Compression-Based Fixed-Parameter Algorithms for Feedback Vertex Set and Edge Bipartization [1]

Jiong Guo [a,2] Jens Gramm [b,3] Falk Hüffner [a,2] Rolf Niedermeier [a]
Sebastian Wernicke [a,4]

[a] *Institut für Informatik, Friedrich-Schiller-Universität Jena, Ernst-Abbe-Platz 2, D-07743 Jena, Germany*

[b] *Wilhelm-Schickard-Institut für Informatik, Universität Tübingen, Sand 13, D-72076 Tübingen, Germany*

Abstract

We show that the NP-complete Feedback Vertex Set problem, which asks for the smallest set of vertices to remove from a graph to destroy all cycles, is deterministically solvable in $O(c^k \cdot m)$ time. Here, m denotes the number of graph edges, k denotes the size of the feedback vertex set searched for, and c is a constant. We extend this to an algorithm enumerating *all* solutions in $O(d^k \cdot m)$ time for a (larger) constant d. As a further result, we present a fixed-parameter algorithm with runtime $O(2^k \cdot m^2)$ for the NP-complete Edge Bipartization problem, which asks for at most k edges to remove from a graph to make it bipartite.

Key words: fixed-parameter tractability, iterative compression, graph algorithm, graph modification problem, feedback set problem

---

Email addresses: guo@minet.uni-jena.de (Jiong Guo), gramm@informatik.uni-tuebingen.de (Jens Gramm), hueffner@minet.uni-jena.de (Falk Hüffner), niedermr@minet.uni-jena.de (Rolf Niedermeier), wernicke@minet.uni-jena.de (Sebastian Wernicke).

[1] An extended abstract of this work appears under the title "Improved Fixed-Parameter Algorithms for Two Feedback Set Problems" in the proceedings of the 9th Workshop on Algorithms and Data Structures (WADS 2005), volume 3608 of *LNCS*, pages 158–168. Springer, Aug. 2005.

[2] Supported by the Deutsche Forschungsgemeinschaft, Emmy Noether research group PIAF (fixed-parameter algorithms), NI 369/4.

[3] Supported by the Deutsche Forschungsgemeinschaft, project OPAL (optimal solutions for hard problems in computational biology), NI 369/2.

[4] Supported by the Deutsche Telekom Stiftung and the Studienstiftung des

1 Introduction

In feedback set problems the task is, given a graph G and a collection C of cycles in G, to find a minimum size set of vertices or edges that meets all cycles in C. We refer to Festa, Pardalos, and Resende [16] for a 1999 survey. In this work we restrict our attention to undirected and unweighted graphs, giving significantly improved exact algorithms for two NP-complete feedback set problems.

- Feedback Vertex Set (FVS): Here, the task is to find a minimum cardinality set of *vertices* that meets all cycles in the graph.
- Edge Bipartization: Here, the task is to find a minimum cardinality set of *edges* that meets all *odd-length* cycles in the graph. [5]

Concerning the FVS problem, it is known that an optimal solution can be approximated to a factor of 2 in polynomial time [3]. The best known *linear*-time approximation algorithm has approximation factor 4 [4]. FVS is MaxSNP-hard [23] (hence, there is no hope for polynomial-time approximation schemes). A question of similar importance as approximability is to ask how fast one can find an *optimal* feedback vertex set. There is a simple and elegant randomized algorithm due to Becker, Bar-Yehuda, and Geiger [5] which solves the FVS problem in $O(c \cdot 4^k \cdot kn)$ time by finding a feedback vertex set of size k with probability at least $1-(1-4^{-k})^{c4^k}$ for an arbitrary constant c. Note that this means that by choosing an appropriate value c, one can achieve any constant error probability independent of k. As to deterministic algorithms, Bodlaender [6] and Downey and Fellows [11] were the first to show that the problem is fixed-parameter tractable, i.e., that the combinatorial explosion when solving it can be confined to the parameter k. An exact algorithm with runtime $O((2k+1)^k \cdot n^2)$ was described by Downey and Fellows [12]. In 2002, Raman, Saurabh, and Subramanian [29] made a significant step forward by proving the upper bound $O(\max\{12^k, (4\log k)^k\} \cdot n^\omega)$ (where n^ω denotes the time to multiply two $n \times n$ integer matrices). This bound was slightly improved to $O((2\log k + 2\log\log k + 18)^k \cdot n^2)$ by Kanj, Pelsmajer, and Schaefer [21] using results from extremal graph theory. Finally, Raman, Saurabh, and Subramanian [30] published an algorithm running in $O((12\log k/\log\log k + 6)^k \cdot n^\omega)$ time.

The central question left open was whether there is an $O(c^k \cdot n^{O(1)})$ time algorithm for FVS for some constant c. We settle this by giving an $O(c^k \cdot mn)$ time

deutschen Volkes.

[5] That is, the deletion of those edges would make the graph bipartite. Note that the task of finding a minimum cardinality set of edges that meets *all* cycles is the well-known Minimum Spanning Tree problem, which can be solved in polynomial time [7].

algorithm. Independently, this result was also shown for $c \approx 10.6$ by Dehne et al. [9]. Surprisingly, although both studies were performed completely independent of each other, the developed algorithms turn out to be quite similar. The advantage of the result by Dehne et al. is a better worst-case upper bound on the constant c, whereas our advantage seems to be a more compact and accessible presentation of the algorithm. Since it seems hard to bring the constant c close to the constant 4 achieved by the randomized algorithm of Becker et al., the described deterministic algorithms for FVS are of more theoretical interest.

Other than the result of Dehne et al. [9] we also show that FVS can be solved deterministically in *linear* time for constant k, a property which also holds for the randomized algorithm. Hence, with a corresponding $O(c^k \cdot m)$ algorithm, we can conclude that FVS is *linear-time* fixed-parameter tractable. Very recently, Fiorini et al. [17] showed, by significant technical expenditure, the analogous result concerning the Graph Bipartization problem (which is basically the same problem as Edge Bipartization, only deleting vertices instead of edges) restricted to planar graphs.

