Theoretical Computer Science 694 (2017) 66–78

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

On the minimum eccentricity shortest path problem ☆

Feodor F. Dragan *, Arne Leitert

Department of Computer Science, Kent State University, Kent, OH 44240, USA

ARTICLE INFO

Article history:
Received 28 September 2016
Received in revised form 26 May 2017
Accepted 3 July 2017
Available online 17 July 2017
Communicated by G.F. Italiano

Keywords:
Graph algorithms
Approximation algorithms
Minimum eccentricity shortest path
Minimum distortion embedding into the line
k-Dominating set
NP-complete and W[2]-hard problems

ABSTRACT

In this paper, we introduce and investigate the *Minimum Eccentricity Shortest Path (MESP)* problem in unweighted graphs. It asks for a given graph to find a shortest path with minimum eccentricity. Let n and m denote the number of vertices and the number of edges of a given graph. We demonstrate that:

- a minimum eccentricity shortest path plays a crucial role in obtaining the best to date approximation algorithm for a minimum distortion embedding of a graph into the line;
- the MESP problem is NP-hard for planar bipartite graphs with maximum degree 3 and W[2]-hard for general graphs;
- a shortest path of minimum eccentricity k can be computed in $\mathcal{O}(n^{2k+2}m)$ time;
- a 2-approximation, a 3-approximation, and an 8-approximation for the MESP problem can be computed in $\mathcal{O}(n^3)$ time, in $\mathcal{O}(nm)$ time, and in $\mathcal{O}(m)$ time, respectively;
- in a graph with a shortest path of eccentricity k, a k-dominating set can be found in $n^{\mathcal{O}(k)}$ time.

1. Introduction

All graphs occurring in this paper are connected, finite, unweighted, undirected, loopless and without multiple edges. For a graph $G=(V,E)$, we use $n=|V|$ and $m=|E|$ to denote the cardinality of the vertex set and the edge set of G. For a vertex v of G, $N_G(v)=\{u\in V \mid uv\in E\}$ is called the *open neighbourhood*, and $N_G[v]=N_G(v)\cup\{v\}$ the *closed neighbourhood* of v.

The *length* of a path from a vertex v to a vertex u is the number of edges in the path. The *distance* $d_G(u,v)$ of two vertices u and v is the length of a shortest path connecting u and v. The distance between a vertex v and a set $S\subseteq V$ is defined as $d_G(v,S)=\min_{u\in S} d_G(u,v)$. The *eccentricity* $\mathrm{ecc}_G(v)$ of a vertex v is $\max_{u\in V} d_G(u,v)$. For a set $S\subseteq V$, its eccentricity is $\mathrm{ecc}_G(S)=\max_{u\in V} d_G(u,S)$.

In this paper, we investigate the following problem.

Definition 1 *(Minimum eccentricity shortest path problem).* For a given graph G, find a shortest path P such that, for each shortest path Q, $\mathrm{ecc}_G(P)\le \mathrm{ecc}_G(Q)$.

☆ This work was partially supported by the NIH grant R01 GM103309. Results of this paper were partially presented at WADS 2015, Victoria, Canada [11].
* Corresponding author.
E-mail addresses: dragan@cs.kent.edu (F.F. Dragan), aleitert@cs.kent.edu (A. Leitert).

http://dx.doi.org/10.1016/j.tcs.2017.07.004
0304-3975/© 2017 Elsevier B.V. All rights reserved.

Although this problem might be of an independent interest (it may arise in determining a "most accessible" speedy linear route in a network and can find applications in communication networks, transportation planning, water resource management and fluid transportation), our interest in this problem stems from the role it plays in obtaining the best to date approximation algorithm for a minimum distortion embedding of a graph into the line. In Section 2, we demonstrate that every graph G with a shortest path of eccentricity k admits an embedding f of G into the line with distortion at most $(8k+2)\,\mathrm{ld}(G)$, where $\mathrm{ld}(G)$ is the minimum line-distortion of G. Furthermore, if a shortest path of G of eccentricity k is given in advance, then such an embedding f can be found in linear time.

This fact augments the importance of investigating the *Minimum Eccentricity Shortest Path* problem (MESP problem) in graphs. Fast algorithms for it will imply fast approximation algorithms for the minimum line distortion problem. Existence of low eccentricity shortest paths in special graph classes will imply low approximation bounds for those classes. For example, all AT-free graphs (and hence all interval, permutation, cocomparability graphs) enjoy a shortest path of eccentricity at most 1 [4], all convex bipartite graphs enjoy a shortest path of eccentricity at most 2 [9].

We prove also that for every graph G with $\mathrm{ld}(G)=\lambda$, the minimum eccentricity of a shortest path of G is at most $\lfloor \frac{\lambda}{2} \rfloor$. Hence, one gets an efficient embedding of G into the line with distortion at most $\mathcal{O}(\lambda^2)$.

In Section 3, we show that the MESP problem is NP-hard for bipartite planar graphs with maximum degree 3, W[2]-hard on general graphs, and that a shortest path of minimum eccentricity k, in general graphs, can be computed in $\mathcal{O}(n^{2k+2}m)$ time. In Section 4, we design, for the MESP problem on general graphs, a 2-approximation algorithm that runs in $\mathcal{O}(n^3)$ time, a 3-approximation algorithm that runs in $\mathcal{O}(nm)$ time and an 8-approximation algorithm that runs in linear time. In Section 5, we will show that in a graph with a shortest path of eccentricity k a k-dominating set can be found in $n^{\mathcal{O}(k)}$ time.

Note that our Minimum Eccentricity Shortest Path problem is close but different from the *Central Path* problem in graphs introduced in [21]. It asks for a given graph G to find a path P (not necessarily shortest) such that any other path of G has eccentricity at least $ecc_G(P)$. The Central Path problem generalizes the Hamiltonian Path problem and therefore is NP-hard even for chordal graphs [20]. Our problem is polynomial time solvable for chordal graphs [10].

In what follows, we will need the following additional notions and notations.

The *diameter* of a graph G is $\mathrm{diam}(G) = \max_{u,v\in V} d_G(u,v)$. The diameter $\mathrm{diam}_G(S)$ of a set $S \subseteq V$ is defined as $\max_{u,v\in S} d_G(u,v)$. A pair of vertices x, y of G is called a *diametral pair* if $d_G(u,v) = \mathrm{diam}(G)$. In this case, every shortest path connecting x and y is called a *diametral path*.

A path P of a graph G is called a *k-dominating path* of G if $ecc_G(P) \le k$. In this case, we say also that P *k-dominates* each vertex of G. A pair of vertices x, y of G is called a *k-dominating pair* if every path connecting x and y has eccentricity at most k.

For a vertex s, let $L_i^{(s)} = \{ v \mid d_G(s,v) = i \}$ denote the vertices with distance i from s. We will also refer to $L_i^{(s)}$ as the i-th layer.

2. Motivation through the line-distortion of a graph

Computing a minimum distortion embedding of a given n-vertex graph G into the line ℓ was recently identified as a fundamental algorithmic problem with important applications in various areas of computer science, like computer vision [22], as well as in computational chemistry and biology (see [16,17]). The *minimum line distortion* problem asks, for a given graph $G=(V,E)$, to find a mapping f of vertices V of G into points of ℓ with minimum number λ such that $d_G(x,y) \le |f(x)-f(y)| \le \lambda\, d_G(x,y)$ for every $x, y \in V$. The parameter λ is called the *minimum line-distortion* of G and denoted by $\mathrm{ld}(G)$. The embedding f is called *non-contractive* since $d_G(x,y) \le |f(x)-f(y)|$ for every $x, y \in V$.

In [2], Bădoiu et al. showed that this problem is hard to approximate within some constant factor. They gave an exponential-time exact algorithm and a polynomial-time $\mathcal{O}(n^{1/2})$-approximation algorithm for arbitrary unweighted input graphs, along with a polynomial-time $\mathcal{O}(n^{1/3})$-approximation algorithm for unweighted trees. In fact, their algorithms achieve line-distortion $O(\lambda^2)$ for general (unweighted) graphs, and line-distortion $O(\lambda^{3/2})$ for unweighted trees, where λ is the minimum line-distortion. In another paper [1], Bădoiu et al. showed that the problem is hard to approximate by a factor $\mathcal{O}(n^{1/12})$, even for weighted trees. They also gave a better polynomial-time approximation algorithm for general weighted graphs, along with a polynomial-time algorithm that approximates the minimum line-distortion λ embedding of a weighted tree by a factor that is polynomial in λ.

