A Bounded Search Tree Algorithm for Parameterized FACE COVER

Faisal N. Abu-Khzam, Henning Fernau, Michael A. Langston

Abstract

The parameterized complexity of the FACE COVER problem is considered. The input to this problem is a plane graph \( G \) with \( n \) vertices. The question asked is whether, for a given parameter value \( k \), there exists a set of \( k \) or fewer faces whose boundaries collectively cover (contain) every vertex in \( G \). Early attempts achieved run times of \( O^*(12^k) \) or worse, by reducing the problem into a special form of DOMINATING SET, namely RED-BLUE DOMINATING SET, restricted to planar graphs.

Here, we consider a direct approach, where some surgical operation is performed on the graph at each branching decision. This paper builds on previous work of the authors and employs a structure theorem of Aksionov et al., with a detailed case analysis, to produce a FACE COVER algorithm that runs in \( O(4.6056^k + n^2) \) time.

We also point to the tight connections with RED-BLUE DOMINATING SET on planar graphs via the annotated version of FACE COVER that we consider in our search tree algorithm. Hence, we get both a \( O(4.6056^k + n^2) \) time algorithm for solving RED-BLUE DOMINATING SET on planar graphs and a polynomial time algorithm for producing a linear kernel for ANNOTATED FACE COVER.

1 Introduction and Preliminaries

The bounded search tree technique is probably the most popular method for the design of efficient fixed-parameter algorithms [7]. Strategies based on this technique are also referred to as “branching algorithms.” Nodes of a search tree correspond to problem instances, and children (or subtrees) of a single node correspond to a number of mutually exclusive decisions that can be taken during the search. The branching factor of a search tree node is the number of subtrees rooted at that node. The branching factor of a search tree algorithm is the maximum branching factor taken over all the nodes in the search tree of that algorithm. In this paper, we apply this strategy to one of the many variants of cover and domination problems (that can be seen as the testbed for parameterized algorithms), more precisely to FACE COVER, described formally below.

It is well-known (due to Euler’s formula) that any planar graph has at least one vertex of degree five or less. This property can be exploited in a variety of ways. In the FACE COVER problem, for example, a vertex of degree at most five belongs to at most five faces, of which one (or more) must be used to cover it. Similarly in PLANAR DOMINATING SET, a vertex of degree at most five must either be in the dominating set or be dominated by one of its neighbors. In each case, when branching is applied, the root of the search tree is guaranteed to have a small branching factor (five for FACE COVER, six for PLANAR DOMINATING SET).

\(^0\)A preliminary version of this paper appeared under the title “Asymptotically Faster Algorithms for Parameterized FACE COVER” in the Proceedings of the first ACiD conference held in Durham (UK), 2005.

\(^*\)Research supported by the Lebanese American University under grant URC-c2004-63.

\(^†\)This research has been supported by the U.S. National Science Foundation under grant CCR-0311500, by the U.S. National Institutes of Health under grants 1-P01-DA-015027-01, 5-U01-AA-013512-02 and 1-R01-MH-074460-01, and by the European Commission under the Sixth Framework Programme.
Great care must be taken, however, if one is to design a branching algorithm around this property. This is because subsequent nodes in the expansion of the search tree are apt to have larger branching factors. See [3] for a detailed discussion of this phenomenon in the case of planar dominating set.

The main result of [1] was to produce an algorithm for the parameterized face cover problem with branching factor five. In this paper, we build upon and improve on this result with the use of a structural theorem quoted below.

But let us first set the scene of this paper by explaining some basic notions of parameterized graph algorithms. We make use of standard notation in graph theory and parameterized algorithmics. For example, \( N(v) \) denotes all neighbors of \( v \), \( N[v] = N(v) \cup \{v\} \), \( deg(v) = |N(v)| \). We also point to the important difference between a planar graph (i.e., a graph that could be embedded in the plane) and a plane graph (i.e., a graph that is embedded in the plane).

The size of a problem instance is denoted by \( n \). The size of the relevant parameter is denoted by \( k \leq n \). A bound of \( O^*(f(k)) \) for a parameterized problem, where \( f \) is some super-polynomial function, means that there is a polynomial function \( p \) such that \( O(f(k)p(n)) \) is the true running time of the algorithm. In other words, a parameterized problem belongs to the class \( \mathcal{FPT} \) of fixed-parameter tractable problems iff it admits an algorithm whose running time can be bounded above by \( O^*(f(k)) \) for some arbitrary function \( f \). Once membership in \( \mathcal{FPT} \) is established, a natural research goal is to ensure better upper bounds on the running times of parameterized algorithms. This is also the direction of this paper. In the case of the branching algorithms we address here, \( f(k) \) will be used to define an upper bound on the number of leaves contained in the search tree generated by the algorithm.

We make use of the following structural theorem.

**Theorem 1** [2] Every connected plane graph with at least two vertices has either

1. two vertices whose degrees sum to at most 5,
2. two vertices at a distance of at most two whose degrees sum to at most 7,
3. a triangular face containing two vertices whose degrees sum to at most 9, or
4. two triangular faces with vertices \( u \) and \( v \) in common, where the degrees of \( u \) and \( v \) sum to at most 11.

We shall focus in our case analysis on the local situation sketched in Fig. 1. We will also call such a pair of vertices \( \{u, v\} \) a 5-6 pair whenever \( deg(u) = 5, deg(v) \in \{5, 6\} \), \( u, w, z \in N[u] \cap N[v] \); we further denote the triangular faces as \( f_w = \{u, w, v\} \) and \( f_z = \{u, v, z\} \). Note that, by virtue of Theorem 1, any planar graph of minimum degree five must have 5-6 pairs.

