# Asymptotically Faster Algorithms for the Parameterized FACE COVER Problem*

Faisal N. Abu-Khzam[1], Henning Fernau[2**] and Michael A. Langston[3]

[1] Division of Computer Science and Mathematics, Lebanese American University, Chouran, Beirut, Lebanon
[2] Wilhelm-Schickard Institut für Informatik, Universität Tübingen, D-72076 Tübingen, Germany & The University of Newcastle, Australia & The University of Hertfordshire, Hatfield, UK
[3] Department of Computer Science, University of Tennessee, Knoxville, TN 37996, USA

**Abstract.** The parameterized complexity of the FACE COVER problem is considered. The input to this problem is a plane graph, $G$, of order $n$. The question asked is whether, for any fixed $k$, there exists a set of $k$ or fewer vertices whose boundaries collectively cover (contain) every vertex in $G$. The fastest previously-published FACE COVER algorithm is achieved with the bounded search tree technique, in which branching requires $O(5^k + n^2)$ time. In this paper, a structure theorem of Aksionov *et al.* is combined with a detailed case analysis to produce a FACE COVER algorithm that runs in $O(4.5414^k + n^2)$ time.

## 1 Introduction and Preliminaries

The bounded search tree technique is probably the most popular method for the design of efficient fixed-parameter algorithms [5]. Strategies based on this technique are also referred to as "branching algorithms." The *branching factor* of a node in a search tree is the number of subtrees rooted at that node. The branching factor of an algorithm is the maximum branching factor taken over all the nodes in the search tree generated by that algorithm.

It is elementary (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. In PLANAR DOMINATING SET, on the other hand, 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). 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 the following structural theorem.

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

- *two vertices whose degrees sum to at most 5,*

* This research has been supported in part by the U.S. National Science Foundation under grants CCR–0075792 and CCR–0311500, by the U.S. Office of Naval Research under grant N00014–01–1–0608, and by the U.S. Department of Energy under contract DE–AC05–00OR22725.

** Communicating author: `fernau@informatik.uni-tuebingen.de`

– *two vertices at a distance of at most two whose degrees sum to at most 7,*
– *a triangular face containing two vertices whose degrees sum to at most 9, or*
– *two triangular faces with vertices u and v in common, where the degrees of u and v sum to at most 11.*

A nontrivial corollary of the above theorem is the following:

**Corollary 1.** *Let $G$ be a plane graph in which every vertex has degree greater than 4. Then $G$ has a pair of adjacent vertices, $u$ and $v$ satisfying:*

1. $deg(u) + deg(v) \in \{10, 11\}$,
2. *$u$ and $v$ are common to two triangular faces $\{u, v, z\}$ and $\{u, w, v\}$, and*
3. *if $w$ and $z$ are the only common neighbors of $u$ and $v$, then one of the following must hold:*
   (a) *at least one element from $N(u) \cup N(v)$ has degree $\leq 6$.*
   (b) *either $w$ or $z$ (or both) belongs to a triangular face that is common to another pair $\{u', v'\}$ that satisfies (1) and (2).*

*Proof.* The existence of a pair $\{u, v\}$ satisfying conditions (1) and (2) is guaranteed by theorem 1. Assume every pair $\{u, v\}$ that satisfies (1) and (2) fails to satisfy (3), then we have the following:

– If two pairs $\{u, v\}$ and $\{u', v'\}$ satisfy conditions (1) and (2), then $\{u, v\} \cap \{u', v'\} = \emptyset$. Moreover $(N(u) \cup N(v)) \cap \{u', v'\} = \emptyset$. This is guaranteed by our assumption (that $(3-a)$ doesn't hold). So it is possible to contract edges $uv$ and $u'v'$ simultaneously.
– Let $E_c = \{uv \mid u \text{ and } v \text{ satisfy (1) and (2) but not (3)}\}$. Any two elements of $E_c$ are independent (in the line graph of $G$). Moreover, contracting all edges of $E_c$ does not reduce the degree of any vertex of $G$ by more than 1. This is guaranteed by our assumption (that $(3-b)$ doesn't hold).

It follows that contracting the edges of $E_c$ gives a planar graph whose minimum degree is bounded below by 6. This is impossible. ∎

To simplify the presentation of our results, we adopt the above notation throughout the sequel. That is, $u$ and $v$ denote a pair of vertices that satisfy conditions (1) and (2) of the previous corollary; $w$ and $z$ denote the common neighbors of $u$ and $v$ such that $f_w = \{u, w, v\}$ and $f_z = \{u, v, z\}$ are triangular faces. We make use of standard notation in graph theory and parameterized algorithmics. 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 superpolynomial 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.

