Bicriteria Approximation for Minimum Dilation Graph Augmentation

Kevin Buchin
Technical University Dortmund, Germany

Maike Buchin
Ruhr University Bochum, Germany

Joachim Gudmundsson
University of Sydney, Australia

Sampson Wong
University of Copenhagen, Denmark

Abstract

Spanner constructions focus on the initial design of the network. However, networks tend to improve over time. In this paper, we focus on the improvement step. Given a graph and a budget k, which k edges do we add to the graph to minimise its dilation? Gudmundsson and Wong [TALG'22] provided the first positive result for this problem, but their approximation factor is linear in k.

Our main result is a $(2\sqrt[r]{2}\ k^{1/r}, 2r)$-bicriteria approximation that runs in $O(n^3 \log n)$ time, for all $r \geq 1$. In other words, if t^* is the minimum dilation after adding any k edges to a graph, then our algorithm adds $O(k^{1+1/r})$ edges to the graph to obtain a dilation of $2rt^*$. Moreover, our analysis of the algorithm is tight under the Erdős girth conjecture.

2012 ACM Subject Classification Theory of computation → Computational geometry

Keywords and phrases Greedy spanner, Graph augmentation

Digital Object Identifier 10.4230/LIPIcs.ESA.2024.36

Related Version *Full Version*: https://arxiv.org/abs/2407.04614

1 Introduction

Let G be a graph embedded in a metric space M. Let $V(G)$, $E(G)$ be the vertices and edges of G. For vertices $u, v \in V(G)$, define $d_M(u, v)$ to be the metric distance between points $u, v \in M$, and define $d_G(u, v)$ to be the shortest path distance between vertices $u, v \in G$. The *dilation* or *stretch* of G is the minimum $t \in \mathbb{R}$ so that for all $u, v \in V(G)$, we have $d_G(u, v) \leq t \cdot d_M(u, v)$.

Dilation measures the quality of a network in applications such as transportation, energy, and communication. For now, we restrict our attention to the special case of low dilation trees.

▶ **Problem 1.** *Given a set of n points V embedded in a metric space M, compute a spanning tree of V with minimum dilation.*

Problem 1 is known across the theory community, as either the minimum dilation spanning tree problem [4, 10, 12], the tree spanner problem [11, 21, 23] or the minimum maximum-stretch spanning tree problem [16, 34, 38]. The problem is NP-hard even if M is an unweighted graph metric [11] or the Euclidean plane [12]. Problem 1 is closely related to tree embeddings of general metrics [5], and has applications to communication networks and distributed systems [38].

32nd Annual European Symposium on Algorithms (ESA 2024).
Editors: Timothy Chan, Johannes Fischer, John Iacono, and Grzegorz Herman; Article No. 36; pp. 36:1–36:15
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

The approximability of Problem 1 is an open problem stated in several surveys and papers [12, 17, 38], and is a major obstacle towards constructing low dilation graphs with few edges [4, 29]. The minimum spanning tree is an $O(n)$-approximation [17] for Problem 1, but no better result is known. Only in the special case where M is an unweighted graph is there an $O(\log n)$-approximation [16].

▶ **Obstacle 2.** *Is there an $O(n^{1-\varepsilon})$-approximation algorithm for Problem 1, for any $\varepsilon > 0$?*

If we no longer restrict ourselves to trees, we can shift our attention to spanners, which are low dilation sparse graphs. An advantage of spanners over minimum dilation trees is that spanners are not affected by Obstacle 2. Spanners obtain significantly better dilation guarantees, at the cost of adding slightly more edges. The trade-off between sparsity and dilation in spanners has been studied extensively [2, 14, 22, 32]. For an overview of the rich history and multitude of applications of spanners, see the survey on graph spanners [1] and the textbook on geometric spanners [37].

Spanner constructions focus on the initial design of the network. However, networks tend to improve over time. In this paper, we focus on the improvement step. Given a graph and a budget k, which k edges do we add to the graph to minimise its dilation?

▶ **Problem 3.** *Given a positive integer k and a metric graph G, compute a set S of k edges so that the dilation of the graph $G' = (V(G), E(G) \cup S)$ is minimised. Note that $S \subseteq V(G) \times V(G)$.*

Narasimham and Smid [37] stated Problem 3 as one of twelve open problems in the final chapter of their reference textbook. For over a decade, the only positive results for Problem 3 were for the special case where $k = 1$ [3, 20, 35, 43]. In 2021, Gudmundsson and Wong [29] showed the first positive result for $k \geq 2$, by providing an $O(k)$-approximation algorithm that runs in $O(n^3 \log n)$ time. A downside of [29] is that their approximation factor is linear in k. However, since Problem 1 is a special case of Problem 3, Obstacle 2 applies to Problem 3 as well.

▶ **Obstacle 4.** *One cannot obtain an $O(k^{1-\varepsilon})$-approximation algorithm for Problem 3 for any $\varepsilon > 0$, without first resolving Obstacle 2.*

One way to circumvent Obstacle 4 is to consider a bicriteria approximation. An advantage of a bicriteria approximation is that we can obtain significantly better dilation guarantees, at the cost of adding slightly more edges.

The goal of our bicriteria problem is to investigate the trade-off between sparsity and dilation. We define the sparsity parameter f to be the number of edges added by our algorithm divided by k. We define the dilation parameter g to be the dilation of our algorithm (which adds fk edges) divided by the dilation of the optimal solution (which adds k edges).

▶ **Problem 5.** *Given a positive integer k, a metric graph G, sparsity $f \in \mathbb{R}$ and dilation $g \in \mathbb{R}$, construct a set S of fk edges so that the dilation of the graph $G' = (V(G), E(G) \cup S)$ is at most gt^*, where t^* is the minimum dilation in Problem 3. Note that $S \subseteq V(G) \times V(G)$.*

We define an (f, g)-bicriteria approximation to be an algorithm for Problem 5 that achieves sparsity f and dilation g.

1.1 Contributions

Our main result is a $(2\sqrt[r]{2}\ k^{1/r}, 2(1+\delta)r)$-bicriteria approximation for Problem 5 that runs in $O(n^3(\log n + \log \frac{1}{\delta}))$ time, for all $r \geq 1$ and $\delta > 0$. In other words, if t^* is the minimum dilation after adding any k edges to a graph, then our algorithm adds $O(k^{1+1/r})$ edges to the graph to obtain a dilation of $2(1+\delta)rt^*$. Our dilation guarantees are significantly better than the previous best result [29], at the cost of adding slightly more edges. For example, if $r = \log(2k)$ we obtain a $(4, 2(1+\delta)\log(2k))$-bicriteria approximation algorithm, which adds $4k$ edges to the graph to obtain a dilation of $2(1+\delta)\log(2k)t^*$. See Table 1.

Our approach uses the greedy spanner construction. The greedy spanner is among the most extensively studied spanner constructions [2, 14, 22, 32]. Therefore, it is perhaps unsurprising that greedy spanner can be used for Problem 5. Nonetheless, we believe that our result shows the utility and versatility of the greedy spanner.

Our main technical contribution is our analysis of the greedy spanner. Our main insight is to construct an auxilliary graph, which we call the girth graph, and to argue that the approximation ratio is bounded by the length of the shortest cycle in the girth graph. Moreover, our analysis of the greedy spanner is tight, up to constant factors. In particular, assuming the Erdős girth conjecture, there is a graph class for which our algorithm is an $(\Omega(k^{1/r}), 2r+1)$-bicriteria approximation.

Assuming $W[1] \neq FPT$, we prove that one cannot obtain a $(h(k), 2-\varepsilon)$-bicriteria approximation, for any computable function h and for any $\varepsilon > 0$. Since one cannot approximate the dilation to within a factor of $(2-\varepsilon)$, The restriction $r \geq 1$ is essentially necessary in our main result.