## 2 The Face Cover Problem

Let us now define the face cover (FC) problem formally as follows:

**Given:** A plane graph \( G = (V, E) \) with face set \( F \) and a positive integer \( k \)

**Question:** Is there a face cover (set) \( C \subseteq F \) with \( |C| \leq k \)?

Here, a face cover is a set of faces whose boundaries contain all vertices of the given plane graph.

Consider a plane graph \( G = (V, E) \) with face set \( F \). If we consider a vertex \( v \) to be given as a set \( F(v) \) of those faces on whose boundaries \( v \) lies, then a face cover set \( C \subseteq F \) corresponds to a hitting set of a hypergraph \( H = (F, E_H) \), where the vertex set \( F \) is the face set of \( G \), and \( E_H = \{F(v) \mid v \in V\} \).
It was shown in [1] that a traditional HS algorithm can then be translated into a FC algorithm, where rather than deleting vertices or faces, they are marked. Vertices that are not yet marked are called active. To formulate this modified problem, we need some additional notation. Let

- $F_a$ collect all active faces;
- $F_a(v)$ collect all active faces incident to vertex $v$; $\text{deg}_a(v) = |F_a(v)|$ be the face degree of $v$;
- $V_a$ collect all active vertices;
- $V_a(f)$ collect all active vertices on the boundary of face $f$; $\text{deg}_a(f) = |V_a(f)|$ be the face size of $f$.

To a graph, we may associate two marking functions $\mu_F$ and $\mu_V$. For a face $f$, $\mu_F(f) =$ active if $f \in F_a$, and $\mu_F(f) =$ marked if $f \notin F_a$. For a vertex $v$, $\mu_V(v) =$ active if $v \in V_a$, and $\mu_V(v) =$ marked if $v \notin V_a$. Observe that the information expressed by $\mu_F$ and by $F_a$ is the same, likewise that by $\mu_V$ and by $V_a$.

Notice that the face degree and the usual notion of the degree of a graph need not coincide, even with simple graphs where all faces are active. This is due to the fact that a face can be incident with the same vertex more than once if we consider multiplicities when we “walk” along the face boundary; consequently, also the face size need not correspond to the length of a closed walk (as just described). Moreover, if we consider multigraphs (graphs with multiple edges), even a vertex with no neighbor but itself can be incident to arbitrarily many faces and hence have an arbitrarily large face degree, see Fig. 2.

Initially, in the classical FACE COVER problem, all vertices and all faces are active. However, in the course of the algorithm, we are dealing with an annotated version of FC, called ANNOTATED FACE COVER (FCANN) in the following, which is more formally:

**Given:** A plane (multi-)graph $G = (V, E)$ with face set $F$, a function $\mu_V : V \rightarrow \{\text{active, marked}\}$, a function $\mu_F : F \rightarrow \{\text{active, marked}\}$, and a positive integer $k$.

**Question:** Is there a set $C \subseteq \{f \in F \mid \mu_F(f) = \text{active}\}$ with $|C| \leq k$ and $\forall v : \mu_V(v) = \text{active} \Rightarrow F_a(v) \cap C \neq \emptyset$?

We use another quite natural notion that is needed in our first branching algorithm and its analysis.

Let $G = (V,E)$ be a plane multigraph (i.e., a multigraph that is embedded in the plane) with marking functions $\mu_V$ and $\mu_F$. Let $v$ be a marked vertex whose only neighbor is $v$ itself, i.e., the only edges incident
to \( v \) are loops. Let \( f_0, \ldots, f_r \) be all faces incident to \( v \). Since we are considering an embedded graph, we can assume that \( f_0 \) contains \( f_1 \) that contains \( f_2 \) etc. in a geometrical sense (i.e., due to the embedding). For \( f_r \), let \( V_r \) be the set of vertices that is geometrically contained in \( f_r \) (excluding \( v \)). Consider the (possibly empty) graph \( G_r \) that is induced by \( V_r \) embedded as before, with \( f_r \) being now the outer face of \( G_r \). Also the marking functions of \( G_r \) is obtained from those of \( G \) by simple restrictions. For \( 0 \leq i < r \), let \( V_i \) be the set of vertices that is geometrically contained in \( f_i \), excluding \( v \) and all vertices from \( V_{i+1} \) up to \( V_r \). Now, the graph \( G_i \) is induced by \( V_i \) and is embedded as before, with \( f_i \) being now the outer face of \( G_i \). Since \( v \) is marked, \( G \) has a face cover set of size \( k \) iff the island graphs \( G_0, \ldots, G_r \) have face cover sets of size \( k_0, \ldots, k_r \) such that \( k = \sum_{i=0}^{r} k_i \). Notice that we could have defined “islands” also in more general terms; however, this is the form we will need in the analysis of our algorithm.\(^1\) Referring to this notion, we also say that an annotated plane multigraph contains no island if it has no marked vertices with loops.

3 Reduction Techniques

In our reduction rules, marked vertices are removed by a sort of a triangulation operation. This geometrical surgery allows us to finally use the fact that each planar graph possesses a vertex of face degree at most five to branch at. Let us first describe some Hitting Set reduction rules in accordance with the notation introduced above. See [8, 9, 11, 12, 13, 14], to mention some recent references.

**Rule 1** (dominated vertex rule) If \( F_a(u) \subseteq F_a(v) \) for some active vertices \( u, v \), then mark \( v \).

