PROC. 2ND IWPEC-06, VOL. 4169 IN LNCS, PP. 203-214, SPRINGER, 2006

Fixed-Parameter Tractability Results for Full-Degree Spanning Tree and Its Dual

Jiong Guo*, Rolf Niedermeier, and Sebastian Wernicke**

Institut für Informatik, Friedrich-Schiller-Universität Jena
Ernst-Abbe-Platz 2, D-07743 Jena, Germany
{guo,niedermr,wernicke}@minet.uni-jena.de

Abstract. We provide first-time fixed-parameter tractability results for the NP-complete problems MAXIMUM FULL-DEGREE SPANNING TREE and MINIMUM-VERTEX FEEDBACK EDGE SET. These problems are dual to each other: In MAXIMUM FULL-DEGREE SPANNING TREE, the task is to find a spanning tree for a given graph that maximizes the number of vertices that preserve their degree. For MINIMUM-VERTEX FEEDBACK EDGE SET the task is to minimize the number of vertices that end up with a reduced degree. Parameterized by the solution size, we exhibit that MINIMUM-VERTEX FEEDBACK EDGE SET is fixed-parameter tractable and has a problem kernel with the number of vertices linearly depending on the parameter k. Our main contribution for MAXIMUM FULL-DEGREE SPANNING TREE, which is W[1]-hard, is a linear-size problem kernel when restricted to planar graphs. Moreover, we present subexponential-time algorithms in the case of planar graphs.

1 Introduction

The NP-complete MAXIMUM FULL-DEGREE SPANNING TREE (FDST) problem is defined as follows.

> **Input**: An undirected graph $G = (V, E)$ and an integer $k \geq 0$.
> **Task**: Find a spanning tree T of G (called *solution tree*) in which at least k vertices have the same degree as in G.

Referring to vertices that maintain their degree as *full-degree* vertices and to the remaining ones as *reduced-degree* vertices, the task basically is to maximize the number of full-degree vertices. FDST is motivated by applications in water distribution and electrical networks [3, 4, 13]. It is a notoriously hard problem and, as such, not polynomial-time approximable within a factor of $O(n^{1/2-\epsilon})$ for any $\epsilon > 0$ unless NP-complete problems have randomized polynomial-time algorithms [3]. The approximability bound is almost tight in that Bhatia et al. [3] provide an algorithm with an approximation ratio of $\Theta(n^{1/2})$. FDST remains

* Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.
** Supported by the Deutsche Telekom Stiftung.

NP-complete in planar graphs; however, polynomial-time approximation schemes (PTAS) are known here [3, 4]. Broersma et al. [4] present further tractability and intractability results for various special graph classes. By way of contrast, the parameterized complexity [7, 10, 14] of FDST has so far been unexplored.

The complement (dual) problem of FDST, called MINIMUM-VERTEX FEEDBACK EDGE SET (VFES), is to find a feedback edge set[1] in a given graph such that this set is incident to as few vertices as possible. In other words, we want to minimize the number of reduced-degree vertices. VFES is motivated by an application of placing pressure-meters in fluid networks [12, 15, 16]. Khuller et al. [12] show that VFES is APX-hard and present a polynomial-time approximation algorithm with ratio $(2 + \epsilon)$ for any fixed $\epsilon > 0$. Moreover, they develop a PTAS for VFES in planar graphs. As with FDST, the parameterized complexity of VFES has so far not been investigated.

Parameterized by the respective solution size k, this work provides first-time parameterized complexity results for FDST and its dual VFES. Somewhat analogously to the study of approximability properties, we observe that FDST seems to be the harder problem when compared to its dual VFES: Whereas VFES is fixed-parameter tractable, FDST is W[1]-hard.[2] More specifically, our findings are as follows:

– VFES has a problem kernel with less than $4k$ vertices and it can be solved in $O(4^k \cdot k^2 + m)$ time for an m-edges graph.
– FDST becomes fixed-parameter tractable in the case of planar graphs. In particular, as the main technical contribution of this paper, we prove a linear-size problem kernel for FDST when restricted to planar graphs.
– For planar graphs, both VFES and FDST are solvable in subexponential time. More specifically, in n-vertices planar graphs VFES is solvable in $O(2^{O(\sqrt{k} \log k)} + k^5 + n)$ time and FDST in $O(2^{O(\sqrt{k} \log k)} + k^5 + n^3)$ time. Herein, we make use of tree decomposition-based dynamic programming.

