Information Processing Letters 181 (2023) 106360

Contents lists available at ScienceDirect

Information Processing Letters

journal homepage: www.elsevier.com/locate/ipl

Improved kernels for tracking paths

Pratibha Choudhary [a,*], Michael T. Goodrich [b,1], Siddharth Gupta [c,2], Hadi Khodabandeh [b], Pedro Matias [b], Venkatesh Raman [d]

[a] Czech Technical University in Prague, Prague, Czech Republic
[b] Dept. of Computer Science, Univ. of California Irvine, Irvine, United States
[c] Dept. of Computer Science, University of Warwick, Coventry, UK
[d] The Institute of Mathematical Sciences, HBNI, Chennai, India

ARTICLE INFO

Article history:
Received 30 September 2021
Received in revised form 4 January 2023
Accepted 5 January 2023
Available online 10 January 2023

Keywords:
Graph algorithms
Kernelization
Fixed-parameter tractability
Planar graphs
Tracking paths

ABSTRACT

Tracking of moving objects is crucial to security systems and networks. Given a graph G, terminal vertices s and t, and an integer k, the Tracking Paths problem asks whether there exists at most k vertices, which if marked as trackers, would ensure that the sequence of trackers encountered in each s-t path is unique. It is known that the problem is NP-hard and admits a kernel (reducible to an equivalent instance) with $\mathcal{O}(k^6)$ vertices and $\mathcal{O}(k^7)$ edges, when parameterized by the size of the output (tracking set) k [4]. In this paper we improve the size of the kernel substantially by providing a kernel with $O(k^2)$ vertices and edges for general graphs and a kernel with $O(k)$ vertices and edges for planar graphs. We do this via a new concept, namely a *tree-sink structure*. We also show that finding a tracking set of size at most $n-k$ for a graph on n vertices is hard for the parameterized complexity class W[1], when parameterized by k.

1. Introduction

Graphs serve as a systematic model for modeling and analysis of many real life problems. One of the commonly studied problems in areas of networks and machine learning is tracking of moving objects. Coordinated path tracking and framework for multi-target tracking have been discussed in [26] and [25]. Tracking of moving objects has been widely studied in networks, wireless sensor networks, neural networks and binary sensor networks [14], [11], [24], [1]. Tracking algorithms can also be used in designing debugging tools in programs and for leakage detection systems.

The problem of target tracking can be modeled as the following graph theoretic problem. Let $G = (V, E)$ be an undirected graph without any self loops or parallel edges with a unique entry vertex (source) s and a unique exit vertex (destination) t. A simple path from s to t is called an s-t path. The Tracking Paths problem asks to find a set of vertices $T \subseteq V$ such that for any two distinct s-t paths, say P_1 and P_2, the sequence of vertices in $T \cap V(P_1)$ as encountered in P_1 is different from the sequence of vertices in $T \cap V(P_2)$ as encountered in P_2. Here T is called a *tracking set* for the graph G, and the vertices in T are called *trackers*. Banik, Katz, Packer and Simakov [6] first studied the problem of tracking paths in graphs, where they focused on distinguishing all shortest s-t paths in a graph and proved that the problem Tracking Shortest Paths is NP-hard and APX-hard. They also gave a 2-approximate algorithm for the same problem in planar graphs, along with some other results. Tracking Shortest Paths was first

* Corresponding author.
E-mail addresses: pratibha.choudhary@fit.cvut.cz (P. Choudhary), goodrich@uci.edu (M.T. Goodrich), siddharth.gupta.1@warwick.ac.uk (S. Gupta), khodabah@uci.edu (H. Khodabandeh), pmatias@uci.edu (P. Matias), vraman@imsc.res.in (V. Raman).

[1] Supported in part by NSF Grant 1815073.
[2] Supported in part by the Engineering and Physical Sciences Research Council (EPSRC) grant EP/V007793/1.

https://doi.org/10.1016/j.ipl.2023.106360
0020-0190/

studied from a parameterized perspective in [3], [5], where the problem was shown to be fixed-parameter tractable (FPT). Tracking Paths is formally defined as follows.

Tracking Paths (G, s, t, k)
Input: An undirected graph $G = (V, E)$ with two distinct vertices s and t, and a non-negative integer k.
Parameter: k
Question: Does there exist a tracking set T of size at most k for G?

Banik et al. [4] proved Tracking Paths to be NP-complete and fixed-parameter tractable by showing the existence of a polynomial kernel. Specifically it was proven that an instance of Tracking Paths can be reduced to an equivalent instance of size $\mathcal{O}(k^7)$ in polynomial time, where k is the desired size of the tracking set.[3] In this paper we give improved kernels for general graphs and planar graphs. We also give the first hardness result for Tracking Paths with respect to parameterized complexity.

Our Contributions and Methods. We give a quadratic kernel for Tracking Paths on general graphs, which is a major improvement from the $\mathcal{O}(k^7)$ kernel given in [4]. We also give a linear kernel for Tracking Paths on planar graphs. Further we prove that deciding if there exists a tracking set of size at most $n-k$, where n is the number of vertices in the graph, is W[1]-hard.

Given an instance (G, s, t, k), we give a polynomial time algorithm that either determines that (G, s, t, k) is a NO instance or produces an equivalent instance with $\mathcal{O}(k^2)$ vertices and $\mathcal{O}(k^2)$ edges, where k is the size of a desired tracking set. This polynomial time algorithm is called a *kernelization algorithm* and the reduced instance is called a *kernel*. For more details about parameterized complexity and kernelization we refer to monographs [16,15].

The kernelization algorithm works along the following lines. Let (G, s, t, k) be an input instance to Tracking Paths. We start the algorithm with a 2-approximate solution for Feedback Vertex Set (FVS) and makes of a useful structural result that we give. Specifically, we prove that if there exists an induced subgraph in G which consists of a tree with all of its leaves adjacent to a single vertex v, then the size of a minimum tracking set for G is at least the number of neighbors of v in this tree minus one. Then if S is an FVS of size at most $2k$, we give bounds on different types of vertices in $G \setminus S$, based on how they share neighbors in S. Combining all these bounds we show the existence of a quadratic kernel for general graphs. We also give a linear kernel for planar graphs. A planar graph is a graph that can be embedded on a two-dimensional plane i.e. it can be drawn on a two dimensional plane in such a way that its edges intersect only at their endpoints and do not cross over each other. Eppstein, Goodrich, Liu and Matias [17] studied Tracking Paths for planar graphs and showed that Tracking Paths remains NP-complete when the graph is planar, and gave a 4-approximation algorithm for this setting. Our linear kernel uses the bound on the number of faces with respect to the size of an optimal tracking set for a planar graph, given in [17].

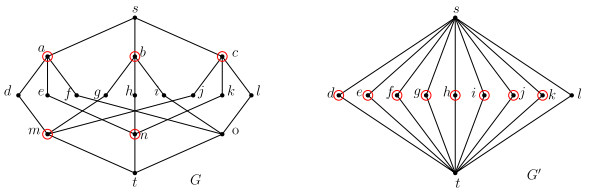

Fig. 1. Graph G' is a minor of graph G. The set of circled vertices represents a minimum tracking set in each graph.

In parameterized complexity it is common to identify tractable parameterizations. For a graph on n vertices, n is a trivial upper bound for the size of a tracking set, so from the perspective of 'distance to triviality' we consider the question of whether we can find a tracking set with $n-k$ trackers. We prove that finding a tracking set of size at most $n-k$ is W[1]-hard on a graph with n vertices, where the parameter is k.