**Rule 2** If \( \deg_a(v) = 1 \) (i.e., \( F_a(v) = \{ f \} \)) and \( v \) is active, then put the unique active incident face \( f \) into the face cover and mark both \( v \) and \( f \).

**Rule 3** (dominated face rule) If \( V_a(f) \subseteq V_a(f') \) for some active faces \( f, f' \), then mark \( f \).

In the aforementioned Hitting Set literature, Rule 1 corresponds to the edge domination rule, Rule 2 corresponds to the tiny edge rule, and Rule 3 is known as the vertex domination rule.

\(^1\)This part is missing in the earlier version of the search tree algorithm for FACE COVER as presented in [1].
Figure 3: Two cases of Rule 5, the left-hand side illustrating the situation where $v \neq w$, and the right-hand side the situation when $v = w$. The dashed line shows the border between the two new faces obtained by splitting $f$ into an active face $f'$ and a marked face $f_m$.

As shown in [1], we have to be cautious with simply deleting vertices and faces, so that we only mark them with the above rules. However, it is indeed possible to simplify the obtained graph with a couple of surgery rules.

**Rule 4** If $u$ and $v$ are two marked vertices with $u \in N(v)$, then merge $u$ and $v$ into a marked vertex $[u, v]$.

Notice that Rule 4 may introduce multiple edges or loops, even though we started off with a simple graph. Namely, in contrast to the common use of the notion of merging two vertices in graphs, we are not simplifying the graph thereafter by deleting “superfluous” edges. Rather, we have to keep them, since we have to preserve the face information. Therefore, “merging two vertices” means “contracting two vertices along one of the connecting edges.” It should also be noticed that there is a certain nondeterminism concerning where to embed the thus possibly newly created degenerate faces. However, it is not difficult to see that it does not matter at all where these faces are put regarding the solution of the ANNOTATED FACE COVER problem at hand.

The following rule is possibly the most intricate one of all rules that are listed here and therefore it is illustrated in Fig. 3.

**Rule 5** If $v$ is a marked vertex with two not necessarily different active neighbors $u, w$ such that $u, v, w$ are all incident to an active face $f$, with one edge connecting $u$ and $v$ and another edge connecting $v$ and $w$ on the boundary of $f$, then partition $f$ into two faces by introducing a new edge between $u$ and $w$, and that edge has to be drawn inside of $f$. This way, our graph may get (additional) multiple edges.

The new face $f_m$ bordered by $u, v, w$ is marked, while the other part $f'$ of what was formerly the face $f$ will be active. $f_m$ is triangular if $u \neq w$.

**Rule 6** If $v$ is incident to only one face and if $v$ is marked, then delete $v$.

Note that Rule 6 applies when the degree of $v$ is in $\{0, 1\}$ and when $v$ is a cut-vertex (which includes the case where $v$ belongs to an acyclic component of the graph).

**Rule 7** If $\deg_a(v) = 0$ and if $v$ is marked, then delete $v$. The new face that will replace all the marked faces that formerly surrounded $v$ will be marked, as well.

If $\deg_a(v) = 0$ and if $v$ is active, then $(G, F, \mu_V, \mu_F, k)$ is a NO-instance of ANNOTATED FACE COVER.
Notice that the second part of the previous rule will never trigger when the original input was a (non-annotated) FC instance.

**Rule 8** Let \( e \) be an edge incident to \( f \) and to \( f' \). If (1) \( f \not\in F_a \) or \( \{f, f'\} \subseteq F_a \) and if (2) \( V_a(f) \subseteq V_a(f') \), then delete \( e \) and merge \( f \) and \( f' \) into a (possibly) new face carrying the former marking \( \mu_F(f') \).

**Lemma 1** The reduction rules for ANNOTATED FACE COVER are sound.

**Proof.** To simplify notation, let \( I = (G = (V, E), F, \mu_V, \mu_F, k) \) be an instance of ANNOTATED FACE COVER and let \( I' = (G' = (V', E'), F', \mu'_V, \mu'_F, k') \) denote the instance obtained from \( I \) by applying a specified reduction rule once. We also use \( F_a, F'_a, V_a, V'_a \) when convenient (as induced by the marking functions).

Rule 1: The rule only affects the vertex marking of \( v \), triggered by the condition on \( u \). If \( C \subseteq F'_a \) is a face cover set, then \( C \) would be also a face cover set of \( I \), since there must be a face \( f \in C \) with \( u \) on its boundary, more specifically, \( f \in F_a(u) \). Since \( F_a(u) \subseteq F_a(v) \), \( v \) is also on the boundary of \( f \). The converse direction is trivial. Notice that the proof almost literally transfers from the corresponding HITTING SET rule as contained in [8].

Rule 2: This rule is evident, since active vertices need to be covered.

Rule 3: This is another HITTING SET rule whose proof is analogous to the one contained in [8].

Rule 4: Notice that there is an obvious bijection between \( F \) and \( F' \) (and the corresponding markings), since no faces are introduced nor destroyed by the operation. Moreover, since \( u, v, [u, v] \) are all marked, a feasible face cover of \( I \) is also feasible for \( I' \) and vice versa.

Rule 5: If \( C \) is a feasible face cover for \( I \), then \( C' = C \) will be a feasible face cover for \( I' \) if \( f \not\in C \), and \( C' = (C \cup \{f\}) \setminus \{f\} \) will be feasible for \( I' \) if \( f \in C \). The converse is similarly seen.

Rule 6: This rule is evident, since marked vertices need not be covered.

Rule 7: This is similarly trivial.

