

1 Retracting Graphs to Cycles

2 **Samuel Haney**

3 Duke University
4 shaney@cs.duke.edu

5 **Mehraneh Liaee**

6 Northeastern University
7 mehraneh@ccs.neu.edu

8 **Bruce M. Maggs**

9 Duke University & Akamai Technologies
10 bmm@cs.duke.edu

11 **Debmalya Panigrahi**

12 Duke University
13 debmalya@cs.duke.edu

14 **Rajmohan Rajaraman**

15 Northeastern University
16 rraj@ccs.neu.edu

17 **Ravi Sundaram**

18 Northeastern University
19 koods@ccs.neu.edu

20 — Abstract —

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

33 **2012 ACM Subject Classification** Theory of computation → Design and analysis of algorithms

34 **Keywords and phrases** Graph algorithms, Graph embedding, Planar graphs, Approximation al-
35 gorithms.

36 **Digital Object Identifier** 10.4230/LIPIcs.ICALP.2019.65

37 **Category** Regular Paper Track A

38 **Related Version** A full version of the paper is available at <https://arxiv.org/abs/1904.11946>

39 **Funding** All the authors were partially supported by NSF grant CCF 1535972. Additional funding
40 information appears below.

41 *Debmalya Panigrahi*: NSF grant CCF 1527084, an NSF CAREER Award CCF 1750140, and the
42 Joint Indo-US Networked Center for Algorithms under Uncertainty.

43 *Ravi Sundaram*: NSF grant CNS 1718286.

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



© Samuel Haney, Mehraneh Liaee, Bruce M. Maggs, Debmalya Panigrahi, Rajmohan Rajaraman,
Ravi Sundaram;
licensed under Creative Commons License CC-BY

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
LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



46 **1** Introduction

47 Originally introduced in 1930 by K. Borsuk in his PhD thesis [5], *retraction* is a fundamental
 48 concept in topology describing continuous mappings of a topological space into a subspace
 49 that leaves the position of all points in the subspace fixed. Over the years, this has developed
 50 into a rich theory with deep connections to fundamental results in topology such as Brouwer’s
 51 Fixed Point Theorem [22]. Inspired by this success, graph theorists have extensively studied a
 52 discrete version of the problem in graphs, where a *retraction* is a mapping from the vertices of
 53 a graph to a given subgraph that produces the identity map when restricted to the subgraph
 54 (i.e., it leaves the subgraph fixed). For a rich history of retraction in graph theory, we refer
 55 the reader to [21]. Define the *stretch* of a retraction to be the maximum distance between
 56 the images of the endpoints of any edge, as measured in the subgraph. We use *stretch- k*
 57 retraction to mean a retraction whose stretch is k ; in particular, a *stretch-1* retraction is a
 58 mapping where every edge of the graph is mapped to either an edge of the subgraph, or both
 59 its ends are mapped to the same vertex of the subgraph¹.

60 In this paper, we study the algorithmic problem of finding a *minimum stretch retraction*
 61 in a graph. This problem belongs to the rich area of metric embeddings, but somewhat
 62 surprisingly, has not received much attention in spite of the deep but non-constructive
 63 results in the graph theory literature. The graph retraction problem has a close resemblance
 64 to the well-studied 0-extension problem [6, 24, 25] (and its generalizations such as metric
 65 labeling [27, 8]), which is also an embedding of a graph G to a metric over a subset of
 66 terminals H with the constraint that each vertex in H maps to itself. The two problems
 67 differ in their objective: whereas 0-extension seeks to minimize the *average* stretch of edges,
 68 graph retraction minimizes the *maximum* stretch. The different objectives lead to significant
 69 technical differences. For instance, a well-studied linear program called the earthmover
 70 LP has a nearly logarithmic integrality gap for 0-extension. In contrast, we show that a
 71 corresponding earthmover LP for graph retraction has integrality gap $\Omega(\sqrt{n})$. A well-studied
 72 problem in the metric embedding literature that considers the maximum stretch objective is
 73 the *minimum bandwidth* problem, where one seeks an isomorphic embedding of a graph into
 74 a line (or cycle) that minimizes maximum stretch. In contrast, in graph retraction, we allow
 75 homomorphic maps² but additionally require a subset of vertices (called the *anchors*) to be
 76 mapped to themselves.