## 2 Active and Marked Faces

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 boundary $v$ 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 of $F$ 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 two more notions: Let

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

Initially, all vertices and all faces are active. So, more formally, we are dealing with an annotated version of FC in the course of the algorithm, i.e.:
Given: A plane (multi-)graph $G = (V, E)$ with face set $F$, a function $\mu_V : V \to \{active, marked\}$, a function $\mu_F : F \to \{active, marked\}$, and a positive integer $k$.
Question: Is there a set $C \subseteq \{f \in F \mid \mu_F(f) = active\}$ with $|C| \leq k$ and $\forall v : \mu_V(v) = active \Rightarrow F_a(v) \cap C \neq \emptyset$?

In addition, marked vertices are shortcut by a sort of triangulation operation. This geometrical surgery allows to finally use the fact that each planar graph possesses a vertex of degree at most five to branch at. Let us explain this idea in more details. In order to do this, let us first describe some HS reduction rules in accordance with the notation introduced above (see [6–8].

**Rule 1** *If $F_a(u) \subseteq F_a(v)$ for some active vertices $u, v$. Then mark $v$.*

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

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

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 $v$. (This way, our graph may get multiple edges or loops.)*

**Rule 5** *If $u$ is a marked vertex with two active neighbors $v, w$ such that $u, v, w$ are all incident to an active face $f$, then partition $f$ into two faces by introducing a new edge between $v$ and $w$, and that edge has to be drawn inside of $f$. (This way, our graph may get multiple edges.)*

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

**Rule 6** *If $deg(v) = 1$ and if $v$ is marked, then delete $v$.*

**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.*

**Rule 8** *If $f$ is a marked face with only one vertex or with two vertices on its boundary, then choose one edge $e$ on the boundary of $f$ and delete $e$. This will destroy $f$.*

**Rule 9** *If $e$ is an edge with two incident marked faces $f, f'$, then delete $e$, i.e., merge $f$ and $f'$; the resulting new face will be also marked.*

Algorithm 1 applies the above reduction rules, and is the branching algorithm described in [1]. We include it here so that we can describe how we modify and improve it. The following analysis 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 graph $G = (V, E)$ with face set $F$ and marking functions $\mu_V$ and $\mu_F$, a positive integer $k$
**Ensure:** YES 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 YES
**else**
  Let $v$ be a vertex of lowest face degree in $G$.
  {One incident face of $v$ must be used to cover $v$.}
  Choose $f \in F_a$ such that $f$ is incident to $v$.
  Mark $f$ and all vertices that are on the boundary of $f$.
  Call the resulting marking functions $\mu'_V$ and $\mu'_F$.
  **if** FC-ST$(G, F, \mu'_V, \mu'_F, k-1)$ **then**
    return YES
  **else**
    Mark $f$, i.e., return FC-ST$(G, F, \mu_V, \mu'_F, k)$
  **end if**
**end if**

**Lemma 1.** *If $G = (V, E)$ is an annotated plane graph 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, then no marked vertex will exist in $G$.*

*Proof.* Assume there is a marked vertex $v$ in $G$. $deg(v) \neq 1$, since otherwise Rule 6 would have been applicable. Hence, $v$ has two neighbors $u$ and $w$. If one of them would be marked, Rule 4 would have applied. Therefore, all neighbors of $v$ must be active. To avoid application of Rule 7, we can assume that $deg_a(v) > 0$. Then, Rule 5 applies $deg_a(v)$ many times. The new triangular-shaped 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 9) applies and deletes $v$.

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

**Lemma 2.** *If $G = (V, E)$ is an annotated plane graph with face set $F$ (seen as instance of the annotated plane graph problem with parameter $k$) that is reduced according to the* FACE COVER *reduction rules listed above, then the only degenerated faces that might exist are active faces $f$ with two incident vertices. Moreover, the two faces that are neighbored with such a degenerated face $f$ via common edges are both marked.*

*Proof.* Let $f$ be a degenerated face. If $f$ has only one vertex $v$ on its boundary, the following will happen:

- If $f$ and $v$ are active and $deg_a(v) = 1$, then Rule 2 applies and puts $f$ into the face cover. Moreover, $f$ and $v$ will become marked. Hence, Rule 6 applies and deletes $v$, so that also $f$ will be replaced by a probably larger face.
- If $f$ and $v$ are active and $deg_a(v) > 1$, then Rule 3 applies and renders $f$ marked.
- If $f$ is active but $v$ is marked, then $V_a(f) = \emptyset$, so that Rule 3 vacuously applies and renders $f$ marked.
- If $f$ is marked, then Rule 8 applies and removes $f$.