Rule 8: The operation should be clear if there is only one vertex on the boundary of \( f \). If \( f \) contains two vertices \( u, v \) on its boundary, then the rule tells us to select an edge \( e \) connecting \( u \) and \( v \). This guarantees that the other face having \( e \) on its boundary also contains at least \( u \) and \( v \) on its boundary. The soundness of the rule follows by the same argument as that for the dominated face rule.

4 A simple Branching Algorithm

Our first branching algorithm (Algorithm 1 below) applies the reduction rules listed above and is very similar to the branching algorithm described in [1]. Note that Algorithm 1 is parameter-driven. In other words, the size of the desired solution (the parameter \( k \)) must be given and used towards finding the solution. However, we can treat it as a minimization algorithm because it can be called iteratively \((O(\log k) \text{ times})\) to find an optimum solution. This is assumed in our algorithm when we need to compute minimum face covers for the island graphs previously described. Another (rather subtle) issue is the following one: In Algorithm 1, first the reduction rules are exhaustively applied. Therefore, as the reader may verify, \( G \) "contains a marked vertex \( v \) whose only neighbor is \( v \) itself" iff \( G \) contains an island due to Rule 8.

The analysis employed in this section highlights the importance of the use of low-degree vertices and the surgical operation, which we assume in the rest of this paper.
Algorithm 1 A simple search tree algorithm for the annotated FACE COVER problem, called FC-ST

Require: an annotated plane multigraph $G = (V,E)$ with face set $F$ and marking functions $\mu_V$ and $\mu_F$, a positive integer $k$

Ensure: $C$ if $G$ has an annotated face cover set $C \subseteq F$ with $|C| \leq k$; NO otherwise

Exhaustively apply the reduction rules.

1. The resulting instance will be also called $G$ (etc.) as before.
2. if $k < 0$ then
   return NO
3. else if $V_a = \emptyset$ then
   return $\emptyset$
4. else if $G$ contains a marked vertex $v$ whose only neighbor is $v$ itself then
   Recursively compute minimum face covers $C_0, \ldots, C_r$ of the island graphs $G_0, \ldots, G_r$
   if $\exists i : C_i = \text{NO} \lor \sum_{i=0}^{r} |C_i| > k$ then
5.   return NO
6.   else
7.     return $\bigcup_{i=0}^{r} C_i$
8. end if

10. else
11.    Let $v$ be a vertex of lowest face degree in $G$.
12.   {One incident active face of $v$ must be used to cover $v$.}
13.    Choose $f \in F_a$ such that $f$ is incident to $v$.
14.    Mark $f$ and all vertices that are on the boundary of $f$.
15.    Call the resulting marking functions $\mu'_V$ and $\mu'_F$.
16. end else

18. Let $C := \text{FC-ST}(G,F,\mu'_V,\mu'_F,k-1)$.
19. if $C \neq \text{NO}$ then
20.   return $C \cup \{f\}$
21. else
22.   Mark $f$ and return $\text{FC-ST}(G,F,\mu_V,\mu'_F,k)$
23. end if
24. end if
Lemma 2 If $G = (V, E, V_a, F_a)$ is an annotated plane multigraph with face set $F$ (seen as an instance of the ANNOTATED FACE COVER problem with parameter $k$) that is reduced according to the FACE COVER reduction rules listed above and that contains no island, then no marked vertex will exist in $G$.

**Proof.** Assume the hypothesis and that there is a marked vertex $v$ in $G$. The following three observations will be used to demonstrate that $v$ must have been deleted in the reduced graph, and hence that no marked vertices exist in $G$.

1. $v \not\in N(v)$, since otherwise $G$ would contain a possibly empty island.

2. $deg_a(v) > 0$, or otherwise Rule 7 would apply and $v$ would have been deleted.

3. All neighbors of $v$ must be active, or Rule 4 would apply and all non-active neighbors would be merged into $v$.

If $v$ is incident to only one face, this face has to be active by Obs. 2 from the list of observations above. (In particular, this is the case when $v$ has no adjacent vertices.) Then, Rule 6 applies and $v$ is removed. So we can assume that $v$ is incident to at least two faces in the following.

If $v$ has only one adjacent vertex $u$, i.e., $N(v) = \{u\}$ with $u \neq v$ by Obs. 1., then we must have a degenerate face $\{u, v\}$, since otherwise, $v$ would be incident to only one face.

We will now show that there cannot exist any degenerate face $\{u, v\}$ if $v$ is marked. By Obs. 3., the vertex $u$ must be active. $\{u, v\}$ could not be active, since either it should have been marked by Rule 3, or it is the only way to cover $u$ (and thus, it is also marked, together with $u$, using Rule 2). However, if $\{u, v\}$ is marked, Rule 8 applies and merges it with a neighboring face, which is again impossible.

Hence, $v$ has (at least) two different neighbors $u$ and $w$ with $v \not\in \{u, w\}$. By Obs. 3., $u$ and $w$ are active. By Obs. 2., $deg_a(v) > 0$. Then, Rule 5 applies $deg_a(v)$ many times. The new (in general triangular) faces that would be introduced by that rule plus the already previously marked faces incident to $v$ would be in fact all faces that are incident to $v$, and all of them are marked. Hence, Rule 7 (in possible cooperation with Rule 8) applies and deletes $v$.

As mentioned within the reduction rules, it might occur that we create degenerate faces (i.e., faces with only one or two incident vertices) in the course of the application of the reduction rules.