We remark that, when restricted to planar graphs, this work amends the so far few examples where both a problem and its dual possess linear-size problem kernels. Other examples we are aware of (again restricted to planar graphs) are VERTEX COVER and its dual INDEPENDENT SET and DOMINATING SET and its dual NONBLOCKER [1, 5, 6].

We omit most proofs in this extended abstract due to a lack of space.

2 Minimum-Vertex Feedback Edge Set

MINIMUM-VERTEX FEEDBACK EDGE SET (VFES), the dual of MAXIMUM FULL-DEGREE SPANNING TREE (FDST), appears to be better tractable from a parameterized point of view than FDST. We subsequently present a simple problem kernelization with a kernel graph whose number of vertices linearly depends

[1] A feedback edge set is a set of edges whose deletion destroys all cycles in a graph.
[2] The reduction is from INDEPENDENT SET and the same as the one in [3, 12].

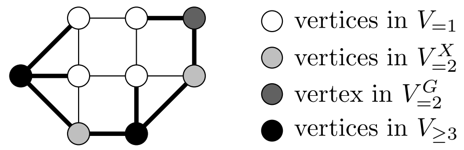


Fig. 1. Given a spanning tree for a graph (bold edges), the proof of the linear kernel for VFES in Theorem 1 partitions them into four disjoint subsets as illustrated above (see proof for details).

on the parameter (Sect. 2.1) and, based on this, an efficient fixed-parameter algorithm (Sect. 2.2) for VFES.

2.1 Data Reduction and Problem Kernel

To reduce a given instance of VFES to a problem kernel, we make use of two very simple data reduction rules already used by Khuller et al. [12].

Rule 1. Remove all degree-one vertices.
Rule 2. For any two neighboring degree-two vertices that do not have a common neighbor, contract the edge between them.

The correctness and the linear running time of these two rules are easy to verify; we call an instance of VFES *reduced* if the reduction rules cannot be applied any more.

Theorem 1. *A reduced instance* $(G = (V, E), k)$ *of* VFES *only has a solution tree if* $|V| < 4k$*.*

Proof. Assume that G has a solution tree T. Let X denote the set of the reduced-degree vertices in T. We partition the vertices in V into three disjoint subsets according to their degree in T, namely $V_{=1}$ contains all degree-1 vertices, $V_{=2}$ contains all degree-2 vertices, and $V_{\geq 3}$ contains all vertices of degree higher than 2. Furthermore, let $V_{=2}^X := V_{=2} \cap X$ and $V_{=2}^G := V_{=2} \setminus V_{=2}^X$. The partition is illustrated in Figure 1. Since G does not contain any degree-1 vertices by Rule 1, every degree-1 vertex in T is a reduced-degree vertex. Hence, T can have at most k leaves and $|V_{=1}| \leq |X| \leq k$. Since T is a tree, this directly implies $|V_{\geq 3}| \leq k - 2$.

As for $V_{=2}$, the vertices in $V_{=1} \cup V_{\geq 3}$ are either directly connected to each other or via a path P consisting of vertices from $V_{=2}$. Because Rule 2 contracts edges between two degree-2 vertices that have no common neighbor in the input graph, at least one of every two neighboring vertices of P has to be a reduced-degree vertex. Clearly, $V_{=2}^X \cup V_1 \subseteq X$. Since T is a tree, it follows that $|V_{=2}^G| \leq |V_{=1} \cup V_{\geq 3} \cup V_{=2}^X| - 1 \leq 2k - 3$.

Overall, this shows that $|V| = |V_{=1} \cup V_{=2}^X \cup V_{=2}^G \cup V_{\geq 3}| < 4k$ as claimed. □

2.2 A Fixed-Parameter Algorithm

The problem kernel obtained in Theorem 1 suggests a simple fixed-parameter algorithm for Minimum-Vertex Feedback Edge Set: For $i = 1, \ldots, k$, we consider all $\binom{4k}{i}$ size-i subsets X of kernel vertices. For each of these subsets, all edges between the vertices in X are removed from the input graph, that is, they become reduced-degree vertices. If the remaining graph is a forest, we have found a solution.[3] The correctness of this algorithm is obvious by its exhaustive nature, but on the running-time side it requires the consideration of $\sum_{i=1}^{k} \binom{4k}{i} > 9.45^k$ vertex subsets. The next theorem shows that we can do better because an exhaustive approach does not need to consider all vertices of the kernel but only those with degree at least three.