Fast exponential-time exact algorithms for computing the line-distortion of a graph were proposed in [5,12,13]. Fomin et al. [13] showed that a minimum distortion embedding of an unweighted graph into the line can be found in time $5^{n+o(n)}$. Fellows et al. [12] gave an $\mathcal{O}(n\lambda^4(2\lambda+1)^{2\lambda})$ time algorithm that for an unweighted graph G and integer λ either constructs an embedding of G into the line with distortion at most λ, or concludes that no such embedding exists. They extended their approach also to weighted graphs obtaining an $\mathcal{O}(n\lambda^{4W}(2\lambda+1)^{2\lambda W})$ time algorithm, where W is the largest edge weight. Thus, the problem of minimum distortion embedding of a given n-vertex graph G into the line ℓ is Fixed Parameter Tractable. Recently, Cygan and Pilipczuk [5] enhanced the $5^{n+o(n)}$ time and $\mathcal{O}^*(2^n)$ space algorithm by Fomin et al. [13] to an algorithm working in $\mathcal{O}(4.383^n)$ time and space.

Heggernes et al. [14,15] initiated the study of minimum distortion embeddings into the line of specific graph classes other than trees. In particular, they gave polynomial-time algorithms for the problem on bipartite permutation graphs and on threshold graphs [15]. Furthermore, in [14], Heggernes et al. showed that the problem of computing a minimum distortion embedding of a given graph into the line remains NP-hard even when the input graph is restricted to a bipartite,

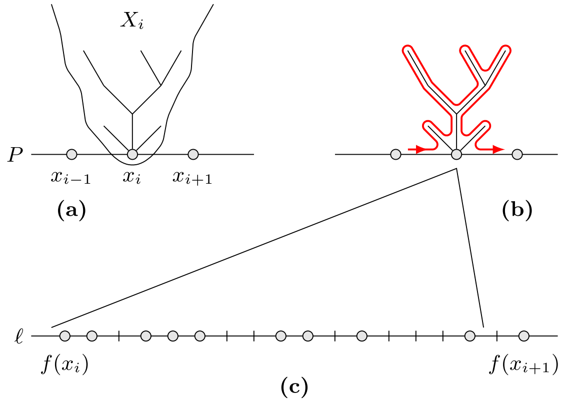

Fig. 1. Illustration to the proof of Theorem 3. (a) The decomposition $\{X_0, X_1, \ldots, X_q\}$ of the vertex set V of G. (b) The upper part of the twice around tour. (c) An embedding f obtained from following the upper part of the twice around tour.

cobipartite, or split graph, implying that it is NP-hard also on chordal, cocomparability, and AT-free graphs. They also gave polynomial-time constant-factor approximation algorithms for split and cocomparability graphs.

Recently, in [9], a more general result for unweighted graphs was proven: for every class of graphs with path-length bounded by a constant, there exists an efficient constant-factor approximation algorithm for the minimum line-distortion problem. As a byproduct, an efficient algorithm was obtained which for each unweighted graph G with $\mathrm{ld}(G) = \lambda$ constructs an embedding with distortion at most $\mathcal{O}(\lambda^2)$. Furthermore, for AT-free graphs, a linear time 8-approximation algorithm for the minimum line-distortion problem was obtained. Note that AT-free graphs contain all cocomparability graphs and hence all interval, permutation and trapezoid graphs.

In this section, we simplify and improve the result of [9]. We show that a minimum eccentricity shortest path plays a crucial role in obtaining the best to date approximation algorithm for the minimum line-distortion problem.

We will need the following simple “Local Density” Lemma which generalizes a Local Density Lemma from [2].

Lemma 2. *For every vertex set $S \subseteq V$ of an arbitrary graph $G = (V, E)$,*

$$|S| - 1 \leq \mathrm{diam}_G(S)\,\mathrm{ld}(G).$$

Proof. Consider an embedding f^* of G into the line ℓ with distortion $\mathrm{ld}(G)$. Let a and b be the leftmost and the rightmost, respectively, in ℓ vertices of S, i.e., $f^*(a) = \min\{f^*(v) \mid v \in S\}$ and $f^*(b) = \max\{f^*(v) \mid v \in S\}$. Consider a shortest path P in G between a and b. Since, for each edge xy of G (and hence of P), $|f^*(x) - f^*(y)| \leq \mathrm{ld}(G)$ holds, we get $f^*(b) - f^*(a) \leq d_G(a, b)\,\mathrm{ld}(G) \leq \mathrm{diam}_G(S)\,\mathrm{ld}(G)$. On the other hand, since all vertices of S are mapped to points of ℓ between $f^*(a)$ and $f^*(b)$, we have $f^*(b) - f^*(a) \geq |S| - 1$. □

The main result of this section is the following.

Theorem 3. *Every graph G with a shortest path of eccentricity k admits an embedding f of G into the line with distortion at most $(8k + 2)\,\mathrm{ld}(G)$. If a shortest path of G of eccentricity k is given in advance, then such an embedding f can be found in linear time.*

Proof. Our embedding is based on the idea from [2]. Let $P = (x_0, x_1, \ldots, x_i, \ldots, x_j, \ldots, x_q)$ be a shortest path of G of eccentricity k. Build a BFS(P, G)-tree T of G (i.e., a Breadth-First-Search tree of G started at path P). Denote by $\{X_0, X_1, \ldots, X_q\}$ the decomposition of the vertex set V of G obtained from T by removing the edges of P. That is, X_i is the vertex set of a subtree (branch) of T growing from vertex x_i of P. See Fig. 1(a) for an illustration. Since eccentricity of P is k, we have $d_G(v, x_i) \leq k$ for every $i \in \{1, \ldots, q\}$ and every $v \in X_i$.

We define an embedding f of G into the line ℓ by performing a preorder traversal of the vertices of T starting at vertex x_0 and visiting first vertices of X_i and then vertices of X_{i+1}, $i = 0, \ldots, q-1$. We place vertices of G on the line in that order, and also, for each $i \in \{0, \ldots, q-1\}$, we leave a space of length $d_T(v_i, v_{i+1})$ between any two vertices v_i and v_{i+1} placed next to each other (this can be done during the preorder traversal). Alternatively, f can be defined by creating a twice around tour of the tree T, which visits vertices of X_i prior to vertices of X_{i+1}, $i = 0, \ldots, q-1$, and then returns to x_0 from x_q along edges of P. Following vertices of T from x_0 to x_q as shown in Fig. 1(b) (i.e., using upper part of the twice around tour), $f(v)$ can be defined as the first appearance of vertex v in that subtour (see Fig. 1(c)).

We claim that f is a (non-contractive) embedding with distortion at most $(8k + 2)\,\mathrm{ld}(G)$. It is sufficient to show that $d_G(x, y) \leq |f(x) - f(y)|$ for every two vertices of G that are placed by f next to each other in ℓ and that $|f(v) - f(u)| \leq (8k + 2)\,\mathrm{ld}(G)$ for every edge uv of G (see, e.g., [2,15]).

Let x, y be arbitrary two vertices of G that are placed by f next to each other in ℓ. By construction, we know that $|f(x) - f(y)| = d_T(x, y)$. Since $d_G(x, y) \le d_T(x, y)$, we get also $d_G(x, y) \le |f(x) - f(y)|$, i.e., f is non-contractive.

Consider now an arbitrary edge uv of G and assume $u \in X_i$ and $v \in X_j$ $(i \le j)$. Note that $d_P(x_i, x_j) = j - i \le 2k + 1$, since P is a shortest path of G and $d_P(x_i, x_j) = d_G(x_i, x_j) \le d_G(x_i, u) + 1 + d_G(x_j, v) \le 2k + 1$. Set $S = \bigcup_{h=i}^{j} X_h$. For any two vertices $x, y \in S$, $d_G(x, y) \le d_G(x, P) + 2k + 1 + d_G(y, P) \le k + 2k + 1 + k = 4k + 1$ holds. Hence, $\text{diam}_G(S) \le 4k + 1$. Consider subtree T_S of T induced by S. Clearly, T_S is connected and has $|S| - 1$ edges. Therefore, $f(v) - f(u) \le 2(|S| - 1)$ since each edge of T_S contributes to $f(v) - f(u)$ at most 2 units. Now, by Lemma 2, $f(v) - f(u) \le 2(|S| - 1) \le 2\,\text{diam}_G(S)\,\text{ld}(G) \le (8k + 2)\,\text{ld}(G)$. □

Recall that a pair x, y of vertices of a graph G forms a k-dominating pair if every path connecting x and y in G has eccentricity at most k. It turns out that the following result is true.

Proposition 4. *If the minimum line-distortion of a graph G is λ, then G has a $\lfloor \frac{\lambda}{2} \rfloor$-dominating pair.*

Proof. Let f be an optimal line embedding for G. This embedding has a first vertex v_1 and a last vertex v_n. Let u be an arbitrary vertex and P an arbitrary path from v_1 to v_n. If u is not on this path, there is an edge $v_i v_j$ of P with $f(v_i) < f(u) < f(v_j)$. Without loss of generality, we can say that $f(u) - f(v_i) \le \lfloor (f(v_j) - f(v_i))/2 \rfloor \le \lfloor \frac{\lambda}{2} \rfloor$. Thus, each vertex is $\lfloor \frac{\lambda}{2} \rfloor$-dominated by each path from v_1 to v_n, i.e., v_1, v_n is a $\lfloor \frac{\lambda}{2} \rfloor$-dominating pair. □

Corollary 5. *For every graph G with $\text{ld}(G) = \lambda$, the minimum eccentricity of a shortest path of G is at most $\lfloor \frac{\lambda}{2} \rfloor$.*

Theorem 3 and Corollary 5 stress the importance of investigating the Minimum Eccentricity Shortest Path problem in graphs. As we will show later, although the MESP problem is NP-hard on general graphs, there are much better (than for the minimum line distortion problem) approximation algorithms for it. We design for the MESP problem on general graphs a 2-approximation algorithm that runs in $\mathcal{O}(n^3)$ time, a 3-approximation algorithm that runs in $\mathcal{O}(nm)$ time and an 8-approximation algorithm that runs in linear time.