The approaches mentioned above address the fixed-parameter tractability of finding *one* solution of size at most k. Parameterized enumeration, i.e., the question whether it is fixed-parameter tractable to find *all* minimal solutions of size at most k, has lately attracted some interest [8,15]. Concerning feedback set problems, Schwikowski and Speckenmeyer studied "classical algorithms" for enumerating minimal solutions [32]. Extending our above algorithm, we show in this work how to enumerate *all* minimal feedback vertex sets of size at most k in $O(c^k \cdot m)$ time.

We also note that, without a change in runtime, both presented FVS algorithms (for finding one minimal solution and enumerating all minimal solutions) can solve a more general problem introduced in [4] where some graph vertices are marked as "blackout" and may not be part of the feedback vertex set.

We now turn our attention to the Edge Bipartization problem, also known as (unweighted) Minimum Uncut. This problem is known to be MaxSNP-hard [26] and can be approximated to a factor of $O(\sqrt{\log n})$ in polynomial time [1]. Another approximation algorithm finds in polynomial time a solution of size $O(k \log k)$, where k is the size of an optimal solution [2]. Assuming Khot's Unique Games Conjecture, it is NP-hard to approximate Edge Bipartization within any constant factor [22]. The problem has applications in genome sequence assembly [27] and VLSI chip design [20].

In a recent breakthrough paper, Reed, Smith, and Vetta [31] proved that the Graph Bipartization problem is solvable in $O(4^k \cdot kmn)$ time, where k

denotes the number of vertices to be deleted for making the graph bipartite. (Actually, it is straightforward to observe that the exponential factor 4^k can be lowered to 3^k by a more careful analysis of the algorithm [19].) Since there is a "parameter-preserving" reduction from EDGE BIPARTIZATION to GRAPH BIPARTIZATION [33], one can use the algorithm by Reed et al. to directly obtain a runtime of $O(3^k \cdot k^3m^2n)$ for EDGE BIPARTIZATION, k denoting the size of the set of edges to be deleted. In this work our main concern is to shrink the combinatorial explosion and the polynomial complexity related to the fixed-parameter tractability of EDGE BIPARTIZATION. We achieve an algorithm running in $O(2^k \cdot m^2)$ time. This shows that we can shrink the combinatorial explosion from 3^k to 2^k and additionally save a cubic-time factor k^3 as well as a linear-time factor n.

From a different perspective, our above results are examples for the versatility of a new algorithmic technique called "iterative compression" [31]. It is discussed in more detail in the following sections, and our paper can also be seen as a gentle introduction to this new tool for fixed-parameter algorithm design. A more extensive introduction to iterative compression can be found in [18].

2 Preliminaries

This work considers undirected graphs $G = (V, E)$ with $n := |V|$ and $m := |E|$. Given a set $E' \subseteq E$ of edges, $V(E')$ denotes the set $\bigcup_{\{u,v\} \in E'} \{u, v\}$ of their endpoints. We use $G[X]$ to denote the subgraph of G induced by the vertices in $X \subseteq V$. For a set of edges $E' \subseteq E$, we write $G \setminus E'$ for the graph $(V, E \setminus E')$. For $u \in V$, we use $N(u)$ to denote the neighbor set $\{v \in V : \{u, v\} \in E\}$. The *length* of a path in a graph is the number of its edges. With a *side* of a bipartite graph G, we mean one of the two classes of an arbitrary but fixed two-coloring of G. An *edge cut* between two disjoint vertex sets in a graph is a set of edges whose removal disconnects these two sets in the graph. For a minimization problem, a feasible solution is called *minimal* if it does not contain another feasible solution as a proper subset and *minimum* if there is no other feasible solution with better measure.

The two problems we study are formally defined as follows:

FEEDBACK VERTEX SET (FVS)
Input: An undirected graph $G = (V, E)$ and a nonnegative integer k.
Task: Find a subset $V' \subseteq V$ of vertices with $|V'| \leq k$ such that each cycle in G contains at least one vertex from V'. (The removal of all vertices in V' from G results in a forest.)

EDGE BIPARTIZATION

Input: An undirected graph $G = (V, E)$ and a nonnegative integer k.
Task: Find a subset $E' \subseteq E$ of edges with $|E'| \leq k$ such that each odd-length cycle in G contains at least one edge from E'. (The removal of all edges in E' from G results in a bipartite graph.)

We investigate FVS and Edge Bipartization in the context of parameterized complexity [12,25] (see [14,24] for surveys). A parameterized problem is *fixed-parameter tractable* if it can be solved in $f(k) \cdot n^{O(1)}$ time where f is a computable function solely depending on the parameter k and not on the input size n.

To the best of our knowledge, Reed et al. [31] were the first to make the following simple but fruitful observation: To show that a minimization problem is fixed-parameter tractable with respect to the size of the solution k, it often suffices to give a fixed-parameter algorithm that, given a size-$(k+1)$ solution, either proves that there is no size-k solution or constructs one. Starting with a trivial instance and inductively applying this compression routine a linear number of rounds to larger instances, one obtains the fixed-parameter tractability of the problem. This method is called *iterative compression*. The main challenge of applying it lies in showing that there is a "fixed-parameter compression routine." It is this hard part where Reed et al. achieved a breakthrough concerning Graph Bipartization. The compression routine, however, is highly problem-specific and no universal standard techniques are known.

3 Algorithm for Feedback Vertex Set

In this section we show that Feedback Vertex Set can be solved in $O(c^k \cdot m)$ time for a constant c by presenting an algorithm based on iterative compression. The central part is the compression routine which, given a feedback vertex set (fvs) X, produces a smaller one if it exists. To make this easier, we restrict our search to solutions that are *disjoint* from the known solution. This restriction can be achieved without loss of generality at the cost of a factor of $2^{|X|}$ in the runtime by using a brute-force enumeration of all partitions of X into a part Y to keep and a part S to exchange in the smaller solution; the vertices in Y can then be immediately deleted, and we arrive at the following task.