77 From an applications standpoint, our original motivation for studying minimum-stretch
 78 graph retraction comes from a distributed systems scenario where the aim is to map processes
 79 comprising a distributed computation to a network of servers where some processes are
 80 constrained to be mapped onto specific servers. The objective is to minimize the maximum
 81 communication latency between two communicating processes in the embedding. Such
 82 anchored embedding problems can be shown to be equivalent to graph retraction for gen-
 83 eral subgraphs, and arise in several other domains including VLSI layout, multi-processor
 84 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.

85 1.1 Problem definition, techniques, and results

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

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

94 The graph retraction problem is easy if the subgraph H is acyclic (see, e.g., [29]); therefore,
 95 the first non-trivial problem is to retract a graph into a cycle. Indeed, this problem is NP-
 96 hard even when H is just a 4-cycle [13]. Given this intractability result, a natural goal is
 97 to obtain an algorithm for retracting graphs to cycles that *approximately* minimizes the
 98 stretch of the retraction. This problem is the focus of our work. While there has been
 99 considerable interest in identifying conditions under which retracting to a cycle with stretch
 100 1 is tractable [17, 21, 37], there has been no work (to the best of our knowledge) on deriving
 101 approximations to the minimum stretch.³

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

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

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

118 Our results for retracting a general graph to a cycle appear in Section 2. We also give
 119 evidence that the gap induced by Sperner's lemma on a grid graph is fundamental, showing an
 120 $\Omega(\min\{k, \sqrt{n}\})$ integrality gap for natural linear and semi-definite programming relaxations
 121 of the problem. To overcome this gap, we focus on the special case of planar graphs, of which
 122 the grid is an example. Retraction in planar graphs has been considered in the past, most
 123 notably in a beautiful paper of Quilliot [30] that uses homotopy techniques to characterize
 124 stretch-1 retractions of a planar graph to a cycle. Quilliot's proof, however, does not yield
 125 an efficient algorithm. In Section 3, we provide an exact algorithm for retraction in planar
 126 graphs by developing the gap induced by Sperner's lemma on a grid into a general lower
 127 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.

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

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

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

145 1.2 Related work

146 **List homomorphisms and constraint satisfaction.** The graph retraction problem is
 147 a special case of the *list homomorphism* problem introduced by Feder and Hell [13], who
 148 established conditions under which the problem is NP-complete. Given graphs G, H , and
 149 $L(v) \subset V(H)$ for each $v \in V(G)$, a list homomorphism of G to H with respect to L is a
 150 homomorphism $f : G \rightarrow H$ with $f(v) \in L(v)$ for each $v \in V(G)$.

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

160 **0-extension, minimum bandwidth, and low-distortion embeddings.** From an ap-
 161 proximation algorithms standpoint, the graph retraction problem is closely related to the
 162 0-extension and minimum bandwidth problems [14, 4, 15, 35, 9, 32]. In the 0-extension prob-
 163 lem, one seeks to minimize the average stretch, which can be solved to an $O(\log k / \log \log k)$
 164 approximation using a natural LP relaxation [6, 11]. In contrast, we give polynomial in-
 165 tegrality gaps for the graph retraction problem. In the minimum bandwidth problem, the
 166 objective is to find an embedding to a line that minimizes maximum stretch, but the con-
 167 straint is that the map must be isomorphic rather than that the anchor vertices must be
 168 fixed. In a seminal result [14], Feige designed the first polylogarithmic-approximation using
 169 a novel concept of volume-respecting embeddings. A slightly improved approximation was
 170 achieved in [10] by combining Feige’s approach with another bandwidth algorithm based on
 171 semidefinite-programming [4]. Interestingly, the minimum bandwidth problem is NP-hard
 172 even for (guest) trees, while graph retraction to (host) trees is solvable in polynomial time.
 173 Conversely, the bandwidth problem is solvable in time $O(n^b)$ for bandwidth b graphs [16],