Although a tracking set is also a feedback vertex set, both are fundamentally very different. A graph may have a small FVS but the tracking set may be arbitrarily larger than the FVS. Moreover, Tracking Paths is more demanding as a problem compared to the classic covering problems studied in graph theory. While covering problems aim at hitting a particular type of structure in graphs, Tracking Paths requires distinguishing each s-t path uniquely using a small set of vertices. Here we mention an important property about this problem that distinguishes it from many other covering problems. It can be shown using an example that the problem is not closed on minors. See Fig. 1.

Observe that graph G' can be obtained from graph G by contracting the following edges: (s, a), (s, b), (s, c), (m, t), (n, t), (o, t). However note that G' requires at least 8 trackers while G can be tracked with just 5 trackers. Since Tracking Paths is not closed under minors, the well known *graph minor theorem* does not apply to it [23,22,7]. Hence, the problem is inherently different from the standard covering problems and the well known techniques of finding some specific obstructions and devising algorithms to hit/cover them, shall not work for the case of Tracking Paths.

Even proving that Tracking Paths is in NP is non-trivial. See [4] for details. A combinatorial generalization of Tracking Paths has been studied in [5], where the input is a set system, and it is required to find a set of elements from the universe that have a unique intersection with each set in the family. The problem has been shown to be a dual of the test cover problem. Tracking Paths was proven to be polynomial time solvable for chordal graphs, tournament graphs and for the case when edges are used as trackers instead of vertices [12], [13]. It is also known from [17] that it is polynomial time solvable for graphs of bounded clique-width (when the clique decomposition is given in advance). Bilò, Gualà, Leucci and Proietti [8] discussed results on Tracking Shortest Paths with multiple source-destination pairs, presenting an $\tilde{\mathcal{O}}(\sqrt{n})$-approximation algorithm for general graphs and NP-hardness for cubic planar graphs. They also give an FPT algorithm parame-

[3] Throughout the paper we assume k to be a non-negative integer.

terized by the maximum number of vertices equidistant from source, for the single source-destination pair scenario. Recent work on Tracking Shortest Paths also includes results on quadratic kernels for general graphs and linear kernel for planar graphs [10]. With regard to approximation, it was recently shown that there exists 6-approximation algorithm for Tracking Paths [9]. A related problem, *Identifying Path Cover* has been discussed in [19] and [20]. *Identifying Path Cover* requires finding a set of paths that cover all the vertices in a graph and uniquely identify each vertex by inclusion in a distinct set of paths.

1.1. Roadmap

Section 2 explains the terms and notations used in the paper. Section 3 analyzes graph structures (disjoint paths, tree-sink structure) that have a strict lower bound in terms of the number of trackers required in them if they appear as subgraphs (not necessarily induced) in the input graph. The lemmas in this section form the basis of the reduction rules used in Sections 4 and 5 to get the respective kernels. Section 4 gives an $\mathcal{O}(k^2)$ kernel for general graphs, where k is the desired size of a tracking set. We start by finding a 2-approximate FVS S for the input graph G. Next the vertices in $V(G) \setminus S$ are categorized on the basis of how they share neighbors in S with other vertices in $V(G) \setminus S$. Section 5 derives an $\mathcal{O}(k)$ kernel for planar graphs using a bound on the number of faces of a planar graph after some preprocessing. Section 6 discusses the W[1]-hardness for the problem of finding whether a graph can be tracked with $n-k$ trackers, where n is the number of vertices in the graph. Finally Section 7 summarizes the results in the paper with some open problems.

2. Preliminaries

A kernelization algorithm is typically obtained using what are called *reduction rules*. These rules transform a given parameterized instance in polynomial time to an equivalent instance, preferably with some bound on the size of the new instance. A rule is said to be *safe* if the resulting graph is a YES instance if and only if the original instance is a YES instance.

Throughout the paper, we assume graphs to have no self loops or multi-edges. When considering tracking set for a graph $G = (V, E)$, we assume that the given graph is an s-t graph, i.e. the graph contains a unique source $s \in V$ and a unique destination $t \in V$ (both s and t are known), and we aim to find a tracking set that can distinguish all simple s-t paths. For a graph/subgraph G, $V(G)$ represents the vertex set of G and $E(G)$ represents the edge set of G, i.e. the set of edges whose both endpoints lie in $V(G)$. If $a, b \in V$ then, unless otherwise stated, $\{a, b\}$ represents the set of vertices a and b, and (a, b) represents an edge between a and b. For a vertex $v \in V$, *neighborhood* of v is denoted by $N(v) = \{x \mid (x, v) \in E\}$. We use $deg(v) = |N(v)|$ to denote the degree of vertex v. For a graph G, we use $G' \subseteq G$ to denote that G' is a subgraph of G. For a vertex $v \in V$ and a subgraph G', $N_{G'}(v) = N(v) \cap V(G')$. Similarly, for a vertex $v \in V$ and a set of vertices V', we use $N_{V'}(v)$ to denote $N(v) \cap V'$. For a subset of vertices $V' \subseteq V$ we use $N(V')$ to denote $\bigcup_{v \in V'} N(v)$. With slight abuse of notation we use $N(G')$ to denote $N(V(G'))$. For a graph G and a set of vertices $S \subseteq V(G)$, $G - S$ denotes the subgraph induced by the vertex set $V(G) \setminus S$. If S is a singleton, we may use $G - x$ to denote $G - S$, where $S = \{x\}$. We use $[m]$ to denote the set of integers $\{1, \ldots, m\}$.

A path in a graph G is a subgraph of G defined as $P = \{v_1.e_1.v_2.e_2.v_3 \ldots v_n\}$, where $v_i \in V(G)$, $e_i = (v_i, v_{i+1}) \in E(G)$, and $v_i \neq v_j$, $e_i \neq e_j$ if $i \neq j$. Let P_1 be a path between vertices a and b, and P_2 be a path between vertices b and c, such that $V(P_1) \cap V(P_2) = \{b\}$. Then we use $P_1 \cdot P_2$ to denote the path between a and c, formed by concatenating paths P_1 and P_2 at b. Two paths P_1 and P_2 are said to be *vertex disjoint* if their vertex sets do not intersect except possibly at the endpoints, i.e. $V(P_1) \cap V(P_2) \subseteq \{a, b\}$, where a and b are the starting and endpoints of the paths. By distance we mean length of the shortest path, i.e. the number of edges in that path. For a graph $G = (V, E)$, an FVS is a set of vertices $S \subseteq V$ such that $G - S$ is a forest.

3. Structural properties

In this section we analyze the problem of distinguishing paths in some specific graph structures along with providing some basic preprocessing steps. We start by recalling some reduction rules and basic results from the previous work on Tracking Paths, followed by giving some new ones. Later we give some important lemmas based on tree-like structures, which form the base for vertex counting arguments for the kernels we give in subsequent sections. Following reduction rules are applied exhaustively as long as they are applicable.

Reduction Rule 1. *Banik et al. [4] If there exists a vertex or an edge that does not participate in any s-t path then delete it.*

In the rest of the paper we assume that each vertex and edge participate in at least one s-t path.

Reduction Rule 2. *If $V \setminus \{s, t\} = \emptyset$, then return a trivial YES instance. Else, if degree of s (or t) is 1 and $N(s) \neq t$ ($N(t) \neq s$), then delete s (t) and label the vertex adjacent to it as s (t).*

Now we recall two reduction rules from [17] that help us bound the number of degree two vertices in the graph.