Finally, we use ideas from our hardness proof to provide a $(4k\log n, 1)$-bicriteria approximation.

Our results are summarised in Table 1. For a technical overview of our results, see Section 2.

Table 1 The table shows the trade-off between sparsity f and dilation g in our bicriteria approximation algorithms for Problem 5. Note that EGC is the Erdős girth conjecture, $h(\cdot)$ is any computable function, and $O_\varepsilon(\cdot)$ hides dependence on ε.

Sparsity (f)	Dilation (g)	Complexity	Reference
1	$(1+\delta)(k+1)$	$O(n^3(\log n + \log \frac{1}{\delta}))$	Gudmundsson and Wong [29]
$2+\varepsilon$	$O_\varepsilon((1+\delta)\log(k))$	$O(n^3(\log n + \log \frac{1}{\delta}))$	$r = O_\varepsilon(\log(k))$ in Theorem 6
4	$2(1+\delta)\log(2k)$	$O(n^3(\log n + \log \frac{1}{\delta}))$	$r = \log(2k)$ in Theorem 6
$2^{1+\varepsilon}\ k^\varepsilon$	$2(1+\delta)\varepsilon^{-1}$	$O(n^3(\log n + \log \frac{1}{\delta}))$	$r = 1/\varepsilon$ in Theorem 6
$2\sqrt{2}\ \sqrt{k}$	$4(1+\delta)$	$O(n^3(\log n + \log \frac{1}{\delta}))$	$r = 2$ in Theorem 6
$4k$	$2(1+\delta)$	$O(n^3(\log n + \log \frac{1}{\delta}))$	$r = 1$ in Theorem 6
Our analysis in Theorem 6 is tight under EGC			Theorem 7
$h(k)$	$2-\varepsilon$	W[1]-hard	Theorem 8
$4k\log n$	1	$O(n^6\log n)$	Theorem 9

1.2 Related work

Due to the difficult nature of Problem 3, most of the literature focuses on the special case where $k = 1$. Farshi, Giannopoulos and Gudmundsson [20] provide an $O(n^4)$ time algorithm, and an $O(n^3)$ time 3-approximation when $k = 1$. Wulff-Nilsen [43] presents an $O(n^3 \log n)$ time algorithm. Luo and Wulff-Nilsen [35] improves the space requirement to linear. Aronov et al. [3] provide a nearly-linear time algorithm in the special case where the graph is a simple polygon and an interior point.

A variant of Problem 3 is to add k edges to a graph to minimise the diameter instead of the dilation. Frati, Gaspers, Gudmundsson and Mathieson [24] provide a fixed parameter tractable 4-approximation for the problem. Bilò, Gualà and Proietti [8] provide bicriteria approximability and inapproximability results. Several special cases have been studied. Demaine and Zadimoghaddam [15] consider adding k edges of length δ, where δ is small relative to the diameter. Große et al. [26] present nearly-linear time algorithms for adding one edge to either a path or a tree in order to minimise its diameter. Follow up papers improve the running time of the algorithm for paths [40] and for trees [6, 42]. Bilò, Gualà, Stefano Leucci and Sciarria [7] extend the linear time algorithm to approximate the minimum diameter when $k > 1$ edges are added to a tree.

Another variant is to add k edges to a graph to minimise the radius. Gudmundsson, Sha and Yao [28] provide a 3-approximation for adding k edges to a graph to minimise its radius. The problem of adding one edge to minimise the radius of paths [30, 41] and trees [27] has also been studied.

A problem closely related to Problem 1 is to compute minimum dilation graphs. In his Master's thesis, Mulzer [36] studies minimum dilation triangulations for the regular n-gon. Eppstein and Wortman [18] provide a nearly-linear time algorithm to compute a minimum dilation star of a set of points. Giannopoulos, Knauer and Marx [25] prove that, given a set of points, it is NP-hard to compute a minimum dilation tour or a minimum dilation path. Aronov et al. [4] show that, given n points, one can construct a graph with $n - 1 + k$ edges and dilation $O(n/(k + 1))$.

Our algorithm for Problem 5 uses the greedy spanner, which is among the most extensively studied spanner constructions. In general metrics, the greedy $(2k - 1)$-spanner has $O(n^{1+1/k})$ edges [2]. In d-dimensional Euclidean space, the greedy $(1 + \varepsilon)$-spanner has $O(n\varepsilon^{-d+1})$ edges [14, 37]. The sparsity-dilation trade-off is (existentially) optimal in both cases [22, 32].

2 Technical overview

We divide our technical overview into six subsections. In Section 2.1, we summarise the previous algorithm of Gudmundsson and Wong [29]. In Section 2.2, we give an overview of our main result, that is, our $(2\sqrt[r]{2}\, k^{1/r}, 2r)$-bicriteria approximation for all $r \geq 1$. In Section 2.3, we present the main ideas for proving our analysis is tight, assuming the Erdős girth conjecture. In Section 2.4, we summarise our proof that it is W[1]-hard to obtain a $(h(k), 2 - \varepsilon)$-bicriteria approximation, for any computable function h and for any $\varepsilon > 0$. In Section 2.5, we present a $(4k \log n, 1)$-bicriteria approximation. In Section 2.6, we summarise the structure of the remainder of the paper.

2.1 Previous algorithm of [29]

Gudmundsson and Wong's [29] algorithm constructs the greedy spanner with a simple modification. The traditional greedy t-spanner takes as input a set of vertices, i.e. an empty graph, however, the modified greedy t-spanner [29] takes as input a set of vertices and edges, i.e. a non-empty graph.

The greedy t-spanner construction has two steps. First, all edges that are not in the initial graph are sorted by their length. Second, the edges are processed from shortest to longest. A processed edge uv is added if $d_G(u, v) > t \cdot d_M(u, v)$, otherwise the edge uv is not added.

For Problem 3, Gudmundsson and Wong's [29] show, in their main lemma, that if the greedy t-spanner adds at least $k + 1$ edges, then $t \leq (k + 1)\, t^*$. Here, t^* is the minimum dilation if k edges are added to our graph. Using this lemma, they then perform a binary search over a multiplicative $(1 + \delta)$-grid for a $t \in \mathbb{R}$ such that the greedy $(1 + \delta)\, t$-spanner adds at most k edges, but the greedy t-spanner adds at least $k + 1$ edges. Therefore, $(1 + \delta)\, t$ is a $(1+\delta)(k+1)$-approximation of t^*, since we can add k edges to obtain a $(1 + \delta)\, t$-spanner and $(1 + \delta)\, t \leq (1 + \delta)(k + 1)\, t^*$.

Next, we briefly summarise the proof that if $k+1$ edges are added by the greedy algorithm, then $t \leq (k+1)\, t^*$. In Lemma 2 of [29], the authors use the $k+1$ greedy edges to construct a set of $k+1$ vectors in a k-dimensional vector space. They define I to be a linearly dependent subset of the $k + 1$ vectors. In Theorem 5 of [29], the authors use the linear dependence property of I to prove that $t \leq |I| \cdot t^*$. Since $|I| \leq k + 1$, they obtain $t \leq (k + 1)\ t^*$. Unfortunately, the vector space approach of [29] fails extend to Problem 5, if the dilation factors g is sublinear in k, even if the sparsity factor f is allowed to be polynomial in k.

2.2 Greedy bicriteria approximation

Our algorithm is the same as the one in [29]. Our difference lies in our analysis of the greedy t-spanner, in particular, in our main lemma.