Task 1 *Given a graph $G = (V, E)$ and an fvs S for G, find a minimum fvs S' for G with $S' \cap S = \emptyset$.*

To solve Task 1, we make use of simple data reduction rules mostly known from the literature [5,29], and then show that the remaining instance is small

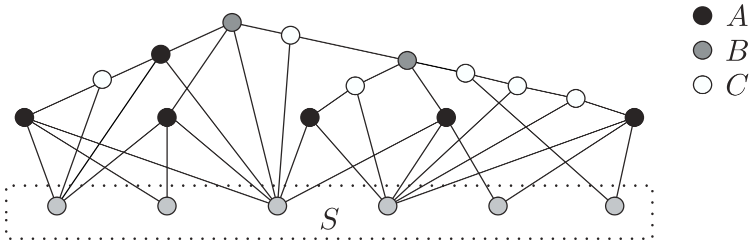

Fig. 1. Partition of the vertices in V' into three disjoint subsets A, B, and C.

enough to be solved in $O(c^k \cdot m)$ time by brute force.

Reduction Rule 1 *In an instance of Task 1, remove degree-1 vertices.*

Reduction Rule 2 *In an instance of Task 1, if there is a degree-2 vertex $v \in S$ with two neighbors v_1 and v_2, where $v_1 \notin S$ or $v_2 \notin S$, then remove v and connect v_1 and v_2. If this creates two parallel edges between v_1 and v_2, then remove the vertex of v_1 and v_2 that is not in S and add it to any solution for the reduced instance.*

Lemma 1 *Reduction Rules 1 and 2 are correct and can be executed in $O(m)$ time.*

PROOF. Reduction Rule 1 is correct because degree-1 vertices are not contained in any cycle. Reduction Rule 2 is correct because deleting any of v_1 and v_2 destroys all cycles that contain v. In the case that a parallel edge is created, exactly one of v_1 and v_2 is in S because S is an fvs of G and $G[S]$ contains no cycle unless there is no solution. Thus, we have to delete the other endpoint since we are not allowed to take vertices from S.

The runtime bound is straightforward to obtain and we omit its proof here. □

Lemma 2 *An instance of Task 1 that is reduced with respect to Reduction Rules 1 and 2 and has a solution S' with $|S'| < |S|$ contains at most $15|S|$ vertices.*

PROOF. Let $G = (V, E)$ be a reduced instance and $V' := V \setminus S$. We partition V' into three subsets, each of which will have a provable size bound linearly dependant on $|S|$ (the partition is illustrated in Figure 1):

$$A := \{v \in V' : |N(v) \cap S| \geq 2\},$$
$$B := \{v \in V' \setminus A : |N(v) \cap V'| \geq 3\},$$
$$C := V' \setminus (A \cup B).$$

To upper-bound the number of vertices in A, consider the bipartite subgraph $G_A := (A \dot\cup S, E_A)$ of G with $E_A := (A \times S) \cap E$. Observe that if there are more than $|S| - 1$ vertices in A, then there is a cycle in G_A: If G_A is acyclic, then G_A is a forest, and, thus, $|E_A| \leq |S| + |A| - 1$. Moreover, since each vertex in A has at least two incident edges in G_A, $|E_A| \geq 2|A|$, which implies that $|A| \leq |S| - 1$ if G_A is acyclic. It follows directly that if $|A| \geq 2|S|$, it is impossible to delete at most $|S|$ vertices from A such that $G'[A \cup S]$ is acyclic.

To upper-bound the number of vertices in B, observe that $G[V']$ is a forest. Furthermore, all leaves of the trees in $G[V']$ are from A since G is reduced with respect to Reduction Rules 1 and 2. By the definition of B, each vertex in B is an internal vertex of degree at least three in the forest induced by V'. Thus, there cannot be more vertices in B than in A, and therefore $|B| < 2|S|$.

Finally, consider the vertices in C. By the definitions of A and B, and since G is reduced, each vertex in C has degree two in $G[V']$ and exactly one neighbor in S. Hence, graph $G[C]$ is a forest consisting of paths and isolated vertices. We now separately upper-bound the number of isolated vertices and those participating in paths.

Each of the isolated vertices in $G[C]$ connects two vertices from $A \cup B$ in $G[V']$ and no pair of vertices from $A \cup B$ is connected by more than one such vertex. Since $G[V']$ is acyclic, this means that the number of isolated vertices in $G[C]$ cannot exceed $|A \cup B| - 1 < 4|S|$.

The total number of vertices participating in paths in $G[C]$ can be upper-bounded as follows: Consider the subgraph $G[C \cup S]$. Each edge in $G[C]$ creates a path between two vertices in S, that is, if $|E(G[C])| \geq |S|$, then there exists a cycle in $G[C \cup S]$. Removing a vertex from $G[C]$ destroys at most two edges from $E(G[C])$. Hence, the total number of edges in $E(G[C])$ must not exceed $6|S|$; otherwise, at least $2|S|$ edges remain after at most $|S|$ vertices from C have been deleted, leaving at least one cycle in the remaining graph.

Altogether, $|V'| = |A| + |B| + |C| < 2|S| + 2|S| + (4 + 6)|S| = 14|S|$. □

Having established the linear size bound for the vertex set of a reduced instance in Lemma 2, we can now provide the compression routine for our Feedback Vertex Set algorithm.

Lemma 3 *Given a graph G and a size-$(k+1)$ fvs X for G, we can decide in $O(c^k \cdot m)$ time for some constant c whether there exists a size-k fvs X' for G and if so provide one.*