Reduction Rule 3. *Eppstein et al. [17] If there exist $a, b, c \in V(G)$ such that $deg(b) = deg(c) = 2$, $b, c \notin \{s, t\}$, $(a, c) \notin E(G)$, and $N(b) = \{a, c\}$, then delete b and introduce an edge between a and c.*

Reduction Rule 4. *Eppstein et al. [17] If there exist $a, b, c \in V(G)$ such that $N(b) = \{a, c\}$ and $(a, c) \in E(G)$ and $b \notin \{s, t\}$, then mark b as a tracker, delete b from G and set k to $k - 1$.*

Next we recall a monotonicity lemma and a corollary from [4], which says that if a subgraph of G cannot be tracked with k trackers, then G cannot be tracked with k trackers either.

Lemma 3.1. *Banik et al. [4] Let $G = (V, E)$ be a graph and $G' = (V', E')$ be a subgraph of G such that $\{s, t\} \subseteq V'$. If T is a tracking set for G and T' is a minimum tracking set for G', then $|T'| \leq |T|$.*

If the above Reduction Rules are applied then the block-cut tree of the graph is a path i.e. each block is a biconnected component [21]. A block-cut tree of a graph G is a tree where each node represents either a biconnected component or cut vertex of G; and a node representing a cut vertex v in G is adjacent to all biconnected components in G that contain v. Note if the block-cut tree is not a path then at least one vertex shall not be participating in any s-t path. Observe that the cut vertices cannot help uniquely identify any s-t path. Hence we can consider each biconnected component individually as a subproblem in itself.

Observation 1. *If Reduction Rules 1, 2, 3 and 4 are applied then the graph is a chain of biconnected components.*

Observation 2. *The tracking set for each biconnected component can be computed individually and tracking set for G is the union of tracking set of each of the biconnected components.*

Definition 1. Let $G' \subseteq G$ be a subgraph. If $u, v \in V(G')$ is a pair of distinct vertices then, for the subgraph G', u is a **local source** and v is a **local destination** if the following hold: (a) there exists a path in G from s to u, say P_{su}, and another path from v to t, say P_{vt}, (b) $V(P_{su}) \cap V(P_{vt}) = \emptyset$, (c) $V(P_{su}) \cap V(G') = \{u\}$ and $V(P_{vt}) \cap V(G') = \{v\}$.

Note that a subgraph may have multiple local source-destination pairs.

Lemma 3.2. *Banik et al. [4][Rephrased] If each vertex and edge in graph G participates in an s-t path, then for a subgraph $G' \subseteq G$ containing at least one edge, it holds that G' contains a local source and a local destination.*

In the rest of the paper the phrase 'subgraph cannot be tracked by k trackers' implies that the paths between a local source and destination in a subgraph cannot be tracked with k trackers. Due to Lemma 3.1 and Lemma 3.2, we have the following two corollaries.

Corollary 1. *Banik et al. [4] If a subgraph of G that contains both s and t cannot be tracked by k trackers, then G cannot be tracked by k trackers either.*

Corollary 2. *If there exists a subgraph G' of G, and there exists a pair of vertices $u, v \in V(G')$, such that u is a local source for G' and v is a local destination for G', and all paths between u and v in G' cannot be tracked by at most k trackers, then G cannot be tracked by at most k trackers.*

Next corollary forms a starting point for the kernelization algorithms.

Corollary 3. *Banik et al. [4] If every vertex and edge in a graph G is part of an s-t path in G then a tracking set T for G is also an FVS for G.*

For the rest of the paper, we assume that the input graph has already been preprocessed using Reduction Rules 1, 2, 3 and 4, and hence the following holds:

1. All vertices and edges in the graph participate in some s-t path.
2. Degree of all vertices in the graph is at least two, and each vertex of degree two has both its neighbors with degree three or higher.
3. There exist at least two s-t paths in the graph.

3.1. Vertex disjoint paths

Here we give a bound on the number of vertex disjoint paths that can exist between a pair of vertices in a graph G, given that G can be tracked with at most k trackers. While in [4] it is proven that there can exist at most $k+3$ vertex disjoint paths between a pair of vertices in G, we improve the bound to $k+1$. The new bound allows easy analysis and computation in future lemmas.

Lemma 3.3. *If $u, v \in V$ and there exists more than $k+1$ vertex disjoint paths between u and v, and the graph induced by these $k+1$ paths along with u and v has u as a local source and v as a local destination, then G cannot be tracked with at most k trackers.*

Proof. For a contradiction assume that there exist $k+2$ vertex disjoint paths $\mathcal{P} = \{P_1, \ldots, P_{k+2}\}$ between a pair of vertices u and v in G i.e. u, v are end points for every path in $\mathcal{P}$, and G can be tracked with at most k trackers. Let G' be the subgraph induced by $V(\mathcal{P})$. Since u is a local source for G' and v is a local destination for G', there exists a path P_{su} that starts at s and ends at u and does not contain any vertex from $G' - \{u, v\}$, and there exists a path P_{vt} that starts at v and ends at t and does not contain any vertex from $G' - \{u, v\}$. See Fig. 2. Consider a pair of paths $P_i, P_j \in \mathcal{P}$. Let P_i and P_j do not contain any trackers (except for u and v). Now consider the s-t paths $P'_i = P_{su} \cdot P_i \cdot P_{vt}$ and $P'_j = P_{su} \cdot P_j \cdot P_{vt}$. Note that P'_i and P'_j contain the same sequence of trackers. Since these paths differ only in the vertices on paths P_i and P_j, at least one vertex (except u and v) either on P_i or P_j has to be a tracker. Thus as long as there are two paths in $\mathcal{P}$ without any trackers, there will be two s-t paths with the same sequence of trackers. Hence, at least $k+1$ paths in $\mathcal{P}$ need a tracker on them. Due to Corollary 2, we know that if G' cannot be tracked with at most k trackers then G cannot be tracked with at most k trackers. This contradicts the initial assumption, and hence completes the proof. □

Lemma 3.4. *If there exists two vertices $u, v \in V$ such that there exists more than $k+1$ vertex disjoint paths between u and v, then G cannot be tracked with at most k trackers.*

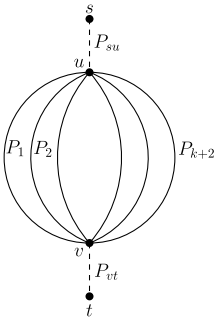

Fig. 2. Vertex disjoint paths between a local source and destination.

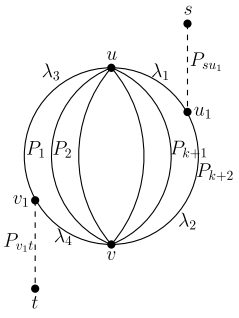

Fig. 3. Vertex disjoint paths between a pair of vertices.

Proof. For a contradiction assume that there exist $k+2$ vertex disjoint paths $\mathcal{P} = \{P_1, \ldots, P_{k+2}\}$ between a pair of vertices u and v in G i.e. u, v are end points for every path in $\mathcal{P}$, and G can be tracked with at most k trackers. Let G' be the subgraph induced by $V(\mathcal{P})$. Due to Lemma 3.2, there exists a local source, say u_1, and a local destination, say v_1, in G'. We consider various cases possible based on the positions of u_1 and v_1 in G'.