Lemma 3 If $G = (V, E, V_a, F_a)$ is an annotated plane multigraph with face set $F$ (seen as an instance of ANNOTATED FACE COVER with parameter $k$) that is reduced according to the FACE COVER reduction rules listed above and that contains no islands, then the only degenerate faces that might exist are active faces $f$ with two incident vertices. More specifically, each vertex pair $\{u, v\}$ has at most one degenerate face incident to both $u$ and $v$. Moreover, the two faces that are neighbored with such a degenerate face $f$ via common edges are both marked.

**Proof.** Let $f$ be a degenerate face. If $f$ has only one vertex $v$ on its boundary, we observe that $v$ cannot be marked, since then we would find possibly empty island graphs, which is excluded by assumption. Hence, $v$ is active.

- If $f$ is marked, then Rule 8 applies and removes $f$.
- If $f$ and $v$ are active and $deg_a(v) > 1$, then Rule 3 applies and renders $f$ marked, so that then the previous case applies.
• If \( f \) and \( v \) are active and \( \text{deg}_{a}(v) = 1 \), then Rule 2 applies and puts \( f \) into the face cover. Moreover, \( f \) and \( v \) will become marked, so that the first case applies.

Hence, in the end a degenerate face with only one vertex on its boundary cannot exist in a reduced instance.

Consider now a degenerate face \( f \) with two vertices \( u \) and \( v \) on its boundary. Since \( G \) contains no island graphs, the boundary of \( f \) is connected. If \( f \) is marked, then Rule 8 applies and removes \( f \). Otherwise, \( f \) is active. Let \( f' \) be one of the faces with which \( f \) shares one edge. Due to the previous analysis, we may assume that \( f' \) has more than one vertex on its boundary. If \( f' \) were active, then Rule 3 would render \( f \) marked (see previous case). Hence, all faces that share edges with \( f \) are marked in a reduced instance.

Finally, if there would be two degenerate faces having the same vertices \( u \) and \( v \) on its boundary, then Rule 3 would mark one of these, and then Rule 8 would apply and merge the faces.

The following proof is a streamlined and simplified version of the arguments found in [1]. We include it here so that we can build upon it in the analysis to follow.

Theorem 2 [1] Algorithm 1 solves the ANNOTATED FACE COVER problem in time \( O^{*}(5^k) \).

Proof. We have to show that, in an annotated (multi-)graph \( G \) without islands, there is always a vertex with face degree at most five. Having found such a vertex \( v \), the heuristic priority of choosing a face incident to the vertex of lowest face degree (as formulated in Algorithm 1, line 15) would let the subsequent branches be made at faces also neighboring \( v \), so that the claim then follows. The (non-annotated) simple graph \( G' = (V, E') \) obtained from the annotated (multi-)graph \( G = (V, E) \) by putting one edge between \( u \) and \( v \) whenever there is some edge between \( u \) and \( v \) in \( G \) is planar; therefore, we can find a vertex \( v \) of degree at most five in \( G' \). However, back in \( G \), an edge \( uv \) in \( G' \) might correspond to two edges connecting \( u \) and \( v \). Notice that according to previous Lemma 3, no more than two edges in \( G \) could correspond to one edge in \( G' \). Hence, there might be up to ten faces neighboring a low-degree vertex \( v \) in \( G \). Lemma 3 also shows that at most five of these faces can be active.

5 An Improved Face Cover Algorithm

We now use Theorem 1 to improve on the running time of the algorithm. Of course, if the auxiliary graph \( G' \) always contains a vertex of degree four, the algorithm achieves a \( 4^k \) branching behavior. So, the only situation that needs to be analyzed is the following one (in \( G' \)): there are two triangular faces neighbored via an edge \( \{u, v\} \) where the sum of the degrees of \( u \) and \( v \) is at most 11. Since we want to analyze the worst case in the sequel, we can assume that \( \text{deg}(u) = 5 \) and \( \text{deg}(v) = 6 \) for any 5-6 pair to be analyzed, as depicted in Fig. 1. As usual, \( T(k) \) denotes the number of leaves in the search tree corresponding to our algorithm. We now discuss the possibilities for vertex \( u \).

1. If some of the edges incident to \( u \) in \( G' \) represent degenerate faces in \( G \) and some correspond to (simple) edges in \( G \), then Lemma 3 implies that the face degree of \( u \) in \( G \) is less than five, so that we automatically get a favorable branching.

2. If all edges incident to \( u \) in \( G' \) represent degenerate faces in \( G \), then this is true in particular for the edge \( uv \) in \( G' \). In order to achieve \( \text{deg}_{a}(v) = \text{deg}(v) (= 6) \), all edges incident to \( v \) must also represent

\[\text{This was the basic concern when stating the } O^{*}(10^k) \text{ (actually, even worse) algorithm for the FACE COVER problem in [7].}\]
Algorithm 2  An advanced search tree algorithm for ANNOTATED FACE COVER, called FC-ST-advanced

**Require:** an annotated plane graph \( G = (V, E) \) with face set \( F \) and marking functions \( \mu_V \) and \( \mu_F \), a positive integer \( k \)

**Ensure:** \( C \) if \( G \) has an annotated face cover set \( C \subseteq F \) with \(|C| \leq k \); NO otherwise

Exhaustively apply the reduction rules.

