How to Catch Marathon Cheaters: New Approximation Algorithms for Tracking Paths

Michael T. Goodrich[1], Siddharth Gupta[2], Hadi Khodabandeh[1], and Pedro Matias[1](✉)

[1] Department of Computer Science, University of California Irvine, Irvine, USA
{goodrich,khodabah,pmatias}@uci.edu

[2] Department of Computer Science, Ben-Gurion University of the Negev, Beersheba, Israel
siddhart@post.bgu.ac.il

Abstract. Given an undirected graph, G, and vertices, s and t in G, the *tracking paths* problem is that of finding the smallest subset of vertices in G whose intersection with any s-t path results in a unique sequence. This problem is known to be NP-complete and has applications to animal migration tracking and detecting marathon course-cutting, but its approximability is largely unknown. In this paper, we address this latter issue, giving novel algorithms having approximation ratios of $(1+\epsilon)$, $O(\lg OPT)$ and $O(\lg n)$, for H-minor-free, general, and weighted graphs, respectively. We also give a linear kernel for H-minor-free graphs.

Keywords: Graph algorithms · Approximation algorithms · Graph minor · Fixed-parameter tractability · Kernelization · Minor-free graphs · Road networks

1 Introduction

In most modern marathons, each runner is provided with a small RFID tag, which is worn on the runner's shoe or embedded in the runner's bib. RFID readers are placed throughout the course and are used to track the progress of the runners [9,36]. In spite these measures, some runners try to cheat by taking shortcuts [37]. To detect all possible course-cutting, we are interested in the combinatorial optimization problem of placing the minimum number of RFID readers in the environment of a marathon to determine every possible path from the start to the finish, including paths that deviate from the official course, just from the sequence of RFID readers that are crossed by a runner taking a given path. In addition to detecting marathon course-cutting, solutions to this optimization problem could also allow for a type of marathon where each runner

The full version of this paper is available in [26]. Our research was supported in part by NSF Grant 1815073 and by the Zuckerman STEM Leadership Program.

A. Lubiw et al. (Eds.): WADS 2021, LNCS 12808, pp. 442–456, 2021.
https://doi.org/10.1007/978-3-030-83508-8_32

could be allowed to map out their own path from the start to finish so long as their path is at least the required length.

Formally, we model a city road network [18,20,21] through which a marathon will be run as an undirected graph, $G = (V, E)$, where V is the set of road intersections and possible RFID reader locations in the city, as well as the placements of the start and finish lines, and E is the set of road segments joining two points in V without having any other elements of V in its interior. Given a start-finish pair, (s, t), of vertices in G, a ***tracking set*** for (s, t) is a subset, T, of V, such that for any s-t path[1] P in G, the sequence $\mathcal{S}^T(P)$ of vertices in T traversed by P uniquely identifies P. In other words, T is a tracking set if $\mathcal{S}^T(P) \neq \mathcal{S}^T(Q)$ for all distinct s-t paths P and Q. We formally define the optimization problem, which is called the ***tracking paths*** problem, as follows:

TRACKING(G, s, t):
Input: An undirected simple graph $G = (V, E)$ and vertices $s, t \in V$.
Output: A smallest tracking set for (s, t) in G.

We denote by WEIGHTEDTRACKING the vertex-weighted version, whose goal is to find a tracking set of least total weight. Further, we denote by k-TRACKING the decision version of TRACKING, which asks whether there exists a tracking set of size at most k (for any given integer k). For conciseness, we refer to the "tracking set of G", when s and t are clear from context.

Related Work. TRACKING has been shown to be NP-Complete [3], even when the input graph is planar [19] or has bounded degree [10]. It is fixed-parameter tractable (FPT): when parameterized by the solution size (a.k.a., the natural parameter), it admits a quadratic kernel in general and a linear kernel when the graph is planar [11] (other parameterizations have been studied in [12]). Further, it admits approximation ratios of 4 [19] for planar graphs and of $2\Delta + 1$ [10] for degree-Δ graphs. Exact polynomial time algorithms exist for bounded clique-width graphs [19], as well as chordal and tournament graphs [10]. For the NP-hard variant of tracking only shortest paths between multiple start-finish pairs, there exists a $O(\sqrt{n \lg n})$-approximation [5]. We refer the reader to the full version of the paper [26] for more details on related work.

Our Contributions. Our results are summarized below:

1. **Linear kernel for H-minor-free graphs**. Previously, we only knew of a linear kernel for planar graphs [11]. This result also immediately implies an efficient $O(1)$-approximation.
2. **$(1 + \epsilon)$-approximation for H-minor-free graphs**. Previous best was a 4-approximation for planar graphs [19].
3. **$O(\lg OPT)$-approximation for** TRACKING, where OPT denotes the cardinality of an optimal tracking set. This is the first algorithm for general graphs with a non-trivial approximation ratio. Previously, we only knew of a $O(\sqrt{n \lg n})$-approximation for tracking shortest paths only [5].

[1] In this paper, paths do not repeat vertices. We denote a path from u to v by u-v.

4. $O(\lg n)$**-approximation for** WeightedTracking. This is the first approximation for weighted graphs, among all variants of Tracking.

Preliminaries. We use standard terminology concerning graphs, approximation algorithms and kernelization, which is detailed in the full version of the paper [26]. For space considerations, content marked with a link symbol "⊛" is provided in more detail and/or proved in the full version of the paper [26].