- When $u = u_1$ and $v = v_1$, or $u = v_1$ and $v = u_1$. Both of these cases are symmetric to each other, and have been proven in Lemma 3.3.
- When $\{u, v\} \cap \{u_1, v_1\} = \emptyset$. First we consider the case when u_1 and v_1 lie on different paths in $\mathcal{P}$. See Fig. 3. We denote the path between u and u_1 (subpath of P_{k+2}) by λ_1, the path between u_1 and v (subpath of P_{k+2}) by λ_2, the path between u and v_1 (subpath of P_1) by λ_3, and the path between v and v_1 (subpath of P_1) by λ_4. We denote the paths in $\mathcal{P} \setminus \{P_1, P_{k+2}\}$ by $\mathcal{P}'$. Any s-t path in G that passes through G' will be one among the following types:

  1. $P_{su_1} \cdot \lambda_1 \cdot P_i \cdot \lambda_4 \cdot P_{v_1t}$, where $P_i \in \mathcal{P}'$
  2. $P_{su_1} \cdot \lambda_2 \cdot P_i \cdot \lambda_3 \cdot P_{v_1t}$, where $P_i \in \mathcal{P}'$
  3. $P_{su_1} \cdot \lambda_1 \cdot \lambda_3 \cdot P_{v_1t}$
  4. $P_{su_1} \cdot \lambda_2 \cdot \lambda_4 \cdot P_{v_1t}$

  Consider the first two cases. Let G'' be the graph induced by $\mathcal{P}'$. Observe that u and v are local source and destination for G'', since there exists a path $P_{su_1} \cdot \lambda_1$ from s to u, and a path $P_{v_1t} \cdot \lambda_4$ from v to t, and these paths intersect with G'' only at u and v. Since there are k paths between u and v in G'', due to Lemma 3.3, these require at least $k-1$ trackers in $V(\mathcal{P}') \setminus \{u, v\}$. If each of the k paths in $\mathcal{P}'$ has a tracker, the paths indicated in cases 3,4 have the same sequence of trackers, and this contradicts the assumption that G has a tracking set of size k. Else, without loss of generality, let P_{k+1} be the path in $\mathcal{P}'$ that is left without a tracker.

  Cases 3,4 denote two vertex disjoint paths between u_1 and v_1 along P_1 and P_{k+2}. Hence, due to Lemma 3.3, there must be a tracker on either $V(\lambda_1) \cup V(\lambda_3) \setminus \{u_1, v_1\}$ or $V(\lambda_2) \cup V(\lambda_4) \setminus \{u_1, v_1\}$. We consider following cases:
  - There exists a tracker in $V(\lambda_1) \setminus \{u_1, v_1\}$: Paths $P_{su_1} \cdot \lambda_1 \cdot \lambda_3 \cdot P_{v_1t}$ and $P_{su_1} \cdot \lambda_1 \cdot P_{k+1} \cdot \lambda_4 \cdot P_{v_1t}$ contain the same set of trackers.
  - There exists a tracker in $V(\lambda_2) \setminus \{u_1, v_1\}$: Paths $P_{su_1} \cdot \lambda_2 \cdot P_{k+1} \cdot \lambda_3 \cdot P_{v_1t}$ and $P_{su_1} \cdot \lambda_2 \cdot \lambda_4 \cdot P_{v_1t}$ contain the same set of trackers.
  - There exists a tracker in $V(\lambda_3) \setminus \{u_1, v_1\}$: Paths $P_{su_1} \cdot \lambda_1 \cdot \lambda_3 \cdot P_{v_1t}$ and $P_{su_1} \cdot \lambda_2 \cdot P_{k+1} \cdot \lambda_3 \cdot P_{v_1t}$ contain the same set of trackers.
  - There exists a tracker in $V(\lambda_4) \setminus \{u_1, v_1\}$: Paths $P_{su_1} \cdot \lambda_1 \cdot P_{k+1} \cdot \lambda_4 \cdot P_{v_1t}$ and $P_{su_1} \cdot \lambda_2 \cdot \lambda_4 \cdot P_{v_1t}$ contain the same set of trackers.

  All the above cases contradict the assumption that G can be tracked with at most k trackers.

  Next we consider the case when both u_1 and v_1 lie on the same path in $\mathcal{P}$. Without loss of generality assume that u_1 and v_1 both lie on P_1. Here note that there exists one path between u_1 and v_1 that is a strict subpath of P_1, and the remaining paths between u_1 and v_1 pass through $\mathcal{P} \setminus P_1$, via vertices u and v. Observe that u and v are a local source and destination for the subgraph G''' induced by $V(\mathcal{P} \setminus P_1)$. Since there are $k+1$ paths between u and v in the subgraph G''', due to Lemma 3.3, at least k trackers are required in $V(G''')$. If there are $k+1$ trackers in $V(G''')$, it contradicts the assumption that G can be tracked with at most k trackers. If there are k trackers in $V(G''')$, without loss of generality, let P_2 be the path without any tracker. Now observe that there are two paths between u_1 and v_1, the one that does not pass through u and v (subpath of P_1) and the one that passes through u and v, through P_2, that do not have any trackers on them. Due to Lemma 3.3, at least one tracker is required on one of these paths. Hence we have a contradiction to the assumption that the graph can be tracked with at most k trackers.
- When $u = u_1$ and $v \neq v_1$, or, $u \neq u_1$ and $v = v_1$. Consider $u = u_1$, and $v \neq v_1$. This case can be argued similar to the second case, except that now $\lambda_1 = u = u_1$. Similarly, if $u \neq u_1$, and $v = v_1$, the case is similar to the second case, except that now $\lambda_4 = v = v_1$.

Note that the case when $u_1 = v_1$, is not possible after application of Reduction Rule 1. Correctness of the proof follows from Corollary 1. Hence the lemma holds. □

Next we give a reduction rule based on Lemma 3.4.

Reduction Rule 5. *Let G' be a subgraph of G, consisting of a pair of vertices a, b such that a, b have i common neighbors, each of degree two. If a is a local source for G' and b is a local destination for G', then arbitrarily mark $i-1$ among the i vertices of degree two as trackers and delete them. If $i > k+1$ return a NO instance, else set k to $k-i-1$.*

Lemma 3.5. *Reduction Rule 5 is safe and can be implemented in polynomial time.*

Proof. Let V_i be the set of i vertices of degree two that are adjacent to a and b and let V_{i-1} be the set of $i-1$ vertices that were marked as trackers and deleted. Let G' be the newly created graph after the deletion of V_{i-1}. We claim that G' has a tracking set of size $k-i-1$ if and only if G has a tracking set of size k. Suppose G' has a tracking set T' of size $k-i-1$. If we add the vertices of V_{i-1} back to G' along with their edges, there exist i vertex disjoint paths between a and b. Since a and b are the local source and destination, due to Lemma 3.4 at least $i-1$ trackers are required on the vertices of V_i. We mark all the vertices in V_{i-1} as trackers. Now all paths between a and b are tracked. Since all other paths were already being tracked by T' in G', $T' \cup V_{i-1}$ is a tracking set of size k for G.

In the other direction let T be a tracking set of size k in G. Let $x \in V_i \setminus V_{i-1}$. We claim that G' has a tracking set $T' = V(G') \cap T$ of size $k-i-1$. Suppose not. Then there exist two s-t paths, say P_1 and P_2, in G' that have the same sequence of trackers. Observe that this implies that P_1 and P_2 are also two paths with the same sequence of trackers in G. Note that the trackers on vertices in V_{i-1} cannot be used to distinguish P_1 and P_2, as that would leave some untracked paths between a and b in G. Thus T is not a tracking set for G, which is a contradiction. This completes the proof of safeness of Reduction Rule 5.

In order to implement Reduction Rule 5, we consider each pair of vertices $u, v \in V(G)$, and compare all their neighbors, to check for common neighbors of degree two. This can be done in $\mathcal{O}(n^4)$ time. Hence the rule can be applied in polynomial time. □

3.2. Tree-sink structure

In this section we discuss a specific graph structure, namely the *tree-sink structure*, and prove a lower bound for the number of trackers required if such a structure exists in an s-t graph.

