Structural Parameterizations of Tracking Paths Problem[⋆]

Pratibha Choudhary[1] and Venkatesh Raman[2]

[1] Indian Institute of Technology Jodhpur, Jodhpur, India.
pratibhac247@gmail.com
[2] Institute of Mathematical Sciences, HBNI, Chennai, India.
vraman@imsc.res.in

Abstract. Given a graph G with source and destination vertices $s, t \in V(G)$ respectively, Tracking Paths asks for a minimum set of vertices $T \subseteq V(G)$, such that the sequence of vertices encountered in each simple path from s to t is unique. The problem was proven NP-hard [3] and was found to admit a quadratic kernel when parameterized by the size of the desired solution [6]. Following recent trends, for the first time, we study Tracking Paths with respect to structural parameters of the input graph, parameters that measure how far the input graph is, from an easy instance. We prove that Tracking Paths admits fixed-parameter tractable (FPT) algorithms when parameterized by the size of vertex cover, and the size of cluster vertex deletion set for the input graph.

Keywords: Tracking Paths · structural parameterization · vertex cover · cluster vertex deletion set · undirected graphs.

1 Introduction

Graph theory plays a fundamental role in modeling many real world problems related to (but not limited to) road networks, traffic monitoring, world wide web, social networks and circuit design. One of the graph theoretic problems studied in recent years is Tracking Paths: Given a graph, find a set of vertices that can help uniquely distinguish all simple paths between a given source and destination in the input graph. The problem finds applications in secure facility object tracking, tracing data packets in network, identifying source of fake news on social media, and tracking objects in wireless sensor networks.

More formally, let $\overrightarrow{V}(P)$ be the sequence of vertices in a path P. A *tracking set* for a graph G with source s and destination t is a subset T of vertices such that for any two distinct s-t paths P_1 and P_2, $\overrightarrow{V}(P_1) \neq \overrightarrow{V}(P_2)$, and the Tracking Paths problem is defined as follows:

Tracking Paths (G, s, t)
Input: An undirected graph $G = (V, E)$ with terminal vertices s and t.
Question: Find a minimum cardinality tracking set T for G.

The problem was first studied by Banik et al. [1], where the problem was restricted to distinguishing all *shortest* s-t paths in a graph. The authors proved the problem NP-hard and APX-hard, and gave a 2-approximation algorithm for Tracking Shortest Paths in planar graphs.

Parameterization of a problem involves associating the problem with an integer k. A parameterized problem is said to admit a *fixed-parameter tractable*(FPT) algorithm if there exists an algorithm with running time of the type $f(k).n^{\mathcal{O}(1)}$, where f is a computable function, k is the parameter and n is the input size. Tracking Shortest Paths was proven to be FPT when parameterized by the size of tracking set [2]. Bilò et al. [4] gave an FPT algorithm for the case when there are multiple sources and destinations, the parameter being the maximum number of vertices equidistant from the source (or destination).

Tracking Paths (not just tracking shortest paths) was proven to be NP-hard for general graphs [3]. Note that from the definition, it is not even clear how to verify if a subset of vertices forms a tracking set, as there can be exponentially many s-t paths. Through an equivalent characterization, a polynomial time algorithm was shown [3] for this task, thus proving the problem NP.

Theorem 1. *[3]* Tracking Paths *belongs to* NP*, i.e. for a graph G and a set of vertices $T \subseteq V(G)$, there exists a polynomial time algorithm to verify if T is a tracking set for G.*

The problem was shown to be FPT when parameterized by the size of tracking set, by showing the existence of a polynomial kernel [3, 6]. A kernel for a parameterized problem is an equivalent instance of the given problem, whose size (of the reduced new instance) is bounded by a function of just the parameter. Kernelization (the process of deriving a kernel) is usually achieved through *Reduction Rules* which are preprocessing operations. A reduction rule is said to be safe if the new instance is equivalent to the original one, i.e. the original instance is a YES instance if and only if the new one is a YES instance. Eppstein et al. [9] studied Tracking Paths for planar graphs. They showed the problem NP-hard and gave a 4-approximation algorithm. They also gave a linear time algorithm for bounded clique-width graphs. Recently we gave polynomial time algorithms for restricted cases of Tracking Paths [5].