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

Consider now a degenerated face $f$ with two vertices $u$ and $v$ on its boundary. 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. 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.

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 $G$, 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 8) 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 $v$, i.e., there might be up to ten faces neighboring $v$ in $G$.[4] Lemma 2 shows that at most five of these faces can be active. ∎

[4] This was the basic concern when stating the $O^*(10^k)$ (actually, even worse) algorithm for the FACE COVER problem in [5].

## 3 An Improved FACE COVER Algorithm

We now use Theorem 1 and its corollary to improve on the running time of the algorithm. Of course, as long as in the auxiliary graph $G'$ constructed according to the previous proof we find a vertex of degree four, we find a $4^k$ branching behaviour and are happy with it. 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 $deg(u) = 5$ and $deg(v) = 6$. In the sequel, $T(k)$ denotes the number of leaves in the search tree corresponding to our algorithm. Moreover, $T(k)$ corresponds to the case where the algorithm is trying to find a face cover of size $\leq k$. We now discuss some possibilities for vertex $u$.

1. If some of the edges incident to $u$ in $G'$ represent degenerated faces in $G$ and some correspond to (simple) edges in $G$, then Lemma 2 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 degenerated faces in $G$, then this is true in particular for the edge $uv$ in $G'$. In order to achieve $deg_a(v) = deg(v)(= 6)$, all edges incident to $v$ must also represent degenerated faces in $G$ (otherwise, the branching would be only better, see Lemma 2).[5]
   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 $G'$). 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 degenerated face $\{u, w\}$ into the cover. Then, both faces $\{v, w\}$ and $\{u, v\}$ will get marked by the dominated face rule 3, so that 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: $T(k) \leq 4.7016^k$. We show in the following section that a bound of $4.5414^k$ is achievable.
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, 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 $w$. 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

[5] 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-degenerated. Branching first on the degenerated 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

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

faces containing $u$ or $v$ are active. 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 $f_1$ and $f_2$.

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 $T(k) \leq 4.6056^k$.

The above branching scenarios are described in Algorithm 2. The time complexity of this algorithm is $O^*(4.7016^k)$. We shall see that Corollary 1 can be used to reduce this bound below $4.5414^k$.