Theorem 2. *For an m-edge instance $(G = (V, E), k)$ of* VFES*, it can be decided in $O(4^k \cdot k^2 + m)$ time whether it has a solution tree.*

Proof. Given an instance $(G = (V, E), k)$ of VFES, we first perform the kernelization which needs $O(m)$ time. By Theorem 1, we know that the remaining graph only has a solution tree if it contains fewer than $4k$ vertices. We now partition the vertices in V according to their degree: vertices in $V_{=2}$ have degree two and $V_{\geq 3}$ contains all vertices with degree at least three. For every size-i subset $X_{\geq 3} \subseteq V_{\geq 3}$, $1 \leq i \leq k$, the following two steps are performed:

> **Step 1.** Remove all edges between vertices in $X_{\geq 3}$. Call the resulting graph $G' = (V, E')$.
> **Step 2.** For each edge $e \in E'$, assign it a weight $w(e) = m + 1$ if it is incident to a vertex in $V_{\geq 3} \setminus X_{\geq 3}$ and a weight of 1, otherwise. Then, find a maximum-weight spanning tree for every connected component of G'. If the total weight of edges that are *not* in a spanning tree is at most $k-i$, then the original VFES instance $(G = (V, E), k)$ has a solution tree and the algorithm terminates; otherwise, the next subset $X_{\geq 3}$ is tried.

To justify Step 2, observe that the edges incident to vertices in $V_{\geq 3} \setminus X_{\geq 3}$ have to be in the solution tree because these vertices preserve their degree. Therefore, if a component in G' contains cycles we can only destroy these by removing edges between a vertex in $X_{\geq 3}$ and a vertex in $V_{=2}$. Moreover, removing such an edge results in exactly one additional reduced-degree vertex. Thus, searching for a maximum-weight spanning tree in the second step leads to a solution tree (the components can easily be reconnected by edges between vertices in $X_{\geq 3}$).

The running time of the algorithm is composed of the linear preprocessing time, the number of subsets that need to be considered, and the time needed for Steps 1 and 2. By Theorem 1, a reduced graph contains less than $4k$ vertices and, hence, $O(k^2)$ edges, upper-bounding the time needed for Steps 1 and 2 by $O(k^2)$. From the proof of Theorem 1, we know that $|V_{\geq 3}| \leq 2k$, and hence the overall running time is bounded above by $O(m + \sum_{1 \leq i \leq k} \binom{2k}{i} k^2) = O(4^k \cdot k^2 + m)$. □

[3] Since the input graph must be connected, we can add some edges from G that are between the vertices in X in order to reconnect connected components with each other and thus obtain a spanning tree.

3 Maximum Full-Degree Spanning Tree in Planar Graphs

The reduction from INDEPENDENT SET to MAXIMUM FULL-DEGREE SPANNING TREE (FDST) that is given in [3] already shows that FDST is W[1]-hard with respect to the number k of full-degree vertices.[4] For *planar* graphs, however, this does not hold; in this section, we show that it is even possible to achieve a linear-size problem kernel for FDST in planar graphs.

While the proof of the linear size upper bound is very involved and technical, the actual computation of the problem kernel is based on several straightforward data reduction rules on a given instance of FDST.[5]

Reduction Rules. Let $N(v)$ denote the neighborhood of a vertex v. For two vertices $v \neq w$ with $|N(v) \cap N(w)| \geq 3$, perform the following reductions:

1. Let u_1, u_2, u_3 be three vertices in $N(v) \cap N(w)$.
  1.1 If $N(u_1) = \{v, w\}$ and additionally either $N(u_2) = \{v, w\}$ or $N(u_2) = \{u_3, v, w\}$, then remove u_2.
  1.2 If $N(u_1) = \{u_2, v, w\}$, $N(u_2) = \{u_1, u_3, v, w\}$, and $N(u_3) = \{u_2, v, w\}$, then remove u_3.

2. Let u_1, u_2, u_3, u_4 be four vertices in $N(v) \cap N(w)$.
  2.1 If $N(u_1) = \{u_2, v, w\}$, $N(u_2) = \{u_1, v, w\}$, $N(u_3) = \{u_4, v, w\}$, and $N(u_4) = \{u_3, v, w\}$, then remove u_3 and u_4.
  2.2 If $N(u_2) = \{u_1, u_3, v, w\}$, $N(u_3) = \{u_2, u_4, v, w\}$, and there is no edge $\{u_1, u_4\}$, then remove the edge $\{u_2, u_3\}$.