2 Structural Properties

Definition 1 (Entry-exit subgraph). *Let (G, s, t) be an instance of* Tracking*. An **entry-exit subgraph** is a triple (G', s', t'), where G' is a subgraph of G, and (s', t') is the **entry-exit pair** corresponding to vertices in C that satisfy the following conditions:*

1. *There exists a path s-s' from s to the **entry** vertex s'*
2. *There exists a path t'-t from the **exit** vertex t' to t*
3. *Paths s-s' and t'-t are vertex-disjoint*
4. *Path s-s' (resp. t'-t) and G' share exactly one vertex: s' (resp. t').*

Notice that the same subgraph G' of G may contain multiple entry-exit pairs.

Definition 2 (Entry-exit cycle). *An **entry-exit cycle** is an entry-exit subgraph (C, s', t'), where C is a cycle (see Fig. 1).*

We say that a vertex v ***tracks*** (C, s', t') if $v \in C \setminus \{s', t'\}$. Moreover, we say that (C, s', t') is ***tracked*** if there exists a tracker in a vertex that tracks it. A cycle C is ***tracked*** if all entry-exit cycles with entry-exit pairs in C are tracked. If C contains either (i) 3 trackers or (ii) s or t and 1 tracker in a non-entry/non-exit vertex, then it must be tracked. We say that these cycles are ***trivially tracked***.

We rely on the following alternative characterization of a tracking set, due to Banik et al. [3, Lemma 2], which establishes Tracking as a covering problem.

Lemma 1 ([3]). *For a graph $G = (V, E)$, a subset $T \subseteq V$ is a tracking set if and only if every simple cycle C in G is tracked with respect to T.*

Reduction Rules. Let us recall some reduction rules previously used to obtain polynomial kernels [3,11] and approximation algorithms [4,10,12,19].

Rule 1. [3] If there exists an edge or vertex that does not participate in any s-t path, remove it from the graph.

Rule 2. [11] If the degree of s (or t) is 1 and $N(s) \neq \{t\}$ ($N(t) \neq \{s\}$), then remove s (t), and label the vertex adjacent to it as s (t).

Rule 3. [19] If there exist adjacent vertices $a, b \notin \{s, t\}$ such that $\deg(a) = \deg(b) = 2$, then contract the edge ab.

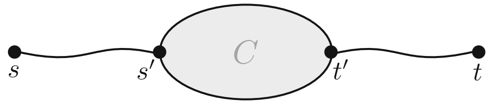

Fig. 1. Entry-exit pair illustration, with entry vertex s' and exit vertex t'.

Definition 3. *We say that an undirected graph G is* ***reduced by Rule X*** *if it cannot be further by reduced Rule X. Further, we say that G is* ***reduced*** *if it is reduced by Rules 1, 2 and 3.*

After exhaustive application of Rules 1 and 2, the graph is either a single edge, (s,t), or all its vertices have degree at least 2. Henceforth, we assume the latter, since the problem becomes trivial in the former case. Rule 3, which precludes the existence of adjacent vertices of degree 2, is used to bound the overall number of degree-2 vertices. Let us highlight a few additional useful consequences of Rule 1.

Remark 1 ([3]). Let G be a graph reduced by Rule 1. Then, every subgraph of G containing at least one edge has at least one entry-exit pair.

Remark 2 ([3]). Let G be a graph reduced by Rule 1. Then, any tracking set of G is also an FVS of G.

Remark 3. Let G be a graph reduced by Rule 1. Then the block-cut tree of G is an s-t path (see Fig. 2).

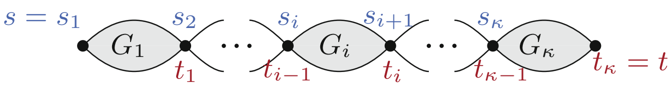

Fig. 2. The block-cut tree of a graph G reduced by Rule 1 (see Remark 4).

In other words, the latter remark says that the graph G that results from exhaustively applying Rule 1 consists of a sequence of $\kappa \geq 1$ biconnected components attached together by cut-vertices in a way that is analogous to series composition in series-parallel graphs. Thus, we can turn an instance (G,s,t) of TRACKING into one or more subproblems on biconnected graphs, (G_i, s_i, t_i), one for each biconnected component, as depicted in Fig. 2.

Remark 4. ⊛ Let G be a graph reduced by Rule 1. Then, an optimal tracking set for (G,s,t) is the ***disjoint union*** of optimal tracking sets for all (G_i, s_i, t_i).

Lower Bounds. We expand on a result by Choudhary and Raman [11], which provides a lower bound on the size of a tracking set, based on the presence of a tree-sink structure in the graph.

Definition 4 **([11]).** *A* ***tree-sink***[2] *in a graph G is a pair (Tr, x), where Tr is a subtree of G with at least two vertices and x, the* ***sink****, a vertex not in Tr that is adjacent to all the leaves*[3] *of Tr in G. We use $G(Tr, x)$ to denote the subgraph induced by (Tr, x). (Notice that this definition does not preclude the adjacency between non-leaf vertices and x.)*

Lemma 2 **([11]).** *Let (Tr, x) be a tree-sink in a reduced graph G, such that $|N_{Tr}(x)| = \delta$. Further let (s', t') be an entry-exit pair of $G(Tr, x)$. Then, if $x \in \{s', t'\}$, any tracking set of G contains at least $\delta - 1$ vertices in $V(Tr)$.*

The above lemma is a generalization of the lower bound given by the maximum number of vertex-disjoint paths between any two vertices [3], and it can be generalized further to obtain a more useful lower bound, established as the maximum degree among non-cut vertices (this follows from [11, Corollary 5]):

Lemma 3 **([11]).** *Let G' be a subgraph of a reduced graph G and x a vertex in G', such that $G' - x$ is connected and $N_{G'}(x) = \delta$. Then, any tracking set of G contains at least $\delta - 2$ vertices in $G' - x$.*

3 *H*-Minor-Free Graphs

A graph is H***-minor-free*** if it does not contain a fixed graph H as a minor. In this section, we present a linear kernel for H-minor-free graphs and use this kernel, as well as some ideas intrinsic to its construction, to design an efficient polynomial-time approximation scheme (EPTAS). An EPTAS is a $(1 \pm \epsilon)$-approximate algorithm whose running time is $O(n^c)$ for an input of size n and a constant c independent of ϵ.