PROOF. Consider the smaller fvs X' as a modification of the larger fvs X. The smaller fvs retains some vertices $Y \subseteq X$ while the other vertices $S := X \setminus Y$ are replaced by $|S| - 1$ new vertices from $V \setminus X$. The idea is to try by brute force all $2^{|X|}$ partitions of X into such sets Y and S. For each partition, the vertices from Y are immediately deleted and it remains to solve Task 1 for $G' := G[V \setminus Y]$ and S. For this, we first check that S does not induce a cycle; otherwise, no S' with $S' \cap S = \emptyset$ can be an fvs for G'. We then apply Reduction Rules 1 and 2 exhaustively to G', which can be done in $O(m)$ time by Lemma 1. We have shown in Lemma 2 that the size of the set of candidates V' for a solution S' with $|S'| < |S|$ is upper-bounded by $14|S|$. Since $|S| \leq k + 1$, $|V'|$ thus only depends on the problem parameter k and not on the input size. We again use brute force and consider each of the at most $\binom{14|S|}{|S|-1}$ possible choices of vertices from V' that can be added to Y to form X'. The test whether a choice of vertices from V' together with Y forms an fvs can easily be done in $O(m)$ time. We can now bound the overall runtime T, where the index i corresponds to a partition of X into Y and S with $|Y| = i$ and $|S| = |X| - i$:

$$
\begin{aligned}
T &= O\left(\sum_{i=0}^{k} \binom{|X|}{i} \cdot \left(O(m) + \binom{14(|X|-i)}{|X|-i-1} \cdot O(m)\right)\right) \\
&= O\left(2^k \cdot m + \sum_{i=0}^{k+1} \binom{k+1}{i} \cdot \binom{14(k+1-i)}{k+1-i} \cdot m\right)
\end{aligned}
$$

and with Stirling's inequality to evaluate the second binomial coefficient,

$$
= O\left(2^k \cdot m + \sum_{i=0}^{k+1} \binom{k+1}{i} (36.7)^{k+1-i} \cdot m\right) = O((1 + 36.7)^k \cdot m),
$$

which gives the lemma's claim with $c \approx 37.7$.[6] □

Theorem 4 Feedback Vertex Set *can be solved in* $O(c^k \cdot mn)$ *time for a constant* c.

PROOF. Given as input a graph G with vertex set $\{v_1, \ldots, v_n\}$, we can solve Feedback Vertex Set for G by iteratively considering the subgraphs $G_i := G[\{v_1, \ldots, v_i\}]$. For $i = 1$, the optimal fvs is empty. For $i > 1$, assume that an optimal fvs X_i for G_i is known. Obviously, $X_i \cup \{v_{i+1}\}$ is an fvs for G_{i+1}. Using Lemma 3, we can in $O(c^k \cdot m)$ time either determine that $X_i \cup \{v_{i+1}\}$ is an optimal fvs for G_{i+1}, or, if not, compute an optimal fvs for G_{i+1}. For $i = n$, we thus have computed an optimal fvs for G in $O(c^k \cdot mn)$ time. □

[6] The value of c can be significantly improved by a more careful analysis in Lemma 2. Indeed, Dehne et al. [9] achieve $c \approx 10.6$.

Theorem 4 shows that FVS is fixed-parameter tractable with the combinatorial explosion bounded from above by c^k for some constant c. Next, we show that FVS is also *linear-time* fixed-parameter tractable (with the combinatorial explosion still bounded by c^k, however for a larger constant c). An analogous result for Graph Bipartization restricted to planar graphs was shown by Fiorini et al. [17], accepting a much worse combinatorial explosion compared to [31].

Theorem 5 Feedback Vertex Set *can be solved in* $O(c^k \cdot m)$ *time for a constant* c.

PROOF. We first calculate in $O(m)$ time a factor-4 approximation as described by Bar-Yehuda et al. [4]. This gives us the precondition for Lemma 3 with $|X| = 4k$ instead of $|X| = k + 1$. Now, we can employ the same techniques as in the proof of Lemma 3 to obtain the desired runtime: we examine 2^{4k} partitions $S \dot\cup Y$ of X, and—by applying the reduction rules and using Lemma 2—for each partition there is some constant c' such that the number of candidate vertices is bounded from above by $c' \cdot |S|$. In summary, there is some constant c such that the runtime of the compression routine is bounded from above by $O(c^k \cdot m)$. Since one of the 2^{4k} partitions must lead to the optimal solution of size k, we need only one call of the compression routine to obtain an optimal solution, which proves the claimed runtime bound. □

Note that any improvement of the approximation factor of a linear-time approximation algorithm for Feedback Vertex Set below 4 will immediately improve the constant c in the runtime of the exact algorithm described in Theorem 5.

4 Enumerating Minimal Feedback Vertex Sets

In this section we show that in $O(c^k \cdot m)$ time for a constant c we can not only find *one* feedback vertex set of size k but even enumerate *all* minimal feedback vertex sets of size at most k. Since, in general, there may be more than $O(c^k \cdot m)$ many such vertex sets, we list *compact representations* of all minimal feedback vertex sets. A compact representation for a set of minimal feedback vertex sets of a graph $G = (V, E)$ is a set $\mathcal{C}$ of pairwise disjoint subsets of V such that choosing exactly one vertex from every set in $\mathcal{C}$ results in a minimal feedback vertex set for G.

Naturally, a set in $\mathcal{C}$ may also contain exactly one vertex; then this vertex is in every minimal feedback vertex set represented by $\mathcal{C}$. This notion of compact representations allows us to easily expand a compact representation to the set of minimal feedback vertex sets it represents and to enumerate the compact representations of all minimal feedback vertex sets within the claimed time bound.

Recall that, in order to compress a size-$(k+1)$ fvs X to a size-k fvs X', the algorithm in Section 3 first tries all partitions of X into Y and S under the assumption that $Y \subseteq X'$ and $S \cap X' = \emptyset$. After deleting the vertices in Y, Reduction Rules 1 and 2 are applied to reduce the instance with respect to its degree-1 and degree-2 vertices. These reduction rules are based on the observation that there is always an optimal solution for FVS without degree-1 and degree-2 vertices (assuming the input graph does not contain a connected component that is a cycle of degree-2 vertices). In contrast, to enumerate *all* minimal feedback vertex sets, the degree-2 vertices cannot be reduced any more because some of these might contain degree-2 vertices. Observe that the number of the vertices with degree higher than two in the graph after deleting the vertices in Y is upper-bounded by $14|S|$ as follows from Lemma 2. Moreover, since degree-1 vertices cannot contribute to a minimal feedback vertex set we can still eliminate all degree-1 vertices as in Section 3. Then compared to finding one feedback vertex set with at most k vertices, the only problem with enumeration is how to deal with degree-2 vertices. The solution to this problem is to use compact representations as detailed in the proof of the following lemma.