For Problem 5, we show, in our main lemma, that if the greedy t-spanner adds at least $fk + 1$ edges, then $t \leq gt^*$. We will specify f and g later. Then, we apply the same binary search procedure to find a $t \in \mathbb{R}$ where the greedy $(1 + \delta)\, t$-spanner adds at most fk edges, but the greedy t-spanner adds at least $fk + 1$ edges. Then $(1 + \delta)\, t$ is an $(f, (1 + \delta)g)$-bicriteria approximation of t^*.

Next, we briefly summarise our new proof that if $fk + 1$ edges are added, then $t \leq gt^*$. In order to extend our analysis to sublinear dilation factors g, we abandon the vector space approach of [29]. Our main idea is to construct an auxiliary graph, which we call the girth graph. The girth graph is an unweighted graph with $2k$ vertices and $fk + 1$ edges. Instead of defining I to be a linearly dependent subset, we define I to be the shortest cycle in the girth graph. We use a classical result in graph theory to choose the values $f = 2\sqrt[r]{2}\ k^{1/r}$ and $g = 2r$, so that $|I| \leq g$. In our final step, we use the cycle property of I to carefully prove $t \leq |I| \cdot t^*$, which implies $t \leq gt^*$.

Putting this all together, we obtain Theorem 6. For a full proof, see Section 3.

▶ **Theorem 6.** *For all $r \geq 1$, there is an $(f, (1 + \delta)g)$-bicriteria approximation for Problem 5 that runs in $O(n^3(\log n + \log \frac{1}{\delta}))$ time, where*

$$f = 2\sqrt[r]{2}\ k^{1/r} \quad \text{and} \quad g = 2r.$$

2.3 Greedy analysis is tight

Our analysis in Theorem 6 is tight. This means one cannot obtain better bounds (up to constant factors) using the greedy spanner. Our proof assumes the Erdős girth conjecture [19].

The girth of an unweighted graph is defined as the number of edges in its shortest cycle. In the proof of Theorem 6, we cite a classical result stating that a graph with n vertices and at least $n^{1+1/r}$ edges has girth at most $2r$. The Erdős girth conjecture states that there are graphs with n vertices, at least $\Omega(n^{1+1/r})$ edges and girth $2r+2$. Several conditional lower bounds have been shown under the Erdős girth conjecture, namely, the sparsity-dilation trade-off of the greedy spanner [2], and the space requirement of approximate distance oracles [39].

We summarise our construction that proves that our analysis is tight. Assuming the Erdős girth conjecture, there exists a graph H with $n = k+1$ vertices, $m = \Omega(n^{1+1/r})$ edges, and girth $2r+2$. We construct a graph G so that if we run the algorithm in Theorem 6, the girth graph of G would be H. We use the properties of H to show that, if there are k edges that can be added to G so that the resulting dilation is t^*, then if we add $m-1$ edges to G using the greedy t-spanner construction, the resulting dilation is at least $(2r+1)\ t^*$.

Putting this all together, we obtain Theorem 7. For a full proof, see .

▶ **Theorem 7.** *For all $r \geq 1$, assuming the Erdős girth conjecture, there is a graph class for which the algorithm in Theorem 6 returns an (f,g)-bicriteria approximation, where*

$$f = \Omega(k^{1/r}) \quad \textit{and} \quad g = 2r+1.$$

2.4 Set cover reduction

Next, we show that the restriction $r \geq 1$ is necessary in Theorem 6. Recall that Theorem 6 states that there is a $(2\sqrt[r]{2}\ k^{1/r}, (1+\delta)\,2r)$-bicriteria approximation algorithm for all $r \geq 1$. We prove that it is W[1]-hard to obtain a $(h(k), 2-\varepsilon)$-bicriteria approximation for any computable function h and for any $\varepsilon > 0$. Our proof is a reduction from set cover.

We summarise our construction of the Problem 5 instance. We show that every set cover instance can be reduced to a Problem 5 instance. We represent each element with a pair of points, and we represent each set with a triple of points. In our Problem 5 instance, we add edges to connect either the pairs or the triples. We show via an exchange argument that we only need to consider adding edges that connect triple. Connecting a triple corresponds to choosing a set, which lowers the dilation of all elements in that set to below the threshold value. Finally, we show that a $(h(k), 2-\varepsilon)$-bicriteria approximation for our Problem 5 would solve set cover within an approximation factor of $h(k)$. However, it is W[1]-hard to obtain an $h(k)$-approximation algorithm for any computatable function h [31].

Putting this all together, we obtain Theorem 8. For a full proof, see the full version of this paper.

▶ **Theorem 8.** *For all $\varepsilon > 0$, assuming FPT $\neq$ W[1], one cannot obtain an (f,g)-bicriteria approximation for Problem 5, where*

$$f = h(k) \quad \textit{and} \quad g = 2-\varepsilon,$$

and $h(\cdot)$ is any computable function.

2.5 Set cover algorithm

Finally, we provide a $(4k \log n, 1)$-bicriteria approximation that runs in $O(n^6 \log n)$ time. Our main idea is to formulate the problem into a set cover instance, and then to apply an $O(\log n)$-approximation algorithm for set cover [13].

We state our algorithm. For each $t \in \mathbb{R}$, we define a set cover instance $\mathcal{I}_t$. We construct the set cover instance $\mathcal{I}_t$ so that its elements are defined by pairs of vertices in $V(G)$, and each set in $\mathcal{I}_t$ is associated with a pair of vertices in $V(G)$. Formally, the elements of $\mathcal{I}_t$ are $\{(u,v) : u,v \in V(G)\}$. The sets of $\mathcal{I}_t$ are $\{S_e : e \in V(G) \times V(G)\}$, where each set S_e contains all pairs (u,v) that have dilation at most t after the edge e is added to G. To formally define S_e, let G_e be the graph if an edge e is added to G, and define $S_e = \{(u,v) : d_{G_e}(u,v) \le t \cdot d_M(u,v)\}$. Now, we can apply the algorithm of [13] on $\mathcal{I}_t$ to obtain a set cover $\mathcal{S}$. If $|\mathcal{S}| < k$, then we can add fewer than k edges to G to reduce its dilation to t, so $t > t^*$. We claim that if $|\mathcal{S}| > 4k^2 \log n$, then $t \le t^*$. Therefore, we perform binary search on t in the same way as [29] to obtain a $(4k \log n, 1)$-bicriteria approximation.

To prove correctness, it remains to show our claim that $|\mathcal{S}| > 4k^2 \log n \implies t \le t^*$. We show the contrapositive. If $t > t^*$, from the definition of t^* there exists k edges that can be added to G to make it a t-spanner. Consider a clique with vertices that are the endpoints of the k edges. Adding these $2k^2$ edges to the graph would make it a t-spanner. Moreover, each t-path, that is, a (u,v)-path with length at most $t \cdot d_M(u,v)$, uses at most one edge in the clique. Therefore, the union of the sets S_e, where e is an edge in the clique, forms a set cover over all pairs of vertices (u,v). The optimal solution of the set cover instance is at most $2k^2$. The algorithm of [13] returns an $O(\log(n^2))$-approximation, since the number of elements and sets in $\mathcal{I}_t$ is $O(n^2)$. Putting this together, our algorithm returns a set cover $\mathcal{S}$ such that $|\mathcal{S}| \le 2k^2 \log(n^2) = 4k^2 \log n$, as required.

Next, we analyse the running time. Computing S_e takes $O(n^3)$ time for each $e \in V(G) \times V(G)$. Therefore, constructing the set cover instance $\mathcal{I}_t$ takes $O(n^5)$ time. The number of elements and the number of sets in $\mathcal{I}_t$ is $O(n^2)$. Therefore, the cubic time algorithm of [13] takes $O(n^6)$ time in total. Finally, performing the $\log n)$ steps in the search brings the total running time to $O(n^6 \log n)$.

Putting this all together, we obtain Theorem 9.