174 while graph retraction to a cycle is NP-complete even when the host cycle has only four
 175 vertices. Nevertheless, it is conceivable that volume-respecting embeddings, in combination
 176 with random projection, could lead to effective approximation algorithms for graph retraction
 177 to a cycle in a manner similar to what was achieved for VLSI layout on the plane [35].

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

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

192 2 Retracting an arbitrary graph to a cycle

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

197 Arguably, the simplest lower bound on the optimal stretch is the *distance-based* bound
 198 $\ell(G, H) = \max_{u, v \in A} d_H(u, v) / d_G(u, v)$, since every retraction places a path of length $d_G(u, v)$
 199 in G on a path of length at least $d_H(u, v)$ in H .

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

206 We now show how to implement the first step of Algorithm 1. Our goal is to embed each
 207 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
 208 following inequality, where $d_\infty(a, b)$ denotes the L_∞ distance between a and b . (That is, for
 209 two points (x_1, y_1) and (x_2, y_2) , $d_\infty((x_1, y_1), (x_2, y_2)) = \max\{|x_1 - x_2|, |y_1 - y_2|\}$.)

$$210 \quad d_\infty(g(u), g(v)) \leq \ell(G, H) d_G(u, v) \tag{1}$$

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

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

214 **Proof.** We incrementally construct the embedding g . Initially, we place the anchors on the
 215 boundary of the grid so that the boundary is isometric to d_H . (This can be done since H is
 216 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
 217 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 $u, v \in V$, $d_\infty(g(u), g(v)) \leq \ell(G, H)d_G(u, v)$.**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

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

221 Let $B_\infty(g(u), r)$ denote the L_∞ ball around $g(u)$ with radius r (note that these balls
 222 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
 223 set $g(v_i) = x$, then inequality 1 holds for all points in $U \cup \{v_i\}$. We now show that this
 224 intersection is nonempty (it is straightforward to find an element in the intersection). The
 225 set of axis aligned squares has Helly number⁴ 2; therefore it is enough to show that for every
 226 $u, u' \in U$, $B_\infty(g(u), \ell(G, H)d_G(u, v_i))$ and $B_\infty(g(u'), \ell(G, H)d_G(u', v_i))$ intersect. Otherwise,

$$227 \quad d_\infty(g(u), g(u')) > \ell(G, H)(d_G(u, v_i) + d_G(u', v_i)) \geq \ell(G, H)d_G(u, u').$$

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

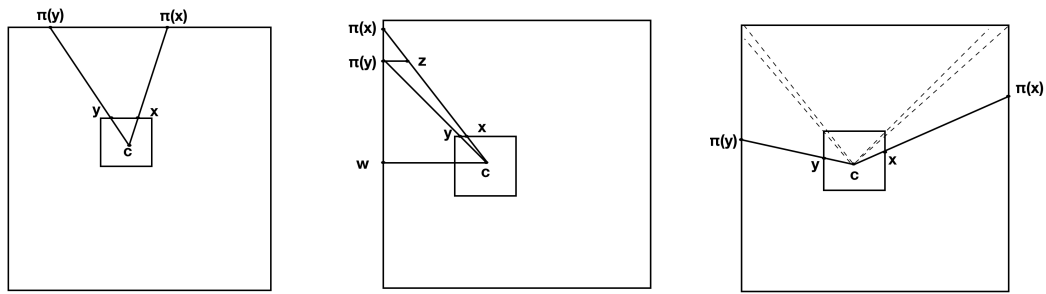
231 In the following lemma, we analyze the projection embedding step of the algorithm.