For a parameterized problem, although output size is a natural parameter, recent years have seen increasing attention on parameters related to structure of the input [8, 10, 12, 14]. So far, parameterized analysis of Tracking Paths has been done only with respect to the output size. In this paper, we study Tracking Paths parameterized by the size of vertex cover and the size of cluster vertex deletion set. For a graph $G = (V, E)$, a *vertex cover* is a set of vertices that covers all edges, i.e. the union of these vertices includes at least one endpoint of each edge in E. Removal of a vertex cover leaves the graph edgeless. For G, *cluster vertex deletion set* is the set of vertices whose removal converts G into a cluster graph: a graph whose each component is a clique.

Edgeless graphs do not need any trackers (since they lack s-t paths) and a tracking set can be found in polynomial time for cluster graphs (proven later in the paper). Hence, it is an interesting question to analyze whether there exists an FPT algorithm to solve Tracking Paths for graphs that are k vertices away from an edgeless graph or a cluster graph. Usually, the quest is to look for the smallest possible parameter for which the problem at hand is fixed-parameter tractable. In general, the size of a minimum vertex cover can be both larger or smaller than the size of a tracking set for a graph. A graph with long paths of degree two vertices can have a vertex cover larger than the size of a tracking set. While a denser graph can have a tracking set bigger than the size of a vertex cover. See Figure 1. Here the circled vertices represent a vertex cover. However, all vertices except s, t need to be part of a tracking set.

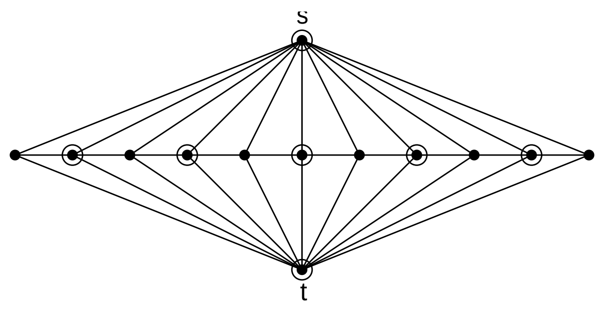

Fig. 1. Graph with tracking set larger than a vertex cover (circled vertices)

Our Approach. The usual challenge with structural parameterization is that the parameter does not drop by rules that utilize properties of the output. We start by applying some known preprocessing rules and then use some structural properties to mark vertices as trackers that definitely need to belong to any tracking set. Then we bound the number of vertices that are left unmarked as a function of the parameter. Finally we try all subsets of the unmarked vertices to find which among them can be trackers.

To design FPT algorithms for the two parameterizations we consider in this paper, we first define an intermediate parameter, and that is the size of what we call a *Dual Connected Modulator*. For a graph $G = (V, E)$, a set of vertices $S \subseteq V$ is a **dual connected modulator** (DCM) if every vertex in $V \setminus S$ has at least two neighbours in S and has an additional property Π. For parameterization by the size of vertex cover or cluster vertex deletion set, it suffices that Π is a disjoint union of cliques.

We will first give an FPT algorithm for Tracking Paths parameterized by the size of a dual connected modulator. Then we show how this algorithm can be used to give FPT algorithms for Tracking Paths parameterized by the size of vertex cover and cluster vertex deletion set.

2 Notations and Definitions

Throughout the paper, we assume graphs to be simple (no self loops or multi-edges). We assume that the input graph contains a unique source s and a unique destination t (s and t are known), and we aim to find a tracking set that can distinguish all simple paths between s and t. Here s and t are also referred as the terminal vertices. If $a, b \in V$, then unless otherwise stated, $\{a, b\}$ represents the set of vertices a, b, and (a, b) represents an edge between a and b. For a vertex $v \in V$, the *neighbourhood* of v is denoted by $N(v) = \{x \mid (x, v) \in E\}$. *Degree* of a vertex v is denoted by $deg(v) = |N(v)|$. For set of vertices V', $G(V')$ denotes the graph induced by vertices in V'. For a subgraph G' of G, $V(G')$ represents the vertex set of G' and $E(G')$ represents those edges whose both endpoints belong to $V(G')$. We use $G' \subseteq G$ to denote that G' is a subgraph of G. For a vertex $v \in V$ and a subgraph G', $N_{G'}(v) = N(v) \cap V(G')$ and $deg_{G'}(v) = |N(v) \cap V(G')|$. For a subset of vertices $V' \subseteq V$ we use $N(V')$ to denote $\bigcup_{v \in V'} N(v)$. For a graph G and a set of vertices $S \subseteq V(G)$, $G - S$ denotes the subgraph induced by the vertex set $V(G) \setminus V(S)$. For $A, B \subseteq V(G)$, $A \uplus B$ denotes that A and B are vertex disjoint partitions of graph G. Let P_1 be a path between vertices a and b, and P_2 be a path between vertices b and c, such that $V(P_1) \cap V(P_2) = \{b\}$. By $P_1.P_2$, we 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 end points, i.e. $V(P_1) \cap V(P_2) \subseteq \{a, b\}$, where a and b are the starting and end points of the paths. For details on parameterized complexity please refer to [7, 8, 11].