▶ **Theorem 9.** *There is an (f,g)-bicriteria approximation for Problem 5, where*

$$f = 4k \log n \quad \textit{and} \quad g = 1.$$

This completes the overview of the main results of this paper.

2.6 Structure of paper

The structure of the remainder of our paper is summarised in Table 2.

Table 2 References for Theorems 6, 7, 8.

	Reference	Proof
$(2\sqrt[r]{2}\,k^{1/r}, (1+\delta)\,2r)$-bicriteria approximation	Theorem 6	Section 3
Theorem 6 analysis is tight	Theorem 7	Full version
$(h(k), 2-\varepsilon)$-bicriteria approximation is W[1]-hard	Theorem 8	Full version

3 Greedy bicriteria approximation

In this section, we will prove Theorem 6. We restate the theorem for convenience.

▶ **Theorem 6.** *For all $r \geq 1$, there is an $(f, (1+\delta)g)$-bicriteria approximation for Problem 5 that runs in $O(n^3(\log n + \log \frac{1}{\delta}))$ time, where*

$$f = 2\sqrt[r]{2}\, k^{1/r} \quad \text{and} \quad g = 2r.$$

Recall from Section 1 that the vertices and edges of G are $V(G)$ and $E(G)$ respectively. Let $e \in V(G) \times V(G)$ be an edge not necessarily in $E(G)$. Let $d_M(e)$ denote the length of the edge e in the metric space M and let $d_G(e)$ denote the shortest path distance between the endpoints of e in the graph G. Consider a minimum dilation graph G^* after adding an optimal set S^* of k edges to G. Let t^* be the dilation of G^*.

Recall from Section 2 that our approach is to use the greedy t-spanner construction. We formalise the construction in the definition below.

▶ **Definition 10.** *Define $G_0 = G$, and for $i \geq 1$, define $G_i = G_{i-1} \cup \{a_i\}$, where a_i is the shortest edge in $V(G) \times V(G)$ satisfying $d_{G_{i-1}}(a_i) > t \cdot d_M(a_i)$. The process halts if no edge a_i exists.*

We have two cases: either the process halts after adding more than fk edges, or after adding at most fk edges. If more than fk edges are added, we show a dilation bound on t. In particular, Lemma 12 states that if there is an edge a_i satisfying $d_{G_{i-1}}(a_i) > t \cdot d_M(a_i)$ for all $1 \leq i \leq fk+1$, then we have the dilation bound $t \leq gt^*$. We will specify the parameters $f, g \geq 1$ later in this section.

Our approach is to construct an auxilliary graph H, which we will also refer to as the girth graph. Define the vertices of H to be $V(H) = \{v_1, \dots, v_{2k}\}$. Each vertex in $V(H)$ corresponds to an endpoint of an edge in the optimal set of k edges S^*. In particular, let $S^* = \{s_1, \dots, s_k\}$, and let $v_{2i-1}, v_{2i} \in V(H)$ correspond to the endpoints of s_i. Define the edges of H to be $E(H) = \{e_1, \dots, e_{fk+1}\}$. We will describe the procedure for constructing each edge e_i.

Consider the greedy edge a_i, see Figure 1. Define $\delta_{G^*}(a_i)$ to be the shortest path between the endpoints of a_i in G^*, shown in grey in Figure 1. Note that $\delta_{G^*}(a_i)$ denotes a path, whereas $d_{G^*}(a_i)$ denotes a length. Suppose that there are no edges in S^* along the path $\delta_{G^*}(a_i)$, for some $1 \leq i \leq fk+1$. Then,

$$t^* \cdot d_M(a_i) \geq d_{G^*}(a_i) = d_G(a_i) \geq d_{G_{i-1}}(a_i) > t \cdot d_M(a_i),$$

so $t < t^* \leq gt^*$, which would already imply Lemma 12.

Therefore, we can assume that $\delta_{G^*}(a_i)$ contains at least one edge in S^*, for every $i = 1, \dots, fk+1$. Consider the edges $\delta_{G^*}(a_i) \cap S^*$, shown in red in Figure 1. Choose a direction for the path $\delta_{G^*}(a_i)$, sort the list of endpoints of $\delta_{G^*}(a_i) \cap S^*$ with respect to this direction, and let the first and last endpoints in the sorted list be v_j and v_ℓ. Another way to characterise v_j (respectively v_ℓ) is that v_j is an endpoint of an edge in $\delta_{G^*}(a_i) \cap S^*$ so that the shortest path between v_j and one of the endpoints of a_i contains no edges in S^* (respectively the other endpoint of a_i). Finally, we define e_i to be the edge in H connecting v_j to v_ℓ. Note that e_i is an undirected, unweighted edge, shown in orange in Figure 1. This completes the construction of H.

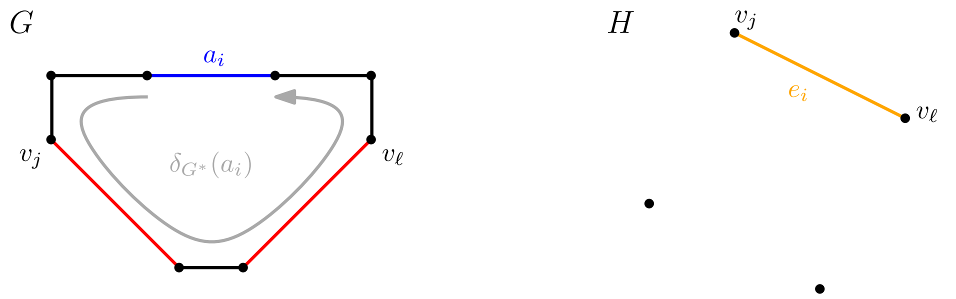

Figure 1 Left: The graph G (black), the greedy edge a_i (blue), the path $\delta_{G^*}(a_i)$ (grey), and the edges $\delta_{G^*}(a_i) \cap S^*$ (red). Right: The girth graph H and the edge e_i (orange).

In Figure 2, we provide a more complete example of a graph G and its girth graph H. The optimal set of $k = 4$ edges is $S^* = \{s_1, s_2, s_3, s_4\}$, which is shown in red. The five greedy edges $\{a_1, a_2, a_3, a_4, a_5\}$ are shown in blue. The first and last endpoints of $\delta_{G^*}(a_1) \cap S^*$ are v_1 and v_3, so $e_1 = v_1v_3$. Similarly, $e_2 = v_3v_5$, $e_3 = v_5v_7$, $e_4 = v_1v_7$ and $e_5 = v_5v_6$.

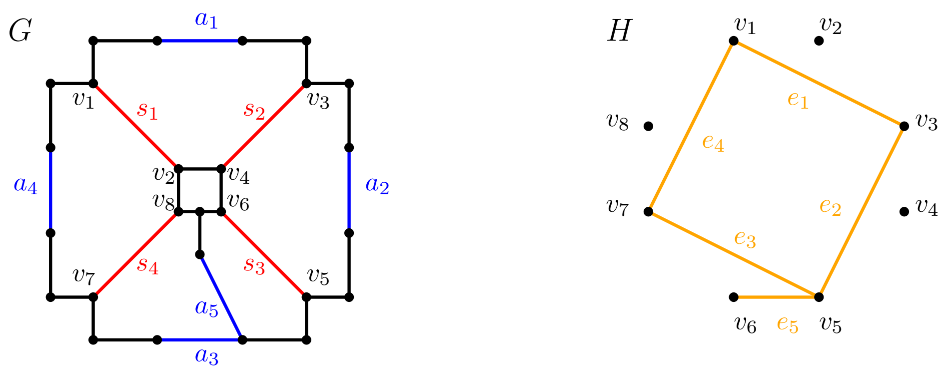

Figure 2 Left: The graph G (black), the optimal edges $s_1, \ldots, s_4$ (red), and the greedy edges $a_1, \ldots, a_5$ (blue). Right: The girth graph H has edges $e_1, \ldots, e_5$ (orange) and a girth of 4.