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

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

238 We first argue that it is sufficient to establish the preceding claim for points on the
 239 boundary of D , at the loss of a factor of $\sqrt{2}$. Let x and y be two arbitrary points in M but
 240 not in the interior of D . Let x' (resp., y') denote the intersection of $R(x)$ (resp., $R(y)$) and
 241 the boundary of D . From elementary geometry, it follows that $d(x', y') \leq d(x, y)$, where d
 242 is the Euclidean distance; since $d_\infty(x, y) \geq d(x, y)/\sqrt{2}$ and $d_\infty(x', y') \leq d(x', y')$, we obtain
 243 $d_\infty(x', y') \leq \sqrt{2}d_\infty(x, y)$. Since $\pi(x) = \pi(x')$ and $\pi(y) = \pi(y')$, establishing the above
 244 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.

245 Consider points x and y on the boundary of D . We consider three cases. In the first
 246 two cases, x and y are on the same side of D . In the first case (Figure 1a), $\pi(x)$ and $\pi(y)$
 247 are on the same side of the boundary of M and segment $\overline{\pi(x)\pi(y)}$ is parallel to segment
 248 \overline{xy} ; then, by similarity of triangle formed by c , x , and y and the one formed by c , $\pi(x)$ and
 249 $\pi(y)$, we obtain that the distance between $\pi(x)$ and $\pi(y)$ is at most $3kd_\infty(x, y)/(16r)$. In
 250 the second case (Figure 1b), $\pi(x)$ and $\pi(y)$ are on same side of the boundary of M , and
 251 segment $\overline{\pi(x)\pi(y)}$ is orthogonal to segment \overline{xy} . In this case, w.l.o.g. assume that $\pi(y)$ is
 252 closer to center c than $\pi(x)$ with respect to d_∞ distance. Let point z be a point on segment
 253 $\overline{c\pi(x)}$ such that segments \overline{xy} and $\overline{\pi(y)z}$ are parallel. From center c extend a line parallel to
 254 segment \overline{xy} until it hits the side of M on which $\pi(x)$ and $\pi(y)$ are. Let w be the intersection.
 255 Using elementary geometry and similarity argument, we have the following:

$$256 \quad \frac{|\overline{\pi(x)\pi(y)}|}{|z\overline{\pi(y)}|} = \frac{|\overline{\pi(x)w}|}{|c\overline{w}|} \leq \frac{k/4}{k/16} = 4 \quad \text{and} \quad \frac{|\overline{z\pi(y)}|}{|\overline{xy}|} = \frac{|\overline{\pi(y)w}|}{r} \leq \frac{k}{4r}$$

257 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
 258 at least half the shortest path between x and y that lies within the boundary of D . This
 259 latter shortest path consists of at most five segments, each residing completely on one side of
 260 the boundary of D . We apply the argument of the first and second case to each of these
 261 segments to obtain that the distance between $\pi(x)$ and $\pi(y)$ is at most $10kd_\infty(x, y)/r$.

262 To complete the proof, we note that distance between anchor nearest (clockwise) to $\pi(x)$
 263 and anchor nearest (clockwise) to $\pi(y)$ is at most one plus the distance between $\pi(x)$ and
 264 $\pi(y)$. Therefore, the $d_H(f(u), f(v))$ is at most $1 + 10\sqrt{2}kd_\infty(g(u), g(v))/r$. ◀

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

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

271 $d_H(f(u), f(v))$ is at most $1 + O(\sqrt{n})\ell(G, H)d_G(u, v)$. Since the distance in H cannot exceed
 272 $k/2$, the claim of the theorem follows. ◀