3 Parameterization by Dual Connected Modulator

In this section, we give an FPT algorithm for Tracking Paths parameterized by the size of a *Dual Connected Modulator*. Recall that for a graph G, a subset of vertices $S \subseteq V(G)$ is a DCM if every vertex in $V(G) \setminus S$ has at least two neighbours in S, and has an additional property Π.

Tracking Paths/DCM (G, s, t, S, k)
Input: An undirected graph $G = (V, E)$ with terminal vertices s and t, and a dual connected modulator $S \subseteq V(G)$ for G, such that $|S| = k$.
Question: Find a minimum cardinality tracking set T for G.

The main idea of the algorithm is to first guess how S intersects with a tracking set T in G, and then for each such guess, analyze the graph structures across the partition $S \uplus (G - S)$ and mark as many vertices as possible, as trackers. In the process, we give an upper bound for the number of vertices left unmarked in $G - S$. Finally, we consider all possible subsets of unmarked vertices in $G - S$ as trackers, and together with the set of already marked vertices, we verify if they form a tracking set for the graph G using Theorem 1. Initially none of the vertices in V are marked as trackers. We start by recalling some preprocessing rules from previous work.

Reduction Rule 1 *[3] If there exists a vertex or an edge that does not participate in any s-t path then delete it.*

Reduction Rule 2 *[6] (Rephrased) If $V \setminus \{s,t\} = \emptyset$, then return an $\emptyset$ as a solution. 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 $s(t)$.*

Reduction Rule 3 *[9] Let $u, v \in V(G)$ such that $deg(u) = deg(v) = 2$, $N(v) = \{u, w\}$, then delete v and introduce an edge between u and w.*

We apply above rules repeatedly as long as they are applicable. We refer to the graph obtained after applying the above rules as a *reduced graph.* Note that in the reduced graph, there is no vertex with degree less than or equal to 1, each vertex and edge participates in an *s-t* path and there are no long degree 2 paths (paths with consequent degree 2 vertices). Throughout the paper, after application of each reduction rule, we retain the notations of G, S and k to refer to the graph, modulator, and size of the modulator. For now, we assume that the application of above reduction rules does not destroy any properties of the modulator. Later, while analyzing specific graph parameters, we shall tweak the rules in order to maintain the modulator properties.

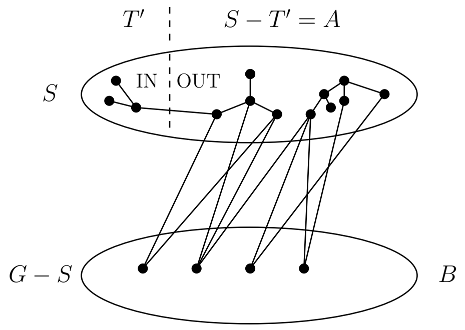

Fig. 2. Disjoint tracking set problem

Let T be a minimum tracking set that will be output by the algorithm, and $T' = T \cap S$ be the subset of T that belongs to S. We attempt to guess the vertices that belong to T', and towards this we simply consider all possible subsets of S. Hence, there are 2^k possible choices for T'. For each guess T', observe that the vertices in $S \setminus T'$ cannot belong to T. Thus we need to find a tracking set for G, that is disjoint from $S \setminus T'$. We refer to this problem as *disjoint tracking set.* See Figure 2. S represents a DCM for G that is received as a part of the input and $G - S$ represents the graph formed after removal of S from G. Recall that each vertex in $G - S$ has two neighbours in S. First we rule out those guesses for T' that can be easily discarded. It is known from [3] that every cycle in the graph must contain a tracker. This implies that the graph induced by $S \setminus T'$ cannot contain a cycle. Thus, we have the following reduction rule.

Reduction Rule 4 *If the graph induced by $S \setminus T'$ is not a forest, reject the current guess for T'.*

Now we define *local source* and *local destination*. For a subgraph $G' \subseteq G$, and vertices $u, v \in V(G')$, u is a local source and v is a local destination if

1. there exists a path in G from s to u, say P_{su},
2. there exists a path from v to t, say P_{vt},
3. $V(P_{su}) \cap V(P_{vt}) = \emptyset$, and
4. $V(P_{su}) \cap V(G') = \{u\}$ and $V(P_{vt}) \cap V(G') = \{v\}$.