Combining Theorem 3 and Corollary 5 with those approximation results, we reproduce a result of [2] and [9].

Corollary 6 *([2,9]). For every graph G with $\text{ld}(G) = \lambda$, an embedding into the line with distortion at most $\mathcal{O}(\lambda^2)$ can be found in polynomial time.*

It should be noted that, since the ratio between the minimum eccentricity of a shortest path and the line-distortion of a graph can be very large (e.g., the line distortion of a complete graph on n vertices is $n - 1$, whereas each shortest path of such a graph has eccentricity 1), the result in Theorem 3 seems to be stronger. Furthermore, one version of our algorithm (that uses an 8-approximation algorithm for the MESP problem) runs in total linear time.

3. Finding an optimal solution

In this section, we will show that finding a minimum eccentricity shortest path is NP-hard and W[2]-hard. Additionally, we will present a pseudo-polynomial time algorithm for finding such a path.

To show NP-hardness, we define the decision version of this problem (k-ESP) as follows: Given a graph G and an integer k, does G contain a shortest path P with eccentricity at most k?

Theorem 7. *The decision version of the Minimum Eccentricity Shortest Path problem is NP-complete.*

Proof. To prove Theorem 7, we use a version of 3-SAT called *Planar Monotone 3-SAT* which was introduced by de Berg and Khosravi in [6]. Consider an instance of 3-SAT given in CNF with the variables $\mathcal{P} = \{p_1, \dots, p_n\}$ and the clauses $\mathcal{C} = \{c_1, \dots, c_m\}$. A clauses is called *positive* if it consists only of positive literals (i.e., $p_a \vee p_b \vee p_c$) and is called *negative* if it consists only of negative literals (i.e., $\neg p_a \vee \neg p_b \vee \neg p_c$). Consider the bipartite graph $\mathcal{G} = (\mathcal{P}, \mathcal{C}, \mathcal{E})$ where $p_i c_j \in \mathcal{E}$ if and only if c_j contains p_i or $\neg p_i$. An instance of 3-SAT is *planar monotone* if each clause is either positive or negative and there is a planar embedding for $\mathcal{G}$ such that all variables are on a (horizontal) line L, all positive clauses are above L, all negative clauses are below L, and no edge is crossing L. Planar Monotone 3-SAT is NP-complete [6].

Now, assume that we are given an instance $\mathcal{I}$ of Planar Monotone 3-SAT with the variables $\mathcal{P} = \{p_1, \dots, p_n\}$ and the clauses $\mathcal{C} = \{c_1, \dots, c_m\}$. Also, let $k = \max\{n, m\}$. We create a graph G as shown in Fig. 2. For each variable p_i create two vertices, one representing p_i and one representing $\neg p_i$. Create one vertex c_i for every clause c_i. Additionally, create two vertices u_0, u_n and, for each i with $0 \le i \le n$, a vertex v_i. Connect each variable vertex p_i and $\neg p_i$ with v_{i-1} and v_i directly with an edge. Connect each clause with the literals contained in it with a path of length k. Also connect v_0 with u_0 and v_n with u_n with a path of length k.

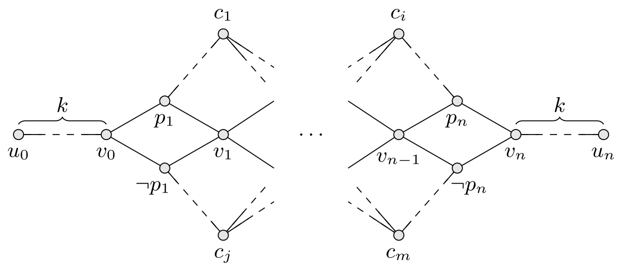

Fig. 2. Reduction from Planar Monotone 3-SAT to k-ESP. Illustration to the proof of Theorem 7.

Recall that, by the definition of $\mathcal{I}$, the corresponding bipartite graph $\mathcal{G}$ has a planar embedding where all variables are on a line. Therefore, we can clearly achieve a planar embedding for G when placing its vertices as shown in Fig. 2.

Note that every shortest path in G not containing v_0 and v_n has an eccentricity larger than k. Also, a shortest path from v_0 to v_n has length $2n$ ($d_G(v_{i-1}, v_i) = 2$, passing through p_i or $\neg p_i$). Since $k \geq n$, no shortest path from v_0 to v_n is passing through a vertex c_i; in this case the minimal length would be $2k + 2$. Additionally, note that, for all vertices in G except the vertices which represent clauses, the distance to a vertex v_i with $0 \leq i \leq n$ is at most k.

We will now show that $\mathcal{I}$ is satisfiable if and only if G has a shortest path with eccentricity k.

First, assume $\mathcal{I}$ is satisfiable. Let $f : \mathcal{P} \to \{T, F\}$ be a satisfying assignment for the variables. As shortest path P we choose a shortest path from v_0 to v_n. Thus, we have to chose between p_i and $\neg p_i$. We will chose p_i if and only if $f(p_i) = T$. Because $\mathcal{I}$ is satisfiable, there is a p_i for each c_j such that either $f(p_i) = T$ and $d_G(c_j, p_i) = k$, or $f(p_i) = F$ and $d_G(c_j, \neg p_i) = k$. Thus, P has eccentricity k.

Next, consider a shortest path P in G of eccentricity k. As mentioned above, P contains either p_i or $\neg p_i$. Now, we define $f : \mathcal{P} \to \{T, F\}$ as follows:

$$f(p_i) = \begin{cases} T & \text{if } p_i \in P, \\ F & \text{else, i.e. } \neg p_i \in P. \end{cases}$$

Because P has eccentricity k and only vertices representing a literal in the clause c_j are at distance k to vertex c_j, f is a satisfying assignment for $\mathcal{I}$. $\square$

While the reduction works in principle for any version of SAT (given as CNF), choosing Planar Monotone 3-SAT allows to construct a planar graph G.

Note that the created graph is bipartite. Set the colour of each vertex v_i to black and of each p_j and $\neg p_j$ to white. For some vertex x on the shortest path from c_i to p_j (or $\neg p_j$), set the colour of x based on its distance to p_j (or $\neg p_j$), i.e., x is white if $d_G(x, p_i)$ is even and black otherwise.

Additionally, V.B. Le[1] pointed out that, by slightly modifying the created graph as follows, it can be shown that the problem remains NP-complete even if the graph has the maximum vertex-degree 3. First, increase k to $k = \max\{2n - 1, m\}$ and update all distances in the graph accordingly. Therefore, $2k + 2 > 4n - 2$. Then, replace each vertex v_i where $1 \leq i \leq n - 1$ with three vertices v_i^-, v_i', and v_i^+ such that $N_G(v_i^-) = \{p_i, \neg p_i, v_i'\}$, $N_G(v_i') = \{v_i^-, v_i^+\}$, and $N_G(v_i^+) = \{p_{i+1}, \neg p_{i+1}, v_i'\}$. Note that a path from v_0 to v_n which, for all i, passes through p_i or $\neg p_i$ has length $4n - 2$. This is still a shortest path, because each path from v_0 to v_n passing through some c_i has length $2k + 2 > 4n - 2$. Also, since $d_G(p_i, p_{i+1}) = 4$, the graph remains bipartite. Next, to limit the degree of a vertex p_i (or $\neg p_i$), instead of connecting it directly to all clauses containing it, make p_i adjacent to the root of a binary tree T_i with height $\lceil \log_2 k \rceil$. Then, connect each clause containing p_i to a leaf of T_i using a path with length $k - \lceil \log_2 k \rceil - 1$ and, last, remove unused branches of T_i. Because this does not effect planarity or colouring, we get:

Corollary 8. *The decision version of the MESP problem remains NP-complete when restricted to planar bipartite graphs with the maximum vertex-degree* 3.

We can slightly modify the MESP problem such that a start vertex s and an end vertex t of the path are given. This is, for a given a graph G and two vertices s and t, find a shortest path P from s to t such that, for each shortest path Q from s to t, $\text{ecc}_G(P) \leq \text{ecc}_G(Q)$. We call this the (s, t)-MESP problem. From the reduction above, it follows that the decision version of this problem is NP-complete, too.

Corollary 9. *The decision version of the* (s, t)*-MESP problem is NP-complete.*

[1] University of Rostock, Germany.