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

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

280 **Proof.** Suppose we triangulate the grid by adding northwest-to-southeast diagonals in each
 281 cell of the grid. Consider the following coloring of the boundary H with 3 colors. Divide H
 282 into three segments, each consisting of a contiguous sequence of at least $\lfloor 4\sqrt{n}/3 \rfloor$ vertices; all
 283 vertices in the first, second, and third segment are colored red, green, and blue, respectively.
 284 Let f be any retraction from G to H . Let c_f denote the following coloring for $G \setminus H$: the
 285 color of u is the color of $f(u)$. By Sperner's Lemma [34], there exists a tri-chromatic triangle.
 286 This implies that there are two vertices within distance at most two in G that are at least
 287 $4\sqrt{n}/3$ apart in the retraction f , resulting in a stretch of at least $2\sqrt{n}/3$. ◀

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

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

307 **3 Retracting a planar graph to a cycle**

308 The main result of this section is the following theorem.

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

311 We begin by presenting some useful definitions and elementary claims in Section 3.1. We
 312 then present an overview of our algorithm in Section 3.2. Finally, we present the algorithm
 313 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.

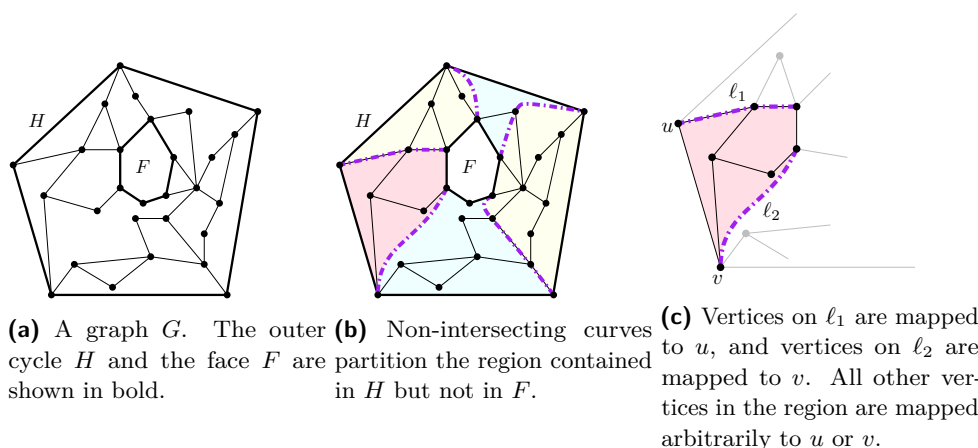
3.2 Overview of our algorithm

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



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

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

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

364 Of course, it is not obvious that a valid set of curves exists for a given face, and, if it
 365 does, how to compute it. We show that if the graph has a stretch-1 retraction, then there is
 366 some face F with k valid curves, and that we can efficiently compute them. Our algorithm
 367 (Algorithm 2) iterates over all faces in the graph, in each case finding the maximum number
 368 of valid curves it can with respect to that face. The number of valid curves we can find is
 369 the length of the shortest cycle surrounding F . If the shortest cycle C surrounding F has
 370 length ℓ , then it is impossible to find more than ℓ valid curves with respect to F : By the
 371 Jordan curve theorem, each curve must intersect C , and by the definition, valid curves do
 372 not intersect each other and can intersect C only at its vertices. Our construction of the
 373 valid curves shows that this is tight (i.e. we can always find ℓ curves). We show that if a
 374 stretch-1 retraction exists, then there is some face for which $\ell = k$. Algorithm 2 gives an
 375 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  $H$   $p_1, \dots, p_\ell$ 
3:   if  $\ell = k$  then
4:     Compute stretch-1 retraction from  $G$  to  $H$  using  $p_1, \dots, p_k$ 
5:   end if
6: end for
7: If no retraction was computed, report no stretch-1 retraction exists

```

376 3.3 Algorithm and analysis

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

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

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

390 ► **Lemma 11.** *Let $\{p_1, \dots, p_k\}$ be a set of curves that are valid with respect to F . Let Z
