $H \cup F \cup \bigcup_i p_i$ excluding the outer face and F. Then, each face $f \in Z$ is bordered by exactly 1 edge of H, and every vertex of $G \setminus \bigcup_i p_i$ is in a unique face of Z.*

Proof. Consider the faces of $H \cup F \cup \bigcup_i p_i$. H and F still define faces since the paths p_i fall in $R_H \setminus F$. Let (u, v) be an edge of H, and consider $X = p_i \cup (u, v) \cup p_j \cup p_F(i, j)$ where p_i is the path containing u, p_j is the path containing v, and $p_F(i, j)$ is the path on the boundary of F between the vertices where i and j meet F such that F is not contained in X. If p_i and p_j are both degenerate (i.e., each is empty), then $(u, v) = p_F(i, j)$. Otherwise X is a simple cycle. We claim that X defines a face. In particular, we show that the path $p_F(i, j)$ contains no other vertex of path p_z for all $z \neq i, j$. Suppose it does and let w be that vertex. Let w' be the vertex adjacent to w on p_z. Then $w' \in R_H \setminus F$, and so $w' \in X$. The other end of path p_z, call it vertex y, is in H, but $y \neq u, v$. By the Jordan curve theorem, $p_z \setminus w$ must cross X. Since the graph is planar, $p_z \setminus w$ must contain a vertex of F, H, p_i, or p_j. Any of these outcomes leads to a contradiction. ◀

▶ **Lemma 12.** *Given a non-outer face F and a set $\{p_1, p_2, \ldots, p_k\}$ of curves that are valid with respect to F, we can construct a stretch-1 retraction from G to H in polynomial time.*

Proof. Let Z be as defined in Lemma 11. For each vertex w on p_i, map w to the unique vertex $v \in H \cap p_i$. Otherwise, map w to u or v, where (u, v) is the unique edge of H contained in the same face of Z as w. Fix a face f of Z. Let (u, v) be the unique edge of H contained in f. Any edge (x, y) contained in f also has $x, y \in f$, and so x and y are each mapped to either u or v. Thus, this retraction to H has stretch 1. ◀

As mentioned earlier, we will show that our construction produces ℓ valid curves for face F, where ℓ is the minimum length cycle surrounding F. So we must show that if a stretch-1 retraction exists, there is some F such that every cycle surrounding F has length at least k.

▶ **Lemma 13.** *Fix a plane embedding of G where H defines the outer face of the embedding and suppose there is a stretch-1 retraction G to H. Then there exists a non-outer face F such that every cycle surrounding F has length at least k.*

Proof. We prove a related claim that implies the statement in the lemma. Fix some stretch-1 retraction of G to H. Then there exists a non-outer face F such that for every cycle C in the set of cycles surrounding F, and for each vertex $v \in H$, there is some vertex of C mapped to v. This implies that each of these cycles has length at least k, since the statement says that vertices of C are mapped to k vertices of H.

The claim is very similar to Sperner's lemma, and the proof is similar as well. Let $\phi : V(G) \to V(H)$ denote the retraction. We associate a score with each cycle C of the graph: Order the vertices of the cycle in clockwise order, denoted $v_1, v_2, \ldots, v_j, v_{j+1} = v_1$. Consider the sequence $\phi(v_1), \ldots, \phi(v_j), \phi(v_{j+1})$. Let the score of C be 0 to start. For each pair $\phi(v_i), \phi(v_{i+1})$, we have: either $\phi(v_i) = \phi(v_{i+1})$, or $\phi(v_i)$ and $\phi(v_{i+1})$ are adjacent in H. If $\phi(v_{i+1})$ is clockwise of $\phi(v_i)$ (i.e. if they are in the same order as on C), add 1 to the score of C. If they are in counterclockwise order, subtract 1. If they are the same vertex, the score remains the same. If $\phi(v_1), \ldots, \phi(v_j)$ does not contain every vertex on the outer cycle, the score of C must be 0, since each edge along the path $\phi(v_1), \ldots, \phi(v_{j+1})$ is traversed exactly the same number of times in each direction. On the other hand, a cycle with a non-zero score must have a score that is divisible by k.