Lemma 6 *Given a graph G and an fvs X for G of size $k+1$, we can enumerate compact representations of all minimal feedback vertex sets for G having size at most k in $O(c^k \cdot m)$ time for a constant c.*

PROOF. We show this lemma by constructing all compact representations in the claimed time bound.

Consider a minimal fvs X' with at most k vertices. In comparison to X, the fvs X' retains some vertices $Y \subseteq X$ and replaces the vertices in $S := X \setminus Y$ by at most $|S|-1$ new vertices from $V \setminus X$. Therefore, we begin with a branching into 2^{k+1} cases corresponding to all such partitions of X.

In each case the compact representation is initialized as $\mathcal{C} := \{\{v\} : v \in Y\}$. As in the proof of Lemma 3, we delete the vertices in Y and all degree-1 vertices from G. Let $G' = (V', E')$ denote the resulting graph. We partition V' into three sets $V'_{\geq 3}$, $V'_{=2}$, and S, where $V'_{\geq 3}$ contains the vertices with degree at least 3 in $V' \setminus S$ and $V'_{=2}$ the degree-2 vertices in $V' \setminus S$. Since the two reduction rules do not change $V'_{\geq 3}$, $|V'_{\geq 3}| \leq 14|S|$ due to Lemma 2. We then make a further branching into at most $\sum_{l=0}^{|S|-1} \binom{14|S|}{l}$ cases; in each case, $\mathcal{C}$ is extended by l one-element sets $\{\{v\} : v \in V'_{\geq 3}\}$ for $0 \leq l \leq |S|-1$. In each of these cases we delete from G' those vertices in $V'_{\geq 3}$ which are added to $\mathcal{C}$ and we successively reduce the degree-1 vertices as they cannot participate in a minimal fvs. Let $G'' = (V'', E'')$ denote the resulting graph and $V''_{=2}$ denote the set of degree-2 vertices in $V'' \setminus S$. If G'' is empty, then we have a compact representation $\mathcal{C}$. Otherwise, the cycles in G'' can only be destroyed by deleting degree-2 vertices.

In G'', we identify every maximal path of vertices $v_1, v_2, \ldots, v_r$ where $v_i \in V''_{=2}$ for $i = 1, \ldots, r$, v_i is adjacent to v_{i+1} for $i = 1, \ldots, r-1$, and both v_1 and v_r are adjacent to vertices in $V'' \setminus V''_{=2}$. Clearly, a minimal feedback vertex set may contain at most *one* vertex from such a path. If it does contain one vertex from a path, then it does not matter which vertex is chosen. Therefore, since we are aiming for a compact representation of a minimal feedback vertex set, we save all these maximal paths in a set $\mathcal{P}$, i.e., $\mathcal{P} := \{\{v_1, v_2, \ldots, v_r\} : v_1, v_2, \ldots, v_r \in V''_{=2} \text{ form a maximal path}\}$.

Having obtained $\mathcal{P}$ in this way, we now show that $|\mathcal{P}| \leq 16 \cdot |S|$. To this end, we define a bipartite graph B which on one side has as vertices the elements of $\mathcal{P}$ and on the other side the vertices in $V'' \setminus V''_{=2}$. An element of $\mathcal{P}$ has an edge to a vertex v in $V'' \setminus V''_{=2}$ iff one endpoint of its corresponding maximal path has an edge to v in G''. Note that there can be multiple edges between an element of $\mathcal{P}$ and a vertex in $V'' \setminus V''_{=2}$. Completing $\mathcal{C}$ to a compact representation of minimal feedback vertex sets having size at most k is now equivalent to selecting at most $|S| - l - 1$ many elements of $\mathcal{P}$ to eliminate all cycles in B.

We can now infer that

$$|\mathcal{P}| \leq |V'' \setminus V''_{=2}| + (|S| - l - 1);$$

otherwise, it would not be possible to remove all cycles from B by deleting $|S| - l - 1$ elements of $\mathcal{P}$. Therefore,

$$|\mathcal{P}| \leq |V'' \setminus V''_{=2}| + (|S| - l - 1) \leq |V'_{\geq 3}| + |S| + |S| = 16|S|.$$

Now, we make the last branching into $\sum_{j=0}^{|S|-l-1} \binom{16|S|}{j}$ cases; each case represents a choice of at most $|S| - l - 1$ elements from $\mathcal{P}$. For a case where the resulting graph by deleting these chosen elements from B is cycle-free, we extend $\mathcal{C}$ by the chosen elements.

Altogether, we have at most 2^{k+1} partitions of X, and for each partition at most $\sum_{l=0}^{|S|-1} \binom{14|S|}{l}$ cases corresponding to the choices of vertices with degree more than two, and then for each possible choice of vertices in $V'_{\geq 3}$, we have further $\sum_{j=0}^{|S|-l-1} \binom{16|S|}{j}$ choices for degree-2 vertices in compact form. In summary, the compact representations can be computed in

$$
\begin{aligned}
&O\left(2^{k+1} \cdot \sum_{l=0}^{k-1} \binom{14k}{l} \cdot \sum_{j=0}^{k-l-1} \binom{16k}{j} \cdot m\right) \\
=&O\left(2^{k+1} \cdot \sum_{l=0}^{k-1} \binom{14k}{l} \cdot \binom{16k}{k-l-1} \cdot m\right) \\
=&O\left(2^{k+1} \cdot \binom{30k}{k-1} \cdot m\right) \\
=&O(c^k \cdot m)
\end{aligned}
$$

time for a constant c. □

Together with Theorem 5, we have the following result:

Theorem 7 *All minimal feedback vertex sets of size at most k can be enumerated in $O(c^k \cdot m)$ time for a constant c.*

5 Algorithm for Edge Bipartization

In this section we present a new algorithm for Edge Bipartization which runs in $O(2^k \cdot m^2)$ time. The algorithm is structurally similar to the $O(3^k \cdot kmn)$ time iterative compression algorithm for Graph Bipartization given by Reed et al. [31,19]. As many other known compression routines (such as the one from Section 3), it starts by enumerating all partitions of the known solution into two parts, one containing vertices to keep in the solution and one containing the vertices to exchange. This is followed by a second step that tries to find a compressed bipartization set under this constraint. By enforcing that the smaller solution is disjoint from the known one, our algorithm for Edge Bipartization does not need the first step, thereby significantly reducing the combinatorial explosion.