Note that the factor k in the reduction above depends on the input size. In [18], it was already mentioned that, for $k = 1$, the problem can be solved in $\mathcal{O}(n^3 m)$ time by modifying an algorithm given in [7]. There, the problem was called *Dominating Shortest Path* problem.

The next algorithm will show that the k-ESP problem remains polynomial for each fixed k. Our algorithm is a generalization of the algorithm mentioned in [18]. It is based on Lemma 10 below. Informally, Lemma 10 states that, if a graph has a shortest path P with eccentricity k starting at s, each layer $L_i^{(s)}$ is dominated by a subpath of P of length at most $2k$.

Lemma 10. *Let $P = \{s = v_0, v_1, \ldots, v_l\}$ be a shortest path with eccentricity k, $v_i \in L_i^{(s)}$, and $P_{i,k} = \{v_{\max\{0,i-k\}}, \ldots, v_{\min\{i+k,l\}}\}$. Then, $L_i^{(s)} \subseteq N_G^k[P_{i,k}]$.*

Proof. Assume there is a vertex $u \in L_i^{(s)} \setminus N_G^k[P_{i,k}]$. Consider any vertex $v_j \in P \setminus P_{i,k}$. By the definition of $P_{i,k}$ it follows that $|i - j| > k$. Thus, because $u \in L_i^{(s)}$ and $v_j \in L_j^{(s)}$, $d_G(v_j, u) \geq |i - j| > k$. This contradicts with P having eccentricity k. □

For Algorithm 1 below, we say a shortest path $\tau = \{v_{i-k}, \ldots, v_j\}$ with $i \leq j \leq i + k$ is a *layer-dominating path for a layer $L_i^{(s)}$* if

- $v_l \in L_l^{(s)}$ for $i - k \leq l \leq j$,
- $j < i + k$ implies that there is no edge $v_j w \in E$ with $w \in L_{j+1}^{(s)}$, and
- $N_G^k[\tau] \supseteq L_i^{(s)}$ with $N_G^k[\tau] := \bigcup_{l=i-k}^{j} N_G^k[v_l]$.

We say that a layer-dominating path $\sigma = \{v_{i-k}, \ldots, v_j\}$ for layer $L_i^{(s)}$ is *compatible* with a layer-dominating path $\tau = \{u_{i+1-k}, \ldots, u_{j'}\}$ for layer $L_{i+1}^{(s)}$ if $j' - j \in \{0, 1\}$ and $v_l = u_l$ for $i + 1 - k \leq l \leq j$. This is, σ and τ share a path of length $j - i - 1 + k$.

Algorithm 1: Determines if there is a shortest path of eccentricity at most k starting at a given vertex s.

Input: A graph $G = (V, E)$ and a positive integer k.
Output: A shortest path with eccentricity at most k if existent in G.

1 Calculate the layers $L_i^{(s)} = \{v \in V \mid d_G(s, v) = i\}$ with $0 \leq i \leq ecc_G(s)$.
2 **if** $ecc_G(s) \leq 2k$ **then**
3   For each shortest path P from s, determine if $ecc_G(P) \leq k$. In this case, **return** P. If there is no such P, then G does not contain a shortest path of eccentricity at most k starting at s.
4 **for** $i = k$ **to** $ecc_G(s) - k$ **do**
5   Create an empty vertex set V'_i.
6   **foreach** *layer-dominating path τ for layer $L_i^{(s)}$* **do**
7   Add a vertex v_τ, representing the path τ, to V'_i.
8 **foreach** $v_\tau \in V'_{ecc_G(s)-k}$ **do**
9   If $N_G^k[\tau] \not\supseteq \bigcup_{j=ecc_G(s)-k}^{ecc_G(s)} L_j^{(s)}$, remove v_τ from $V'_{ecc_G(s)-k}$.
10 Create a graph $G' = (V', E')$ with $V' = V'_k \cup \ldots \cup V'_{ecc_G(s)-k}$ and $E' = \{v_\sigma v_\tau \mid \sigma$ is compatible with $\tau\}$.
11 G contains a shortest path of eccentricity at most k starting at s if and only if G' contains a path from a vertex $v_\sigma \in V'_k$ to a vertex $v_\tau \in V'_{ecc_G(s)-k}$.

Theorem 11. *Algorithm 1 determines if there is a shortest path of eccentricity at most k starting from a given vertex s in $\mathcal{O}(n^{2k+1} m)$ time.*

Proof *(Correctness).* To show the correctness of the algorithm, we need to show that line 11 is correct. Without loss of generality, we can assume that $ecc_G(s) > 2k$. Otherwise, the algorithm would have stopped in line 3.

First, assume that there is a shortest path $P = \{s = u_0, \ldots, u_l\}$ of length l in G with $ecc_G(P) \leq k$. Note that $ecc_G(s) - k \leq l \leq ecc_G(s)$. Then, by Lemma 10, each subpath $\tau = \{u_{i-k}, \ldots, u_j\}$ ($k \leq i \leq ecc_G(s) - k$, $j = \min\{l, i + k\}$) is a layer-dominating path for layer $L_i^{(s)}$. Additionally, if $j = l$, then $N_G^k[\tau] \supseteq \bigcup_{j=ecc_G(s)-k}^{ecc_G(s)} L_j^{(s)}$. Thus, the algorithm creates a vertex $v_\tau \in V'_i$ in line 7 which represents a subpath of P for each i with $k \leq i \leq ecc_G(s) - k$. If $v_\tau \in V'_i$ and $v_\sigma \in V'_{i+1}$ represent subpaths of P, v_τ and v_σ are adjacent in G' because τ and σ are compatible. Therefore, there is a path in G' from a vertex in V'_k to a vertex in $V'_{ecc_G(s)-k}$.

Next, assume that G' contains a path P' from a vertex $u \in V'_k$ to a vertex $v \in V'_{ecc_G(s)-k}$. Each vertex $v_\sigma \in V'_i \cap P'$ represents a layer-dominating path for layer $L_i^{(s)}$ in G. By definition of layer-dominating paths, if $v_\sigma \in V'_i$ is adjacent to $v_\tau \in V'_{i+1}$, the paths σ and τ in G (of length $2k$) can be combined to a longer path (of length $2k + 1$). If τ has length less than $2k$, it is a subpath of σ. Thus, P' represents a path P in G from s to a vertex $w \in L_q^{(s)}$ with $ecc_G(s) - k \leq q \leq ecc_G(s)$.

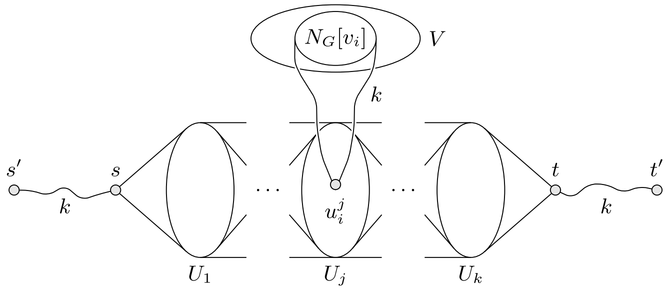

Fig. 3. Reduction from Dominating Set to k-ESP. Illustration to the proof of Theorem 13.

Each vertex $v_\tau \in V'_i \cap P'$ represents a layer-dominating path τ for layer $L_i^{(s)}$. Because of line 9, $v_\tau \in V'_{\text{ecc}_G(s)-k}$ implies $N_G^k[\tau] \supseteq \bigcup_{j=\text{ecc}_G(s)-k}^{k} L_j^{(s)}$. Thus, P is a shortest path starting from s with $\text{ecc}_G(P) \leq k$. □

Proof *(Complexity).* If $\text{ecc}_G(s) \leq 2k$, the algorithm stops after line 3. In this case there are at most $\mathcal{O}(n^{2k})$ shortest paths starting from s. Thus, finding a shortest path with eccentricity k can be done in $\mathcal{O}(n^{2k}m)$ time by deciding in $\mathcal{O}(m)$ time if a path has eccentricity k.

Next, assume $\text{ecc}_G(s) > 2k$. The graph can only contain $\mathcal{O}(n^{2k+1})$ layer-dominating paths because each such path has at most $2k+1$ vertices in it. Therefore, creating the vertices of G' (lines 4–9) can be done in $\mathcal{O}(n^{2k+1}m)$ time.

Store the found layer-dominating paths in a forest structure $\mathcal{T}$ as follows. For each vertex v of G, $\mathcal{T}$ contains a tree T_v rooted at v of depth at most $2k$. This tree T_v stores all layer-dominating paths of G starting at v. Any node u in T_v (including the root v) at depth less than $2k$ has as the children all neighbours w of u in G such that $d_G(s,w) = d_G(s,w) + 1$. Every node of T_v represents a unique path of G corresponding to the path of T_v from the root v to this node. A leaf t of T_v has a pointer to a layer-dominating path τ (and, hence, to the corresponding vertex v_τ in G') if the path $\tau = \{v, \ldots, t\}$ from the root v to the leaf t forms a layer-dominating path τ in G.