Next, we claim that the score of cycle C is the same as the sum of the scores of the cycles defining the faces contained in C. To see this, consider the total contribution to the scores of these cycles from any fixed edge. If the edge is not in cycle C, it is a member of exactly 2 faces contained in C, and contributes either 0 to both faces, or -1 to one and 1 to the other. Edges in C are each a member of just one face surrounded by C. Therefore, the score of cycle C is the same as the sum of scores of its surrounded faces. Since the score of cycle H is k, there must be some face f that has non-zero score.

Finally, we show that there is some face with nonzero score such that every cycle surrounding the face also has nonzero score. Suppose this is not the case. Then, every face with a non-zero score is surrounded by a cycle with score 0, which implies that the sum of all scores of faces with non-zero scores is 0. This is a contradiction, since it implies that the sum of scores of all internal faces in the graph is 0. ◀

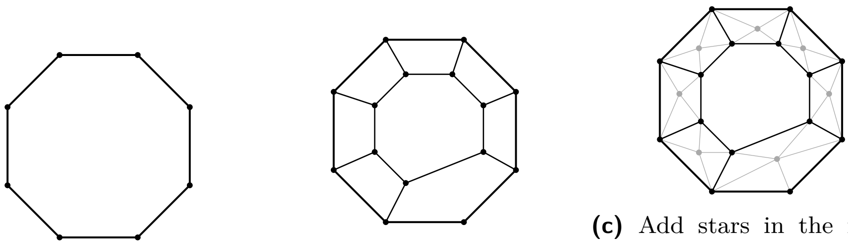
We complete the section by giving a construction of the largest set of valid curves with respect to some face F, and show that the number of curves equals the length of the shortest cycle surrounding F. Our curves will be disjoint paths in a supergraph $G_\Delta(F)$ of G. It is necessary to relate the maximum number of disjoint paths to the length of the shortest cycle surrounding F. The following lemma, proved in full paper [18], establishes this connection. We believe this lemma should be known, but cannot find it in the relevant literature.

▶ **Lemma 14.** *Let G be a triangulated graph. The graph induced by any minimum s-t vertex cut is the shortest simple cycle separating s and t.*

If G was already triangulated, we could compute a set of vertex disjoint paths from F to H (note that a set of vertex disjoint paths yields a set of valid curves). By Menger's theorem and Lemma 14, we would find ℓ paths, where ℓ is the shortest cycle surrounding F. G may not be triangulated, so instead we could first triangulate G and then compute the paths. However, the number of paths we find in this case is the length of the shortest cycle surrounding F in the triangulation of G, which may be smaller than ℓ. We prevent this from happening by producing a triangulation that adds vertices as well as edges.

▶ **Lemma 15.** *Fix a planar embedding of G with H as the outer face, and let F be other face. Then we can compute ℓ valid curves in polynomial time, where ℓ is the length of the shortest cycle surrounding F.*

Proof. We build a triangulated graph $G_\Delta(F)$ from the planar embedding of G. First, add vertices and edges to every face of G, excluding the outer face and F. We do this such that (1) every face except F and the outer face is a triangle, and (2) the distance between any $u, v \in G$ is preserved. From each face with more than 3 edges, we create one new face that has one fewer edge. One step of this iterative construction is shown in Figure 3.



(a) Some face F' with y>3 edges.

(b) Add a new cycle C with $y-1$ edges inside F' along with connecting edges.

(c) Add stars in the newly created faces, except the one formed by C. Distances between vertices of the original face are preserved.

Figure 3 Iteratively triangulate faces.

Note that distances are preserved inductively, and we make progress by reducing the size of some face. The graph we produce has 3 edges bordering each face, except for the outer face and F. In all, the number of vertices and edges added to each face of G is polynomial in the number of edges bordering the face.