**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:** YES 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 = ∅ then
  return YES
else
  Let u be a vertex of lowest face degree in G.
  {One incident face of u must be used to cover u.}
  if deg_a(u) ≤ 4 then
    Choose f ∈ F_a such that f is incident to u.
    Mark f and all vertices that are on the boundary of f.
    Call the resulting marking functions μ'_V and μ'_F.
    if FC-ST-advanced(G, F, μ'_V, μ'_F, k − 1) then
      return YES
    else
      Mark f, i.e., return FC-ST-advanced(G, F, μ_V, μ'_F, k)
    end if
  else
    {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 degenerated then
      execute FC-ST-case-1
    else if no active faces incident with v are degenerated then
      execute FC-ST-case-2
    else
      {all active faces incident with u are degenerated; only one active face incident with v is degenerated}
      execute FC-ST-case-3
    end if
  end if
end if
```

**Algorithm 3** The code of FC-ST-case-1

Let $G' = G \setminus \{u, v\}$ and mark the face to which (formally) $u, v$ belonged; modify $F, \mu_V, \mu_F$ accordingly, yielding $F'$, $\mu'_V$ and $\mu'_F$.
**if** FC-ST-advanced$(G', F', \mu'_V, \mu'_F, k-1)$ **then**
  return YES
**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.
    **if** FC-ST-advanced$(G', F, \mu'_V, \mu'_F, k-2)$ **then**
      return YES
    **end if**
  **end for**
  return NO
**end if**

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

Let $G' = G \setminus \{u, v\}$ and mark the face 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.
  **if** FC-ST-advanced$(G', F', \mu'_V, \mu'_F, k-1)$ **then**
    return YES
  **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.
  **if** FC-ST-advanced$(G', F, \mu'_V, \mu'_F, k-2)$ **then**
    return YES
  **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 $\mu'_F$.
  **if** FC-ST-advanced$(G', F', \mu'_V, \mu'_F, k - |\{f, g\}|)$ **then**
    return YES
  **end if**
**end for**
return NO

### 3.1 The Degenerated Faces Case

Let us look a bit closer at the case that all edges incident with $u$ and $v$ represent degenerated faces. 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 the degenerated face $\{u, w\}$ is put into the cover, then both faces $\{v, w\}$ and $\{u, v\}$ will get marked by the dominated face rule 3, so that for branching at $v$, four further cases need to be considered. Moreover, notice that our reduction rules will produce the following situation thereafter:[6]
  - $u$ and $v$ will be deleted.
  - All faces formerly neighboring $u$ or $v$ will be merged into one large marked face $f$.
  - On the boundary of $f$, $w$ will be marked together with one vertex out of $\{v_1, v_2, v_3, z\}$.
- If the the degenerated face $\{u, z\}$ is put into the cover, then a similar analysis applies. This means that (according to the previous reasoning) we are left with the following situations:
  - $u$ and $v$ will be deleted.
  - All faces formerly neighboring $u$ or $v$ will be merged into one large marked face $f$.
  - On the boundary of $f$, $z$ will be marked together with one vertex out of $\{v_1, v_2, v_3, w\}$.

  Now, observe that the situation that marks $z$ together with $w$ (on the boundary of $f$) has already been found before. Hence, we only need to consider three $T(k-2)$-branches here.
- The other cases are treated as before, giving three more $T(k-1)$-branches.

Altogether, we get as a recurrence for the search tree size:

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

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

### 3.2 Dealing with Non-degenerated Faces

Consider the case where all faces containing $u$ and $v$ are non-degenerated. We distinguish the following sub-cases:

1. 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 $T(k) \leq 3T(k-1) + 6T(k-2) \leq 4.3723^k$.
2. The degree of $w$ (or $z$) is $\leq 6$. One of the 6 faces of $w$ is used in the cover. There are (potentially) 3 faces that are not common with $u$ or $v$. Let $f_1$ be the face common to $u$ and $w$ but not $v$, and let $f_2$ be the face common to $v$ and $w$ but not $u$. We branch according to the following cases:
   (a) $f_1$ is selected. Then to cover $v$, we are left with 4 cases (since the two faces $f_w$ and $f_z$ become dominated). This sub-case produces $4T(k-2)$-branches.
   (b) $f_2$ is selected. Then to cover $u$, we are left with 2 cases (2 and not 3 since $f_1$ was selected in the previous case). This sub-case leads to $2T(k-2)$-branches.
   (c) $f_w$ is selected and covers $u$,$v$, and $w$. This gives a $T(k-1)$-branch.

[6] In fact, the marked vertices will again disappear by further application of reduction rules, but this will be neglected in the following argument.

(d) None of the 3 sub-cases above is applied. So one of the 3 faces containing $w$ is selected. After each selection, and since we assumed none of $f_1, f_2$ and $f_w$ is selected, we are left with the following: $u$ has 3 faces that can be selected, one of which is $f_z$, and $v$ has 4 faces that can be selected, one of which is $f_z$. The corresponding equation is: $3(T(k-2)+6T(k-3))$

Therefore, in case 2 the run time is given by $T(k) \leq T(k-1)+9T(k-2)+18T(k-3) \leq 4.1817^k$, which is smaller than $4.5414^k$.