Definition 2. A **tree-sink structure** in a graph G is a subgraph G' such that $V(G') = V(R) \cup \{x\}$, where R is a tree with at least two vertices, and all of its leaves are adjacent to x. Here R is the **tree** while x is the **sink** of the tree-sink structure.

Note that x may or may not be adjacent to the non-leaf vertices of R. See Fig. 4. We prove that if the sink x is adjacent to more than $k+1$ vertices in R, then G cannot be tracked with at most k trackers. We start with a simple case when the graph G itself is a tree-sink structure and either s or t is the sink.

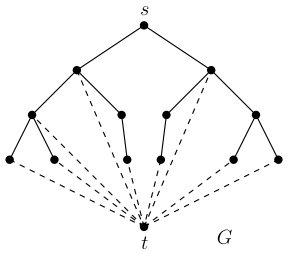

Fig. 4. Tree-sink structure: Solid lines are edges of tree, and dashed lines are edges between the sink t and vertices of the tree.

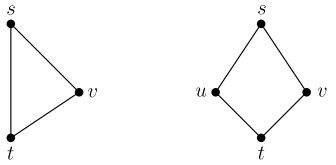

Fig. 5. Possible cases when the sink t has two neighbors in the tree induced by $G-t$.

Lemma 3.6. *Let G be an s-t graph that forms a tree-sink structure with $x \in V(G)$ as the sink and $x \in \{s, t\}$. If $|N(x)| = \delta$, then $\delta - 1$ trackers are required in G, and these trackers have to be in $V(R)$, where R is the tree induced by $G - x$.*

Proof. Without loss of generalization we assume that $x = t$. We root R at the source vertex s. Consider that graph G has already been preprocessed using Reduction Rules 1, 2 and 3.

We prove the lemma by induction on the value of δ. Observe that due to Reduction Rule 1, 2 and 3, $\delta = 1$ is not possible. Thus the base case for induction is when $\delta = 2$. Note that in this case G is either a triangle or a four cycle. See Fig. 5. Consider the case when G is a triangle. Due to Reduction Rule 4, the vertex $v \in V(G) \setminus \{s, t\}$ is marked as a tracker and deleted. Consider the case when G is a four cycle. Observe that there exist two vertices, say u, v, of degree two each, adjacent to s and t. Due to Reduction Rule 5, one among u and v is marked as a tracker and deleted. Note that in both the cases, after application of the corresponding reduction rules, G comprises only the edge (s, t). Due to Reduction Rule 2, this is a trivial YES instance. Hence, when $\delta = 2$, exactly one tracker is required in G. This proves that the claim holds for the base case.

Next, for induction hypothesis, we assume that the claim holds for $\delta = i$, i.e. if the sink is adjacent to i vertices, then $i-1$ trackers are required in G. Consider the case when $\delta = i+1$. Note that here $\delta \geq 3$. Due to Reduction Rule 1, all leaves in R are adjacent to t, R being the tree induced by $G - t$. Consider a leaf vertex, say $v_1 \in V(R)$, that is at maximum distance from s. Since $deg(v_1) = 2$, due to Reduction Rule 3, the degree of its parent node, say v_p, is at least 3. Thus either v_p has another child node, or v_p is adjacent to t. We analyze both the possibilities:

- *Case I*: v_p has another child node, say v_2. Since v_1 is at maximum distance possible from s, v_2 is a leaf node in R. Observe that the graph G' induced by v_1, v_2, v_p and t has v_p as a local source and t as a local desti-

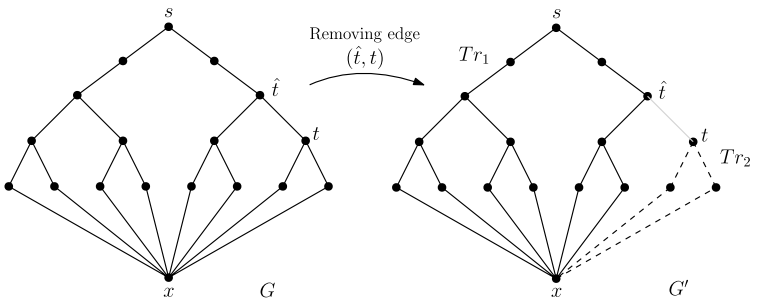

Fig. 6. Removing edge $(\hat{t}, t)$ from G creates two tree-sink structures in G', with trees R_1 (shown with solid lines) and R_2 (shown in dashed lines) and sink x.

nation, and $deg(v_1) = deg(v_2) = 2$. Due to Reduction Rule 5, either v_1 or v_2 will be marked a tracker and deleted. This reduces the value of δ from $i+1$ to i, while using one tracker.

- *Case II*: v_p is adjacent to t. Observe that v_1, v_p and t form and triangle and $deg(v_1) = 2$. Due to Reduction Rule 4, v_1 will be marked as a tracker and deleted. This reduces the value of δ from $i+1$ to i, while using one tracker.

Now $\delta = i$. Due to the induction hypothesis, we know that when $\delta = i$, then $i-1$ trackers are required in G. Since we already used a tracker in both the above cases, the total number of trackers required when $\delta = i+1$, is i. Since the sink is t itself, all the trackers need to be in $V(R)$. This completes the proof. □

Next we give a corollary which makes the above lemma more usable for the sake of our future arguments.

Corollary 4. *Let G be a graph and G' be a subgraph of G such that G' induces a tree-sink structure with $v \in V(G')$ as its sink. If $|N(v) \cap V(G')| = \delta$, and v is either a local source or a local destination for G', then the size of a tracking set for G is at least $\delta - 1$. Further these $\delta - 1$ trackers need to be in $V(G') \setminus \{v\}$.*

Proof. Consider the subgraph G'. Without loss of generality, we assume that v is a local destination for G'. Let $u \in V(G')$ be a local source corresponding to the local destination v. Due to Lemma 3.6, we have that $\delta - 1$ trackers are required in $V(G') \setminus \{v\}$ to track all paths between u and v. From Corollary 2, if in a subgraph all paths between a local source and destination cannot be tracked with k trackers then the graph cannot be tracked with k trackers. Hence if $k < \delta - 1$, then G cannot be tracked with at most k trackers. Thus the size of a tracking set for G is at least $\delta - 1$. It follows from Lemma 3.6 that these $\delta - 1$ trackers need to be in $V(G') \setminus \{v\}$. □

The next lemma generalizes the result in Corollary 4. We prove that regardless of where s and t lie in graph G, if G forms a tree-sink structure, then the size of the tracking set for G is at least the number of neighbors of the sink in the tree minus one.

Lemma 3.7. *If an s-t graph G forms a tree-sink structure such that $x \in V(G)$ is the sink and $G - x$ induces a tree and $|N(x)| = \delta$, then the size of a tracking set for G is at least $\delta - 1$, and at least $\delta - 2$ trackers are required in $G - x$.*

Proof. Let R be the tree induced by $V(G \setminus \{x\})$. The case when $x \in \{s, t\}$ has been proven in Lemma 3.6. Consider the case when $s, t \in V(R)$. We start by rooting the tree at s. Now create a graph G' by removing the edge between t and its parent vertex, say $\hat{t}$, in R. Observe that in G', there exists a tree, say R_1 that can be considered rooted at s, consisting of all those vertices in $V(R)$ that are not descendants of t in R, with all its leaves adjacent to the vertex x. There exists another tree, say R_2, rooted at t, consisting of all of its descendants in R, with all of its leaves adjacent to x. See Fig. 6. We denote the graph induced by $V(R_1) \cup \{x\}$ by G_1, and the graph induced by $V(R_2) \cup \{x\}$ by G_2. Let δ_1 be the number of leaves in R_1, and δ_2 be the number of leaves in R_2. Note that $\delta_1 + \delta_2 = \delta$.