Finally, we add vertices s and t, and edges from s to each vertex of F and from t to each vertex of C. The resulting graph is triangulated, and we call this graph $G_\triangle(F)$.

At this point, we can find the maximum set of vertex disjoint paths between s and t in $G_\triangle(F)$, by setting vertex capacities to 1 and computing a max flow between s and t. Because we have preserved distances between vertices of G in our construction of $G_\triangle(F)$, the length of the minimum cycle surrounding F must be ℓ. Therefore, the number of disjoint paths we find must also be ℓ. Finally, we claim that this set of disjoint paths from F to H in $G_\triangle(F)$ is a set of valid curves for G. This is because G is a subgraph of $G_\triangle(F)$, and therefore the criteria for valid curves are still met after removing the vertices and edges of $G_\triangle(F) \setminus G$. ◀

We conclude by tying together the pieces of the section to show we proved Theorem 8.

Proof of Theorem 8. Fix a face F. By Lemma 14, we determine the set of ℓ disjoint paths from F to H where the surrounding minimum cycle is of length ℓ. By Lemma 13, there is a stretch-1 retraction only if there exists a face F whose surrounding min-cycle is of length k. So if there is no stretch-1 retraction, we find $< k$ disjoint paths for all faces, and our algorithm returns "no". Otherwise, there exists a face F for which the surrounding min-cycle is of length k, and this gives a set of k valid paths. Then, by Lemma 12, the retraction that we construct has stretch 1. ◀

4 Open problems

Our work leaves several interesting directions for further research. First, we would like to determine improved upper and/or lower bounds on the best approximation factor achievable for retracting a general graph to a cycle. Second, we would like to explore extending our approach for planar graphs (Section 3) and Euclidean metrics (details in the full paper [18]) to more general graphs and high-dimensional metrics. Another open problem is that of finding approximation algorithms for retracting a general guest graph to an arbitrary host graph over a subset of anchor vertices, for which we present a hardness result in the full paper [18].

References

1. Mark Anthony Armstrong. *Basic Topology.* Springer Science & Business Media, 2013.
2. Mihai Badoiu, Kedar Dhamdhere, Anupam Gupta, Yuri Rabinovich, Harald Räcke, R. Ravi, and Anastasios Sidiropoulos. Approximation algorithms for low-distortion embeddings into low-dimensional spaces. In *SODA*, pages 119–128, 2005.
3. Aaron Bernstein, Karl Däubel, Yann Disser, Torsten Mütze Max Klimm, and Frieder Smolny. Distance-preserving graph contractions. In *ITCS*, 2018.
4. A. Blum, G. Konjevod, R. Ravi, and S. Vempala. Semi-definite relaxations for minimum bandwidth and other vertex-ordering problems. *Theoretical Computer Science*, 235(1):25 – 42, 2000.
5. Karol Borsuk. *On retractions and related sets.* PhD thesis, University of Warsaw, 1930.
6. G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension problem. *SIAM Journal on Computing*, 34(2):358–372, 2004.
7. Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. A linear programming formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on Discrete Math.*, 18:608–625, 2004.
8. J. Chuzhoy and J. S. Naor. The hardness of metric labeling. *SIAM Journal on Computing*, 36(2):498–515, 2006.
9. Chandan K. Dubey, Uriel Feige, and Walter Unger. Hardness results for approximating the bandwidth. *Journal of Computer and System Sciences*, 77(1):62–90, 2011.
10. John Dunagan and Santosh Vempala. On euclidean embeddings and bandwidth minimization. In *APPROX-RANDOM*, pages 229–240, 2001.
11. Jittat Fakcharoenphol, Chris Harrelson, Satish Rao, and Kunal Talwar. An improved approximation algorithm for the 0-extension problem. In *SODA*, pages 257–265, 2003.
12. T. Feder, P. Hell, P. Jonsson, A. Krokhin, and G. Nordh. Retractions to pseudoforests. *SIAM Journal on Discrete Mathematics*, 24(1):101–112, 2010.
13. Tomas Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial Theory*, 72(2):236 – 250, 1998.
14. Uriel Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of Computer and System Sciences*, 60(3):510 – 539, 2000.
15. Anupam Gupta. Improved bandwidth approximation for trees. In *SODA*, pages 788–793, 2000.
16. Eitan M. Gurari and Ivan Hal Sudborough. Improved dynamic programming algorithms for bandwidth minimization and the mincut linear arrangement problem. *Journal of Algorithms*, 5(4):531–546, 1984.
17. Geňa Hahn and Claude Tardif. *Graph homomorphisms: structure and symmetry*, pages 107–166. Springer Netherlands, 1997.
18. Samuel Haney, Mehraneh Liaee, Bruce M. Maggs, Rajmohan Rajaraman, Debmalya Panigrahi, and Ravi Sundaram. Retracting graphs to cycles. 2019. `arXiv:1904.11946`.
19. Mark D. Hansen. Approximation algorithms for geometric embeddings in the plane with applications to parallel processing problems. In *FOCS*, pages 604–609, 1989.
20. Stephan Held, Bernhard Korte, Dieter Rautenbach, and Jens Vygen. Combinatorial optimization in VLSI design. In *Combinatorial Optimization - Methods and Applications*, pages 33–96. 2011.
21. P. Hell and J. Nesetril. *Graphs and Homomorphisms.* Oxford Lecture Series in Mathematics and Its Applications. Oxford University Press, 2004.
22. J.G. Hocking and G.S. Young. *Topology.* Addison-Wesley series in mathematics. Dover Publications, 1961.
23. Matoušek J. Embedding finite metric spaces into normed spaces. volume 212. Springer, 2002.
24. H. Karloff, S. Khot, A. Mehta, and Y. Rabani. On eathmover distance, metric labeling, and 0-extension. *SIAM Journal on Computing*, 39(2):371–387, 2009.