For a subgraph G', we can check if a pair of vertices $a, b \in V(G')$ forms a local source-destination pair if there exists disjoint paths from s to a and b to t in the graph $G \setminus G' \cup \{a, b\}$, in quadratic time using the disjoint path algorithm from [13]. The concept of local source-destination pair has been used to obtain efficient algorithms for Tracking Paths (see [3,5,9]). If u, v form a local source-destination pair for a subgraph G', we refer to them as a local s-t pair. Next we recall the following lemmas (rephrased) from [3].

Lemma 1. *In a reduced graph G, any induced subgraph G' consisting of at least one edge contains a local source and a local destination.*

Lemma 2. *In a subgraph $G' \subseteq G$, if all paths between a local s-t pair cannot be tracked with at most x trackers, then G cannot be tracked with at most x trackers.*

Note that a subgraph can have more than one local s-t pairs. Now we can analyze subgraphs in G and identify trackers with respect to the local s-t pairs in that subgraph. Next we give two rules that help mark some vertices as trackers, and thus reduce the number of unmarked vertices in a graph.

Reduction Rule 5 [3] *If abc is a triangle in G such that a, b is a local s-t pair for the triangle abc and $c \notin S$, then mark c as a tracker. While considering the disjoint version, if $c \in S \setminus T'$, then reject the current guess for T'.*

Reduction Rule 6 *If there exists a subgraph $G' \subseteq G$, such that G' is a clique, with $a, b \in V(G')$ as a local s-t pair for G', then all vertices in $V(G') \setminus \{a, b\}$ need to be marked as trackers. Further, if a, b is the only local s-t pair for G', then delete all vertices in $V(G') \setminus \{a, b\}$.*

3.1 Finding a Disjoint Tracking Set

Let $A = S \setminus T'$ and $B = V \setminus S$. Note that our aim is to find a tracking set that is disjoint from A, i.e. $T \subseteq T' \cup B$. We first look at some structures induced by vertices in S and B that tries to force some vertices in B as trackers and mark them. We start with the following reduction rule.

[3] Proof of safeness and polynomial time application of Reduction Rules can be found in the full version of the paper.

Reduction Rule 7 *Let $a, b \in T'$. If a, b form a local s-t pair for a subgraph $G' \subseteq G[\{a, b\} \cup A]$ (the subgraph induced by A and a, b) and G' induces a cycle, then reject T', and move to the next guess.*

Since S is a DCM, each vertex in B is adjacent to at least 2 vertices in S. We categorize the vertices in B based on whether their neighbours lie in A or $S - A$ as follows:

- V_1: The set of vertices that have at least two neighbours in A.
- V_2: The set of vertices that have at least one neighbour in A and at least one in $S - A$.
- V_3: The set of vertices that have at least two neighbours in $S - A$.

Observe that $B = V_1 \cup V_2 \cup V_3$. There can be vertices that belong to more than one category, but as we shall see, this does not affect the outcome of the algorithm. Consider the case in which a pair of vertices $u, v \in S$ is adjacent to two vertices $w, x \in B$. Observe that the vertices u, v, w, x induce a C_4, say C. Due to Lemma 1, there exists a local s-t pair in the subgraph C. Now we analyze each of the above listed vertex sets in B, and we consider the possibility of each pair of vertices in a C_4 being an local s-t pair for that C_4. Then we mark all those vertices as trackers that necessarily need to belong to T, and we bound the number of unmarked vertices in $V \setminus S$.

Bounding $V_1 - T$ Consider a set of vertices in B that have two neighbours u, v in A, i.e. $S \setminus T'$. See Figure 3. Here, u, v cannot be trackers, as $A \cap T = \emptyset$.

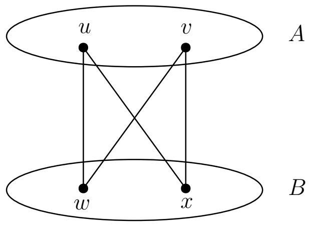

Fig. 3. A pair of vertices in A adjacent to two vertices in B forms a C_4

Lemma 3. ⊛[4] *The number of vertices in $V_1 - T$ can be bounded by $\binom{k}{2}$.*

Bounding $V_2 - T$ Here we consider the set of vertices in B that have one neighbour in $S \setminus A$ and one neighbour in A. Since $u \in S \setminus A, v \in A$, u is already marked as a tracker and v cannot be a tracker.