Unlike the minimum FVS problem, which also consists of covering cycles, Tracking is not minor-closed [11] (i.e., an optimal solution for a minor of G may require more trackers than an optimal solution for G), so the powerful framework of bidimensionality [22] cannot be used to obtain either linear kernels [30] or PTASs for H-minor-free graphs [14]. Moreover, Tracking does not possess the "local" properties required by Baker's technique to develop EPTASs for planar graphs [2], or apex-minor-free graphs [17].

Linear Kernel. The following theorem about the sparsity of H-minor-free graphs will be helpful throughout the section.

Theorem 1 **(Mader [31]).** *Any simple H-minor-free graph with n vertices has at most $\sigma_H n$ edges, where σ_H depends solely on $|V(H)|$.*

We now give the following lemma concerning a relationship between the sizes of the vertex sets in certain bipartite minor-free graphs.

[2] This is illustrated in [11], or in the full version of the paper [26].
[3] We consider a leaf in an unrooted tree to be any vertex of degree 1.

Lemma 4. ⊛ *Let $B = (U \cup V, E)$ be a simple H-minor-free bipartite graph, such that: (i) every vertex in V has degree at least 2, and (ii) there exist at most δ neighbors in common between any pair u_1, u_2 in U, i.e., $|N(u_1) \cap N(u_2)| \leq \delta$ for all $u_1, u_2 \in U$. Then $|V| \leq \delta\sigma_H|U|$.*

Next, we give a lemma which will be useful throughout the paper.

Lemma 5. ⊛ *Let F be an FVS of a reduced graph G. Then $|V(G - F)| \leq 4|X| - 5$, where X is the cut set defined by $(F, G - F)$, consisting of edges with endpoints in both F and $G - F$.*

We will use Lemmas 4 and 5 above to give, in the next lemma, a linear kernel for a biconnected reduced H-minor-free graph.

Lemma 6. *Let G be a biconnected reduced H-minor-free graph with start s and finish t. Then, G has at most $(16\sigma_H^2 + 8\sigma_H + 1)OPT - 5$ vertices and at most $(20\sigma_H^2 + 11\sigma_H)OPT - 6$ edges, where OPT denotes the size of an optimal tracking set of G.*

Proof. Let T^* be an optimal tracking set of (G, s, t), i.e., $|T^*| = OPT$. Note that $G - T^*$ is a forest, since T^* is an FVS of G. We assume that $|T^*| \geq 2$, since otherwise one could check, in polynomial time, which vertex of G belongs to T^*. We now give some claims about the structure of G:

Claim 1: Let u_1, u_2 be two vertices in T^*. There exist at most 2 trees in $G - T^*$ that are adjacent[4] to both u_1 and u_2.

Claim 2: Every tree in $G - T^*$ is adjacent to at least 2 vertices in T^*.

Claim 3: Every tree in $G - T^*$ contains at most 2 vertices adjacent to the same vertex in T^*.

The first claim follows from Lemma 3. If there existed 3 or more trees adjacent to both u_1 and u_2, then the graph G', induced by u_1, u_2 and the trees, would require at least 1 tracker in $V(G') \setminus \{u_1\}$ and 1 tracker in $V(G') \setminus \{u_2\}$, contradicting the feasibility of T^*. The last claim also follows from Lemma 3 in a similar fashion. The second claim follows from the fact that G is biconnected.

To show the bound on the size of the vertex set and the edge set of G, we construct a new graph as follows. Let us contract each tree Tr in $G - T^*$ into a ***tree vertex*** v_{Tr}. Let F be the set of all tree vertices. Note that this operation may create parallel edges between a vertex in T^* and a tree vertex, but never between two vertices in T^* or F. Furthermore, we remove any edges between vertices in T^*. The resulting graph is bipartite, with vertex set partitioned into T^* and F, and is H-minor-free (since the class of minor-free graphs is minor-closed). By Claims 1 and 2, any 2 vertices in T^* have at most 2 common neighbors, and every vertex in F is adjacent to at least 2 vertices in T^*. Hence, by Lemma 4,

$$|F| \leq 2\sigma_H|T^*|.$$

[4] In this context, a tree is adjacent to v if it includes a vertex that is adjacent to v.

As a consequence of Claim 3, there are at most 2 parallel edges between a vertex in T^* and a vertex in F. Thus, by Theorem 1, the set of edges, X, in the bipartite graph is at most

$$2 \cdot \sigma_H(|F| + |T^*|) \leq (4\sigma_H^2 + 2\sigma_H)|T^*|.$$

Notice that X is the cut set defined by $(T^*, G - T^*)$, consisting of edges with endpoints in both T^* and $G - T^*$. Hence, by Lemma 5, $|V(G - T^*)| \leq 4|X| - 5$, giving us:

$$|V(G)| \leq (16\sigma_H^2 + 8\sigma_H + 1)|T^*| - 5.$$

The edges of G consist of (a) edges in $G - T^*$ (at most $|V(G - T^*)| - 1$), (b) the cut set X, and (c) edges with both endpoints in T^* (at most $\sigma_H|T^*|$ by Theorem 1). Thus,

$$\begin{aligned} |E(G)| &\leq (4|X| - 6) + |X| + (\sigma_H|T^*|) \\ &\leq (20\sigma_H^2 + 11\sigma_H)|T^*| - 6. \end{aligned}$$

□

By Remark 4 and the application of the above lemma to each biconnected component of a reduced graph, we obtain the following.