25 A. V. Karzanov. Minimum 0-extension of graph metrics. *European Journal of Combinatorics*, 19:71–101, 1998.

26 Claire Kenyon, Yuval Rabani, and Alistair Sinclair. Low distortion maps between point sets. *SIAM Journal on Computing*, 39(4):1617–1636, 2009.

27 J. Kleinberg and É Tardos. Approximation algorithms for classification problems with pairwise relationships. *Journal of the ACM*, 49:616–639, 2002.

28 Jiří Matoušek and Anastasios Sidiropoulos. Inapproximability for metric embeddings into R^d. In *FOCS*, pages 405–413, 2008.

29 R. Nowakowski and I. Rival. A fixed edge theorem for graphs with loops. *Journal of Graph Theory*, 3:339–350, 1979.

30 Alain Quilliot. A retraction problem in graph theory. *Discrete Mathematics*, 54(1):61 – 71, 1985.

31 Daniel Rotter and Jens Vygen. d-dimensional arrangement revisited. *Information Processing Letters*, 113(13):498–505, 2013.

32 James B. Saxe. Dynamic-programming algorithms for recognizing small-bandwidth graphs in polynomial time. *SIAM on Journal Matrix Analysis Applications*, 1:363–369, 1980.

33 Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network localization. *Mathematical Programming*, 109(2-3):367–384, 2007.

34 E. Sperner. Neuer beweis für die invarianz der dimensionszahl und des gebietes. *Abhandlungen aus dem Mathematischen Seminar der Universität Hamburg*, 6(1):265–272, 1928.

35 Santosh Vempala. Random projection: A new approach to VLSI layout. In *FOCS*, pages 389–395, 1998.

36 Narayan Vikas. Compaction, retraction, and constraint satisfaction. *SIAM Journal on Computing*, 33(4):761–782, 2004.

37 Narayan Vikas. A complete and equal computational complexity classification of compaction and retraction to all graphs with at most four vertices and some general results. *Journal of Computer and System Sciences*, 71(4):406 – 439, 2005.