Lemma 4. ⊛ *The number of vertices in $V_2 - T$ can be bounded by $2\binom{k}{2}$.*

[4] Proofs of Lemmas marked with ⊛ can be found in the full version of the paper.

Bounding $V_3 - T$ Here we consider the set of vertices in B that have two neighbours in $S \setminus A$. Since $u, v \in S \setminus A$, both u and v are already marked as trackers. Since we have already analyzed the vertices in B that have at least one neighbour in A, here we restrict ourselves to only those vertices of B that are adjacent to only $S - A$. Let G' be the subgraph induced by $(S \setminus A) \cup V_3$.

If a pair of vertices $u, v \in S \setminus A$ are adjacent to a pair of vertices $w, x \in B$, they induce a C_4, say C. We create an empty set V_3', which will be used to identify those vertices of V_3 which might later be needed to be marked as trackers. We check for the possibility of each pair of vertices in $V(C)$ being a local s-t pair in the following sequence:

1. If w, x form a local s-t pair: Here u, v already serve as trackers to distinguish the two paths in C between w and x. Even if $(u, v) \in E(G)$, we do not need any more trackers.
2. If u, w form a local s-t pair: Here v already serves as a tracker to distinguish the paths $u.w$ and $u.x.v.w$. If $(u, v) \in E(G)$, then we mark x as a tracker to distinguish the paths $u.v.w$ and $u.x.v.w$.
3. If u, v form a local s-t pair: If $(u, v) \in E(G)$, then we need to mark both w, x as trackers, in order to distinguish the paths $u.v$, $u.w.v$ and $u.x.v$. If $(w, x) \in E(G)$, then also we need to mark both w, x as trackers, in order to distinguish the paths $u.w.v$, $u.x.w.v$ and $u.w.x.v$. If $(u, v), (w, x) \notin E(G)$, we arbitrarily mark one among w, x as a tracker, and add the other vertex to V_3'.

We add to V_3' those vertices from V_3 that are adjacent to a unique pair of vertices from T'. Thus $|V_3'| \leq \binom{k}{2}$. If $V \setminus S$ (hence V_3) is an independent set, for any path in G', at least every alternate vertex is a tracker. Hence, if a pair of paths between a local source and destination in G', has the same sequence of trackers, then the cycle induced by the distinct portions of these paths is of length at most four. Else due to Reduction Rule 6, each component in $G - S$ has at most two unmarked vertices. Hence, if $u, v \in T'$, then there exists a path of length at most two between u and v, in $G(V_3 \cup \{u, v\})$, and all vertices in the path are unmarked. Thus if two such paths exist between a pair of vertices in T' that forms a local s-t pair, then we mark a vertex as a tracker. In $\binom{k}{2}.n^{\mathcal{O}(1)}$ time we can find if two such paths exist between a pair of vertices in T' that forms a local s-t pair. We arbitrarily mark one vertex on one of the paths as a tracker[5], and the two vertices from the other subpath are added to V_3'. Hence $|V_3'| \leq 2\binom{k}{2}$.

Next we prove that we need not consider vertices from $V_3 \setminus V_3'$ as trackers.

Lemma 5. ⊛ *There exists a k sized tracking set for G if and only if there is one that is a subset of $V_1 \cup V_2 \cup V_3' \cup T'$.*

Now each pair of vertices in S is adjacent to at most one unmarked vertex or a vertex from V_3'. Note that $|V_1 \cup V_2 \cup V_3'| \leq 5\binom{k}{2}$. We consider all possible subsets of these $\mathcal{O}(k^2)$ vertices, include them with the already marked trackers to form a tracking set, and check the validity of that tracking set using Theorem 1.

[5] Proof of correctness can be found in the full version of the paper.

Thus the overall algorithm takes $2^{\mathcal{O}(k^2)}.n^{\mathcal{O}(1)}$ time. While solving the disjoint problem, for each new guess for T', if the size of the tracking set found is less than that of the tracking set found with respect to the previous guess, we discard the previous disjoint solution, and retain the new one, else we discard the current one. Hence we have the following theorem.

Theorem 2. *For a graph G with a known dual connected modulator of size k,* Tracking Paths *can be solved in $2^{\mathcal{O}(k^2)}.n^{\mathcal{O}(1)}$ time.*

4 Parameterization by Vertex Cover

In this section we give an FPT algorithm for Tracking Paths when the parameter k is the size of a known vertex cover for the given graph.