Next, define J to be the shortest cycle in H, and define $I = \{j : e_j \in J\}$. Therefore, the girth of H is $|J| = |I|$. Note that H has $2k$ vertices and $fk + 1$ edges, so J is guaranteed to exist if $f \geq 2$.

We use a classical result in graph theory to set the parameters f and g.

▶ **Lemma 11.** *A graph with n vertices and at least $n^{1+1/r} + 1$ edges has girth at most $2r$.*

Proof. The lemma is a classical result [9]. Lemma 2 of [33] provides a self-contained proof. ◀

With Lemma 11 in mind, we set $f = 2\sqrt[r]{2}\ k^{1/r}$ and $g = 2r$, where $r \geq 1$. Then, the graph H has $2k$ vertices, $(2k)^{1+1/r} + 1$ edges, and therefore H has girth $|I| \leq g = 2r$. Having defined the girth graph H, the indices I, and the parameters f and g, the next step is to prove Lemma 12.

▶ **Lemma 12.** *If a_j exists for all $j = 1, \ldots, fk + 1$, then $t \leq gt^*$.*

We divide the proof of Lemma 12 into three lemmas. In Lemma 13, we construct a path. In Lemma 14, we lower bound the length of the path. In Lemma 15, we upper bound the length of the path. We start defining the path.

▶ **Lemma 13.** *Let $i = \max I$. There is a path in G between the endpoints of a_i using only edges in*

$$\{G \cap \delta_{G^*}(a_j) : j \in I\} \cup \{a_j : j \in I \setminus \{i\}\}.$$

Proof. Recall that $J = \{e_j : j \in I\}$ is a cycle in H. After removing the edge e_i, there is still a path in $J \subseteq H$ between the endpoints of e_i. Let the vertices along this path be $w_1, \ldots, w_m$, where $e_i = w_1 w_m$, and $w_\ell w_{\ell+1} \in J$ for all $\ell = 1, \ldots, m-1$. In Figure 3, the path $w_1, \ldots, w_m$ is shown in orange. Let the endpoints of a_i be $a_i(0)$ and $a_i(1)$. Our approach is to use the path $w_1, \ldots, w_m \subset H$ to construct a path between $a_i(0)$ and $a_i(1)$ that only uses edges in $\{G \cap \delta_{G^*}(a_j) : j \in I\} \cup \{a_j : j \in I \setminus \{i\}\}$.

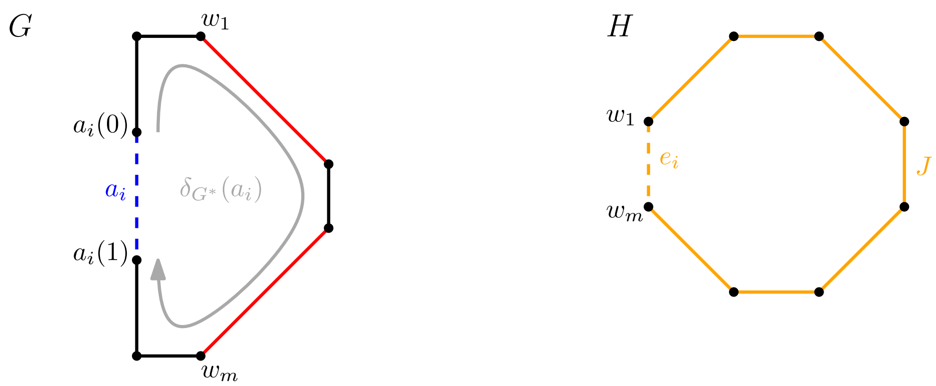

Figure 3 Left: The graph G (black), the greedy edge a_i (blue), the path $\delta_{G^*}(a_i)$ (grey), and the edges $\delta_{G^*}(a_i) \cap S^*$ (red). Right: The girth graph H, the cycle J (orange), and edge e_i (dashed).

First, we consider the edge $e_i = w_1 w_m$. Recall from the definition of $V(H)$ that w_1 and w_m are endpoints of edges in the optimal set S^*. Moreover, from the definition of $e_i \in E(H)$, we know that w_1 and w_m are the first and last endpoints of S^* along the path $\delta_{G^*}(a_i)$. The path $\delta_{G^*}(a_i)$ is shown in grey in Figure 3. The endpoints of $\delta_{G^*}(a_i)$ are $a_i(0)$ and $a_i(1)$. Therefore, the subpath of $\delta_{G^*}(a_i)$ between $a_i(0)$ and w_1 only uses edges in G and no edges in $S^* = G^* \setminus G$. Therefore, the subpath only uses edges in $G \cap \delta_{G^*}(a_i)$. The subpath from $a_i(0)$ to w_1 is shown in black in Figure 3. Similarly, there is a path between w_m and $a_i(1)$ using only edges in $G \cap \delta_{G^*}(a_i)$.

Next, we consider the edge $e_j = w_\ell w_{\ell+1}$, where $1 \le \ell \le m-1$, $j \in I$ and $j < i$. Let the endpoints of a_j be $a_j(0)$ and $a_j(1)$. From the definition of $e_j = w_\ell w_{\ell+1}$, there is a subpath of $\delta_{G^*}(a_j)$ between w_ℓ and $a_j(0)$ that only uses edges in $G \cap \delta_{G^*}(a_j)$. Similarly, there is subpath of $\delta_{G^*}(a_j)$ between $a_j(1)$ and $w_{\ell+1}$ that only uses edges in $G \cap \delta_{G^*}(a_j)$. Therefore, there is a path between w_ℓ and $w_{\ell+1}$ that uses only edges in $\{G \cap \delta_{G^*}(a_j)\} \cup a_j$.

See Figure 4 for an example. Consider the edge $e_1 = w_1 w_2$. There is a path between w_1 and w_2 that only uses edges in $\{G \cap \delta_{G^*}(a_j)\}$, which are black edges, and the blue edge a_1. Similarly arguments apply for $e_2 = w_2 w_3$ and $e_3 = w_3 w_4$.

The final step is to put it all together. There is a path between $a_i(0)$ and w_1 that only uses edges in $G \cap \delta_{G^*}(a_i)$. For $\ell = 1, \ldots, m-1$, there is a path between w_ℓ and $w_{\ell+1}$ that only uses edges in $\{G \cap \delta_{G^*}(a_j)\} \cup a_j$, where $j \in I \setminus \{i\}$. There is a path between w_m and $a_i(1)$ that only uses edges in $G \cap \delta_{G^*}(a_i)$. Therefore, there is a path between $a_i(0)$ and $a_i(1)$ that only uses edges in $\{G \cap \delta_{G^*}(a_j) : j \in I\} \cup \{a_j : j \in I \setminus \{i\}\}$, as required. ◀

In Lemma 14, we show a lower bound on the length of the path in Lemma 13.