Note that x is a local destination for G_1. Hence by Corollary 4, since R_1 has δ_1 many leaves, the size of a tracking set for G is at least $\delta_1 - 1$, and all these trackers must be in $V(R_1 - x)$.

Note that x is a local source for G_2. Hence by Corollary 4, since R_2 has δ_2 many leaves, the size of a tracking set for G is at least $\delta_2 - 1$, and all these trackers must be in $V(R_2 - x)$.

If there exists at least $\delta_1 + \delta_2 - 1$ trackers in G, then the lemma holds. Else there exist $\delta_1 - 1$ trackers in $V(R_1 - x)$ and $\delta_2 - 1$ trackers in $V(R_2 - x)$. Hence, there exists exactly one path in G_1, say P_1, from s to x that does not contain any trackers, and exactly one path in G_2, say P_2, from x to t without trackers. Consider the path $P' = \{s\} \cdot P_1 \cdot \{x\} \cdot P_2 \cdot \{t\}$. Note that if G contains a total of $\delta_1 + \delta_2 - 2$ trackers, then x is not a tracker and hence P' does not contain any trackers. Recall the edge e that was initially removed between t and its parent, $\hat{t}$, in R. Consider the path in G_1 from s to $\hat{t}$, say $P_{s\hat{t}}$. We consider the following two cases.

- $P_{s\hat{t}}$ is a subpath of the path P_1. Consider the paths $\{s\} \cdot P_1 \cdot \{x\} \cdot P_2 \cdot \{t\}$, and $\{s\} \cdot P_{s\hat{t}} \cdot \{t\}$. Observe that both these paths have no trackers. Hence one more tracker is needed, either in $V(P_1)$ or $V(P_2)$ in order to distinguish them in G.
- $P_{s\hat{t}}$ is not a subpath of the path P_1. If $P_{s\hat{t}}$ does not have a tracker, both the paths $\{s\} \cdot P_1 \cdot \{x\} \cdot P_2 \cdot \{t\}$ and $\{s\} \cdot P_{s\hat{t}} \cdot \{t\}$ do not contain any trackers. If $P_{s\hat{t}}$ has a tracker, let $t_r \in V(P_{s\hat{t}})$ be the tracker that is closest

to $\hat{t}$. Since $\delta - 1$ is the minimum number of trackers required in R_1, there exists a path from t_r to x (and not passing through s) in G_1 that does not contain any trackers. Lets denote this path by $P_{t_r x}$. Let P_{st_r} be the path from s to t_r that is a subpath of $P_{s\hat{t}}$. Now observe that paths $\{s\} \cdot P_{st_r} \cdot P_{t_r x} \cdot \{x\} \cdot P_2$ and $\{s\} \cdot P_{s\hat{t}} \cdot \{t\}$ have the same set of trackers. Hence in both the cases discussed one more tracker is required in G.

Thus the total number of required trackers in G is at least $\delta_1 + \delta_2 - 2 + 1$, i.e. $\delta - 1$. Since the sink can be a tracker as well, a tree-sink structure requires at least $\delta - 2$ trackers in the vertex set of the tree. □

Lemma 3.7 along with Corollary 1 gives us the following corollary.

Corollary 5. *In a graph G, if there exists a subgraph G' and a vertex $v \in V(G')$, such that G' forms a tree-sink structure with v as a sink, and $|N(v) \cap V(G')| = \delta$ then the size of a tracking set for G is at least $\delta - 1$. Further at least $\delta - 2$ trackers are required to be in $V(G') \setminus \{v\}$.*

Observe that if a vertex v in a subgraph G' has δ neighbors in G' and $G' - v$ is connected, then there exists a tree-sink structure in G' such that v is the sink that is adjacent to δ vertices of the tree. Hence we have the following two corollaries following from Corollary 5.

Corollary 6. *Let G' be subgraph of a reduced graph G and $v \in V(G')$ be a vertex such that $G' - v$ is connected and $N_{G'}(v) = \delta$. Then any tracking set of G contains at least $\delta - 1$ trackers and $\delta - 2$ of them are required to be in $G' - v$.*

Corollary 7. *If there exists a non-cut vertex in a reduced graph G with degree δ then the size of a tracking set for G is at least $\delta - 2$.*

Reduction Rule 6. *If there exists a non-cut vertex of degree more than $k + 2$, return a trivial NO-instance.*

The safeness of Reduction Rule 6 follows from Corollary 7. Henceforth we assume that the input graph has already been preprocessed by the reduction rules stated so far.

4. Quadratic kernel for general graphs

In this section we show that an instance (G, k) of Tracking Paths can be reduced to an equivalent instance (G', k') such that if (G, k) is a YES instance then $|V(G')| = \mathcal{O}(k'^2)$, $|E(G')| = \mathcal{O}(k'^2)$ and $k' \leq k$. We achieve this by expanding on the notion of tree-sink structures (see 3.2), allowing us to bound the maximum degree among non-cut vertices (Corollary 7). We start by applying Reduction Rules 1, 2, 3, 4, 5 and 6. If the instance is not termed a NO instance by any of the reduction rules, then recall that the resultant graph is a series of biconnected components and tracking set for each component can be computed individually (Observations 1, 2). Recall from Corollary 3, that the size of a minimum tracking set T for G is at least the size of a minimum FVS for G. We start by finding a 2-approximate feedback vertex set S, using [2]. From Corollary 3, we have the following reduction rule.

Reduction Rule 7. *Banik et al. [4] Apply the algorithm of [2] to find a 2-approximate solution S for* Feedback Vertex set. *If $|S| > 2k$, then return that the given instance is a NO instance.*

Observe that $\mathcal{F} = G \setminus S$ is a forest. Now we try to bound the number of vertices and edges in $\mathcal{F}$ for the case when all s-t paths in G can be tracked with at most k trackers. In general, by 'tree' we mean a tree in the forest $\mathcal{F}$. When referring to a tree-sink structure, by 'tree' we mean the tree that forms the tree-sink structure. After applying Reduction Rule 7, the kernelization algorithm applies the following rule.

Reduction Rule 8. *If the number of vertices (resp. edges) is more than $4k^2 + 9k - 5$ (resp. $5k^2 + 10k - 6$), return a trivial NO-instance.*

It can be seen that the rule is applicable in polynomial time. Next, we show the safeness of Reduction Rule 8.