Tracking Paths/Vertex Cover (G, s, t, S, k)
Input: An undirected graph $G = (V, E)$ with terminal vertices s and t, and a vertex cover $S \subseteq V(G)$ for G, such that $|S| = k$.
Question: Find a minimum cardinality tracking set T for G.

We start by applying Reduction Rules 1 and 2. Observe that these rules delete vertices/edges from the input graph, and hence do not tamper with the vertex cover S. Since Reduction Rule 3 introduces a new edge in the graph, we tweak the rule in order to maintain that S is a vertex cover of size at most k.

Reduction Rule 8 *Let $u, v \in V(G)$ such that $deg(u) = deg(v) = 2$, $N(v) = \{u, w\}$, then delete v and introduce an edge between u and w. If $u, w \notin S$, set $S = S \cup \{u\}$.*

Note that if $u, w \notin S$, then v necessarily belongs to S. Since deletion of v reduces the size of S by one, we can safely add u (or w) to S without increasing the value of k.

Now observe that due to Reduction Rules 1 and 8, each vertex in $G - S$ has at least two neighbours in S. Hence S is a DCM for G. Thus we can apply the algorithm for DCM to derive an FPT algorithm for parameterization by the size of a vertex cover for a graph. Hence we have the following theorem.

Theorem 3. *For a graph G with a known vertex cover of size k,* Tracking Paths *can be solved in $2^{\mathcal{O}(k^2)}.n^{\mathcal{O}(1)}$ time.*

5 Parameterization by Cluster Vertex Deletion set

In this section we give an FPT algorithm for Tracking Paths when the parameter k is the size of a cluster vertex deletion set for the given graph.

Tracking Paths/Cluster Vertex Deletion Set (G, s, t, S, k)
Input: An undirected graph $G = (V, E)$ with terminal vertices s and t, and a cluster vertex deletion set $S \subseteq V(G)$ for G, such that $|S| = k$.
Question: Find a minimum cardinality tracking set T for G.

First we apply the Reduction Rules 1, 2 and 3 as explained below.

- Reduction Rule 1: Observe that $G - S$ is a cluster graph. Thus for each component in $G - S$, either all the vertices participate in an s-t path, or none of them do. Thus, when we apply Reduction Rule 1, it might lead to deletion of some vertices/edges from S, or some components from $G - S$. Note that this operation does not affect the properties of S or $G - S$.
- Reduction Rule 2: The application of this rule may delete vertices/edges from S and/or single vertex components from $G - S$. Observe that this does not affect the properties of S or $G - S$.
- Reduction Rule 3: Let $u, v \in V(G)$ be two vertices such that $deg(u) = deg(v) = 2$. If both $u, v \in S$, we can apply the rule and its does not affect properties of S or $G - S$. Consider the case in which one vertex among u, v belongs to S while the other belongs to $G - S$. In such a case, we necessarily delete the vertex that belongs to $G - S$. Suppose $u \in S$ and $v \in G - S$. Then v must belong to a component in $G - S$ that comprises only a single edge. After application of the reduction rule, the component of v shall comprise of only a single vertex. Observe that this does not affect the properties of S and $G - S$.

We also apply Reduction Rules 5 and 6 to mark required vertices in $G - S$ as trackers. Note that while applying all above reduction rules, it has been maintained that $G - S$ is a cluster graph and $|S| \leq k$.

Next, we try to mark as many vertices as possible as trackers in $G - S$, such that for the unmarked vertices S is a DCM. We create two sets $X = Y = \emptyset$. We use X to maintain the unmarked vertices in $G - S$, while ensuring that they have two neighbours each in S, and we use Y to maintain some other vertices that might need to be marked as trackers. Now we identify local s-t pairs in each component. Due to Lemma 1, each component (having at least one edge) in $G - S$ has at least one local s-t pair. After the application of Reduction Rule 6, for each local s-t pair in a component, all the remaining vertices shall be marked as trackers.