 391 denote the set of faces of $H \cup F \cup \bigcup_i p_i$ excluding the outer face and F . Then, each face
 392 $f \in Z$ is bordered by exactly 1 edge of H , and every vertex of $G \setminus \bigcup_i p_i$ is in a unique face of
 393 Z .*

394 **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
 395 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
 396 the path containing u , p_j is the path containing v , and $p_F(i, j)$ is the path on the boundary
 397 of F between the vertices where i and j meet F such that F is not contained in X . If p_i and
 398 p_j are both degenerate (i.e., each is empty), then $(u, v) = p_F(i, j)$. Otherwise X is a simple
 399 cycle. We claim that X defines a face. In particular, we show that the path $p_F(i, j)$ contains
 400 no other vertex of path p_z for all $z \neq i, j$. Suppose it does and let w be that vertex. Let w'
 401 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
 402 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
 403 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
 404 these outcomes leads to a contradiction. ◀

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

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

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

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

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

65:12 Retracting Graphs to Cycles

423 The claim is very similar to Sperner's lemma, and the proof is similar as well. Let
424 $\phi : V(G) \rightarrow V(H)$ denote the retraction. We associate a score with each cycle C of the
425 graph: Order the vertices of the cycle in clockwise order, denoted $v_1, v_2, \dots, v_j, v_{j+1} = v_1$.
426 Consider the sequence $\phi(v_1), \dots, \phi(v_j), \phi(v_{j+1})$. Let the score of C be 0 to start. For each
427 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 .
428 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
429 of C . If they are in counterclockwise order, subtract 1. If they are the same vertex, the score
430 remains the same. If $\phi(v_1), \dots, \phi(v_j)$ does not contain every vertex on the outer cycle, the
431 score of C must be 0, since each edge along the path $\phi(v_1), \dots, \phi(v_{j+1})$ is traversed exactly
432 the same number of times in each direction. On the other hand, a cycle with a non-zero
433 score must have a score that is divisible by k .

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

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

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

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

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

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

464 **Proof.** We build a triangulated graph $G_\Delta(F)$ from the planar embedding of G . First, add
465 vertices and edges to every face of G , excluding the outer face and F . We do this such that
466 (1) every face except F and the outer face is a triangle, and (2) the distance between any
467 $u, v \in G$ is preserved. From each face with more than 3 edges, we create one new face that
468 has one fewer edge. One step of this iterative construction is shown in Figure 3.

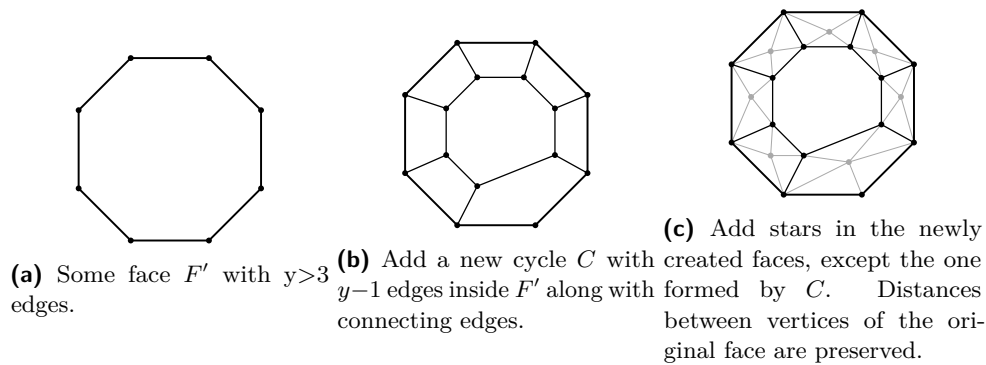


Figure 3 Iteratively triangulate faces.

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