\( \{ \) The resulting instance will be also called \( G \) (etc.) as before. \( \} \)

if \( k < 0 \) then
    return NO

else if \( V_a = \emptyset \) then
    return \( \emptyset \)

else if \( G \) contains a marked vertex \( v \) whose only neighbor is \( v \) itself then
    Recursively compute minimum face covers \( C_0, \ldots, C_r \) of the island graphs \( G_0, \ldots, G_r \)
    if \( \exists i : C_i = \text{NO} \lor \sum_{i=0}^{r} |C_i| > k \) then
        return NO
    else
        return \( \bigcup_{i=0}^{r} C_i \)
    end if

else
    Let \( u \) be a vertex of lowest face degree in \( G \).
    \( \{ \) One incident face of \( u \) must be used to cover \( u \). \( \} \)
    if \( \text{deg}_a(u) \leq 4 \) then
        Choose \( f \in F_a \) such that \( f \) is incident to \( u \).
        Mark \( f \) and all vertices that are on the boundary of \( f \).
    \end if

    Call the resulting marking functions \( \mu'_V \) and \( \mu'_F \).
    Let \( C := \text{FC-ST-advanced}(G, F, \mu'_V, \mu'_F, k - 1) \)
    if \( C \neq \text{NO} \) then
        return \( C \cup \{ f \} \)
    else
        execute FC-ST-case-1
    end if

    \( \{ \) Let \( N(u) = \{ u_1, u_2, v, w, z \} \) be the neighbors of \( u \) and similarly \( N(v) = \{ v_1, v_2, v_3, u, w, z \} \). \( \} \)
    If all active faces incident with \( v \) are degenerate then
        execute FC-ST-case-1
    else if no active faces incident with \( v \) are degenerate then
        execute FC-ST-case-2
    else
        execute FC-ST-case-3
    end if

end if
Algorithm 3 The code of FC-ST-case-1

Let \( G' = G \setminus \{u, v\} \) and mark the face \( f \) to which (formally) \( u, v \) belonged; modify \( F, \mu_V, \mu_F \) accordingly, yielding \( F', \mu'_V \) and \( \mu'_F \). Let \( C := \text{FC-ST-advanced}(G', F', \mu'_V, \mu'_F, k - 1) \)
if \( C \neq \text{NO} \) then
return \( C \cup \{f\} \)
else
for all unordered vertex pairs \( \{x, y\} \) such that \( x \in N(u) \setminus \{v\} \) and \( y \in N(v) \setminus \{x, u\} \) do
Modify \( \mu'_V \) so that \( x \) and \( y \) are the only vertices of \( N(u) \cup N(v) \setminus \{u, v\} \) that are marked.
Let \( C := \text{FC-ST-advanced}(G', F', \mu'_V, \mu'_F, k - 2) \)
if \( C \neq \text{NO} \) then
return \( C \cup \{f\} \)
end if
end for
end if
return NO

Algorithm 4 The code of FC-ST-case-2

Let \( G' = G \setminus \{u, v\} \) and mark the face \( f \) to which (formally) \( u, v \) belonged; modify \( F, \mu_V, \mu_F \) accordingly, yielding \( F', \mu'_V \) and \( \mu'_F \).
for all vertices \( x \in \{w, z\} \) do
Modify \( \mu'_V \) so that \( x \) is the only vertex of \( N(u) \cup N(v) \setminus \{u, v\} \) that is marked.
Let \( C := \text{FC-ST-advanced}(G', F', \mu'_V, \mu'_F, k - 1) \)
if \( C \neq \text{NO} \) then
return \( C \cup \{f\} \)
end if
end for
for all vertices \( x \in \{u_1, u_2\} \) and \( y \in \{v_1, v_2, v_3\} \) do
Modify \( \mu'_V \) so that \( x \) and \( y \) are the only vertices of \( N(u) \cup N(v) \setminus \{u, v\} \) that are marked.
Let \( C := \text{FC-ST-advanced}(G', F', \mu'_V, \mu'_F, k - 2) \)
if \( C \neq \text{NO} \) then
return \( C \cup \{f\} \)
end if
end for
return NO

Algorithm 5 The code of FC-ST-case-3

for all faces \( f \in F_a(u), g \in F_a(v) \) do
Mark \( f \) and \( g \) and all vertices in \( V_a(f) \cup V_a(g) \); call the modified marking functions \( \mu'_V \) and \( \mu'_F \).
Let \( C := \text{FC-ST-advanced}(G', F, \mu'_V, \mu'_F, k - |\{f, g\}|) \)
if \( C \neq \text{NO} \) then
return \( C \cup \{f, g\} \)
end if
end for
return NO
degenerate faces in $G$. Otherwise, the branching would be only better as seen in Lemma 3. More precisely, if $\deg_a(v) = 5$ and $\deg(v) = 6$, this can only be if only four out of the six faces incident to $v$ are non-degenerate. Branching first on the degenerate face $uv$ and then (in the case that $uv$ is not taken into the face cover) on all remaining four possibilities to cover $u$ times four possibilities to cover $v$ gives the recursion

$$T(k) \leq T(k - 1) + 16T(k - 2) \leq 4.5312^k.$$  

Let us therefore return to the case when all edges incident to $v$ must also represent degenerate faces in $G$. We are dealing with the case that the edge $\{u, v\}$ is neighboring the marked triangular faces $f_w = \{u, w, v\}$ and $f_z = \{u, v, z\}$ (in $G'$, see Fig. 1). So, we have the following alternatives for covering $u$ and $v$; as usual, we consider first all cases of covering the small-degree vertex $u$.

- Take the degenerate face $\{u, w\}$ into the cover. Then, both faces $\{v, w\}$ and $\{u, v\}$ will get marked by the dominated face rule 3, as they are dominated by the face $\{v, z\}$, since $u$ and $v$ are marked in the discussed branch. Thus, for branching at $v$, only four further cases need to be considered. This is hence yielding four $T(k - 2)$-branches.
- Quite analogously, the case when we take $\{u, z\}$ into the cover can be treated, leading to another four $T(k - 2)$-branches.
- Otherwise, three possibilities remain to cover $u$, leading us to three $T(k - 1)$-branches.