Lemma 4.1. *If S is an FVS for a reduced graph G then $|V(G - S)| \leq 4|X| - 5$, where X is the cut set defined by $(S, G - S)$ consisting of edges with endpoints in both S and $G - S$.*

Proof. Let V_1, V_2, V_3 be the set of vertices in $G - S$ with degrees one, two and three or greater respectively. Observe that since $G - S$ is a forest, $|V_1| \leq |X|$ (due to Reduction Rules 1, 2) and $|V_3| \leq |V_1| - 2 = |X| - 2$. We use V^* to denote the vertices that are incident with the edge set X. Let V_2' be the set of vertices in V_2 that are not adjacent to S. Observe that $V_2 \setminus V_2'$ is bounded by $|X|$. Due to Reduction Rules 3, 4, it is known that V_2' induces an independent set. Hence, $|V_2'|$ is bounded by $|V_1 + V_3| - 1$ and the number of vertices in $V^* \cap V_2$, since vertices of V_2' can alternate with vertices of $V^* \cap V_2$. It follows that $|V_2'| \leq |V_1| + |V_3| + |V_2 \cap V^*| - 1$. Thus we have $|V(G - S)| \leq 2|V_1| + 2|V_2 \cap V^*| + 2|V_3| - 1$. Since $|V_1| + |V_2 \cap V^*| \leq |X|$, we have $|V(G - S)| \leq 4|X| - 5$. □

Lemma 4.2. *Let G be a biconnected reduced graph, with start s and finish t. Then, G has at most $4OPT^2 + 9OPT - 5$ vertices and at most $5OPT^2 + 10OPT - 6$ edges, where OPT denotes the size of an optimal tracking set of G.*

Proof. Let T^* be an optimal tracking set of G, i.e., $|T^*| = OPT$. Since every tracking set is an FVS of a reduced graph (see Corollary 3), we can apply Lemma 4.1 and obtain $|V(G - T^*)| \leq 4|X| - 5$, where X is the set of edges with endpoints in both T^* and $G - T^*$. By the fact that G is biconnected and by Corollary 7, G has maximum degree $OPT + 2$ and, hence, $|X| \leq OPT(OPT + 2)$. Note that $|V(G)| = OPT + |V(G - T^*)|$. It follows that

$$|V(G)| \leq 4OPT^2 + 9OPT - 5$$

The edges of G consist of edges with no endpoint in T^* (at

most $|V(G-T^*)-1|$) and edges with at least one endpoint in T^* (at most $OPT(OPT+2)$), giving us
$|E(G)| \leq 5OPT^2 + 10OPT - 6$ □

We can now apply Lemma 4.2 individually to each biconnected component, giving us the following.

Lemma 4.3. *Any reduced graph G with start s and finish t has at most $4OPT^2 + 9OPT - 5$ vertices and at most $5OPT^2 + 10OPT - 6$ edges, where OPT denotes the size of an optimal tracking set of G.*

Proof. Let G_i denote the i^{th} biconnected component of G, with entry-exit vertices s_i, t_i (see Reduction Rule 2). Further, let OPT_i denote the size of a minimum tracking set of (G_i, s_i, t_i). It follows from Reduction Rule 2 that $OPT = \sum_i OPT_i$. Moreover, Lemma 4.2 gives us $|V(G_i)| \leq p(OPT_i)$, where $p(x) = 4x^2 + 9x - 5$. Thus,

$$\begin{aligned}
|V(G)| &\leq \sum_i |V(G_i)| \\
&\leq \sum_i p(OPT_i) && \text{(Lemma 4.2)} \\
&\leq p\left(\sum_i OPT_i\right) && (p \text{ is degree-2 polynomial}) \\
&= p(OPT) && \text{(Observation 2)}
\end{aligned}$$

The number of edges in G can be upper bounded in a similar manner. □

Lemma 4.4. *Reduction Rule 8 is safe and can be applied in polynomial-time.*

Correctness of all reduction rules leads us to the following theorem.

Theorem 1. Tracking Paths *admits a kernel of size bounded by $4k^2 + 9k - 5$ vertices and $5k^2 + 10k - 6$ edges.*

We show that Theorem 1 implies an $O(\sqrt{n})$-approximation algorithm (output all the vertices in the kernel).

Lemma 4.5. *There exists an $O(\sqrt{n})$-approximation algorithm for* Tracking Paths.

Proof. Let G be an input graph, $|V(G)| = n$. The approximation algorithm runs the kernelization algorithm for Tracking Paths on G for non-negative integer values of k starting from 1, as long as the algorithm gives a trivial NO instance or until $k = \sqrt{n}$. Let k be the smallest integer value for which the kernelization algorithm returns an $\mathcal{O}(k^2)$ kernel. It holds that $k \leq OPT$ where OPT is the size of a minimum tracking set for G. If $k > \sqrt{n}$, then we return the whole graph as an approximate tracking set. Note that here the approximation ratio is $\mathcal{O}(\sqrt{n})$. Else, $k \leq \sqrt{n}$. This implies that $\mathcal{O}(k^2) \leq \sqrt{n}.k$ since $\mathcal{O}(k^2) \leq n$. Since $k \leq OPT$, it follows that $\mathcal{O}(k^2) \leq \sqrt{n}.OPT$. □

5. Linear kernel for planar graphs

In this section we show that an instance (G, k) of Tracking Paths, where G is a planar graph, can be reduced to an equivalent instance (G', k') such that if (G, k) is a YES instance then $|V(G')| = \mathcal{O}(k)$, $|E(G')| = \mathcal{O}(k)$ and $k' \leq k$. A linear kernel for planar graphs is derived by using an observation from [17].

We apply Reduction Rules 1, 2, 3, 4, and 5, followed by the another reduction rule that bounds the number of faces/regions in a reduced graph.

Lemma 5.1. *Eppstein et al. [17] The number of faces $|F|$ in a reduced planar graph is at most $2.OPT + 1$, where OPT is the number of trackers in an optimum tracking set in the graph.*

It follows from Lemma 5.1 that if the number of faces exceeds $2k+1$ then the graph does not have a tracking set of size k. Hence we have the following reduction rule.

Reduction Rule 9. *In a reduced planar graph G, if the number of faces $|F| > 2k + 1$, then return a trivial NO instance.*

Theorem 2. Tracking Paths *admits a kernel of size $\mathcal{O}(k)$ in planar graphs.*

Proof. Let $G = (V, E)$ be a reduced planar graph that is a YES instance and F be the set of faces/regions in G. Let $V_{\geq 3}$ be the set of vertices with degree at least 3 and V_2 be the set of vertices with degree equal to 2. Observe that after applying Reduction Rules 1 and 2 there are no vertices of degree one in the graph. Further, due to Reduction Rules 3 and 5 each vertex of degree two has vertices with degree three or more as its neighbors. First we construct a graph $G' = (V', E')$ by short-circuiting (the vertex is deleted and an edge is introduced between its neighbors) all vertices in V_2. Due to Reduction Rules 4, 5 the short-circuiting does not create parallel edges. Note that the number of faces $|F|$ in G is the same as that in G'. Further, $|V'| = |V_{\geq 3}|$, $|V_2| \leq |E'|$ and $|E| \leq 2|E'|$. We now bound the size of $V_{\geq 3}$ in G'. Since summation of degrees of vertices in a graph is twice the number of edges, and degree of all vertices in G' is at least three,

$$|E| \geq 3|V'|/2 \tag{1}$$

Due to Euler's theorem,

$$\begin{aligned}
|F| &= |E'| - |V'| + 2 \\
&\geq |V'|/2 + 2 \text{ (Due to Equation (1))}
\end{aligned}$$

Hence,

$$\begin{aligned}
|V_{\geq 3}| = |V'| &\leq 2(|F| - 2) \\
&\leq 2(2k + 1 - 2) \text{ (Due to Reduction Rule 9)} \\
&\leq 4k - 2
\end{aligned}$$

Thus $|V_2| = |E'| = |F| + |V'| - 2 \leq 2k + 1 + 4k - 2 - 2 = 6k - 3$. The total number of vertices in G is $|V| = |V_2| +$

$|V_{\geq 3}| \leq 6k - 3 + 4k - 2 = 10k - 5$. Since $|E| \leq 2|E'|$, $|E| \leq 12k - 6$. Hence, we have a kernel with $\mathcal{O}(k)$ vertices and $\mathcal{O}(k)$ edges. □

6. Hardness result

Here we show that finding a tracking set of size at most $n - k$ for a graph G with n vertices is W[1]-hard.