3. Let u_1, u_2, u_3, u_4, u_5 be five vertices in $N(v) \cap N(w)$.
  If $N(u_2) = \{u_1, u_3, v, w\}$, $N(u_3) = \{u_2, v, w\}$, $N(u_4) = \{u_5, v, w\}$, and $N(u_5) = \{u_4, v, w\}$, then remove u_3.

The five subcases are illustrated in Figure 2. Omitting a formal proof here, exhaustively applying these rules yields a graph that has a spanning tree with k full-degree vertices iff the original graph has a spanning tree with k full-degree vertices, that is, the reduction rules are "correct."

As we shall show in the remainder of this section, a planar graph that is reduced with respect to the reduction rules is a linear-size problem kernel for MAXIMUM FULL-DEGREE SPANNING TREE in planar graphs:

Theorem 3. *For a given planar n-vertex graph G, let k be the maximum number of full-degree vertices in any spanning tree for G. If G is reduced with respect to the given reduction rules, then $n = O(k)$, that is, we have a linear-size problem kernel for* FDST *on G that can be computed in $O(n^3)$ time.*

The proof of this theorem is quite involved. We basically achieve it by contradiction, that is, we assume that we are given an optimal solution tree to FDST

[4] W[1]-hardness stands for (presumable) parameterized intractability. Refer to [7, 10, 14] for details.

[5] Note that the reduction rule applies to an arbitrary graph but the "linear-size kernel" performance-guarantee is only shown for planar graphs.

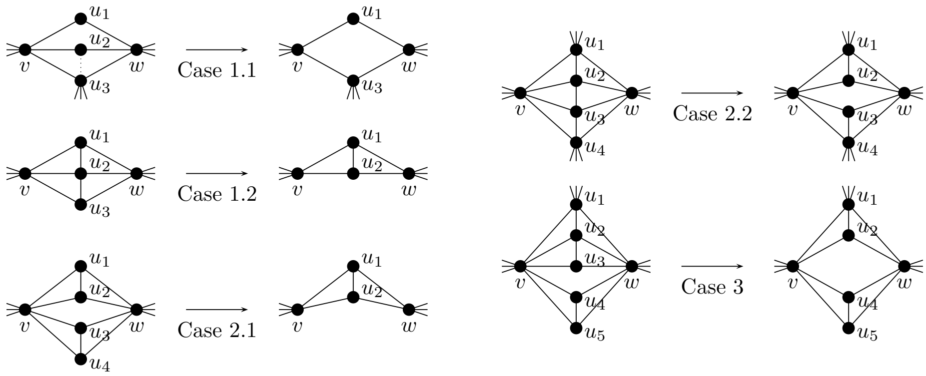


Fig. 2. The five subcases of the reduction rule for planar FDST that yields a linear-size problem kernel.

on G and show that either $|V| = O(k)$ must hold or this optimality is contradicted.[6] Throughout this section, we denote the set of full-degree vertices in the optimal solution tree by F (note that $|F| = k$).

To this end, the following lemma is very helpful.

Lemma 4. *Every vertex in G has distance at most 2 to a vertex in F.* □

Using this lemma, our strategy to prove Theorem 3 is for every vertex in F to partition the set of vertices that have distance at most 2 to it into two sets and separately upper-bound their size. More specifically, Section 3.1 introduces the concept of *region decompositions* which use the set F to divide the input graph into small areas.[7]

In Section 3.2, it is shown that the number of regions in a region decomposition is bounded above by $O(|F|)$ (Lemma 9) and that in the reduced graph, every region contains only a constant number of vertices (Lemma 13). Overall, this shows that there are at most $O(|F|)$ vertices lying inside regions (Proposition 14). Essentially following the same strategy in Section 3.3, we upper-bound the number of vertices that do not lie inside regions by $O(|F|)$ (Proposition 18). The linear kernel claimed in Theorem 3 directly follows from the $O(|F|)$ upper bounds on the number of vertices in regions and outside regions.

3.1 Neighborhood Partition and Region Decomposition

This section prepares the proof of the size-$O(|F|)$ upper bound of a reduced graph by introducing two partitions of the vertices in $V \setminus F$. One partition is

[6] This sort of proof strategy has first been used in work dealing with the MAX LEAF SPANNING TREE problem [8, 9].

[7] The proof concept is, in this sense, similar to the one for the linear-size problem kernel for DOMINATING SET in planar graphs due to Alber et al. [1].