Analyzing this branching scenario gives us: $T(k) \leq 4.7016^k$. However, a closer look at the first and second cases of the above branching scenario reveals that $\{u, w\}$ and $\{v, z\}$ together cover exactly the vertices $\{u, v, w, z\}$; this has exactly the same effect as selecting $\{u, z\}$ and $\{v, w\}$. In other words, both cases lead to reducing $k$ by 2, and to marking exactly the same set of vertices. This means that we can completely neglect one of the two branches (being equivalent). Hence, we get as a recurrence for the search tree size:

$$T(k) \leq 3T(k - 1) + 7T(k - 2)$$

which gives the estimate $T(k) \leq 4.5414^k$.

3. If all edges incident to $u$ (in $G'$) correspond to simple edges in $G$, then we can assume (according to our previous reasonings) that also all edges incident to $v$ (in $G'$) correspond to simple edges in $G$. To further simplify our analysis, we note the following:

**Observation 1** If neither $u$ nor $v$ belongs to a marked face in $G$, then $u$ and $v$ could have at most three common faces in $G'$.

To see this, let $f$ be a face containing $u, v$, and $w$, where $\{u, v\}$ is a 5-6 pair with common neighbors $w$ and $z$. If $f$ is different from $f_w$, then $f_w$ is dominated by $f$. This immediately implies that $f_w$ is marked, which contradicts our assumption that all faces containing $u$ or $v$ are active (otherwise, we would have found a more favorable branching situation at a vertex of face degree of at most four). Since four out of the five faces of $u$ must contain a vertex from $\{w, z\}$, only one face could be common to $u$ and $v$ besides $f_w$ and $f_z$. If $f_w$ and $f_z$ are the only common faces of $u$ and $v$, then we propose the following branching:
• Start branching at the active triangular faces \( f_w \) and \( f_z \) (in \( G' \)). This gives two \( T(k-1) \)-branches.
• Then, branch at the remaining three active faces surrounding \( u \), followed (each time) by branches according to the remaining four active faces surrounding \( v \); overall, this gives 12 \( T(k-2) \)-branches.

Analyzing this branching gives us \( T(k) \leq 4.6056^k \).

Now assume that there is a further common face between \( u \) and \( v \) (besides the triangular faces). Then we can branch at the three common faces of \( u \) and \( v \) first, followed by two times three branches (at \( u \) and \( v \), resp.) to consider all cases for covering \( u \) and \( v \). Hence, we have \( T(k) \leq 3T(k-1) + 6T(k-2) \leq 4.3723^k \).

The above-mentioned branching scenarios are described in Algorithm 2. The overall time complexity of this algorithm is \( O^*(4.6056^k) \).

6 Face Cover, Red-Blue Dominating Set, and Kernelization

We will now deal with the following variation of the DOMINATING SET problem, called the RED-BLUE DOMINATING SET problem: Given a graph \( G = (V, E) \) with \( V \) partitioned as \( V_{\text{red}} \cup V_{\text{blue}} \) and a positive integer \( k \), is there a red/blue dominating set \( D \subseteq V_{\text{red}} \) with \( |D| \leq k \), i.e., \( V_{\text{blue}} \subseteq N(D) \)?

The RED-BLUE DOMINATING SET problem (restricted to planar instances) can be also solved with our algorithm. Formally, we only must (after arbitrarily embedding the given red-blue graph into the plane)

1. mark all faces of the given plane graph \( G \);
2. subdivide all edges \( uv \) of \( G \) with \( u \in V_{\text{red}} \) and \( v \in V_{\text{blue}} \) by (intermediately) introducing new (black) vertices \([u, v] \);
3. consider all blue vertices to be active;
4. for all red vertices \( u \) do:
   (a) if \([u, v_1], \ldots, [u, v_r] \) are all neighbors of \( u \) in clockwise order, then introduce new edges \( v_1v_2, \ldots, v_{r-1}v_r, v_pv_1 \);
   (b) remove all edges incident to \( u \); this creates a new face \( f_u \) with \( u, v_1, \ldots, v_r \) on its boundary;
   (c) finally, remove \( u \) and consider \( f_u \) to be active;
5. contract all edges \([u, v]v \) and call the merger of \([u, v] \) and \( v \) again \( v \), hence keeping its “active” marking; this would finally remove all black intermediate vertices, and we arrive at an instance of ANNOTATED FACE COVER where all vertices are active.

Note that the resulting instance of ANNOTATED FACE COVER could have loops and multiple edges. This is illustrated in figure 6.

The reduction above gives a branching algorithm for planar RED-BLUE DOMINATING SET:

**Corollary 1** RED-BLUE DOMINATING SET, restricted to planar graphs, has an algorithm that runs in time \( O^*(4.6056^k) \).
Figure 4: An example illustrating steps 2, 4(a), 4(b)&(c), and 5 in the reduction from RED-BLUE DOMINATING SET to ANNOTATED FACE COVER.
This is an obvious improvement over the previous algorithm described in [7] running in time $O^*(12^k)$. Notice that there are also competing (asymptotically even better) FPT algorithms for FC and planar RED-BLUE DOMINATING SET that run in time $O^*(e^{\sqrt{k}})$. However, the constant $c$ is quite huge (current “record” seems to be $c \leq 2^{24.551}$, see [10]).