Corollary 1. *After application of Reduction Rule 6 at most two vertices in each component of $G - S$ are left unmarked.*

First, we consider the components in $G - S$ that contain s or t, or both. Let G' be a component in $G - S$ such that $s \in V(G')$ ($t \in V(G')$). Due to Lemma 1, G' contains a local s-t pair, say s, a (b, t). Due to Reduction Rule 6, all vertices in $V(G') \setminus \{s, a\}$ ($V(G') \setminus \{b, t\}$) shall be marked as trackers. If $deg_S(a) \geq 2$ ($deg_S(b) \geq 2$), we add a (b) to the set X, else we add a (b) to the set Y. Note that $|Y| \leq 2$. Henceforth, by 'components' we mean components in $G - S$, and we assume that none of the components in $G - S$ contain s or t. Now we analyze different types of components in $G - S$ based on their sizes.

5.1 Components with at least three vertices

Due to Lemma 1, each component has a local s-t pair. Since we already analyzed the components that contain s or t, a local s-t pair in a component necessarily needs to be adjacent to S in order to connect with the rest of the graph. Thus, each component that does not contain s, t, has at least two neighbours in S. We consider different cases based on the number of vertices in each component of $G - S$ that have neighbours in S.

Components with exactly two vertices with neighbours in S Let $G' \subseteq G - S$ be a component with exactly two vertices, say u, v, that have neighbours in S. Due to Lemma 1, G' has at least one local s-t pair and since $s, t \notin V(G')$, u, v shall form an local s-t pair for G' as these are the only vertices that connect G' with rest of the graph. Due to application of Reduction Rule 6, all vertices in $V(G') \setminus \{u, v\}$ shall have been deleted. Thus if a component has only two neighbours in S, it can consist of at most two vertices. The analysis for such components is explained in Section 5.2.

Components with three or more vertices with neighbours in S Let G' be a component in $G - S$. Due to Reduction Rule 6, if there are two disjoint local s-t pairs, then all vertices in that component shall be marked as trackers. We need not analyze such components further. Henceforth, we assume that if a component in $G - S$ has more than one local s-t pairs, then these pairs overlap. Consider a component G' in $G - S$, with two local s-t pairs. Since all local s-t pairs overlap, at most three vertices in G', say a, b, c, form these two local s-t pairs. Consider the case in which a, b is a local s-t pair and b, c is another local s-t pair. Observe that both c and a shall be marked as trackers due to Reduction Rule 6. Since here all local s-t pairs in a component overlap, at most one vertex in the component will be left unmarked. If such unmarked vertices have at least two neighbours in S, we add them to X. Else, we prove that they need not be part of an optimum tracking set.

Lemma 6. $\circledast$ *If there exists a component $G' \subseteq G - S$ with only one unmarked vertex v, and $deg_S(v) = 1$, then such a vertex need not be marked as a tracker.*

Now we are left with components that have two unmarked vertices. Clearly a pair of unmarked vertices in a component shall be a local s-t pair for that component. Let G' be a component with only one local s-t pair, say a, b, but more than two vertices with neighbours in S. Consider the following cases:

1. Both a and b have at least two neighbours in S: We add both a, b to X.
2. One among a, b, say b, has two neighbours in S, while a has only one neighbour in S: We add b to X. If b is marked as a tracker while application of the DCM algorithm, due to Lemma 6, a need not be considered as a tracker, and hence can be ignored. Else, if b is left unmarked in the DCM algorithm, we can account for a by doubling the bound obtained for unmarked vertices.
3. Both a and b have only one neighbour in S: Let $c \in N_S(a)$ and $d \in N_S(b)$. We introduce an additional vertex v_{ab}, and introduce edges (v_{ab}, c) and (v_{ab}, d) to $E(G)$. We also add v_{ab} to X. If v_{ab} is eventually part of a solution, then we can arbitrarily mark either a or b, say a, as a tracker. Not that in order

to distinguish paths by their vertex sequences, we can mark c along with a, thus ruling out the necessity of marking both a and b as trackers.

5.2 Single vertex and single edge components

If a component in $G - S$ consists of a single vertex v, then due to Reduction Rules 1 and 2, v has at least two neighbours in S. We include v in X, i.e. $X = X \cup \{v\}$.

If a component consists of a single edge $(a, b) \in E(G - S)$, due to Reduction Rules 1 and 2, both a, b have a neighbour in S. Due to Reduction Rule 3, it is not possible that $deg(a) = deg(b) = 2$. Thus, at least one vertex in each single edge component has at least two neighbours in S. If both a, b have two neighbours each in S, then we set $X = X \cup \{a, b\}$. Suppose (a, b) is an edge component and b has two neighbours in S, while a has only one neighbour in S. Observe that if while applying the algorithm for DCM, b is among the $\mathcal{O}(k^2)$ bounded vertices that are left unmarked, then we can simply double this bound in order to account for vertices like a (Note that this does not change the asymptotic bound of the running time). Else, if while applying the algorithm for DCM, if b is marked as a tracker, we show that an optimum tracking set for G need not contain a.