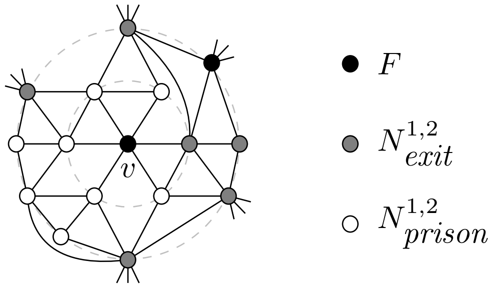
"local" in that for every vertex in F, it concerns vertices within distance at most 2 to it; we partition this *2-neighborhood* into so-called "exit vertices" and "prison vertices." The other partition, called *region decomposition*, is somewhat more "global" in that it concerns the union of 2-neighborhoods for some pairs of vertices in F. These partitions are subsequently used in Sections 3.2 and 3.3 to show the desired linear size of a reduced planar graph.

As to the notation used, for a set $V' \subseteq V$ the subgraph of G *induced* by V' is denoted by $G[V'] = (V', E')$. For a vertex $v \in V$, we denote the vertices having distance 2 to v by $N_2(v)$. By $N_{1,2}(v)$, we denote $N(v) \cup N_2(v)$. Furthermore, we set $N^1(v) := N(v) \setminus F$ and let $N^2(v)$ denote all vertices that have distance exactly 2 to v in the induced graph $G[(V \setminus F) \cup \{v\}]$. Finally, $N^{1,2}(v) := N^1(v) \cup N^2(v)$. This section and the following always consider some fixed embedding of G in the plane (hence, we call G plane instead of planar).

As already mentioned, Lemma 4 shows that every vertex in G has distance at most 2 to a full-degree vertex. Thus, if we can upper-bound the number of vertices in $\bigcup_{v \in F} N^{1,2}(v)$ by $O(|F|)$, the linear problem kernel follows as claimed. In order to do this, we partition the vertices in $N^{1,2}(v)$ into two subsets, separately upper-bounding their sizes in Sections 3.2 and 3.3:

$$N^{1,2}_{exit}(v) := \{u \in N^{1,2}(v) \mid u \in N(w) \text{ for a vertex } w \notin N^{1,2}(v)\},$$
$$N^{1,2}_{prison}(v) := N^{1,2}(v) \setminus N^{1,2}_{exit}(v).$$

For a subset $V' \subseteq V$, we use $N^{1,2}_{exit}(V')$ to denote $\bigcup_{v \in V'} N^{1,2}_{exit}(v)$. The intuition of the partition is that the "exit" vertices have edges to vertices that lie outside of $N^{1,2}(v)$ whereas the "prison" vertices have no edge to a vertex different from $v \cup N^{1,2}(v)$.[8] As an example, the following illustration shows the partition of a neighborhood $N^{1,2}(v)$ into $N^{1,2}_{exit}(v)$ and $N^{1,2}_{prison}(v)$:



As it turns out, every vertex in $N^{1,2}_{exit}(v)$ is "caught" between two vertices in F, that is, for these vertices a stronger variant of Lemma 4 holds.

Lemma 5. *Given a vertex $v \in F$, every vertex $u \in N^{1,2}_{exit}(v)$ lies on a length-at-most-five path between v and an other vertex $w \in F$ (where $w \neq v$).* □

Lemma 5 can be used to eventually upper-bound the size of $N^{1,2}_{exit}(F)$ because the length-at-most-five paths form "bounded areas" between vertices from F called "regions."

[8] See [1] for a similar notion in the context of Dominating Set.

Definition 6. *A* region $R(v,w)$ *between two vertices* $v,w \in F$ *is a closed subset of the plane with the following properties:*

1. *The* boundary *of* $R(v,w)$ *is formed by two paths between* v *and* w*; the length of each path is at most five.*
2. *All vertices which lie* strictly *inside of* $R(v,w)$*—that is, they do not lie on the boundary—are from* $N^{1,2}(v) \cup N^{1,2}(w)$*. (Observe that no vertex from* F *lies strictly inside of a region.)*

To denote the vertices that lie in $R(v,w)$*, we use* $V(R(v,w))$*, that is,* $V(R(v,w))$ *is the union of the boundary vertices and the vertices lying strictly inside of* $R(v,w)$*.*