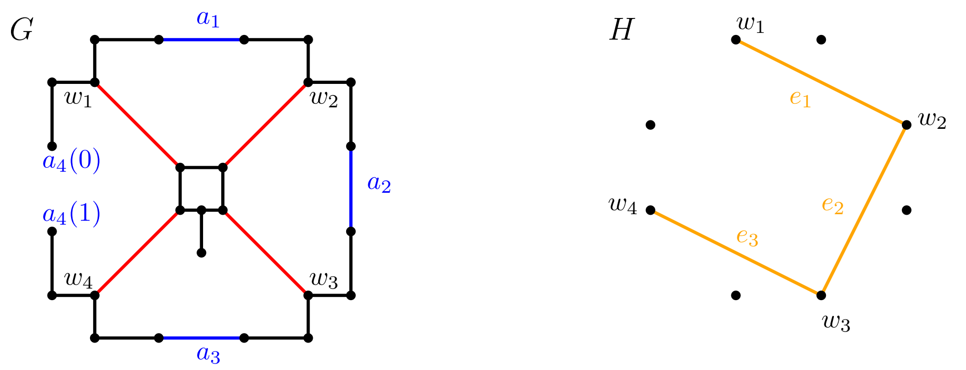

Figure 4 The path w_1, w_2, w_3, w_4 is shown on the right. There is a path between $a_4(0)$ and $a_4(1)$ only using the blue edges a_1, a_2, a_3 and black edges in $\delta_{G^*}(a_1), \delta_{G^*}(a_2), \delta_{G^*}(a_3)$ or $\delta_{G^*}(a_4)$.

▶ **Lemma 14.** *The length of the path in Lemma 13 is at least $t \cdot d_M(a_i)$.*

Proof. From Definition 10, we have $d_{G_{i-1}}(a_i) > t \cdot d_M(a_i)$. Therefore, any path in G_{i-1} between the endpoints of a_i has length at least $t \cdot d_M(a_i)$. It suffices to show that the path is in G_{i-1}. By Lemma 13, all of the edges in the path are in $\{G \cap \delta_{G^*}(a_j) : j \in I\}$ or $\{a_j : j \in I \setminus \{i\}\}$. But $\{G \cap \delta_{G^*}(a_j) : j \in I\} \subseteq G \subseteq G_{i-1}$ and $\{a_j : j \in I \setminus \{i\}\} \subseteq G_{i-1}$. So the path is in G_{i-1} and its length is at least $t \cdot d_M(a_i)$. ◀

In Lemma 15, we upper bound the length of the path in Lemma 13.

▶ **Lemma 15.** *If $t > gt^*$, then the length of the path in Lemma 13 is at most $|I| \cdot t^* \cdot d_M(a_i)$.*

Proof. Given a set of edges E, let total(E) denote the total sum of edge lengths in E. Recall that the path in Lemma 13 only uses edges in $\{G \cap \delta_{G^*}(a_j) : j \in I\} \cup \{a_j : j \in I \setminus \{i\}\}$. A naïve approach to prove the lemma is to bound $\text{total}(\{\delta_{G^*}(a_j) : j \in I\} \cup \{a_j : j \in I \setminus \{i\}\})$. Note that G is removed from the first set of braces. We have

$$\begin{aligned}\text{total}(\{\delta_{G^*}(a_j) : j \in I\}) &\leq \textstyle\sum_{j \in I} t^* \cdot d_M(a_j) \leq |I| \cdot t^* \cdot d_M(a_i),\\ \text{total}(\{a_j : j \in I \setminus \{i\}\}) &= \textstyle\sum_{j \in I \setminus \{i\}} d_M(a_j) \leq (|I| - 1) \cdot d_M(a_i).\end{aligned}$$

Therefore, the total length of the path is at most $(|I| \cdot t^* + |I| - 1) \cdot d_M(a_i) < |I| \cdot (t^* + 1) \cdot d_M(a_i)$. Since $(t^* + 1) \leq 2t^*$, we have proven Lemma 15 up to a factor of 2. This analysis would already yield an $(f, 2g)$-bicriteria approximation. However, to shave off the factor of 2 and obtain a tight analysis, we need a more sophisticated argument.

We strengthen our upper bound by re-introducing G back into the first set of braces, in other words, by bounding $\text{total}(\{G \cap \delta_{G^*}(a_j)\})$. Since $G = G^* \setminus S^*$, we write

$$\text{total}(\{G \cap \delta_{G^*}(a_j)\}) = \text{total}(\{\delta_{G^*}(a_j)\}) - \text{total}(\{S^* \cap \delta_{G^*}(a_j)\}).$$

We have two cases, depending on the size of $\text{total}(\{S^* \cap \delta_{G^*}(a_j)\})$.

Case 1. $\text{total}(\{S^* \cap \delta_{G^*}(a_j)\}) < (1 - \frac{1}{|I|}) \cdot d_M(a_j)$ for some $j \in I$. Then for every $s \in \{S^* \cap \delta_{G^*}(a_j)\}$, it holds that $d_M(s) < d_M(a_i)$. Therefore, $d_{G_{j-1}}(s) \leq t \cdot d_M(s)$, since a_j is the shortest edge in G_{j-1} satisfying $d_{G_{j-1}}(a_j) > t \cdot d_M(a_j)$. Let the endpoints of a_j be $a_j(0)$ and $a_j(1)$. Let the edges of $\delta_{G^*}(a_j) \cap S^*$ be $s_1, \ldots, s_m$, and let the endpoints of s_i be w_{2i-1} and w_{2i}. Assume without loss of generality that the endpoints $w_1, \ldots, w_{2m}$ are in sorted order along the path $\delta_{G^*}(a_j)$. Then,

$$\begin{aligned}
d_{G_{j-1}}(a_j) &\leq d_{G_{j-1}}(a_j(0), w_1) + \textstyle\sum_{i=1}^m d_{G_{j-1}}(w_{2i-1}, w_{2i}) + \sum_{i=1}^{m-1} d_{G_{j-1}}(w_{2i}, w_{2i+1}) \\
&\quad + d_{G_{j-1}}(w_{2m}, a_j(1)) \\
&= d_{G_{j-1}}(a_j(0), w_1) + \textstyle\sum_{i=1}^m d_{G_{j-1}}(s_i) + \sum_{i=1}^{m-1} d_{G_{j-1}}(w_{2i}, w_{2i+1}) \\
&\quad + d_{G_{j-1}}(w_{2m}, a_j(1)) \\
&\leq d_{G_{j-1}}(a_j(0), w_1) + \textstyle\sum_{i=1}^{m-1} d_{G_{j-1}}(w_{2i}, w_{2i+1}) + d_{G_{j-1}}(w_{2m}, a_j(1)) \\
&\quad + \textstyle\sum_{i=1}^m t \cdot d_M(s_i) \\
&\leq d_G(a_j(0), w_1) + \textstyle\sum_{i=1}^{m-1} d_G(w_{2i}, w_{2i+1}) + d_G(w_{2m}, a_j(1)) \\
&\quad + \textstyle\sum_{i=1}^m t \cdot d_M(s_i) \\
&= d_{G^*}(a_j(0), w_1) + \textstyle\sum_{i=1}^{m-1} d_{G^*}(w_{2i}, w_{2i+1}) + d_{G^*}(w_{2m}, a_j(1)) \\
&\quad + \textstyle\sum_{i=1}^m t \cdot d_M(s_i) \\
&< d_{G^*}(a_j) + \textstyle\sum_{i=1}^m t \cdot d_M(s_i),
\end{aligned}$$

where the first line uses the triangle inequality, the second line uses $s_i = w_{2i-1}w_{2i}$, the third line uses $d_{G_{j-1}}(s) \leq t \cdot d_M(s)$, the fourth line uses $G \subset G_{i-1}$, the fifth line uses that all the subpaths no longer use edges in S^*, and the sixth line uses that all edges are a subset of the edges in $\delta^*_G(a_j)$. Therefore,

$$\begin{aligned}
t \cdot d_M(a_j) < d_{G_{j-1}}(a_j) &\leq d_{G^*}(a_j) + \textstyle\sum_{i=1}^m t \cdot d_M(s_i) \\
&\leq t^* \cdot d_M(a_j) + t \cdot \mathrm{total}(\{S^* \cap \delta_{G^*}(a_j)\}) \\
&= t^* \cdot d_M(a_j) + t \cdot (1 - \tfrac{1}{|I|}) \cdot d_M(a_j).
\end{aligned}$$

Simplifying, we get $t < t^* + t - \frac{t}{|I|}$, which implies $t < |I| \cdot t^* = gt^*$. But this contradicts $t > gt^*$ in the lemma statement. Therefore, only Case 2 remains.