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

475 At this point, we can find the maximum set of vertex disjoint paths between s and t in
 476 $G_{\Delta}(F)$, by setting vertex capacities to 1 and computing a max flow between s and t . Because
 477 we have preserved distances between vertices of G in our construction of $G_{\Delta}(F)$, the length
 478 of the minimum cycle surrounding F must be ℓ . Therefore, the number of disjoint paths we
 479 find must also be ℓ . Finally, we claim that this set of disjoint paths from F to H in $G_{\Delta}(F)$
 480 is a set of valid curves for G . This is because G is a subgraph of $G_{\Delta}(F)$, and therefore the
 481 criteria for valid curves are still met after removing the vertices and edges of $G_{\Delta}(F) \setminus G$. ◀

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

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

490 4 Open problems

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

499 — References

- 500 1 Mark Anthony Armstrong. *Basic Topology*. Springer Science & Business Media, 2013.
- 501 2 Mihai Badoiu, Kedar Dhamdhere, Anupam Gupta, Yuri Rabinovich, Harald Räcke, R. Ravi,
502 and Anastasios Sidiropoulos. Approximation algorithms for low-distortion embeddings into
503 low-dimensional spaces. In *SODA*, pages 119–128, 2005.
- 504 3 Aaron Bernstein, Karl Däubel, Yann Disser, Torsten Mütze Max Klimm, and Frieder Smolny.
505 Distance-preserving graph contractions. In *ITCS*, 2018.
- 506 4 A. Blum, G. Konjevod, R. Ravi, and S. Vempala. Semi-definite relaxations for minimum
507 bandwidth and other vertex-ordering problems. *Theoretical Computer Science*, 235(1):25 – 42,
508 2000.
- 509 5 Karol Borsuk. *On retractions and related sets*. PhD thesis, University of Warsaw, 1930.
- 510 6 G. Calinescu, H. Karloff, and Y. Rabani. Approximation algorithms for the 0-extension
511 problem. *SIAM Journal on Computing*, 34(2):358–372, 2004.
- 512 7 Chandra Chekuri, Sanjeev Khanna, Joseph Naor, and Leonid Zosin. A linear programming
513 formulation and approximation algorithms for the metric labeling problem. *SIAM Journal on*
514 *Discrete Math.*, 18:608–625, 2004.
- 515 8 J. Chuzhoy and J. S. Naor. The hardness of metric labeling. *SIAM Journal on Computing*,
516 36(2):498–515, 2006.
- 517 9 Chandan K. Dubey, Uriel Feige, and Walter Unger. Hardness results for approximating the
518 bandwidth. *Journal of Computer and System Sciences*, 77(1):62–90, 2011.
- 519 10 John Dunagan and Santosh Vempala. On euclidean embeddings and bandwidth minimization.
520 In *APPROX-RANDOM*, pages 229–240, 2001.
- 521 11 Jittat Fakcharoenphol, Chris Harrelson, Satish Rao, and Kunal Talwar. An improved approx-
522 imation algorithm for the 0-extension problem. In *SODA*, pages 257–265, 2003.
- 523 12 T. Feder, P. Hell, P. Jonsson, A. Krokhn, and G. Nordh. Retractions to pseudoforests. *SIAM*
524 *Journal on Discrete Mathematics*, 24(1):101–112, 2010.
- 525 13 Tomas Feder and Pavol Hell. List homomorphisms to reflexive graphs. *Journal of Combinatorial*
526 *Theory*, 72(2):236 – 250, 1998.
- 527 14 Uriel Feige. Approximating the bandwidth via volume respecting embeddings. *Journal of*
528 *Computer and System Sciences*, 60(3):510 – 539, 2000.
- 529 15 Anupam Gupta. Improved bandwidth approximation for trees. In *SODA*, pages 788–793,
530 2000.
- 531 16 Eitan M. Gurari and Ivan Hal Sudborough. Improved dynamic programming algorithms for