We remark that a similar runtime of $2^k \cdot n^{O(1)}$ for Edge Bipartization can be achieved by first reducing the input instance to Graph Bipartization [33], and then exploiting a solution disjointness property analogous to the algorithm presented below. This, however, involves several nontrivial modifications to the algorithm of Reed et al. whereas we give a self-contained presentation here. Moreover, our proof reveals details about the structure of Edge Bipartization that might be of independent interest.

The following lemma provides some central insight into the structure of a minimal edge bipartization set.

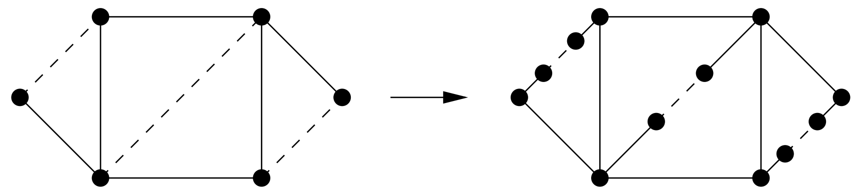

Fig. 2. A graph (left) with an edge bipartization set X (*dashed lines*). To be able to assume without loss of generality that a bipartization set smaller than X is disjoint from X, we subdivide each edge in X by two vertices and choose the middle edge from each thus generated path as the new X (right).

Lemma 8 *Given a graph $G = (V, E)$ and a minimal edge bipartization set X for G, the following two properties hold:*

(1) For every odd-length cycle C in G, $|E(C) \cap X|$ is odd.
(2) For every even-length cycle C in G, $|E(C) \cap X|$ is even.

PROOF. For each edge $e = \{u, v\} \in X$, note that u and v are on the same side of the bipartite graph $G \setminus X$ since otherwise we do not need e to be in X and X would not be minimal. Consider a cycle C in G. The edges in $E(C) \setminus X$ are all between the two sides of $G \setminus X$, while the edges in $E(C) \cap X$ are between vertices of the same side as argued above. In order for C to be a cycle, however, this implies that $|E(C) \setminus X|$ is even. Since $|E(C)| = |E(C) \setminus X| + |E(C) \cap X|$, we conclude that $|E(C)|$ and $|E(C) \cap X|$ have the same parity. $\square$

As mentioned above, it is helpful to assume that an edge bipartization set which is smaller than a given edge bipartization set X is disjoint from X. This can be assumed without loss of generality by applying a simple input transformation (see Figure 2). Since this transformation preserves the parities of the lengths of all cycles, it is easy to see that the thus transformed graph has an edge bipartization set with i edges iff the original graph has an edge bipartization set with i edges. Moreover, for each edge bipartization set Y for the transformed graph there is an edge bipartization set of the same size that is disjoint from X, which can be obtained by replacing every edge in $Y \cap X$ by one of its two adjacent edges.

The following simple definition is the only remaining prerequisite for the central lemma of this section.

Definition 9 *Let $G = (V, E)$ be a graph and $X \subseteq E$. A mapping $\Phi : V(X) \to \{A, B\}$ is called* valid partition of $V(X)$ *if for each $\{u, v\} \in X$, we have $\Phi(u) \neq \Phi(v)$.*

Lemma 10 *Consider a graph $G = (V, E)$ and a minimal edge bipartization set X for G. For a set of edges $Y \subseteq E$ with $X \cap Y = \emptyset$, the following are*

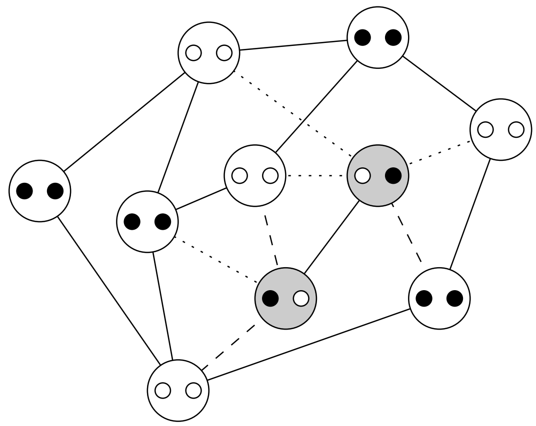

Fig. 3. A graph G with two disjoint edge bipartization sets X marked by *dotted lines* and Y marked by *dashed lines*. The left circle in each vertex denotes a two-coloring C_X for $G \setminus X$, and the right circle denotes a two-coloring C_Y for $G \setminus Y$. Vertices v with $C_X(v) \neq C_Y(v)$ are shaded, corresponding to Φ as defined in the proof of Lemma 10.

equivalent:

(1) Y is an edge bipartization set for G.
(2) There is a valid partition Φ of $V(X)$ such that Y is an edge cut in $G \setminus X$ between $A_\Phi := \Phi^{-1}(A)$ and $B_\Phi := \Phi^{-1}(B)$.

PROOF. (2) $\Rightarrow$ (1): Consider any odd-length cycle C in G. It suffices to show that $E(C) \cap Y \neq \emptyset$. Let $s := |E(C) \cap X|$. By Property (1) in Lemma 8, s is odd. Without loss of generality, we assume that $E(C) \cap X = \{\{u_0, v_0\}, \ldots, \{u_{s-1}, v_{s-1}\}\}$ with vertices v_i and $u_{(i+1) \bmod s}$ being connected by a path in $C \setminus X$. Since Φ is a valid partition, we have $\Phi(u_i) \neq \Phi(v_i)$ for all $0 \leq i < s$. With s being odd, this implies that there is a pair $v_i, u_{(i+1) \bmod s}$ such that $\Phi(v_i) \neq \Phi(u_{(i+1) \bmod s})$. Since the removal of Y destroys all paths in $G \setminus X$ between A_Φ and B_Φ, we obtain that $E(C) \cap Y \neq \emptyset$.