Case 2. $\mathrm{total}(\{S^* \cap \delta_{G^*}(a_j)\}) \geq (1 - \frac{1}{|I|}) \cdot d_M(a_j)$ for all $j \in I$. Let L be the length of the path in Lemma 13. Then,

$$\begin{aligned}
L &\leq \mathrm{total}(\{G \cap \delta_{G^*}(a_j) : j \in I\}) + \mathrm{total}(\{a_j : j \in I \setminus \{i\}\}) \\
&= \mathrm{total}(\{\delta_{G^*}(a_j) : j \in I\}) - \mathrm{total}(\{S^* \cap \delta_{G^*}(a_j) : j \in I\}) + \textstyle\sum_{j \in I \setminus \{i\}} d_M(a_j) \\
&\leq \textstyle\sum_{j \in I} t^* \cdot d_M(a_j) - \sum_{j \in I}(1 - \frac{1}{|I|}) \cdot d_M(a_j) + \sum_{j \in I \setminus \{i\}} d_M(a_j) \\
&= \textstyle\sum_{j \in I} t^* \cdot d_M(a_j) - (1 - \frac{1}{|I|}) \cdot d_M(a_i) + \sum_{j \in I \setminus \{i\}} \frac{1}{|I|} \cdot d_M(a_j) \\
&\leq |I| \cdot t^* \cdot d_M(a_i) - (1 - \tfrac{1}{|I|}) \cdot d_M(a_i) + (\tfrac{|I|-1}{|I|}) \cdot d_M(a_i) \\
&= |I| \cdot t^* \cdot d_M(a_i),
\end{aligned}$$

where the first line uses Lemma 13, the second line uses $G = G^* \setminus S^*$, the third line uses the assumption from the case distinction, and fourth, fifth and sixth lines simplify the expression. Therefore, $L \leq |I| \cdot t^* \cdot d_M(a_i)$, as required. ◀

Combining Lemmas 13-15, we obtain Lemma 12, which we will restate for convenience.

▶ **Lemma 12.** *If a_j exists for all $j = 1, \ldots, fk + 1$, then $t \leq gt^*$.*

Finally, we use Lemma 12 to prove Theorem 6. The idea is to combine the sparsity bound in the case where the greedy construction halts after adding at most fk edges, with the dilation bound $t \leq gt^*$ in the case where the greedy construction halts after adding at least $fk + 1$ edges.

▶ **Theorem 6.** *For all $r \geq 1$, there is an $(f, (1+\delta)g)$-bicriteria approximation for Problem 5 that runs in $O(n^3(\log n + \log \frac{1}{\delta}))$ time, where*

$$f = 2\sqrt[r]{2}\ k^{1/r} \quad \text{and} \quad g = 2r.$$

Proof. First, we describe the decision algorithm. Given any $t \in \mathbb{R}$, the decision algorithm is to construct the greedy t-spanner as described in Definition 10. If at most fk edges are added, then we continue searching over dilation values that are less than t. If at least $fk+1$ edges are added, then we continue searching over dilation values that are greater than t.

Second, we perform a binary search to obtain an $(f, (1+\delta)g)$-bicriteria approximation for Problem 5. Given a set of vertices, Gudmundsson and Wong [29] show how to (implicitly) binary search a set of $O(n^4)$ critical values, so that the dilation of any graph with those vertices will be within a factor of $O(n)$ of one of the critical values. We refine the search to a multiplicative $(1+\delta)$-grid. As a result, we obtain a $t \in \mathbb{R}$ where a greedy t-spanner adds at least $fk+1$ edges, but a greedy $(1+\delta)t$-spanner adds at most fk edges. By Lemma 12, we have $t \leq gt^*$, so $(1+\delta)t \leq (1+\delta)gt^*$. The greedy $(1+\delta)t$-spanner adds at most fk edges to the graph and its dilation is at most $(1+\delta)gt^*$, so we have an $(f, (1+\delta)g)$-bicriteria-approximation.

Third, we analyse the running time. The running time of the decision algorithm is $O(n^3)$ [29]. We perform the binary search by first calling the decider $O(\log n)$ times on the critical values, and an additional $O(\log \frac{1}{\delta})$ times on the multiplicative $(1+\delta)$-grid. ◀

4 Conclusion

We provide bicriteria approximation algorithms for the problem of adding k edges to a graph to minimise its dilation. Our main result is a $(2\sqrt[r]{2}\ k^{1/r}, 2r)$-bicriteria approximation for all $r \geq 1$, that runs in $O(n^3 \log n)$ time. Our analysis is tight and it is W[1]-hard to obtain a $(h(k), 2-\varepsilon)$-bicriteria approximation for any computable function h and for any $\varepsilon > 0$. We provide a simple $(4k^2 \log n, 1)$-bicriteria approximation.

We conclude with directions for future work. Problem 1 remains open. In particular, Obstacle 2 asks: is there an $\varepsilon > 0$ for which there is an $O(n^{1-\varepsilon})$-approximation algorithm for the minimum dilation spanning tree problem? The linear approximation factor of Problem 3 cannot be improved unless Obstacle 4 is resolved. An alternative way to circumvent Obstacle 4 is to consider Problem 3 in the special case of unweighted graph metrics. Finally, Problem 5 offers several directions for future work. Can one obtain a trade-off between sparsity and dilation that is better than the greedy t-spanner construction? What is the sparsity-dilation trade-off when $1 < f < 2$? Can the approximation factor or the running time of Theorem 9 be improved?

References

1 Abu Reyan Ahmed, Greg Bodwin, Faryad Darabi Sahneh, Keaton Hamm, Mohammad Javad Latifi Jebelli, Stephen G. Kobourov, and Richard Spence. Graph spanners: A tutorial review. *Computer Science Review*, 37:100253, 2020. doi:10.1016/j.cosrev.2020.100253.

2 Ingo Althöfer, Gautam Das, David P. Dobkin, Deborah Joseph, and José Soares. On sparse spanners of weighted graphs. *Discrete & Computational Geometry*, 9:81–100, 1993. doi:10.1007/BF02189308.

3 Boris Aronov, Kevin Buchin, Maike Buchin, Bart M. P. Jansen, Tom de Jong, Marc J. van Kreveld, Maarten Löffler, Jun Luo, Rodrigo I. Silveira, and Bettina Speckmann. Connect the dot: Computing feed-links for network extension. *Journal of Spatial Information Science*, 3(1):3–31, 2011. doi:10.5311/JOSIS.2011.3.47.

4 Boris Aronov, Mark de Berg, Otfried Cheong, Joachim Gudmundsson, Herman J. Haverkort, Michiel H. M. Smid, and Antoine Vigneron. Sparse geometric graphs with small dilation. *Computational Geometry*, 40(3):207–219, 2008. doi:10.1016/j.comgeo.2007.07.004.

5 Mihai Badoiu, Piotr Indyk, and Anastasios Sidiropoulos. Approximation algorithms for embedding general metrics into trees. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2007. URL: http://dl.acm.org/citation.cfm?id=1283383.1283438.

6 Davide Bilò. Almost optimal algorithms for diameter-optimally augmenting trees. *Theoretical Computer Science*, 931:31–48, 2022. doi:10.1016/j.tcs.2022.07.028.

7 Davide Bilò, Luciano Gualà, Stefano Leucci, and Luca Pepè Sciarria. Finding diameter-reducing shortcuts in trees. In *Proceedings of the 18th Algorithms and Data Structures Symposium*, WADS 2021.