Observe that in the abstract we claim a running time of $O(4.6056^k + p(n))$ for some polynomial $p(n)$ for FACE COVER. We can of course refer to the quadratic-time preprocessing derived in [1]. There is another way to deduce a linear size kernel for ANNOTATED FACE COVER; however, it is not clear if such a small kernel exists for FACE COVER itself. Namely, in the long version of [6], it was concluded:

**Corollary 2** PLANAR RED/BLUE DOMINATING SET admits a problem kernel of size $67 \cdot k$ (where the size of a problem instance is measured in terms of the vertices of the graph).

This kernelization can be used to produce a small linear kernel for ANNOTATED FACE COVER as follows: Given an instance of ANNOTATED FACE COVER, we can produce an equivalent instance of RED-BLUE DOMINATING SET, restricted to planar graphs, by interpreting faces by means of “face vertices”, as exhibited in [10]. Now, we can use the result of Cor. 2 to get another equivalent planar graph instance of RED-BLUE DOMINATING SET with at most $67k$ vertices. As described above, such a graph can be translated back into an equivalent instance of ANNOTATED FACE COVER with at most $67k$ vertices plus faces.

**Corollary 3** ANNOTATED FACE COVER admits a problem kernel of size $67 \cdot k$ (where the size of a problem instance is measured in terms of the number of vertices plus the number of faces of the given plane graph).

Now, observe that the average vertex degree of a planar graph is less than six (this follows by the same Euler-type argument that shows the existence of a vertex of degree five in a planar graph). Hence, the number of faces of a plane graph is less than six times the number of vertices in the graph. Therefore, we can conclude:

**Corollary 4** ANNOTATED FACE COVER admits a problem kernel of size $56 \cdot k$ (where the size of a problem instance is measured in terms of the number of faces of the given plane graph).

**Proof.** Let $n$, $m$, $c$, and $f$ be the number of vertices, edges, components, and faces, resp., of a given plane graph that is reduced according to the previous procedure. It is well-known that $f + n = m + c + 1$. Since $m \leq 3n - 6c$, we can derive: $f + n \leq 3n - 5c + 1$. As we argued before, edges may play the role of degenerate faces. Hence, in a reduced plane multigraph, we have the estimate: $f + n \leq 6n - 11c + 1$. Since $c = 1$ yields the worst case, we conclude: $f + n \leq 6n - 10$, i.e., $n \geq f/5 + 2$.

By the previous corollary, $n + f \leq 67k$. Hence, $(6/5)f + 2 \leq 67k$, which entails the claim.

7 Concluding Remarks

Bienstock and Monma [4] considered a variant of the FACE COVER problem where some preselected vertices need not be covered. This variant can be solved by our algorithm, as well, since it evidently gives a restriction of the ANNOTATED FACE COVER problem. More generally speaking, Fernau and Juedes [10] introduced a (restricted) form of “planar logic” that can be also solved with a search tree approach along the lines presented in this paper, yielding also $O^*(4.6056^k)$ algorithms for these problems.
It might well be that one can further improve the running times for FACE COVER and RED-BLUE DOMINATING SET with the use of more information about neighborhoods of low-degree vertices. We fear, however, that the corresponding algorithms would be rather complicated. Observe that Algorithm 2 is already much more complicated than Algorithm 1. Hence, another research goal might be to devise still simpler branching algorithms that nonetheless provide provably better running times. Indeed, we attempted to do this in the preliminary conference version of this paper where, unfortunately, our proof of Corollary 2 turned out to be flawed (we inadvertently disregarded vertices of degree five that are not part of 5-6 pairs).

Let us mention that one of the main exploits of our search tree algorithm is the connection of FACE COVER to HITTING SET. This has been also exploited in the famous exact DOMINATING SET algorithm of F. V. Fomin, F. Grandoni, and D. Kratsch [11], and seems to be a very promising general idea to tackle this type of problems.

Notice that we cannot use our kernelization result to speed up our search tree algorithm at the expense of using exponential space with memoization techniques, as successfully exercised in the case of VERTEX COVER, see [5]. The problem is that the reduction rules used in [6] do change the graph instance in a way that preprocessing small induced subgraphs is not possible: namely, small instances produced after repeated branching and kernelization might not show up as induced subgraphs of the original graph.

As a general message and final comment, it can be seen that structural results (Theorem 1 being a recent example) can quite often be useful to derive algorithmic results in the area of parameterized algorithms. Notice that there is no need to worry about the constructivity of the proof giving the existence of interesting local structures; once this existence is established, one can rather straightforwardly look for the local structures in question in polynomial time. Conversely, it might be worthwhile to see if one can improve on the $O^*(8^k)$ algorithm for PLANAR DOMINATING SET developed in [3] by establishing deeper structural results for the planar black-and-white graphs investigated there. This is of particular importance for cover and domination problems that seem to be the yardstick of current research in parameterized algorithms.

Acknowledgements: The authors are grateful for very constructive feedback from the unknown referees.

References


Faisal N. Abu-Khzam
Division of Computer Science and Mathematics, Lebanese American University,
Chouran, Beirut, Lebanon
faisal.abukhzam@lau.edu.lb

Henning Fernau³
Universität Trier, FB 4—Abteilung Informatik,
54286 Trier, Germany
fernau@uni-trier.de

Michael A. Langston
Department of Computer Science, University of Tennessee,
Knoxville, TN 37996, USA
langston@cs.utk.edu

³Most of the work has been done while this author was affiliated with: University of Hertfordshire, Hatfield, Herts AL10 9AB, UK & Universität Tübingen, D-72076 Tübingen, Germany