Theorem 2. k-Tracking *admits a kernel for H-minor-free graphs of size bounded by* $(16\sigma_H^2 + 8\sigma_H + 1)k - 5$ *vertices and* $(20\sigma_H^2 + 11\sigma_H)k - 6$ *edges.*

Corollary 1. Tracking *admits a $O(1)$-approximation for H-minor-free graphs.*

Even though we develop a $(1 + \epsilon)$-approximation in the next section, the latter corollary can be more useful in practice, when running time is a concern.

EPTAS. Given the unsuitability of bidimensionality and Baker's technique discussed earlier, we shall resort to the use of balanced separators. Our algorithm relies on ***balanced separators***, sets of vertices whose removal partitions the graph into two roughly equal-sized parts. Ungar [33] first showed that every n-vertex planar graph has a balanced separator of size $O(\sqrt{n}\lg^{3/2} n)$. This was later improved by Lipton and Tarjan [28] to $\sqrt{8n}$, and Goodrich [25] showed how to compute these recursively in linear time. The Lipton-Tarjan separator theorem has been further refined (e.g., see [13,15]) and generalized to bounded-genus graphs (e.g., see [16,24]) as well as to H-minor-free graphs (e.g., see [1,32]).

Theorem 3 (Minor-free Separator Theorem [1]). *Let G be an H-minor-free graph with n vertices, where H is a simple graph with $h \geq 1$ vertices. Then a balanced separator for G of size at most $c_H^1\sqrt{n}$ can be found in $O(h^{O(1)}n^{O(1)})$ time, where c_H^1 is a positive constant depending solely on h.*

We use the Minor-free Separator Theorem recursively to decompose the graph into a set $\mathcal{R}$ of edge-disjoint subgraphs, called ***regions***. The vertices of a region $R \in \mathcal{R}$ which belong to at least one other region are called ***boundary vertices*** and the set of these vertices is denoted by $\partial(R)$. The remaining vertices of R are called ***interior vertices*** and are denote by $int(R)$.

Definition 5 (Relaxed r-division). *A **relaxed** r**-division** of an n-vertex graph G is a decomposition of G into $\Theta(n/r)$ regions, each of which has at most r vertices, such that the total number boundary vertices is $O(n/\sqrt{r})$.*

Computing a relaxed r-division is the first step in Frederickson's algorithm [23] for constructing an r-division in a planar graph, a decomposition which additionally requires every region to have $O(\sqrt{r})$ boundary vertices (we won't need this property). Both decompositions can easily be generalized to any class of graphs that is characterized by the existence of sublinear balanced separators, which includes H-minor-free graphs.

Theorem 4 (Minor-free Separator Theorem (3) + Frederickson [23]). *There is an $O(n \lg n)$ algorithm that, given an H-minor-free graph G and a positive integer r, computes a relaxed r-division of G.*

Our strategy will be to (i) construct a relaxed r-division of a smaller graph, K, which is itself an $O(1)$-approximate tracking set, (ii) solve optimally for each region, and (iii) combine the solutions for each region into a solution for the original graph with quality comparable to that of an optimal solution. This approach has been used to obtain EPTASs for minimum FVS [6,39], maximum independent set [29] and minimum vertex cover [8]. However, and in contrast to these problems, the step of constructing a close to optimal solution from the solutions of each region is not obvious. Indeed, the difficulty of this step emerges from the very "nonlocal" structure of Tracking, which requires special attention to the location of (s,t) in the graph, in addition to the nonlocal structure of cycles. Our EPTAS is as follows:

1. Compute a linear kernel K of G by reducing it with Rules 1, 2, 3, such that an optimal tracking set of K is $\Omega(|V(K)|)$ (see Corollary 1).
2. Compute a relaxed r-division $\mathcal{R}$ of K with $r = (2c_H^1 c_H^2 (c_H^3 + 1)/\epsilon)^2$, for any choice of $\epsilon > 0$ and constants $c_H^1, c_H^2, c_H^3 > 0$ specified later.
3. For each region R in $\mathcal{R}$, compute an optimal tracking set $OPT(R)$ for the subset of entry-cycles (with respect to (s,t)) which are completely contained in R.
4. Output $T = \bigcup_{R \in \mathcal{R}} (OPT(R) \cup \partial(R) \cup \mathcal{N}(R))$.
   Here, $\mathcal{N}(R) := N_{\Pi(R)}(\partial(\Pi(R)))$ defines an appropriate neighborhood of the boundary vertices of R, where $\Pi(R)$ is the subgraph of R consisting of the union of each path in R that: (i) is not an edge, (ii) has $\partial(R)$ vertices as endpoints, and (iii) traverses no ***internal*** vertices that are in $OPT(R)$. We let $\partial(\Pi(R)) := \partial(R) \cap \Pi(R)$. See Fig. 3.

We will now give the details of the algorithm and its correctness. We refer to the Reduction Rules defined in Sect. 2. As a reminder, after exhaustive application of Rules 1 and 2, the graph is either a single edge between s and t, or all its vertices have degree at least 2. Henceforth, we will assume the latter, since a minimum tracking set is trivial in the former. Notice that none of the reduction rules introduce trackers, so there is no lifting required at the end of our algorithm, i.e., adding back any trackers introduced during the reduction.

Observation 1. *No entry-exit cycles are removed during Rules 1, 2 or 3, so a tracking set of the resulting kernel K is a tracking set of the input graph G. Therefore, any minimum tracking set of K is also a minimum tracking set of G.*

Next, we explain how to compute in polynomial time optimal tracking sets for each region in a relaxed r-division of a kernel K.