Theorem 3. *For general graphs, the problem of finding a tracking set of size at most $n - k$ in a graph of n vertices is* W*[1]-hard when parameterized by k.*

Proof. Tracking Paths has been shown NP-hard by a reduction from Vertex Cover in [4]. Specifically it has been shown that given a graph $G = (V, E)$ on n vertices one can construct in polynomial time a graph $G' = (V', E')$, where $|V'| = n' = |V| + |E| + 5$, such that G has a vertex cover of size k if and only if G' has a tracking set of size $k + |E| + 2$. It follows that G has an independent set of size k if and only if G' has a tracking set of size $n - k + |E| + 2$, i.e. $n' - k - 3$. Hence G has an independent set of size $k + 3$ if and only if G' has a tracking set of size $n' - k$. Since Independent Set is W[1]-hard [16], it follows that the problem of finding a tracking set of size at most $n - k$ is W[1]-hard as well. □

7. Conclusions

In this paper, we have given an $\mathcal{O}(k^2)$ sized kernel for Tracking Paths in general graphs improving from the previous $\mathcal{O}(k^7)$ bound. This is obtained using a lower bound for the number of trackers in what we call a tree-sink structure. This structure and the lower bound have already been used to obtain a linear kernel for H-minor free graphs and to obtain efficient approximation algorithms for the problem in H-minor free and general graphs [21].

It would be nice to show lower bound for the kernel size of Tracking Paths. Note that for the related problem FVS a conditional lower bound (no $\mathcal{O}(k^{2-\epsilon})$) is already known for the size of kernel in general graphs [18]. Another interesting open problem is to obtain improved FPT algorithms for the problem when parameterized by the solution size.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] J. Aslam, Z. Butler, F. Constantin, V. Crespi, G. Cybenko, D. Rus, Tracking a moving object with a binary sensor network, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, Association for Computing Machinery, New York, NY, USA, 2003, pp. 150–161.

[2] V. Bafna, P. Berman, T. Fujito, A 2-approximation algorithm for the undirected feedback vertex set problem, SIAM J. Discrete Math. 12 (1999) 289–297.

[3] A. Banik, P. Choudhary, Fixed-parameter tractable algorithms for tracking set problems, in: Algorithms and Discrete Applied Mathematics - 4th International Conference, Proceedings, CALDAM 2018, Guwahati, India, February 15-17, 2018, 2018, pp. 93–104.

[4] A. Banik, P. Choudhary, D. Lokshtanov, V. Raman, S. Saurabh, A polynomial sized kernel for tracking paths problem, Algorithmica 82 (2020) 41–63.

[5] A. Banik, P. Choudhary, V. Raman, S. Saurabh, Fixed-parameter tractable algorithms for tracking shortest paths, Theor. Comput. Sci. 846 (2020) 1–13, https://doi.org/10.1016/j.tcs.2020.09.006, https://www.sciencedirect.com/science/article/pii/S0304397520305053.

[6] A. Banik, M.J. Katz, E. Packer, M. Simakov, Tracking paths, Discrete Appl. Math. 282 (2020) 22–34.

[7] D. Bienstock, M.A. Langston, Chapter 8 algorithmic implications of the graph minor theorem, in: Network Models, in: Handbooks in Operations Research and Management Science, vol. 7, Elsevier, 1995, pp. 481–502.

[8] D. Bilò, L. Gualà, S. Leucci, G. Proietti, Tracking routes in communication networks, Theor. Comput. Sci. 844 (2020) 1–15, https://doi.org/10.1016/j.tcs.2020.07.012.

[9] V. Blažej, P. Choudhary, D. Knop, J.M. Křišťan, O. Suchý, T. Valla, Constant factor approximation for tracking paths and fault tolerant feedback vertex set, Discrete Optim. 47 (2023) 100756, https://doi.org/10.1016/j.disopt.2022.100756.

[10] V. Blažej, P. Choudhary, D. Knop, J.M. Křišťan, O. Suchý, T. Valla, Polynomial kernels for tracking shortest paths, Inf. Process. Lett. 179 (2023) 106315, https://doi.org/10.1016/j.ipl.2022.106315.

[11] Chih-Yu Lin, Wen-Chih Peng, Yu-Chee Tseng, Efficient in-network moving object tracking in wireless sensor networks, IEEE Trans. Mob. Comput. 5 (2006) 1044–1056.

[12] P. Choudhary, Polynomial time algorithms for tracking path problems, in: Combinatorial Algorithms - 31st International Workshop, Proceedings, IWOCA 2020, Bordeaux, France, June 8-10, 2020, 2020, pp. 166–179.

[13] P. Choudhary, Polynomial time algorithms for tracking path problems, CoRR, https://arxiv.org/abs/2002.07799, arXiv:2002.07799, 2020.

[14] A. Čivilis, C.S. Jensen, S. Pakalnis, Tracking of Moving Objects with Accuracy Guarantees, Springer Berlin Heidelberg, Berlin, Heidelberg, 2007, pp. 285–309.

[15] M. Cygan, F.V. Fomin, L. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, S. Saurabh, Parameterized Algorithms, 1st ed., Springer Publishing Company, Incorporated, 2015.

[16] R.G. Downey, M.R. Fellows, Parameterized Complexity, Springer-Verlag, 1999.

[17] D. Eppstein, M.T. Goodrich, J.A. Liu, P. Matias, Tracking paths in planar graphs, in: 30th International Symposium on Algorithms and Computation, ISAAC 2019, December 8-11, 2019, Shanghai University of Finance and Economics, Shanghai, China, 2019, pp. 54:1–54:17.

[18] F.V. Fomin, D. Lokshtanov, S. Saurabh, M. Zehavi, Kernelization: Theory of Parameterized Preprocessing, Cambridge University Press, 2019.

[19] F. Foucaud, M. Kovse, On graph identification problems and the special case of identifying vertices using paths, in: S. Arumugam, W.F. Smyth (Eds.), Combinatorial Algorithms, 23rd International Workshop, IWOCA 2012, Tamil Nadu, India, July 19-21, 2012, in: Revised Selected Papers, Springer, 2012, pp. 32–45.

[20] F. Foucaud, M. Kovse, Identifying path covers in graphs, J. Discret. Algorithms 23 (2013) 21–34.

[21] M.T. Goodrich, S. Gupta, H. Khodabandeh, P. Matias, How to catch marathon cheaters: new approximation algorithms for tracking paths, in: A. Lubiw, M. Salavatipour (Eds.), Algorithms and Data Structures, Springer International Publishing, Cham, 2021, pp. 442–456.

[22] N. Robertson, P.D. Seymour, Graph minors. v. excluding a planar graph, J. Comb. Theory, Ser. B 41 (1986) 92–114.

[23] N. Robertson, P.D. Seymour, Graph minors. xiii: The disjoint paths problem, J. Comb. Theory, Ser. B 63 (1995) 65–110.

[24] G. Schram, F. van der Linden, B. Kröse, F. Groen, Visual tracking of moving objects using a neural network controller, Robot. Auton. Syst. 18 (1996) 293–299, IAS-4 Conference.
[25] B. Song, T.Y. Jeng, E. Staudt, A.K. Roy-Chowdhury, A stochastic graph evolution framework for robust multi-target tracking, in: Computer Vision – ECCV 2010, Springer Berlin Heidelberg, 2010, pp. 605–619.
[26] Q. Zhang, L. Lapierre, X. Xiang, Distributed control of coordinated path tracking for networked nonholonomic mobile vehicles, IEEE Trans. Ind. Inform. 9 (2013) 472–484.