Now, given a layer-dominating path $\sigma = \{v_{i-k}, v_{i-k+1}, \ldots, v_j\}$, we can determine all layer-dominating paths τ which σ is compatible with in $\mathcal{O}(m)$ time as follows. Take the tree $T_{v_{i-k+1}}$ in $\mathcal{T}$ and, following path σ, decent to node v_j of $T_{v_{i-k+1}}$ representing path $\{v_{i-k+1}, \ldots, v_j\}$. Then, leaves of $T_{v_{i-k+1}}$ attached to v_j have pointer to all paths τ which σ is compatible with.

Since G' has at most $\mathcal{O}(n^{2k+1})$ vertices, creating G' (in line 10) takes at most $\mathcal{O}(n^{2k+1}m)$ time. Thus, the overall running time for Algorithm 1 is $\mathcal{O}(n^{2k+1}m)$. □

Algorithm 1 determines if there is a shortest path of eccentricity at most k for a given vertex s. If a start vertex is not given, iterating Algorithm 1 over each vertex will determine if there is a shortest path of eccentricity at most k in a given graph G in $\mathcal{O}(n^{2k+2}m)$ time. If k is unknown, a path with minimum eccentricity can be found by trying different values for k starting with 1. Then, the runtime is $\mathcal{O}(n^4m) + \mathcal{O}(n^6m) + \ldots + \mathcal{O}(n^{2k+2}m) = \mathcal{O}(n^{2k+2}m)$.

Corollary 12. *If a given graph G contains a shortest path with eccentricity k, the MESP problem can be solved for G in $\mathcal{O}(n^{2k+2}m)$ time, even if k is unknown.*

Next, we will show that the problem is W[2]-hard. Therefore, we do not expect that MESP is Fixed Parameter Tractable, i.e., there is probably no algorithm that finds an optimal solution in $f(k)n^{\mathcal{O}(1)}$ time.

Theorem 13. *The Minimum Eccentricity Shortest Path problem is W[2]-hard.*

Proof. To show W[2]-hardness, we will make a parametrised reduction from the Dominating Set problem which is known to be W[2]-complete [8].

Consider a given instance (G,k) of the Dominating Set problem where $G = (V,E)$ is a graph with $V = \{v_1, v_2, \ldots, v_n\}$. Based on (G,k), we will construct an instance (H,k) of the MESP problem with a graph H as follows. Start with an empty graph and add a copy of V, i.e., only add the vertices without edges between them. Add k sets of vertices $U_1, U_2, \ldots, U_k$ with $U_i = \{u_1^i, u_2^i, \ldots, u_n^i\}$ and, for each j with $1 \leq j < k$, make all vertices in U_j adjacent to all vertices in U_{j+1}. Add the vertices s, s', t, and t' and connect s with s' and t with t', respectively, with a path of length k. Additionally, make s adjacent to all vertices in U_1 and make t adjacent to all vertices in U_k. Connect each vertex $u_i^j \in U_j$ with each vertex in $N_G[v_i] \subseteq V$ with a path of length k for all j with $1 \leq j \leq k$. Fig. 3 gives an illustration.

Because $d_H(s,s') = d_H(t,t') = k$, each shortest path in H not containing s and t has an eccentricity larger than k. Also, a shortest path from s to t has length $k+1$, intersects all sets U_j, and does not intersect V.

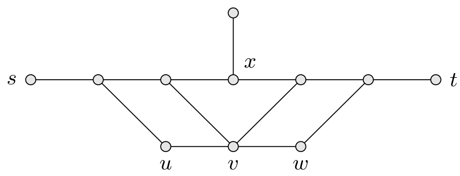

Fig. 4. Example for Lemma 14 and Lemma 15. The shortest path from s to t which contains x has eccentricity 1 and the distance from x to v is 2. The shortest path from s to t which contains u and w has eccentricity 3.

First, assume that H has a shortest path P with eccentricity k. By definition of P and construction of H, for every $v \in V$, there is a vertex $u_i^j \in P$ such that $d_H(v, u_i^j) = d_H(v, P) = k$ and, hence, $v \in N_G[v_i]$. Therefore, the set $D = \{ v_i \in V \mid$ there is a j with $u_i^j \in P\}$ is a dominating set for G with cardinality at most k.

Next, assume there is a dominating set D for G with cardinality at most k. Without loss of generality, let $D = \{v_1, v_2, \ldots, v_k\}$. Then, we define $P = \{s, u_1^1, u_2^2, \ldots, u_k^k, t\}$. By construction of H, each vertex $v \in N_G[v_i]$ is at distance k to u_i^i. Thus, because D is a dominating set, there is a vertex $u_i^i \in P$ for each vertex $v \in V$ with $d_H(v, u_i^i) = k$. Therefore, P has eccentricity k in H. $\square$

4. Approximation algorithms

In this section, we will present different approximation algorithms. The algorithms differ in their approximation factor and runtime. The algorithms are based on the following two Lemmas.

Lemma 14. *In a graph G, let P be a shortest path from s to t of eccentricity at most k. For each layer $L_i^{(s)}$, there is a vertex $p_i \in P$ such that the distance from p_i to each vertex $v \in L_i^{(s)}$ is at most $2k$. Additionally, $p_i \in L_i^{(s)}$ if $i \leq d_G(s,t)$, and $p_i = t$ if $i \geq d_G(s,t)$.*

Proof. For each vertex v, let $p(v) \in P$ be a vertex with $d_G(p(v), v) \leq k$.

For each $i \leq d_G(s,t)$, let $p_i \in P \cap L_i^{(s)}$ be the vertex in P with distance i to s. For an arbitrary vertex $v \in L_i^{(s)}$, let $j = d_G(s, p(v))$. Because $\mathrm{ecc}_G(P) \leq k$ and P is a shortest path, $|i - j| \leq k$. Thus, $d_G(p_i, v) \leq d_G(p_i, p(v)) + d_G(p(v), v) \leq 2k$.

Let $L' = \{ v \mid d_G(s,v) \geq d_G(s,t)\}$. Because P has eccentricity at most k, $d_G(p,t) \leq k$ for all $p \in \{ p(v) \mid v \in L'\}$. Therefore, $d_G(t,v) \leq 2k$ for all $v \in L'$. $\square$

Lemma 15. *If G has a shortest path of eccentricity at most k from s to t, then every path Q with $s \in Q$ and $d_G(s,t) \leq \max_{v \in Q} d_G(s,v)$ has eccentricity at most $3k$.*

Proof. Let P be a shortest path from s to t with $\mathrm{ecc}_G(P) \leq k$ and Q an arbitrary path with $s \in Q$ and $d_G(s,t) \leq \max_{v \in Q} d_G(s,v)$. Without loss of generality, we can assume that Q starts at s. Also, let u be an arbitrary vertex. Since $\mathrm{ecc}_G(P) \leq k$, there is a vertex $p \in P$ with $d_G(u,p) \leq k$. Because $d_G(s,t) \leq \max_{v \in Q} d_G(s,v)$, there is a vertex $q \in Q$ with $d_G(s,p) = d_G(s,q)$. By Lemma 14, the distance between p and q is at most $2k$. Thus, the distance from q to u is at most $3k$. $\square$

Corollary 16. *For a given graph G and two vertices s and t, each shortest path from s to t is a 3-approximation for the (s,t)-MESP problem.*

Note that the bounds given in Lemma 14 and Lemma 15 are tight. Fig. 4 gives an example.

Our first approximation algorithm, Algorithm 2, is based on Lemma 15 and gives a 3-approximation. For each possible start vertex, it finds a shortest path with maximal length. Out off all these paths, it selects the one with the smallest eccentricity.

Algorithm 2: A 3-approximation for the MESP problem.

Input: A graph $G = (V, E)$.
Output: A shortest path with eccentricity at most $3k$, where k is the minimum eccentricity of all paths in G.

```
foreach s ∈ V do
    Find a vertex v for which the distance to s is maximal. Also find a shortest path P(s) from s to v.
    Calculate k(s) = ecc_G(P(s)).
Among all computed paths P(s), select one for which k(s) is minimal.
```

Theorem 17. *Algorithm 2 calculates a 3-approximation for the MESP problem in $\mathcal{O}(nm)$ time.*

Proof. Assume a given graph G has a shortest path P from s to t with $\mathrm{ecc}_G(P) = k$ and s is the vertex selected by the loop in line 1. Let v be a vertex such that $d_G(s, v)$ is maximal (line 2). Because $d_G(s, v)$ is maximal, $d_G(s, t) \leq d_G(s, v)$. Thus, by Lemma 15, each path from s to v has eccentricity at most $3k$, i.e. $k(s) \leq 3k$ (line 3). Therefore, the eccentricity of the path selected in line 4 is also at most $3k$.