3. The degree of a neighbor $v_1$ of $v$ is $\leq 6$. Let $f_1$ and $f_2$ be the faces common to $v$ and $v_1$. We may assume $f_1$ and $f_2$ do not contain $u$ (We already dealt with the case where $u$ and $v$ have a common face other than $f_w$ and $f_z$.). We branch according to the following cases:
   (a) One of the faces $f_1$ and $f_2$ is selected. Each selection leads to covering $v$ and marking the faces $\{u, v, z\}$ and $\{u, w, v\}$ being dominated. Thus we are left with 3 faces to cover $u$. The corresponding run time is $6T(k-2)$ (3 for each case).
   (b) Neither $f_1$ nor $f_2$ is selected. We have 4 faces to cover $v_1$, and in each case we have: 2 faces to cover $v$ and $u$ simultaneously, and 2 faces to cover $v$ but not $u$ and 3 faces to cover $u$ but not $v$. This leads to a run time equal $8T(k-2)+24T(k-3)$.

   In case 3, the run time is $T(k) \leq 14T(k-2)+24T(k-3) \leq 4.4095^k$, which is smaller than $4.5414^k$.
4. The degree of a neighbor $u_1$ of $u$ is $\leq 6$. This case is similar to the previous case. Basically the same branching strategy (interchanging the roles of $u$ and $v$) would lead to the estimate

$$T(k) \leq 16T(k-2)+16T(k-3) \leq 4.4287^k.$$

5. According to Corollary 1, we are left with the case where $w$ (or $z$) is a common neighbor of two pairs $\{u, v\}$ and $\{u', v'\}$ that satisfy the conditions stated in the corollary. In analogy to $u, v$, let $w'$ and $z'$ denote the vertices that are incident with the two triangular faces neighboring the edge $u'v'$. We can assume $w = w'$ in accordance with our corollary. We branch within two cases:
   (a) $f_w$ is in the cover: therefore face $\{u', w, v'\}$ is dominated: so we cover $u'$ and $v'$ by the branching rule: either face $f_{z'} = \{u', v', z'\}$ is in the cover or a pair of faces containing $u'$ and $v'$ (other than $\{u', w', v'\}$ and $\{u', v', z'\}$) is in cover. This leads to $T(k-2)+12T(k-3)$ branches.
   (b) $f_w$ is not in the cover. So we branch with two cases: either face $f_z$ $(= \{u, v, z\})$ is in cover or a pair of faces containing $u$ and $v$ (other than $f_w$ and $f_z$ is in the cover. This leads to $T(k-1)+12T(k-2)$ branches.

   Therefore, in case 5, we get: $T(k) \leq T(k-1)+13T(k-2)+12T(k-3) \leq 4.4903^k$. Again, this is smaller than $4.5414^k$.

The analysis described in the last two subsections leads to the following:

**Theorem 3.** FACE COVER *has an algorithm that runs in time* $O^*(4.5414^k)$.

## 4 Remarks

We conclude with the following final 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.

The RED-BLUE DOMINATING SET (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)

– mark all faces of the graph
– attach to each red vertex an active loop
– inflate a loop attached to the red vertex $v$ along the edges that connect $v$ with its blue neighbors until the loop (seen as a region in the plane) touches all these neighbors; then, we can finally remove $v$. The active faces of the resulting graph correspond to red vertices.

The above reduction gives a branching algorithm for the planar RED-BLUE DOMINATING SET, whose run time is $O^*(4.5414^k)$. This is an obvious improvement over the previous algorithm described in [5].

Finally, observe that it might well be that we can further improve on the running times for FACE COVER and RED-BLUE DOMINATING SET by further strengthened versions of Cor. 1. However, the corresponding algorithms would be rather complicated, as we fear. Notice that already Alg. 2 was much more complicated than Alg. 1 that we presented in the first place, and that the final algorithm that lead to the claimed running time of $O^*(4.5414^k)$ has a quite complicated branching structure. Hence, another aim of research could be to come up with still simple branching algorithms which nonetheless enjoy provable nice running times.

## References

1. F. N. Abu-Khzam and M. A. Langston. A direct algorithm for the parameterized face cover problem. In *Proceedings of the First International Workshop on Parameterized and Exact Computation (IWPEC 2004)*, volume 3162 of *Lecture Notes in Computer Science*, pages 213–222. Springer-Verlag, 2004.
2. V. A. Aksionov, O. V. Borodin, L. S. Mel'nikov, G. Sabidussi, M. Stiebitz, and B. Toft. Deeply asymmetric planar graphs. Technical Report PP–2000–14, University of Southern Denmark, IMADA, Odense, Denmark, September 2000.
3. J. Alber, H. Fan, M. R. Fellows, H. Fernau, R. Niedermeier, F. Rosamond, and U. Stege. Refined search tree techniques for the PLANAR DOMINATING SET problem. In J. Sgall, A. Pultr, and P. Kolman, editors, *Mathematical Foundations of Computer Science (MFCS 2001)*, volume 2136 of *Lecture Notes in Computer Science*, pages 111–122. Springer, 2001. Long version to appear in Journal of Computer and System Sciences.
4. D. Bienstock and C. L. Monma. On the complexity of covering vertices by faces in a planar graph. *SIAM J. Sci. Comput.*, 17:53–76, 1988.
5. R. G. Downey and M. R. Fellows. *Parameterized Complexity.* Springer-Verlag, 1999.
6. H. Fernau. A top-down approach to search-trees: Improved algorithmics for 3-Hitting Set. Technical Report TR04-073, Electronic Colloquium on Computational Complexity, 2004.
7. R. Niedermeier and P. Rossmanith. An efficient fixed-parameter algorithm for 3-Hitting Set. *Journal of Discrete Algorithms*, 1:89–102, 2003.
8. K. Weihe. Covering trains by stations or the power of data reduction. In R. Battiti and A. A. Bertossi, editors, *Algorithms and Experiments ALEX 98*, pages 1–8. `http://rtm.science.unitn.it/alex98/proceedings.html`, 1998.