(1) $\Rightarrow$ (2): Let $C_X : V \to \{A, B\}$ be a two-coloring of the bipartite graph $G \setminus X$ and $C_Y : V \to \{A, B\}$ a two-coloring of the bipartite graph $G \setminus Y$. Define

$$\Phi : V \to \{A, B\},\ v \mapsto \begin{cases} A & \text{if } C_X(v) = C_Y(v), \\ B & \text{otherwise.} \end{cases}$$

We show that $\Phi|_{V(X)}$ (that is, Φ with domain restricted to $V(X)$) is a valid partition with the desired property (see Figure 3 for an example).

First we show that $\Phi|_{V(X)}$ is a valid partition. Consider any edge $\{u, v\} \in X$. There must be at least one even-length path in $G \setminus X$ from u to v; otherwise, $\{u, v\}$ would be redundant as X would not be minimal. Therefore, $C_X(u) =$

$C_X(v)$. In $G \setminus Y$, the vertices u and v are connected by an edge, and therefore $C_Y(u) \neq C_Y(v)$. It follows that $\Phi(u) \neq \Phi(v)$.

Since both C_X and C_Y change in value when going from a vertex to its neighbor in $G \setminus (X \cup Y)$, the value of Φ is constant along any path in $G \setminus (X \cup Y)$. Therefore, there can be no path from any $u \in A_\Phi$ to any $v \in B_\Phi$ in $G \setminus (X \cup Y)$, that is, Y is an edge cut between A_Φ and B_Φ in $G \setminus X$. □

Theorem 11 Edge Bipartization *can be solved in $O(2^k \cdot m^2)$ time.*

PROOF. Given as input a graph G with edge set $\{e_1, \ldots, e_m\}$, we can apply iterative compression to solve Edge Bipartization for G by iteratively considering the graphs G_i induced by the edge set $\{e_1, \ldots, e_i\}$ for $i = 1, \ldots, m$. For $i = 1$, the optimal edge bipartization set is empty. For $i > 1$, assume that an optimal edge bipartization set X_{i-1} with $|X_{i-1}| \leq k$ for G_{i-1} is known. If X_{i-1} is not an edge bipartization set for G_i, then we consider the set $X_{i-1} \cup \{e_i\}$, which obviously is a minimal edge bipartization set for G_i. Using Lemma 10, we can in $O(2^{k'} \cdot k'i)$ time (where $k' := |X_{i-1} \cup \{e_i\}| \leq k+1$) either determine that $X_{i-1} \cup \{e_i\}$ is an optimal edge bipartization set for G_i or otherwise compute an optimal edge bipartization set X_i for G_i. This process can be aborted if $|X_i| > k$, since then no solution exists. Summing over all iterations, we have an algorithm that computes an optimal edge bipartization set for G in $O(\sum_{i=1}^{m} 2^{k+1} \cdot ki) = O(2^k \cdot km^2)$ time.

It remains to describe the compression routine that, given a graph and a minimal edge bipartization set X of size k', either computes a smaller edge bipartization set Y in $O(2^{k'} \cdot k'm)$ time or proves that no such Y exists. For this, we first apply the input transformation from Figure 2 which allows us to assume the prerequisite of Lemma 10 that $Y \cap X = \emptyset$. We then enumerate all $2^{k'}$ valid partitions Φ of $V(X)$ and determine a minimum edge cut between A_Φ and B_Φ until we find an edge cut Y of size $k' - 1$. Each of the minimum cut problems can individually be solved in $O(k'm)$ time with the Edmonds–Karp algorithm that goes through k' rounds, each time finding a flow augmenting path [7]. By Lemma 10, Y is an edge bipartization set. Furthermore, if no such Y is found, we know that k' is minimum.

With the same technique as used by Hüffner [19, Section 4.1] to improve the runtime of the iterative compression algorithm for Graph Bipartization, the runtime can be improved from $O(2^k \cdot km^2)$ to $O(2^k \cdot m^2)$. For this, one uses a Gray code to enumerate the valid partitions in such a way that consecutive partitions differ in only one element. For each of these (but the first one), one can then solve the flow problem by a constant number of augmentation operations on the previous network flow in $O(m)$ time. □

6 Conclusion

We presented significantly improved results on the fixed-parameter tractability of Feedback Vertex Set and Edge Bipartization. To our belief, the iterative compression strategy due to Reed et al. [31] employed in this work will become an important tool in the design of efficient fixed-parameter algorithms [25].

We proved that FVS is even solvable in *linear* time for constant parameter value k. Employing a completely different technique, a similar result could very recently be shown for Graph Bipartization restricted to planar graphs (where the problem remains NP-complete) [17]. For general Graph Bipartization as well as for Edge Bipartization, this remains an open question for future research.

Further, we leave it open to explore the practical performance of the described algorithms. Initial experiments with the closely related Graph Bipartization problem [19] make us believe that our algorithms have potential for applications and experiments. To this end, it would also be useful to develop data reduction rules and kernelizations (see [12–14,24,25]) for both problems.

It would be interesting to further investigate parameterized enumeration aspects of feedback set problems. Schwikowski and Speckenmeyer [32] study "classical algorithms" for enumerating feedback set problems—it remains to see how our corresponding result for FVS may contribute to additional "parameterized achievements" in this direction. Also, the enumeration of edge- and vertex bipartization sets appears to be an interesting topic for future research.

Finally, it remains a long-standing open problem whether Feedback Vertex Set on *directed* graphs is fixed-parameter tractable. The answer to this question would mean a significant breakthrough in the field. Currently, we are only aware of fixed-parameter tractability results for the special case of tournament graphs [28,10].

Acknowledgements

The authors would like to thank the anonymous referees of *WADS* 2005 for helpful and inspiring comments and an anonymous referee of the *Journal of Computer and System Sciences* for helping to improve the presentation of results in this work.

References

[1] A. Agarwal, M. Charikar, K. Makarychev, and Y. Makarychev. $O(\sqrt{\log n})$ approximation algorithms for min UnCut, min 2CNF deletion, and directed cut problems. In *Proc. 37th STOC*, pages 573–581. ACM Press, 2005.