532 bandwidth minimization and the mincut linear arrangement problem. *Journal of Algorithms*,
533 5(4):531–546, 1984.
- 534 17 Geña Hahn and Claude Tardif. *Graph homomorphisms: structure and symmetry*, pages
535 107–166. Springer Netherlands, 1997.
- 536 18 Samuel Haney, Mehraneh Liaee, Bruce M. Maggs, Rajmohan Rajaraman, Debmalya Panigrahi,
537 and Ravi Sundaram. Retracting graphs to cycles. 2019. [arXiv:1904.11946](https://arxiv.org/abs/1904.11946).
- 538 19 Mark D. Hansen. Approximation algorithms for geometric embeddings in the plane with
539 applications to parallel processing problems. In *FOCS*, pages 604–609, 1989.
- 540 20 Stephan Held, Bernhard Korte, Dieter Rautenbach, and Jens Vygen. Combinatorial optimiza-
541 tion in VLSI design. In *Combinatorial Optimization - Methods and Applications*, pages 33–96.
542 2011.
- 543 21 P. Hell and J. Nešetřil. *Graphs and Homomorphisms*. Oxford Lecture Series in Mathematics
544 and Its Applications. Oxford University Press, 2004.
- 545 22 J.G. Hocking and G.S. Young. *Topology*. Addison-Wesley series in mathematics. Dover
546 Publications, 1961.
- 547 23 Matoušek J. Embedding finite metric spaces into normed spaces. volume 212. Springer, 2002.
- 548 24 H. Karloff, S. Khot, A. Mehta, and Y. Rabani. On earthmover distance, metric labeling, and
549 0-extension. *SIAM Journal on Computing*, 39(2):371–387, 2009.

- 550 25 A. V. Karzanov. Minimum 0-extension of graph metrics. *European Journal of Combinatorics*,
551 19:71–101, 1998.
- 552 26 Claire Kenyon, Yuval Rabani, and Alistair Sinclair. Low distortion maps between point sets.
553 *SIAM Journal on Computing*, 39(4):1617–1636, 2009.
- 554 27 J. Kleinberg and É Tardos. Approximation algorithms for classification problems with pairwise
555 relationships. *Journal of the ACM*, 49:616–639, 2002.
- 556 28 Jiří Matoušek and Anastasios Sidiropoulos. Inapproximability for metric embeddings into R^d .
557 In *FOCS*, pages 405–413, 2008.
- 558 29 R. Nowakowski and I. Rival. A fixed edge theorem for graphs with loops. *Journal of Graph*
559 *Theory*, 3:339–350, 1979.
- 560 30 Alain Quilliot. A retraction problem in graph theory. *Discrete Mathematics*, 54(1):61 – 71,
561 1985.
- 562 31 Daniel Rotter and Jens Vygen. d-dimensional arrangement revisited. *Information Processing*
563 *Letters*, 113(13):498–505, 2013.
- 564 32 James B. Saxe. Dynamic-programming algorithms for recognizing small-bandwidth graphs in
565 polynomial time. *SIAM on Journal Matrix Analysis Applications*, 1:363–369, 1980.
- 566 33 Anthony Man-Cho So and Yinyu Ye. Theory of semidefinite programming for sensor network
567 localization. *Mathematical Programming*, 109(2-3):367–384, 2007.
- 568 34 E. Sperner. Neuer beweis für die invarianz der dimensionszahl und des gebietes. *Abhandlungen*
569 *aus dem Mathematischen Seminar der Universität Hamburg*, 6(1):265–272, 1928.
- 570 35 Santosh Vempala. Random projection: A new approach to VLSI layout. In *FOCS*, pages
571 389–395, 1998.
- 572 36 Narayan Vikas. Compaction, retraction, and constraint satisfaction. *SIAM Journal on*
573 *Computing*, 33(4):761–782, 2004.
- 574 37 Narayan Vikas. A complete and equal computational complexity classification of compaction
575 and retraction to all graphs with at most four vertices and some general results. *Journal of*
576 *Computer and System Sciences*, 71(4):406 – 439, 2005.