8 Davide Bilò, Luciano Gualà, and Guido Proietti. Improved approximability and non-approximability results for graph diameter decreasing problems. *Theor. Comput. Sci.*, 417:12–22, 2012.

9 Béla Bollobás. *Extremal graph theory.* Courier Corporation, 2004.

10 Aléx F. Brandt, Miguel F. A. de Gaiowski, Pedro J. de Rezende, and Cid C. de Souza. Computing minimum dilation spanning trees in geometric graphs. In *Proceedings of the 21st Annual International Computing and Combinatorics Conference*, COCOON 2015. doi:10.1007/978-3-319-21398-9_24.

11 Leizhen Cai and Derek G. Corneil. Tree spanners. *SIAM Journal on Discrete Mathematics*, 8(3):359–387, 1995. doi:10.1137/S0895480192237403.

12 Otfried Cheong, Herman J. Haverkort, and Mira Lee. Computing a minimum-dilation spanning tree is NP-hard. *Computational Geometry*, 41(3):188–205, 2008. doi:10.1016/j.comgeo.2007.12.001.

13 Vasek Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979. doi:10.1287/moor.4.3.233.

14 Gautam Das, Paul J. Heffernan, and Giri Narasimhan. Optimally sparse spanners in 3-dimensional euclidean space. In *Proceedings of the 9th Annual Symposium on Computational Geometry*, SoCG 1993. doi:10.1145/160985.160998.

15 Erik D. Demaine and Morteza Zadimoghaddam. Minimizing the diameter of a network using shortcut edges. In *Proceedings of the 12th Scandinavian Workshop on Algorithm Theory*, SWAT 2010. doi:10.1007/978-3-642-13731-0_39.

16 Yuval Emek and David Peleg. Approximating minimum max-stretch spanning trees on unweighted graphs. *SIAM Journal of Computing*, 38(5):1761–1781, 2008. doi:10.1137/060666202.

17 David Eppstein. Spanning trees and spanners. In *Handbook of Computational Geometry*, pages 425–461. Elsevier, 2000. doi:10.1016/b978-044482537-7/50010-3.

18 David Eppstein and Kevin A. Wortman. Minimum dilation stars. In *Proceedings of the 21st Annual Symposium on Computational Geometry*, SoCG 2005. doi:10.1145/1064092.1064142.

19 Paul Erdős. On some extremal problems in graph theory. *Israel Journal of Mathematics*, 3:113–116, 1965.

20 Mohammad Farshi, Panos Giannopoulos, and Joachim Gudmundsson. Improving the stretch factor of a geometric network by edge augmentation. *SIAM Journal of Computing*, 38(1):226–240, 2008. doi:10.1137/050635675.

21 Sándor P. Fekete and Jana Kremer. Tree spanners in planar graphs. *Discrete Applied Mathematics*, 108(1-2):85–103, 2001. doi:10.1016/S0166-218X(00)00226-2.

22 Arnold Filtser and Shay Solomon. The greedy spanner is existentially optimal. *SIAM Journal on Computing*, 49(2):429–447, 2020. doi:10.1137/18M1210678.

23 Fedor V. Fomin, Petr A. Golovach, and Erik Jan van Leeuwen. Spanners of bounded degree graphs. *Information Processing Letters*, 111(3):142–144, 2011. doi:10.1016/j.ipl.2010.10.021.

24 Fabrizio Frati, Serge Gaspers, Joachim Gudmundsson, and Luke Mathieson. Augmenting graphs to minimize the diameter. *Algorithmica*, 72(4):995–1010, 2015. doi:10.1007/s00453-014-9886-4.

25 Panos Giannopoulos, Christian Knauer, and Dániel Marx. Minimum-dilation tour (and path) is NP-hard. In *Proceedings of the 23rd European Workshop on Computational Geometry*, EuroCG 2007.

26 Ulrike Große, Christian Knauer, Fabian Stehn, Joachim Gudmundsson, and Michiel H. M. Smid. Fast algorithms for diameter-optimally augmenting paths and trees. *International Journal of Foundations of Computer Science*, 30(2):293–313, 2019. doi:10.1142/S0129054119500060.

27 Joachim Gudmundsson and Yuan Sha. Algorithms for radius-optimally augmenting trees in a metric space. In *Proceedings of the 17th Algorithms and Data Structures Symposium*, WADS 2021. doi:10.1007/978-3-030-83508-8_33.

28 Joachim Gudmundsson, Yuan Sha, and Fan Yao. Augmenting graphs to minimize the radius. In *Proceedings of the 32nd International Symposium on Algorithms and Computation*, ISAAC 2021. doi:10.4230/LIPIcs.ISAAC.2021.45.

29 Joachim Gudmundsson and Sampson Wong. Improving the dilation of a metric graph by adding edges. *ACM Transactions on Algorithms*, 18(3):20:1–20:20, 2022. doi:10.1145/3517807.

30 Christopher Johnson and Haitao Wang. A linear-time algorithm for radius-optimally augmenting paths in a metric space. *Computational Geometry*, 96:101759, 2021. doi:10.1016/j.comgeo.2021.101759.

31 Karthik C. S., Bundit Laekhanukit, and Pasin Manurangsi. On the parameterized complexity of approximating dominating set. *J. ACM*, 66(5):33:1–33:38, 2019. doi:10.1145/3325116.

32 Hung Le and Shay Solomon. Truly optimal Euclidean spanners. In *Proceedings of the 60th Annual Symposium on Foundations of Computer Science*, FOCS 2019. doi:10.1109/FOCS.2019.00069.

33 Stefano Leucci. Graph spanners. *Lecture Notes for the Course on "Distributed and Sequential Graph Algorithms"*, 2019.

34 Lan Lin and Yixun Lin. Optimality computation of the minimum stretch spanning tree problem. *Applied Mathematics and Computation*, 386:125502, 2020. doi:10.1016/j.amc.2020.125502.

35 Jun Luo and Christian Wulff-Nilsen. Computing best and worst shortcuts of graphs embedded in metric spaces. In *Proceedings of the 19th International Symposium on Algorithms and Computation*, ISAAC 2008. doi:10.1007/978-3-540-92182-0_67.

36 Wolfgang Mulzer. Minimum dilation triangulations for the regular n-gon. *Master's thesis Freie Universität Berlin, Germany*, 2004.

37 Giri Narasimhan and Michiel H. M. Smid. *Geometric spanner networks.* Cambridge University Press, 2007.

38 David Peleg. Low stretch spanning trees. In *Proceedings of the 27th International Symposium of Mathematical Foundations of Computer Science*, MFCS 2002. doi:10.1007/3-540-45687-2_5.

39 Mikkel Thorup and Uri Zwick. Approximate distance oracles. *Journal of the ACM*, 52(1):1–24, 2005. doi:10.1145/1044731.1044732.

40 Haitao Wang. An improved algorithm for diameter-optimally augmenting paths in a metric space. *Computational Geometry*, 75:11–21, 2018. doi:10.1016/j.comgeo.2018.06.004.

41 Haitao Wang and Yiming Zhao. A linear-time algorithm for discrete radius optimally augmenting paths in a metric space. *International Journal of Computational Geometry and Applications*, 30(3&4):167–182, 2020. doi:10.1142/S0218195920500089.

42 Haitao Wang and Yiming Zhao. Algorithms for diameters of unicycle graphs and diameter-optimally augmenting trees. *Theoretical Computer Science*, 890:192–209, 2021. doi:10.1016/j.tcs.2021.09.014.

43 Christian Wulff-Nilsen. Computing the dilation of edge-augmented graphs in metric spaces. *Computational Geometry*, 43(2):68–72, 2010. doi:10.1016/j.comgeo.2009.03.008.