[2] A. Avidor and M. Langberg. The multi-multiway cut problem. In *Proc. 9th SWAT*, volume 3111 of *LNCS*, pages 273–284. Springer, 2004.

[3] V. Bafna, P. Berman, and T. Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM Journal on Discrete Mathematics*, 3(2):289–297, 1999.

[4] R. Bar-Yehuda, D. Geiger, J. Naor, and R. M. Roth. Approximation algorithms for the feedback vertex set problem with applications to constraint satisfaction and Bayesian inference. *SIAM Journal on Computing*, 27(4):942–959, 1998.

[5] A. Becker, R. Bar-Yehuda, and D. Geiger. Randomized algorithms for the Loop Cutset problem. *Journal of Artificial Intelligence Research*, 12:219–234, 2000.

[6] H. L. Bodlaender. On disjoint cycles. *International Journal of Foundations of Computer Science*, 5(1):59–68, 1994.

[7] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2nd edition, 2001.

[8] P. Damaschke. Parameterized enumeration, transversals, and imperfect phylogeny reconstruction. *Theoretical Computer Science*, 351(3):337–350, 2006.

[9] F. Dehne, M. R. Fellows, M. A. Langston, F. Rosamond, and K. Stevens. An $O(2^{O(k)}n^3)$ FPT algorithm for the undirected feedback vertex set problem. In *Proc. 11th COCOON*, volume 3595 of *LNCS*, pages 859–869. Springer, 2005. Long version to appear in *Theory of Computing Systems*.

[10] M. Dom, J. Guo, F. Hüffner, R. Niedermeier, and A. Truß. Fixed-parameter tractability results for feedback set problems in tournaments. In *Proc. 6th CIAC*, LNCS. Springer, 2006. To appear.

[11] R. G. Downey and M. R. Fellows. Fixed-parameter tractability and completeness. *Congressus Numerantium*, 87:161–187, 1992.

[12] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, 1999.

[13] M. R. Fellows. Blow-ups, win/win's, and crown rules: Some new directions in FPT. In *Proc. 29th WG*, volume 2880 of *LNCS*, pages 1–12. Springer, 2003.

[14] M. R. Fellows. New directions and new challenges in algorithm design and complexity, parameterized. In *Proc. 8th WADS*, volume 2748 of *LNCS*, pages 505–520. Springer, 2003.

[15] H. Fernau. On parameterized enumeration. In *Proc. 8th COCOON*, volume 2383 of *LNCS*, pages 564–573. Springer, 2002.

[16] P. Festa, P. M. Pardalos, and M. G. C. Resende. Feedback set problems. In D. Z. Du and P. M. Pardalos, editors, *Handbook of Combinatorial Optimization, Vol. A*, pages 209–258. Kluwer, 1999.

[17] S. Fiorini, N. Hardy, B. Reed, and A. Vetta. Planar graph bipartization in linear time. In *Proc. 2nd GRACO*, Electronic Notes in Discrete Mathematics, 2005. Long version to appear in *Discrete Applied Mathematics*.

[18] J. Guo. *Algorithm Design Techniques for Parameterized Graph Modification Problems*. PhD thesis, Institut für Informatik, Friedrich-Schiller-Universität Jena, Germany, 2006.

[19] F. Hüffner. Algorithm engineering for optimal graph bipartization. In *Proc. 4th WEA*, volume 3503 of *LNCS*, pages 240–252. Springer, 2005.

[20] A. B. Kahng, S. Vaya, and A. Zelikovsky. New graph bipartizations for double-exposure, bright field alternating phase-shift mask layout. In *Proc. Asia and South Pacific Design Automation Conference*, pages 133–138, 2001.

[21] I. Kanj, M. Pelsmajer, and M. Schaefer. Parameterized algorithms for feedback vertex set. In *Proc. 1st IWPEC*, volume 3162 of *LNCS*, pages 235–247. Springer, 2004.

[22] S. Khot. On the power of unique 2-prover 1-round games. In *Proc. 34th STOC*, pages 767–775. ACM Press, 2002.

[23] C. Lund and M. Yannakakis. The approximation of maximum subgraph problems. In *Proc. 20th ICALP*, volume 700 of *LNCS*, pages 40–51. Springer, 1993.

[24] R. Niedermeier. Ubiquitous parameterization—invitation to fixed-parameter algorithms. In *Proc. 29th MFCS*, volume 3153 of *LNCS*, pages 84–103. Springer, 2004.

[25] R. Niedermeier. *Invitation to Fixed-Parameter Algorithms*. Oxford University Press, 2006.

[26] C. H. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *Journal of Computer and System Sciences*, 43(3):425–440, 1991.

[27] M. Pop, D. S. Kosack, and S. L. Salzberg. Hierarchical scaffolding with Bambus. *Genome Research*, 14(1):149–159, 2004.

[28] V. Raman and S. Saurabh. Parameterized algorithms for feedback set problems and their duals in tournaments. *Theoretical Computer Science*, 351(3):446–458, 2006.

[29] V. Raman, S. Saurabh, and C. R. Subramanian. Faster fixed parameter tractable algorithms for undirected feedback vertex set. In *Proc. 13th ISAAC*, volume 2518 of *LNCS*, pages 241–248. Springer, 2002.

[30] V. Raman, S. Saurabh, and C. R. Subramanian. Faster algorithms for feedback vertex set. In *Proc. 2nd GRACO*, Electronic Notes in Discrete Mathematics, 2005.

[31] B. Reed, K. Smith, and A. Vetta. Finding odd cycle transversals. *Operations Research Letters*, 32(4):299–301, 2004.

[32] B. Schwikowski and E. Speckenmeyer. On enumerating all minimal solutions of feedback problems. *Discrete Applied Mathematics*, 117(1–3):253–265, 2002.

[33] S. Wernicke. On the algorithmic tractability of single nucleotide polymorphism (SNP) analysis and related problems. Diplomarbeit, WSI für Informatik, Universität Tübingen, 2003.