It is easy to see that line 2 and line 3 run in $\mathcal{O}(m)$ time for a given s. Therefore, the overall runtime for the algorithm is $\mathcal{O}(nm)$. □

For Algorithm 3 below, we say the *layer-wise eccentricity* of a shortest path Q from s to t is ϕ if, for each layer $L_i^{(s)}$, $\max\left\{ d_G(q_i, u) \mid u \in L_i^{(s)} \right\} \leq \phi$ where $q_i \in Q \cap L_i^{(s)}$ if $i \leq d_G(s,t)$ and $q_i = t$ if $i > d_G(s,t)$. By Lemma 14, a shortest path with eccentricity k has a layer-wise eccentricity of $2k$. Therefore, determining a shortest path with minimum layer-wise eccentricity gives a 2-approximation for the MESP problem. To find such a path, Algorithm 3 computes, for each vertex s, the maximal distance of a vertex v to all other vertices u in the same layer $L_i^{(s)}$ and uses a modified BFS to find a shortest path with minimal layer-wise eccentricity starting at s.

Algorithm 3: A 2-approximation for the MESP problem.

Input: A graph $G = (V, E)$.
Output: A shortest path with eccentricity at most $2k$, where k is the minimum eccentricity of all paths in G.

1 Calculate the distances $d_G(u, v)$ for all vertex pairs u and v, including $L_i^{(u)} = \{ v \in V \mid d_G(u, v) = i \}$ with $0 \leq i \leq \mathrm{ecc}_G(u)$ for each u.
2 **foreach** $s \in V$ **do**
3   Set $\phi(s) := 0$.
4   **for** $i := 1$ **to** $\mathrm{ecc}_G(s)$ **do**
5   **foreach** $v \in L_i^{(s)}$ **do**
6   Set $\phi(v) := \max_{u \in L_i^{(s)}} d_G(u, v)$.
7   Let $N_G^-(v) = L_{i-1}^{(s)} \cap N_G(v)$ denote the neighbours of v in the previous layer. Set $\phi(v) := \max\{\min_{u \in N_G^-(v)} \phi(u), \phi(v)\}$.
8   Set $\phi^+(v) := \max\{ d_G(u, v) \mid d_G(s, u) \geq i\}$.
9   Calculate a BFS-tree $T(s)$ starting from s. If multiple vertices u are possible as parent for a vertex v, select one with the smallest $\phi(u)$.
10   Let t be the vertex for which $\phi'(t) := \max\{\phi(t), \phi^+(t)\}$ is minimal. Set $k(s) := \phi'(t)$
11 Among all computed pairs s and t, select a pair (and corresponding path in $T(s)$) for which $k(s)$ is minimal.

Theorem 18. *Algorithm 3 calculates a 2-approximation for the MESP problem in $\mathcal{O}(n^3)$ time.*

Proof *(Correctness).* Assume a given graph G has a shortest path P from s to t with $\mathrm{ecc}_G(P) = k$ and s is the vertex selected by the loop starting in line 2. Let Q be a shortest path from s to v.

We will now show that lines 4 to 8 calculate for each v the minimal $\phi(v)$ such that there is a shortest path Q from s to v with a layer-wise eccentricity $\phi(v)$.

By induction, assume this is true for all vertices $u \in L_j^{(s)}$ with $j \leq i - 1$. Now let v be an arbitrary vertex in $L_i^{(s)}$. Line 6 calculates the maximal distance $\phi(v)$ from v to all other vertices in $L_i^{(s)}$. Since v is the only vertex in $Q \cap L_i^{(s)}$ for every shortest path Q from s to v, the layer-wise eccentricity of each Q is at least $\phi(v)$. Let u be a neighbour of v in the previous layer. By induction hypothesis, $\phi(u)$ is optimal. Therefore, $\phi(v) := \max\{\min_{u \in N_G^-[v]} \phi(u), \phi(v)\}$ (line 7) is optimal for v.

Since line 9 selects the vertex u with the smallest $\phi(u)$ as parent for v, each path Q from s to v in $T(s)$ has an optimal layer-wise eccentricity of $\phi(v)$. Line 8 calculates the maximal distance from v to all vertices in $\{u \mid d_G(s, u) \geq d_G(s, v)\}$. Thus, $\mathrm{ecc}_G(Q) \leq \phi'(v)$ and line 10 and line 11 select a shortest path which has an eccentricity at most $\phi'(v)$.

By Lemma 14, we know that P has a layer-wise eccentricity of at most $2k$. Thus, the path Q from s to t in $T(s)$ has a layer-wise eccentricity of at most $2k$. Additionally, Lemma 14 says that t $2k$-dominates all vertices in $\{v \mid d_G(s, v) \geq d_G(s, t)\}$. Therefore, $\mathrm{ecc}_G(Q) \leq 2k$. Thus, the path selected in line 11 is a shortest path with eccentricity at most $2k$. □

Proof *(Complexity).* Line 1 runs in $\mathcal{O}(nm)$ time. If the distances are stored in an array, they can be later accessed in constant time. Therefore, line 6 and line 8 run in $\mathcal{O}(n)$ time for a given s and v or in $\mathcal{O}(n^3)$ time overall. For a given s, line 7 runs in $\mathcal{O}(m)$ time and, therefore, has an overall runtime of $\mathcal{O}(nm)$. Line 9 has an overall runtime of $\mathcal{O}(nm)$, line 11 takes $\mathcal{O}(n^2)$ time, and line 10 runs in $\mathcal{O}(n)$ time. Adding all together, the total runtime is $\mathcal{O}(n^3)$. □

For the case that a start vertex s for a shortest path is given, Algorithm 3 can be simplified by having only one iteration of the loop starting in line 2. Then, the runtime is $\mathcal{O}(nm)$.

Corollary 19. *A 2-approximation for the (s, t)-MESP problem can be computed in $\mathcal{O}(nm)$ time.*

Algorithm 2 and Algorithm 3 both iterate over all vertices of the graph to find the best start vertex. Lemma 20 will show that a constant factor approximation can be found with a simple algorithm which starts at an arbitrary vertex. However, the approximation factor will be much higher.

Lemma 20. *Let G be a graph having a shortest path of eccentricity k. Let x be a vertex most distant from some arbitrary vertex, and y be a vertex most distant from x. Then, x, y is a $8k$-dominating pair of G.*

Proof. Let p be an end vertex of a shortest path of eccentricity k in a given graph G. By Lemma 14, the diameter of each layer $L_i^{(p)}$ in G is at most $4k$. Assume, x is most distant from an arbitrary vertex s.

If there is a layer containing both s and x, then $d_G(s,x) \leq 4k$. By the choice of x, each vertex of G is within distance at most $4k$ from s, hence, within distance at most $8k$ from x. Evidently, in this case, x, y is a $8k$-dominating pair of G.

Assume now, without loss of generality, that $x \in L_i^{(p)}$ and $s \in L_l^{(p)}$ with $i < l$. Consider an arbitrary vertex v of G which belongs to a layer with an index smaller than i. We show that $d_G(x,v) \leq 8k$. As $L_i^{(p)}$ separates v from s, a shortest path $P(s,v)$ of G between s and v must have a vertex u in $L_i^{(p)}$. We have $d_G(s,x) \geq d_G(s,v) = d_G(s,u) + d_G(u,v)$ and, by the triangle inequality, $d_G(s,x) \leq d_G(s,u) + d_G(u,x)$. Hence, $d_G(u,v) \leq d_G(u,x)$ and, since both u and x belong to same layer $L_i^{(p)}$, $d_G(u,x) \leq 4k$. That is, $d_G(x,v) \leq d_G(x,u) + d_G(u,v) \leq 2d_G(u,x) \leq 8k$.

If $d_G(x,y) \leq 8k$ then, by the choice of y, each vertex of G is within distance at most $8k$ from x. Hence, x, y is a $8k$-dominating pair of G. So, assume that $d_G(x,y) > 8k$, i.e., the layer $L_j^{(p)}$ with $i < j$ contains y. Repeating the arguments of the previous paragraph, we can show that $d_G(y,v) \leq 8k$ for every vertex v that belongs to a layer with an index greater than j.

Consider now an arbitrary path P of G connecting vertices x and y. P has a vertex in every layer $L_h^{(p)}$ with $i \leq h \leq j$. Hence, for each vertex v of G that belongs to layer $L_h^{(p)}$ ($i \leq h \leq j$), there is a vertex $u \in P \cap L_h^{(p)}$ such that $d_G(v,u) \leq 4k$. As $d_G(v,x) \leq 8k$ for each vertex v from $L_{i'}^{(p)}$ with $i' < i$ and $d_G(v,y) \leq 8k$ for each vertex v from $L_{j'}^{(p)}$ with $j' > j$, we conclude that $\mathrm{ecc}_G(P) \leq 8k$. □

Corollary 21. *An 8-approximation for the MESP problem can be calculated in linear time.*

5. Solving *k*-domination using a MESP

In [18], an $\mathcal{O}(n^7)$ time algorithm was presented which finds a minimum dominating set for graphs containing a shortest path with eccentricity 1. Using a similar approach, we will generalize this result to find a minimum k-dominating set for graphs containing a shortest path with eccentricity k.