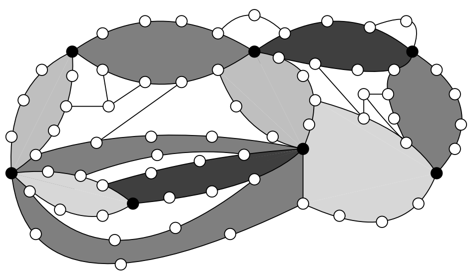
Using this definition, a plane graph can be partitioned into a number of regions by a so-called *region decomposition.*

Definition 7. *An* F-region decomposition *of* G *is a set* $\mathcal{R}$ *of regions* $R(v,w)$ *with* $v,w \in F$ *such that the following holds:*

1. *Except for* v *and* w*, no vertex from* $V(R(v,w))$ *belongs to* F*.*
2. *There is no vertex that lies strictly inside of more than one region from* $\mathcal{R}$*. (The boundaries of regions may touch each other.)*

For an F*-region decomposition* $\mathcal{R}$*, we let* $V(\mathcal{R}) := \bigcup_{R \in \mathcal{R}} V(R)$*. An* F*-region decomposition* $\mathcal{R}$ *is called* maximal *if there is no region* $R \notin \mathcal{R}$ *such that* $\mathcal{R}' := \mathcal{R} \cup \{R\}$ *is an* F*-region decomposition with* $V(\mathcal{R}) \subsetneq V(\mathcal{R}')$*.*

An example of a (maximal) F-region decomposition is the following (the full-degree vertices in F are colored black, the shaded areas are the regions of the decomposition):



Our notions of "region" and "region decomposition" are similar to the technique used by Alber et al. [1] for proving a linear-size kernel for Dominating Set in planar graphs. However, the problem structure of FDST makes the proof somewhat more involved: Whereas Alber et al. were able to bound their regions by length-3 paths, the FDST problem appears to require longer bounds (that is, length-5 paths in our proofs). The reason for this is that in Dominating Set, a vertex affects only its direct neighborhood whereas the full-degree property affects vertices in the 2-neighborhood.

The following lemma shows that it is possible to construct a maximal F-region decomposition $\mathcal{R}$ satisfying $N^{1,2}_{exit}(F) \subseteq V(\mathcal{R})$. By subsequently bounding the number of vertices that lie in regions in Section 3.2, this allows us to upper-bound the number of vertices in $N^{1,2}_{exit}(F)$ so that we only have to deal with vertices from $N^{1,2}_{prison}(F)$ in Section 3.3.

Lemma 8. *For a plane graph $G = (V, E)$, there exists a maximal F-region decomposition $\mathcal{R}$ of G such that $N^{1,2}_{exit}(F) \subseteq V(\mathcal{R})$.* □

3.2 Bounding the Number of Vertices in Regions

In this section, we show that the F-region decomposition $\mathcal{R}$ from Lemma 8 has $O(|F|)$ vertices lying inside of regions (Proposition 14). The proof of this is achieved in several steps: First, Lemma 9 shows that the total number of regions is $O(|F|)$. Then, Proposition 12 upper-bounds the number of length-2 paths that can occur between two vertices in the reduced graph; this proposition is heavily used to prove that every region contains at most $O(1)$ vertices in Lemma 13. Finally, Proposition 14 follows from Lemmas 9 and 13.

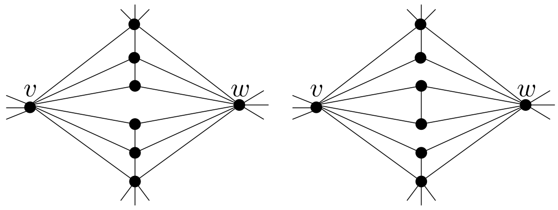
Lemma 9. *For $|F| \geq 3$, the maximal F-region decomposition $\mathcal{R}$ in Lemma 8 consists of at most $6|F| - 12$ regions.* □

To show the constant number of vertices in each region, we make use of the following structure:

Definition 10. *Let v and w be two distinct vertices in a plane graph G. A* diamond $D(v, w)$ *is a closed area of the plane that is bounded by two length-2 paths between v and w such that every vertex that lies inside of the closed area is a neighbor of both v and w. If i vertices lie strictly inside of a diamond, then the diamond is said to have $(i + 1)$* facets *(a facet is an area enclosed by two length-2 paths).*

Lemma 11. *A reduced plane graph does not contain a diamond with more than 5 facets.* □

The only two possible 5-facet diamonds (the worst-case diamonds, so to say) that might remain after the data reduction are the following:



The absence of diamonds with too many facets is central to showing the linear size of the problem kernel for FDST: At most 2 vertices of any diamond can be full-degree, and hence the possibility of arbitrarily large diamonds would prohibit a provable upper bound of the reduced graph. For the same reason, it is important that there cannot be arbitrarily many length-2 paths between two vertices of the input graph. Therefore, Lemma 11 is generalized in order to obtain a upper bound on the number of length-2 paths between two vertices.