Lemma 7. ⊛ *Let $\mathcal{C}(R)$ be the set of all entry-exit cycles in G whose vertices are a subset of $V(R)$, where R is a subgraph of G. Then one can compute a minimum subset of $V(R)$ that tracks every entry-cycle of $\mathcal{C}(R)$ in $O(2^{|V(R)|} \cdot n^{O(1)})$ time.*

Let us now argue that our algorithm computes a $(1+\epsilon)$-approximate tracking set. Let $T = \bigcup_{R \in \mathcal{R}} (OPT(R) \cup \partial(R) \cup \mathcal{N}(R))$ be the output of the algorithm.

Lemma 8. ⊛ *T is a tracking set of the input graph G.*

Let us denote by OPT the size of an optimal tracking set of the input graph G. To argue that $|T| \leq (1+\epsilon)OPT$, we will need to argue that the set of trackers in the special neighborhoods defined by $\mathcal{N}(R)$, for all regions R, have small cardinalities, i.e., roughly equal to $O(\epsilon OPT)$. This is the key argument to our EPTAS, which the next lemma addresses. Its proof is not immediately obvious, since the number of neighbors of all boundary vertices could be $\Omega(OPT)$, a consequence of the quadratic gap between $|\partial(R)|$ and $|V(R)|$.

Lemma 9. ⊛ $|\mathcal{N}(R)| \leq c_H^3 |\partial(\Pi(R))|$, *where* $c_H^3 \geq 9\sigma_H^2 + 3\sigma_H$.

Proof. (Sketch) The set of untracked cycles between 2 regions R and R', which must exist in $\Pi(R) \cup \Pi(R')$, induces a forest on either region if we remove $\partial(R)$ and $\partial(R')$. Using arguments similar to those in the proof of Lemma 6, we can show that the bipartite graph with bipartition $(F, \partial(\Pi(R)))$ has the properties required by Lemma 4, but also that there exists $O(1)$ edges between a tree and a boundary vertex, where F is the set of trees in $\Pi(R) - \partial(\Pi(R))$. As a consequence, we can get an appropriate bound on the number of edges in this bipartite graph, from which the lemma follows. (See [26] for details.) □

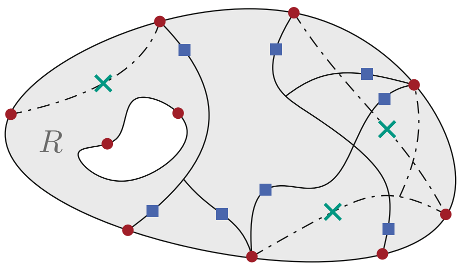

Fig. 3. Illustration of $\Pi(R)$ and of $\mathcal{N}(R)$ for a region R in a relaxed r-division $\mathcal{R}$. Vertices in $\partial(R)$ are depicted in red circles. $\Pi(R)$ consists of the union of all boundary-to-boundary paths in R (solid black), which are not edges and do not traverse $OPT(R)$ (green crosses). The dashed lines represent paths in $R - \Pi(R)$. $\mathcal{N}(R)$ is depicted in blue squares. (Color figure online)

Before proving that the output of our algorithm is a $(1+\epsilon)$-approximate tracking set, let us first recall a result from Frederickson [23, Lemma 1], which concerns the sum, for each boundary vertex b of the number of regions $\Delta(b)$ containing b in a relaxed r-division $\mathcal{R}$ of a planar graph. Even though this result was given in the context of planar graphs, it can easily be generalized to any graph whose subgraphs G' admit balanced separators of size $O(\sqrt{|V(G')|})$. We denote the set of all boundary vertices by $\partial(\mathcal{R})$. Further, let $B(\mathcal{R}) = \sum_{b\in\partial(\mathcal{R})}(\Delta(b)-1)$.

Lemma 10 ([23]). *Let $\mathcal{R}$ be a relaxed r-division of an n-vertex graph whose subgraphs G' admit balanced separators of size at most $c\sqrt{|V(G')|}$. Then $B(\mathcal{R}) \leq c\cdot n/\sqrt{r}$, for a constant c independent of r and n.*

We will use the latter lemma to bound the overall number of trackers in the next theorem.

Theorem 5. Tracking *admits an EPTAS for H-minor-free graphs.*

Proof. Consider the algorithm given at the beginning of the section. As a reminder, let $T = \bigcup_{R\in\mathcal{R}}(OPT(R)\cup\partial(R)\cup\mathcal{N}(R))$ be the output of the algorithm, for a relaxed r-division $\mathcal{R}$ of a kernel K of G, where $OPT(R)$ is the optimal tracking set computed with respect to entry-exit cycles in R. By Lemma 8, T is a tracking set. Next, we argue about the approximation ratio. By a union bound,

$$|T| \leq |\partial(\mathcal{R})| + \sum_{R\in\mathcal{R}}|OPT(R)| + \sum_{R\in\mathcal{R}}|\mathcal{N}(R)|.$$