Recall that, for a graph $G = (V, E)$, a vertex set D is a *k-dominating set* if $N_G^k[D] = V$. Additionally, D is a *minimum* k-dominating set if there is no k-dominating set D' for G with $|D'| < |D|$.

Lemma 22. *Let D be a minimum k-dominating set of a graph G and let G have a shortest path of eccentricity at most k starting at a vertex s. Then, for all non-negative integers $i \leq \mathrm{ecc}_G(s)$,*

$$\left| D \cap \bigcup_{l=i-k}^{i+k} L_l^{(s)} \right| \leq 6k + 1.$$

Proof. Let $P = \{s = v_0, v_1, \ldots, v_j\}$ be a shortest path with $j \leq \mathrm{ecc}_G(s)$ and $\mathrm{ecc}_G(P) \leq k$. Also, let $D_i = D \cap \bigcup_{l=i-k}^{i+k} L_l^{(s)}$ be a set of k-dominating vertices in the layers $L_{i-k}^{(s)}$ to $L_{i+k}^{(s)}$. Because D is k-dominating, D_i can only k-dominate vertices in the layers $L_{i-2k}^{(s)}$ to $L_{i+2k}^{(s)}$. By Lemma 10, these layers are also k-dominated by $P_i = \{v_{i-3k}, \ldots, v_{i+3k}\}$. Thus,

$$N_G^k[D_i] \subseteq \bigcup_{l=i-2k}^{i+2k} L_l^{(s)} \subseteq N_G^k[P_i].$$

Assume, $|D_i| > |P_i|$. Note that $|P_i| \leq 6k + 1$. Then, there is a k-dominating set $D' = (D \setminus D_i) \cup P_i$ such that $|D| > |D'|$. Thus, D is not a minimum k-dominating set. □

Based on Lemma 22, Algorithm 4 below computes a minimum k-dominating set for a given graph $G = (V, E)$ with a shortest path of eccentricity k starting at a vertex s as follows. In the i-th iteration, the algorithm knows all vertex sets S for which there is a vertex set S' such that (i) $S = S' \cap (L_{i-k}^{(s)} \cup \ldots \cup L_{i-1+k}^{(s)})$, (ii) the set $S^* = S \cup (S' \cap L_{i-1-k}^{(s)})$ k-dominates $L_{i-1}^{(s)}$ and has cardinality at most $6k + 1$, and (iii) S' k-dominates the layers $L_0^{(s)}$ to $L_{i-1}^{(s)}$. Due to Lemma 22, a set S^* with a larger cardinality cannot be a subset of a minimum dominating set of G and, hence, neither can be S or S'. Each such set S is stored as a pair (S, S') in a set X_{i-1} where S' is a corresponding set with minimum cardinality, i.e., X_{i-1} does not contain two pairs (S, S') and (T, T') with $S = T$. Note that, since S' has minimum cardinality, it does not contain any vertices from any layer $L_j^{(s)}$ with $j > i - 1 + k$. We will show later that, this way, X_{i-1} always contains a pair (S, S') such that S' is subset of some minimum k-dominating set.

Then, for each pair $(S, S') \in X_{i-1}$, the algorithm computes all sets $S \cup U$ which k-dominate the layer $L_i^{(s)}$ and have cardinality at most $6k+1$. For such a set, the sets $R = (S \cup U) \setminus L_{i-k}^{(s)}$ and $R' = S' \cup U$ are created and, if the set X_i does not contain a pair (P, P') with $P = R$, stored as the pair (R, R') in X_i. In the case that X_i already contains such a pair (P, P'), either, if $|R'| < |P'|$, (P, P') is replaced by (R, R') or, if $|R'| \geq |P'|$, (R, R') is not added to X_i.

Note that $L_{i+k}^{(s)} = \emptyset$ for $i > \mathrm{ecc}_G(s) - k$. Therefore, the algorithm can stop after $\mathrm{ecc}_G(s) - k$ iterations.

Algorithm 4: Determines a minimum k-dominating set in a given graph G containing a shortest path of eccentricity k starting at s.

Input: A graph G, an integer k, and a vertex s which is start vertex of a shortest path with eccentricity k.
Output: A minimum k-dominating set.

1 Compute the layers $L_0^{(s)}, L_1^{(s)}, \ldots, L_{\mathrm{ecc}_G(s)}^{(s)}$.
2 Create the set $X_0 := \{(S, S) \mid S \subseteq N_G^k[s]; 0 < |S| \leq 6k+1\}$.
3 **for** $i := 1$ **to** $\mathrm{ecc}_G(s) - k$ **do**
4 Create $X_i := \emptyset$.
5 **foreach** $(S, S') \in X_{i-1}$ **do**
6 **foreach** $U \subseteq L_{i+k}^{(s)}$ *with* $|S \cup U| \leq 6k+1$ **do**
7 **if** $N_G^k[S \cup U] \supseteq L_i^{(s)}$ **then**
8 $R := (S \cup U) \setminus L_{i-k}^{(s)}$
9 $R' := S' \cup U$
10 **if** *There is no pair* $(P, P') \in X_i$ *with* $P = R$ **then**
11 Insert (R, R') into X_i.
12 **if** *There is a pair* $(P, P') \in X_i$ *with* $P = R$ *and* $|R'| < |P'|$ **then**
13 Replace (P, P') in X_i by (R, R').
14 Among all pairs $(S, S') \in X_{\mathrm{ecc}_G(s)-k}$ for which S' k-dominates G, determine one with minimum $|S'|$, say (D, D').
15 Output D'.

Theorem 23. *For a given graph G and a vertex s which is start vertex of a shortest path with eccentricity k, Algorithm 4 determines a minimum k-dominating set in $n^{\mathcal{O}(k)}$ time.*

Proof *(Correctness).* To prove the correctness, we show by induction that, for each i with $0 \leq i \leq \mathrm{ecc}_G(s) - k$, there is a minimum k-dominating set D and a pair $(S, S') \in X_i$ such that $S' = D \cap \bigcup_{l=0}^{i+k} L_l^{(s)}$. If this is true for $i = \mathrm{ecc}_G(s) - k$, then $S' = D$. Hence, if (S, S') is a pair in $X_{\mathrm{ecc}_G(s)-k}$ such that S' k-dominates G and has minimum cardinality, then S' is a minimum k-dominating set of G.

By construction in line 2, X_0 contains all pairs (S, S') such that S' is a vertex set with cardinality at most $6k+1$ which k-dominates $L_0^{(s)}$. Thus, the base case is true. Next, by induction hypothesis and by definition of the pairs (S, S'), there is a minimum dominating set D and a pair $(S, S') \in X_{i-1}$ such that $S = S' \cap \bigcup_{l=i-k}^{i+k-1} L_l^{(s)} = D \cap \bigcup_{l=i-k}^{i+k-1} L_l^{(s)}$. Let $M = D \cap L_{i+k}^{(s)}$. By Lemma 22, $\left| D \cap \bigcup_{l=i-k}^{i+k} L_l^{(s)} \right| = |S \cup M| \leq 6k+1$. Therefore, there is an iteration of the loop starting in line 6 with $U = M$. Because $S \cup M = D \cap \bigcup_{l=i-k}^{i+k} L_l^{(s)}$, $S \cup M$ k-dominates $L_i^{(s)}$, i.e., $N_G^k[S \cup M] \supseteq L_i^{(s)}$. Thus, the algorithm creates a pair (R, R') with $R' = D \cap \bigcup_{l=0}^{i+k} L_l^{(s)}$ (see line 7 to line 9).

Assume that there is a pair $(P, P') \neq (R, R')$ with $P = R$ and $|P'| \leq |R'|$, i.e., (R, R') will not be stored in X_i or replaced by (P, P') (see line 10 to line 13). Because $P = R$, $P' \cap \bigcup_{l=i-k+1}^{i+k} L_l^{(s)} = D \cap \bigcup_{l=i-k}^{i+k} L_l^{(s)}$. Let $D' = P' \cup (D \cap \bigcup_{l=i-k+1}^{\mathrm{ecc}_G(s)-k} L_l^{(s)})$. By definition, P' k-dominates $\bigcup_{l=0}^{i} L_l^{(s)}$. Thus, D' k-dominates $\bigcup_{l=0}^{i} L_l^{(s)}$. Note that $D' \supseteq D \cap (\bigcup_{l=i-k+1}^{\mathrm{ecc}_G(s)} L_l^{(s)})$. Thus, D' also k-dominates $\bigcup_{l=i+1}^{\mathrm{ecc}_G(s)} L_l^{(s)}$. Therefore, D' is a minimum k-dominating set and there is a pair $(P, P') \in X_i$ such that $P' = D' \cap \bigcup_{l=0}^{i+k} L_l^{(s)}$. $\square$

Proof *(Complexity).* For a given i, there are no two pairs (S, S') and (T, T') in X_i with $S = T$ (see line 10 to line 13). Thus, for each set $U \subseteq L_{i+k}^{(s)}$, $S \cup U \neq T \cup U$. Additionally, since S and T intersect at most $2k$ consecutive layers, $S \neq T$ for all pairs $(S, S') \in X_i$ and $(T, T') \in X_j$ with $|i - j| \geq 2k$. Therefore, a set $S \cup U$ is processed at most $\mathcal{O}(k)$ times by the loop starting in line 6. Hence, because there are at most n^{6k+1} sets $S \cup U$ with $|S \cup U| \leq 6k+1$, the loop starting in line 6 has at most $\mathcal{O}(n^{6k+1}k)$ iterations.