Proposition 12. *Let v and w be two vertices in a reduced plane graph G such that an area $A(v,w)$ of the plane is enclosed by two length-2 paths between v and w. If neither the middle vertices of the enclosing paths nor any vertex inside of the area are contained in F, then the following holds:*

1. *If $v, w \notin F$, at most eight length-2 paths from v to w lie inside of $A(v,w)$.*
2. *If $v \notin F$ or $w \notin F$, at most sixteen length-2 paths from v to w lie inside of $A(v,w)$.* □

Using the upper bound on the number of length-2 paths between two vertices that Proposition 12 establishes, it is possible to bound above the number of vertices that can lie inside of a region.

Lemma 13. *Every region $R(v,w) \in \mathcal{R}$ contains $O(1)$ vertices.* □

In conjunction with the $O(|F|)$ upper bound on the number of regions that was established in Lemma 9, this directly allows us to bound above the number of vertices that lie inside of regions.

Proposition 14. *The total number of vertices lying inside of regions of $\mathcal{R}$ is $O(|F|)$.* □

3.3 Bounding the Number of Vertices Outside of Regions

In the last section, we have bounded above the number of vertices that lie inside of regions of a maximal F-region decomposition $\mathcal{R}$. It remains to bound above the vertices that do not lie in regions. By Lemma 8, every one of these remaining vertices is in $N^{1,2}_{prison}(v)$ for some vertex $v \in F$. To bound the number of these vertices by $O(|F|)$, we essentially follow the same strategy as in the last section: We show that each vertex in $N^{1,2}_{prison}(v)$ lies in one of $O(|F|)$ so-called "prison areas" and that each such prison area contains a constant number of vertices.

Definition 15. *Given a maximal F-region decomposition, a* prison area *for a vertex $v \in F$ is a closed area of the plane with the following properties:*

1. *All vertices that lie strictly inside of the area are from $N^{1,2}_{prison}(v)$.*
2. *The area cannot be extended to include any vertex from $N^{1,2}_{prison}(v)$ without violating the first condition.*

Analogously to the preceding section, we first show that the number of prison areas is upper-bounded by $O(|F|)$ and then show that each area contains a constant number of vertices.

Lemma 16. *Given a maximal F-region decomposition $\mathcal{R}$, the vertices that do not lie inside of any region of $\mathcal{R}$ form at most $12|F| - 24$ prison areas (again, we assume $|F| \geq 3$).* □

Lemma 17. *Every prison area contains $O(1)$ vertices.* □

Proposition 18. *The number of vertices lying outside of regions of $\mathcal{R}$ is bounded above by $O(|F|)$.* □

4 A Tree Decomposition-Based Algorithm

There exists a tree decomposition-based algorithm that solves both Minimum-Vertex Feedback Edge Set and Maximum Full-Degree Spanning Tree in $O((2\omega)^{3\omega} \cdot \omega \cdot n)$ time on an n-vertex graph with a given tree decomposition of width ω.[9] We omit its description here due to lack of space.

Theorem 19. *For an n-vertex graph G with a given width-ω tree decomposition (T, X), both* Minimum-Vertex Feedback Edge Set *and* Maximum Full-Degree Spanning Tree *can be solved in $O((2\omega)^{3\omega} \cdot \omega \cdot n)$ time.* □

Using the linear problem kernels for VFES and FDST that were established in Theorems 1 and 3, Theorem 19 implies subexponential-time algorithms for these problems when restricted to planar graphs:

Theorem 20. *In n-vertex planar graphs,* Minimum-Vertex Feedback Edge Set *is solvable in $O(2^{O(\sqrt{k}\log k)} + k^5 + n)$ time and* Maximum Full-Degree Spanning Tree *in $O(2^{O(\sqrt{k}\log k)} + k^5 + n^3)$ time, where k is the number of degree-reduced vertices or full-degree vertices, respectively.*

Proof. Both problems have linear-size kernels in planar graphs, that is, the reduced graphs have at most $O(k)$ vertices after performing the respective kernelization. The claim follows from Theorem 19 and the facts that planar $O(k)$-vertex graphs have treewidth $\omega = O(\sqrt{k})$ and that a corresponding tree decomposition of width $3\omega/2$ can be computed in $O(k^5)$ time [17]. □

5 Conclusion

Our main technical contribution is the proof of the linear-size problem kernel for FDST in planar graphs. It is easily conceivable that there is room for significantly improving the involved worst-case constant factors—the situation is comparable (but perhaps even more technical) with analogous results for Dominating Set in planar graphs [1, 5]. Having obtained linear problem kernel sizes for a problem and its dual, the way for applying the lower bound technique for kernel size due to Chen et al. [5] now seems open.