Let $n' = |V(K)|$ be the number of vertices in K. Clearly, $|\partial(\mathcal{R})| \leq B(\mathcal{R})$. Moreover, we have that $\sum_{R\in\mathcal{R}}|\partial(R)| \leq 2B(\mathcal{R})$, so by Lemma 9, we have:

$$\sum_{R\in\mathcal{R}}|\mathcal{N}(R)| \leq 2c_H^3 B(\mathcal{R}).$$

Let T^* be an optimal tracking set of K, i.e., $|T^*| = OPT$ (by Observation 1). Since T^* is a tracking set, but not necessarily an optimal one, for all entry-exit cycles within any region $R\in\mathcal{R}$, we have that $|OPT(R)| \leq |T^*\cap V(R)|$. Thus,

$$\sum_{R\in\mathcal{R}}|OPT(R)| \leq OPT + B(\mathcal{R}).$$

Overall, for $r = (2c_H^1 c_H^2(c_H^3+1)/\epsilon)^2$,

$$\begin{aligned}
|T| &\leq OPT + 2(c_H^3+1)B(\mathcal{R}) \\
&\leq OPT + 2c_H^1(c_H^3+1)n'/\sqrt{r} && \text{(Lemma 10, Theorem 3)} \\
&\leq OPT + 2c_H^1 c_H^2(c_H^3+1)OPT/\sqrt{r} && (\text{Theorem 2}, c_H^2 \geq 16\sigma_H^2+8\sigma_H+1) \\
&= (1+\epsilon)OPT.
\end{aligned}$$

Step 1 of the algorithm takes $O(n^{O(1)})$ time, since it consists of applying Rules 1, 2, 3. Step 2 can be done in $O(n \lg n)$ time [23]. Step 3 takes $O(2^r \cdot n^{O(1)})$ time, by Lemma 7. Finally, step 4 takes $O(n^{O(1)})$ time. Overall, these amount to $O(2^{O(1/\epsilon^2)} n^{O(1)})$. □

4 General Graphs

In this section, we derive an $O(\lg n)$-approximation algorithm for WEIGHTEDTRACKING on general graphs, as well as an $O(\lg OPT)$-approximation algorithm for TRACKING.

We reduce an instance (G, s, t, w') of WEIGHTEDTRACKING, for a weight function $w' : V(G) \to \mathbb{Q}$, into an instance $(\mathcal{U}, \mathcal{X}, w)$ of SETCOVER, which asks for the sub-collection of $\mathcal{X}$ of minimum total weight, whose union equals the universe $\mathcal{U}$. Here, $(\mathcal{U}, \mathcal{X})$ defines a set system, i.e., a collection $\mathcal{X}$ of subsets of a set $\mathcal{U}$, and w is the weight function $w : \mathcal{X} \to \mathbb{Q}$. It is well known that there exist greedy polynomial-time algorithms achieving approximation ratios of $(1 + \ln M)$ or of $(1 + \Delta)$ [35,38], where M is the size of the largest set in $\mathcal{X}$ and Δ is the maximum number, over all elements u in $\mathcal{U}$, of sets in $\mathcal{X}$ that contain u.

Let $\mathcal{C}$ be the set of all entry-exit cycles in our input graph G, which we assume w.l.o.g. to be reduced by Rule 1. Further, let $\mathcal{C}_F$ be the set of all entry-exit cycles in G, each of which contains at most 2 vertices from the subset $F \subseteq V$. That is, $\mathcal{C}_F := \{(C, s', t') \in \mathcal{C} : |C \cap F| \leq 2\}$. Our algorithm is as follows.

1. Compute a 2-approximate FVS F of G (see [35,38]).
2. Use the greedy algorithm of [35,38] to compute an approximate set covering, $S \subseteq V(G)$, for an instance $(\mathcal{U}, \mathcal{X}, w)$ of SETCOVER where:
  (i) the universe, $\mathcal{U}$, of elements to be covered is $\mathcal{C}_F$
  (ii) the collection of covering sets, $\mathcal{X}$, is a 1-1 correspondence with $V(G)$, where each covering set with corresponding vertex v is the subset of $\mathcal{C}_F$ which are tracked by v, that is,

$$\mathcal{X} = \{\{(C, s', t') \in \mathcal{C}_F \mid v \text{ tracks } (C, s', t')\}\}_{v \in V(G)}.$$

  (iii) the weight function w is the weight function w' defined for WEIGHTEDTRACKING, given the 1-1 correspondence between $\mathcal{X}$ and $V(G)$.
3. Output $T = S \cup F$.

We can show that $|\mathcal{C}_F| = O(n^{O(1)})$. From the observation that every tracking set F is an FVS (see Remark 2), it follows that there are at most $O(n^{O(1)})$ entry-exit cycles not tracked by F. Thus, our claim follows (details in [26]).

Theorem 6. ⊛ WEIGHTEDTRACKING *admits an* $O(\lg n)$*-approximation.*

Unweighted Graphs. We show that the dual of the above set cover formulation has bounded VC-dimension [27,34]. This immediately improves the approximation ratio to $O(\lg OPT)$ for TRACKING (unweighted version) as a consequence of a result by Brönnimann and Goodrich [7], which establishes an approximation-ratio of $O(d \lg(dc))$ for unweighted set cover instances with dual VC-dimension d and optimal covers of size at most c.

Let $(\mathcal{U}, \mathcal{X})$ be a set system and Y a subset of $\mathcal{U}$. We say that Y is ***shattered*** if $\mathcal{X} \cap Y = 2^Y$, where $\mathcal{X} \cap Y := \{X \cap Y \mid X \in \mathcal{X}\}$. In other words, Y is shattered if the set of intersections of Y with each $X \in \mathcal{X}$ contains all the possible subsets of Y. The set system $(\mathcal{U}, \mathcal{X})$ has ***VC-dimension*** d if d is the largest integer for which there exists a subset $Y \subseteq \mathcal{U}$, of cardinality $|Y| = d$, that can be shattered.