Lemma 7. ⊛ *If there exists a component in $G-S$ that comprises an edge (a, b), and b belongs to an optimum tracking set for G, then it can be determined in polynomial time whether a needs be marked as a tracker.*

Now all vertices in $G - S$ are either already marked as trackers, or need not be marked as trackers, or have been added to the set X. We consider all vertices in X, and apply the algorithm for DCM. In the final step of the DCM algorithm, where we consider all subsets of unmarked vertices and verify if they form a tracking set (along with the marked ones), we include the vertices in Y along with the unmarked vertices. This does not affect the bounds as $|Y| \leq 2$. Observe that we ignore all other vertices in $G - S$, except the ones in X, while analyzing different cases in the DCM algorithm. Hence, we have the following theorem.

Theorem 4. *For a graph G with a known cluster vertex deletion set of size k,* Tracking Paths *can be solved in $2^{\mathcal{O}(k^2)}.n^{\mathcal{O}(1)}$ time.*

6 Conclusions

In this paper, we study structural parameterizations of the Tracking Paths problem. We prove that Tracking Paths is FPT when parameterized by the size of vertex cover or the size of cluster vertex deletion set. We do so by giving a generalized algorithm for the case when the modulator has some specific properties. It would be interesting to explore if the running time of our algorithms can be improved further. Future scope involves studying Tracking Paths structural parameterization with respect to more parameters like odd cycle traversal, feedback vertex set and distance to chordal graph.

References

1. Banik, A., Katz, M.J., Packer, E., Simakov, M.: Tracking paths. In: Algorithms and Complexity - 10th International Conference, CIAC 2017. pp. 67–79 (2017)
2. Banik, A., Choudhary, P.: Fixed-parameter tractable algorithms for tracking set problems. In: Algorithms and Discrete Applied Mathematics - 4th International Conference, CALDAM 2018, Guwahati, India, February 15-17, 2018, Proceedings. pp. 93–104 (2018)
3. Banik, A., Choudhary, P., Lokshtanov, D., Raman, V., Saurabh, S.: A polynomial sized kernel for tracking paths problem. Algorithmica **82**(1), 41–63 (2020)
4. Bilò, D., Gualà, L., Leucci, S., Proietti, G.: Tracking routes in communication networks. In: Censor-Hillel, K., Flammini, M. (eds.) Structural Information and Communication Complexity. pp. 81–93. Springer International Publishing, Cham (2019)
5. Choudhary, P.: Polynomial time algorithms for tracking path problems. In: Combinatorial Algorithms - 31st International Workshop, IWOCA 2020, Bordeaux, France, June 8-10, 2020, Proceedings. pp. 166–179 (2020)
6. Choudhary, P., Raman, V.: Improved kernels for tracking path problems. CoRR **abs/2001.03161** (2020), `http://arxiv.org/abs/2001.03161`
7. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: Parameterized Algorithms. Springer Publishing Company, Incorporated, 1st edn. (2015)
8. Downey, R.G., Fellows, M.R.: Fundamentals of Parameterized Complexity. Springer Publishing Company, Incorporated (2013)
9. Eppstein, D., Goodrich, M.T., Liu, J.A., Matias, P.: 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. pp. 54:1–54:17 (2019)
10. Fellows, M.R., Jansen, B.M., Rosamond, F.: Towards fully multivariate algorithmics: Parameter ecology and the deconstruction of computational complexity. European Journal of Combinatorics **34**(3), 541 – 566 (2013), combinatorial Algorithms and Complexity
11. Flum, J., Grohe, M.: Parameterized Complexity Theory (Texts in Theoretical Computer Science. An EATCS Series). Springer-Verlag, Berlin, Heidelberg (2006)
12. Jansen, B.M., Kratsch, S.: Data reduction for graph coloring problems. Information and Computation **231**, 70 – 88 (2013), fundamentals of Computation Theory
13. Kawarabayashi, K., Kobayashi, Y., Reed, B.: The disjoint paths problem in quadratic time. Journal of Combinatorial Theory, Series B **102**(2), 424 – 435 (2012)
14. Majumdar, D., Raman, V.: Structural parameterizations of undirected feedback vertex set: FPT algorithms and kernelization. Algorithmica **80**(9), 2683–2724 (2018)