Another interesting line of future research might be to investigate whether our problem kernel result for planar graphs can be lifted to superclasses of planar graphs—corresponding results for Dominating Set in this direction are reported in [11]. Perhaps even more importantly, it would be interesting to pursue experimental studies (similar to Bhatia et al. [3]) with real-world data in order to explore the practical usefulness of our algorithms and problem kernelizations.

[9] Broersma et al. [4] have already shown that both VFES and FDST are solvable in linear time for graphs with bounded treewidth by expressing the problems in monadic second-order logic. The linear-time solvability follows then from a result of Arnborg et al. [2]. However, this approach does not provide an exact dependency of the running time on the treewidth. Therefore, their result cannot imply a subexponential-time algorithm.

References

1. J. Alber, M. R. Fellows, and R. Niedermeier. Polynomial-time data reduction for Dominating Set. *Journal of the ACM*, 51:363–384, 2004.
2. S. Arnborg, J. Lagergren, and D. Seese. Easy problems for tree-decomposable graphs. *Journal of Algorithms*, 12:308–340, 1991.
3. R. Bhatia, S. Khuller, R. Pless, and Y. Sussmann. The full degree spanning tree problem. *Networks*, 36:203–209, 2000.
4. H. J. Broersma, A. Huck, T. Kloks, O. Koppius, D. Kratsch, H. Müller, and H. Tuinstra. Degree-preserving trees. *Networks*, 35:26–39, 2000.
5. J. Chen, H. Fernau, I. A. Kanj, and G. Xia. Parametric duality and kernelization: Lower bounds and upper bounds on kernel size. In *Proc. of 22nd STACS*, volume 3404 of *LNCS*, pages 269–280. Springer, 2005.
6. F. K. H. A. Dehne, M. R. Fellows, H. Fernau, E. Prieto, and F. A. Rosamond. Nonblocker: Parameterized algorithmics for minimum dominating set. In *Proc. of 32nd SOFSEM*, volume 3831 of *LNCS*, pages 237–245. Springer, 2006.
7. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, Berlin, 1999.
8. V. Estivill-Castro, M. R. Fellows, M. A. Langston, and F. A. Rosamond. FPT is P-time extremal structure I. In *Proc. 1st ACiD*, pages 1–41, 2005.
9. M. R. Fellows, C. McCartin, F. A. Rosamond, and U. Stege. Coordinatized kernels and catalytic reductions: An improved FPT algorithm for Max Leaf Spanning Tree and other problems. In *Proc. of 20th FSTTCS*, volume 1974 of *LNCS*, pages 240–251. Springer, 2000.
10. J. Flum and M. Grohe. *Parameterized Complexity Theory.* Springer-Verlag, 2006.
11. F. V. Fomin and D. M. Thilikos. Fast parameterized algorithms for graphs on surfaces: Linear kernel and exponential speed-up. In *Proc. 31st ICALP*, volume 3142 of *LNCS*, pages 581–592. Springer, 2004.
12. S. Khuller, R. Bhatia, and R. Pless. On local search and placement of meters in networks. *SIAM Journal on Computing*, 32(2):470–487, 2003.
13. M. Lewinter. Interpolation theorem for the number of degree-preserving vertices of spanning trees. *IEEE Transactions on Circuits Systems I: Fundamental Theory and Applications*, 34(2):205–205, 1987.
14. R. Niedermeier. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, 2006.
15. L. E. Ormsbee. Implicit network calibration. *Journal of Water Resources Planning Management*, 115:243–257, 1989.
16. L. E. Ormsbee and D. J. Wood. Explicit pipe network calibration. *Journal of Water Resources Planning Management*, 112:116–182, 1986.
17. P. D. Seymour and R. Thomas. Call routing and the ratcatcher. *Combinatorica*, 14(2):217–241, 1994.