The dual problem of an unweighted instance $(\mathcal{U}, \mathcal{X})$ of SetCover is finding a ***hitting set*** of minimum size, where a hitting set is a subset of $\mathcal{U}$ that has a non-empty intersection with every set in $\mathcal{X}$. In our case, it corresponds to finding the smallest subset of entry-exit cycles that covers every vertex, where a vertex is covered if it tracks least one entry-exit cycle in the subset. This is equivalent to an unweighted instance of SetCover with set system $(V, \mathcal{C}_F^*)$, where $V = V(G)$ and $\mathcal{C}_F^* := \{V(C) \setminus \{s', t'\} : (C, s', t') \in \mathcal{C}_F\}$ is the collection of sets, one for each entry-exit cycle, of vertices which can track that entry-exit cycle.

Lemma 11. *The set system $(V, \mathcal{C}_F^*)$ has VC-dimension at most 9.*

Proof. We show that there exists no subset $Y \subseteq V$ of size $|Y| \geq 10$ that can be shattered by $\mathcal{C}_F^*$. Since every element of $\mathcal{C}_F^*$ contains at most 2 vertices from F (by definition of $\mathcal{C}_F$), we cannot have more than 2 vertices from F in Y (since we would then require an entry-exit cycle containing at least 3 vertices in F to shatter Y). Thus, the lemma follows if we show that no subset $Y \subseteq V \setminus F$ of size $|Y| \geq 8$ can be shattered by $\mathcal{C}_F^*$. Let us assume, by contradiction, that this is possible. Then, if $Y \subseteq V \setminus F$ is to be shattered by $\mathcal{C}_F^*$, there must exist 2 entry-exit cycles (C_1, s_1', t_1') and (C_2, s_2', t_2') in $\mathcal{C}_F$, such that[5]:

- C_1 traverses all vertices of Y, say in the order $y_1, y_2, \ldots, y_{|Y|}$ (for all $y_j \in Y$),
- C_2 traverses every other vertex of Y traversed by C_1, say $Y' = \{y_2, y_4, \ldots, y_{|Y|}\}$, but not necessarily in the same order (we assume w.l.o.g. $|Y|$ is even).

Consider the graph consisting of the union of the cycles C_1, C_2. Let us contract every shared edge between C_1, C_2. Note that C_1 remains a cycle that traverses Y and C_2 remains a cycle that traverses Y' but not any vertex of $Y \setminus Y'$. So we can safely assume that C_1 and C_2 do not share any edges. Thus, the union of C_1, C_2 is a graph with $|C_1| + |C_2| - |Y|/2$ vertices and $|C_1| + |C_2|$ edges. Since both entry-exit cycles are in $\mathcal{C}_F$, each of C_1, C_2 shares at most 2 vertices with F. Let us remove such vertices, say there's $k \leq 4$ of them. The result is a graph with $|C_1| + |C_2| - |Y|/2 - k$ vertices and, at best, $|C_1| + |C_2| - 2k$ edges (the removed vertices cannot be in Y, so they have degree 2). In order for this graph to be acyclic (since F is an FVS by Remark 2, and our contractions preserve cycles) we would then require $|Y| < 8$ (since any acyclic graph with n vertices has at most $n - 1$ edges), a contradiction. $\square$

[5] This is illustrated in the full version of the paper [26].

The above lemma, combined with the result of Brönnimann and Goodrich [7] gives us the following.

Theorem 7. Tracking *admits an* $O(\lg OPT)$*-approximation, where* OPT *is the size of an optimal tracking set.*

References

1. Alon, N., Seymour, P.D., Thomas, R.: A separator theorem for graphs with an excluded minor and its applications. In: Ortiz, H. (ed.) STOC, pp. 293–299. ACM (1990). https://doi.org/10.1145/100216.100254
2. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. J. ACM **41**(1), 153–180 (1994). https://doi.org/10.1145/174644.174650
3. Banik, A., Choudhary, P., Lokshtanov, D., Raman, V., Saurabh, S.: A polynomial sized kernel for tracking paths problem. Algorithmica **82**(1), 41–63 (2019). https://doi.org/10.1007/s00453-019-00602-8
4. Banik, A., Katz, M.J., Packer, E., Simakov, M.: Tracking paths. In: Fotakis, D., Pagourtzis, A., Paschos, V.T. (eds.) CIAC 2017. LNCS, vol. 10236, pp. 67–79. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-57586-5_7
5. Bilò, D., Gualà, L., Leucci, S., Proietti, G.: Tracking routes in communication networks. Theor. Comput. Sci. **844**, 1–15 (2020). https://doi.org/10.1016/j.tcs.2020.07.012
6. Borradaile, G., Le, H., Zheng, B.: Engineering a PTAS for minimum feedback vertex set in planar graphs. In: Kotsireas, I., Pardalos, P., Parsopoulos, K.E., Souravlias, D., Tsokas, A. (eds.) SEA 2019. LNCS, vol. 11544, pp. 98–113. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-34029-2_7
7. Brönnimann, H., Goodrich, M.T.: Almost optimal set covers in finite VC-dimension. Discret. Comput. Geom. **14**(4), 463–479 (1995). https://doi.org/10.1007/BF02570718
8. Chiba, N., Nishizeki, T., Saito, N.: Applications of the Lipton and Tarjan's planar separator theorem. J. Inf. Process **4**(4), 203–207 (1981)
9. Chokchai, C.: Low cost and high performance UHF RFID system using Arduino based on IoT applications for marathon competition. WPMC **2018**, 15–20 (2018). https://doi.org/10.1109/WPMC.2018.8713018
10. Choudhary, P.: Polynomial time algorithms for tracking path problems. In: Gąsieniec, L., Klasing, R., Radzik, T. (eds.) IWOCA 2020. LNCS, vol. 12126, pp. 166–179. Springer, Cham (2020). https://doi.org/10.1007/978-3-030-48966-3_13
11. Choudhary, P., Raman, V.: Improved kernels for tracking path problems. CoRR abs/2001.03161 (2020). http://arxiv.org/abs/2001.03161
12. Choudhary, P., Raman, V.: Structural parameterizations of tracking paths problem. In: Cordasco, G., Gargano, L., Rescigno, A.A. (eds.) CEUR. CEUR Workshop Proceedings, vol. 2756, pp. 15–27. CEUR-WS.org (2020)
13. Chung, F.R.: Separator theorems and their applications. Universität Bonn, Institut für Ökonometrie und Operations Research (1988)
14. Demaine, E.D., Hajiaghayi, M.T.: Bidimensionality: new connections between FPT algorithms and PTASs. In: SODA, pp. 590–601. SIAM (2005)
15. Djidjev, H., Venkatesan, S.M.: Reduced constants for simple cycle graph separation. Acta Informatica **34**(3), 231–243 (1997). https://doi.org/10.1007/s002360050082