Next, we show that a single iteration of the loop starting in line 6 requires at most $\mathcal{O}(m)$ time. It takes at most $\mathcal{O}(m)$ time to check if $N_G^k[S \cup U] \supseteq L_i^{(s)}$ (line 7) and at most $\mathcal{O}(n)$ time to construct (R, R'). Determining if X_i contains a pair (P, P') with $P = R$ and (if necessary) replacing it can be achieved in $\mathcal{O}(n)$ time as follows. One way is to use a tree-structure similar to the one we used for Algorithm 1. An other (less memory efficient) option is to use an array A_i of size n^{6k+1} where each element can store a pair (S, S'). To determine the index of a pair, assume that each vertex of G has a unique identifier in the range from 0 to $n-1$. Additionally, assume that the vertices in a set S are ordered by their identifier. Therefore, each set S can be represented by a unique $(6k+1)$-digit number with base n. This number is the index of a pair (S, S') in A_i. Hence, it takes at most $\mathcal{O}(n)$ time to add (R, R') to X_i and (if necessary) replace a pair (P, P').

Therefore, the total runtime of the algorithm is $\mathcal{O}(n^{6k+1}km)$. □

If the start vertex s is unknown, one can use Algorithm 1 to, first, find a shortest path with eccentricity k and, then, use Algorithm 4 to find a minimum k-dominating set.

6. Conclusion

We have shown that, if a graph has a shortest path of eccentricity at most k, we can compute an $(8k+2)$-approximation for the line distortion of the graph. Motivated by this result, we investigated the Minimum Eccentricity Shortest Path problem. We have shown that the problem is NP-hard even if a start-end vertex pair is given and presented a pseudo-polynomial time algorithm solving it which runs in $\mathcal{O}(n^{2k+2}m)$ time if the given graph contains a shortest path with eccentricity k. We also gave constant factor approximation algorithms. The best one computes a 2-approximation in $\mathcal{O}(n^3)$ time and the fastest one computes an 8-approximation in linear time. Additionally, we presented an algorithm which, for a graph containing a shortest path with eccentricity k, computes a minimum k-dominating set in $n^{\mathcal{O}(k)}$ time.

The problem can be naturally split into two subproblems. First, find the start and end vertices of an optimal path. Second, for a given vertex pair, find a shortest path between them with the minimum eccentricity. We know that the second subproblem remains NP-hard. However, is it possible to determine the start and end vertices of an optimal path efficiently? That is, can we find two vertices s and t in polynomial time such that a path from s to t has the minimum eccentricity in G?

We have shown that one can compute a constant factor approximation in linear time. Therefore, MESP is in APX. It remains an open question if the problem is APX-complete.

Notes added in proof

1. When this paper was under review (some of its results were already presented at WADS 2015 [11]), we have learned about a follow-up paper by Birmelé et al. [3]. They showed that our linear-time method (the double-BFS procedure) described in Lemma 20 is in fact a 5-approximation algorithm for the MESP problem and that it can be extended to obtain a linear-time 3-approximation algorithm.
2. Using our Local Density Lemma (Lemma 2), one can improve (see [19]) our approximation ratio from [9] for the minimum line-distortion problem on AT-free graphs from 8 to 6.

Acknowledgements

We would like to thank anonymous reviewers for many useful suggestions and comments.

References

[1] M. Bădoiu, J. Chuzhoy, P. Indyk, A. Sidiropoulos, Low-distortion embeddings of general metrics into the line, in: Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC 2005), Baltimore, MD, USA, May 22–24, ACM, 2005, pp. 225–233.
[2] M. Bădoiu, K. Dhamdhere, A. Gupta, Y. Rabinovich, H. Räcke, R. Ravi, A. Sidiropoulos, Approximation algorithms for low-distortion embeddings into low-dimensional spaces, in: Proceedings of the Sixteenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2005), Vancouver, British Columbia, Canada, January 23–25, SIAM, 2005, pp. 119–128.
[3] E. Birmelé, F. de Montgolfier, L. Planche, Minimum eccentricity shortest path problem: an approximation algorithm and relation with the k-laminarity problem, in: COCOA 2016: 10th Annual International Conference on Combinatorial Optimization and Applications, Hong Kong, China, December 16–18, 2016, in: Lecture Notes in Computer Science, vol. 10043, 2016, pp. 216–229.
[4] D.G. Corneil, S. Olariu, L. Stewart, Linear time algorithms for dominating pairs in asteroidal triple-free graphs, SIAM J. Comput. 28 (1997) 292–302.
[5] M. Cygan, M. Pilipczuk, Bandwidth and distortion revisited, Discrete Appl. Math. 160 (2012) 494–504.
[6] M. de Berg, A. Khosravi, Optimal binary space partitions in the plane, in: COCOON 2010: 16th Annual International Computing and Combinatorics Conference, Nha Trang, Vietnam, July 19–21, 2010, in: Lecture Notes in Computer Science, vol. 6196, 2010, pp. 216–225.
[7] J.S. Deogun, D. Kratsch, Diametral path graphs, in: WG 1995: 21st International Workshop on Graph-Theoretic Concepts in Computer Science, in: Lecture Notes in Computer Science, vol. 1017, 1995, pp. 344–357.
[8] R.G. Downey, M.R. Fellows, Fundamentals of Parameterized Complexity, Springer, 2013.
[9] F.F. Dragan, E. Köhler, A. Leitert, Line-distortion, bandwidth and path-length of a graph, in: SWAT 2014: 14th Scandinavian Symposium and Workshops on Algorithm Theory, Copenhagen, Denmark, July 2–4, 2014, in: Lecture Notes in Computer Science, vol. 8503, 2014, pp. 146–257, Algorithmica 77 (2017) 686–713.

[10] F.F. Dragan, A. Leitert, Minimum eccentricity shortest paths in some structured graph classes, in: WG 2015: 41st International Workshop on Graph-Theoretic Concepts in Computer Science, Munich, Germany, June 17–19, 2015, in: Lecture Notes in Computer Science, vol. 9224, 2015, pp. 189–202, J. Graph Algorithms Appl. 20 (2016) 299–322.
[11] F.F. Dragan, A. Leitert, On the minimum eccentricity shortest path problem, in: WADS 2015: 14th International Symposium on Algorithms and Data Structures, Victoria, Canada, August 5–7, 2015, in: Lecture Notes in Computer Science, vol. 9214, 2015, pp. 276–288.
[12] M.R. Fellows, F.V. Fomin, D. Lokshtanov, E. Losievskaja, F.A. Rosamond, S. Saurabh, Distortion is fixed parameter tractable, in: ICALP 2009: 36th International Colloquium on Automata, Languages and Programming, Rhodes, Greece, July 5–12, 2009, in: Lecture Notes in Computer Science, vol. 5555, 2009, pp. 463–474.
[13] F.V. Fomin, D. Lokshtanov, S. Saurabh, An exact algorithm for minimum distortion embedding, Theoret. Comput. Sci. 412 (2011) 3530–3536.
[14] P. Heggernes, D. Meister, Hardness and approximation of minimum distortion embeddings, Inform. Process. Lett. 110 (2010) 312–316.
[15] P. Heggernes, D. Meister, A. Proskurowski, Computing minimum distortion embeddings into a path of bipartite permutation graphs and threshold graphs, Theoret. Comput. Sci. 412 (2011) 1275–1297.
[16] P. Indyk, Algorithmic applications of low-distortion geometric embeddings, in: FOCS 2001: 42nd Annual Symposium on Foundations of Computer Science, Las Vegas, NV, USA, October 14–17, 2001, IEEE, 2001, pp. 10–35.
[17] P. Indyk, J. Matousek, Low-distortion embeddings of finite metric spaces, in: Handbook of Discrete and Computational Geometry, second ed., CRC press, 2004, pp. 177–196.
[18] D. Kratsch, Domination and total domination on asteroidal triple-free graphs, Discrete Appl. Math. 99 (2000) 111–123.
[19] A. Leitert, Tree-Breadth of Graphs with Variants and Applications, Dissertation Kent State University, August 2017.
[20] H. Müller, Hamiltonian circuits in chordal bipartite graphs, Discrete Math. 156 (1996) 291–298.
[21] P.J. Slater, Locating central paths in a graph, Transp. Sci. 16 (1982) 1–18.
[22] J.B. Tenenbaum, V. de Silva, J.C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.