16. Djidjev, H.N.: A linear algorithm for partitioning graphs of fixed genus. Serdica. Bulgariacae mathematicae publicationes **11**(4), 369–387 (1985)
17. Eppstein, D.: Diameter and treewidth in minor-closed graph families. Algorithmica **27**(3), 275–291 (2000). https://doi.org/10.1007/s004530010020
18. Eppstein, D., Goodrich, M.T.: Studying (non-planar) road networks through an algorithmic lens. In: SIGSPATIAL. GIS, ACM (2008). https://doi.org/10.1145/1463434.1463455
19. Eppstein, D., Goodrich, M.T., Liu, J.A., Matias, P.: Tracking paths in planar graphs. In: Lu, P., Zhang, G. (eds.) ISAAC. LIPIcs, vol. 149, pp. 54:1–54:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2019). https://doi.org/10.4230/LIPIcs.ISAAC.2019.54
20. Eppstein, D., Gupta, S.: Crossing patterns in nonplanar road networks. In: SIGSPATIAL. GIS, ACM (2017). https://doi.org/10.1145/3139958.3139999
21. Eppstein, D., Khodabandeh, H.: On the edge crossings of the greedy spanner. CoRR abs/2002.05854 (2020). https://arxiv.org/abs/2002.05854
22. Fomin, F.V., Demaine, E.D., Hajiaghayi, M.T., Thilikos, D.M.: Bidimensionality. In: Kao, M.-Y. (ed.) Encyclopedia of Algorithms, pp. 203–207. Springer, Heidelberg (2016). https://doi.org/10.1007/978-1-4939-2864-4_47
23. Frederickson, G.N.: Fast algorithms for shortest paths in planar graphs, with applications. SIAM J. Comput. **16**(6), 1004–1022 (1987). https://doi.org/10.1137/0216064
24. Gilbert, J.R., Hutchinson, J.P., Tarjan, R.E.: A separator theorem for graphs of bounded genus. J. Algorithms **5**(3), 391–407 (1984). https://doi.org/10.1016/0196-6774(84)90019-1
25. Goodrich, M.T.: Planar separators and parallel polygon triangulation. J. Comput. Syst. Sci. **51**(3), 374–389 (1995)
26. Goodrich, M.T., Gupta, S., Khodabandeh, H., Matias, P.: How to catch marathon cheaters: new approximation algorithms for tracking paths. CoRR abs/2104.12337 (2021). https://arxiv.org/abs/2104.12337
27. Haussler, D., Welzl, E.: ϵ-nets and simplex range queries. Discret. Comput. Geom. **2**(2), 127–151 (1987). https://doi.org/10.1007/BF02187876
28. Lipton, R.J., Tarjan, R.E.: A separator theorem for planar graphs. SIAM J. Appl. Math. **36**(2), 177–189 (1979)
29. Lipton, R.J., Tarjan, R.E.: Applications of a planar separator theorem. SIAM J. Comput. **9**(3), 615–627 (1980). https://doi.org/10.1137/0209046
30. Lokshtanov, D.: Kernelization, bidimensionality and kernels. In: Kao, M.-Y. (ed.) Encyclopedia of Algorithms, pp. 1006–1011. Springer, Heidelberg (2016). https://doi.org/10.1007/978-1-4939-2864-4_526
31. Mader, W.: Homomorphiesätze für graphen. Math. Ann. **178**(2), 154–168 (1968)
32. Reed, B.A., Wood, D.R.: A linear-time algorithm to find a separator in a graph excluding a minor. ACM Trans. Algorithms **5**(4), 39:1–39:16 (2009). https://doi.org/10.1145/1597036.1597043
33. Ungar, P.: A theorem on planar graphs. J. Lond. Math. Soc. **1**(4), 256–262 (1951)
34. Vapnik, V.N., Chervonenkis, A.Y.: On the uniform convergence of relative frequencies of events to their probabilities. In: Vovk, V., Papadopoulos, H., Gammerman, A. (eds.) Measures of Complexity, pp. 11–30. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-21852-6_3
35. Vazirani, V.V.: Approximation Algorithms. Springer, Heidelberg (2001)
36. Want, R.: An introduction to RFID technology. IEEE Pervasive Comput. **5**(1), 25–33 (2006). https://doi.org/10.1109/MPRV.2006.2

37. Wikipedia contributors: Marathon course-cutting (2019). https://en.wikipedia.org/wiki/Marathon_course-cutting. Accessed 16 Feb 2021
38. Williamson, D.P., Shmoys, D.B.: The Design of Approximation Algorithms. Cambridge University Press, Cambridge (2011)
39. Zheng, B.: Approximation schemes in planar graphs. Ph.D. thesis, Oregon State University (2018). https://ir.library.oregonstate.edu/concern/graduate_thesis_or_dissertations/7w62ff609. Accessed 07 Jan 2021