A quadratic kernel for feedback vertex set.

Stéphan Thomassé*
Université Montpellier II - LIRMM,
161 rue Ada, 34392 Montpellier Cedex, France
thomasse@lirmm.fr

Abstract

We prove that given an undirected graph G on n vertices and an integer k, one can compute in polynomial time in n a graph G' with at most $5k^2+k$ vertices and an integer k' such that G has a feedback vertex set of size at most k iff G' has a feedback vertex set of size at most k'. This result improves a previous $O(k^{11})$ kernel of Burrage et al. [6], and a more recent cubic kernel of Bodlaender [3]. This problem was communicated by Fellows in [5].

1 Introduction

One efficient way of dealing with NP-hard problems is to identify a parameter which contains its computational hardness. For instance, instead of asking for a minimum vertex cover in a graph - a classical NP-hard optimization question - one can ask for an algorithm which would decide, in $O(f(k).n^d)$ time for some fixed d, if a graph of size n has a vertex cover of size at most k. If such an algorithm exists, the problem is called *fixed-parameter tractable*, or FPT for short. An extensive litterature is devoted to FPT, the reader is invited to read for instance [10], [17] or [11].

Kernelization is a natural way of proving that a problem is FPT. Formally, a *kernelization algorithm* receives as input an instance G, k of the parameterized problem, and outputs, in polynomial time in the size of the instance, an instance G', k' such that

- $k' \leq k$,
- the size of G' only depends on k,
- the instances G, k and G', k' are both true or both false.

The reduced instance G', k' is a *kernel* of G, k. The existence of a kernelization algorithm clearly implies the existence of an FPT algorithm since one can kernelize the instance, and then solve the reduced instance G', k' using any (valid) algorithm, hence giving an $O(f(k) + n^d)$ algorithm. A classical result asserts that being FPT is indeed equivalent to having kernelization. However, the proof of this result does not imply that the size of the reduced instance G' is small with respect to k. A much more constrained condition than simply being FPT is then to be able to reduce to an instance with polynomial size in k. And indeed, in the parameterized problems zoology, an important distinction is done between three classes: W[1]-hard, FPT, and polynomial kernelization.

Maybe of more interest than a new refinement in the hardness hierarchy is the fact that kernelization deals with reduction rules. Indeed, no branching is allowed in the process, since we start with an instance G, k and output another instance G', k'. Hence, the involved reduction rules can serve as a preprocessing stage in any attempt to solve the problem. In other words, kernelization can be seen as the preliminar non-branching computation of the instance, and thus is a very generic tool. Being then able to reduce the instance to linear or quadratic size in k with reasonable constants is a very good first step towards the solution.

One of the long standing questions in polynomial kernelization was the Feedback Vertex Set problem whose input is a graph G and an integer k and whose output is true if one can remove at most k vertices to G to form a forest, and false otherwise. In a more formal way, a *feedback vertex set* of a graph $G = (V, E)$ is a subset $S \subseteq V$ such that $G \setminus S$ is acyclic, or equivalently is a forest. Note that our graphs may have loops and multiple edges. However, since edges with multiplicity more than two are irrelevant for our purpose, we will implicitely assume that all multiple edges are double edges. In practice, if any edge with multiplicity more than two appears in our graph operations, we will make use of a tacit reduction rule which reduces its multiplicity to two.

The Feedback Vertex Set problem was first shown to be FPT by Downey and Fellows [9]. Some faster FPT algorithms were provided in [2], [10], [1], [18], [15], [19], [14], [8]. Razgon [20] gave an exact algorithm in $O(1.8899^n)$ time, improved by Fomin et

*research supported by the french ANR-project "Graph decompositions and algorithms (GRAAL)" ANR-06-BLAN-0148

al. [12] to $O(1.7548^n)$. The polynomial kernelization of FEEDBACK VERTEX SET was solved by Burrage et al. [6] in $O(n^{11})$, and improved to $O(n^3)$ by Bodlaender [3]. This latter result used and generalized the reduction rules of the $O(n^{11})$ kernelization to obtain a cubic kernel after an intricate argument.

We provide in this paper a quadratic kernel for FEEDBACK VERTEX SET which reduces the input graph G to a graph G' with size at most $5k^2+k$. Our reduction rules are very much inspired from their proofs apart from a new reduction rule (Rule 5). The noticeable fact concerning this new rule is that it is heavily based on combinatorial optimization results. Namely, we make use of the very classical Hall's matching theorem, but also of a less known min/max result of Gallai [13] concerning disjoint A-paths in a graph (which is in fact equivalent to the maximum matching problem in general graphs). The fact that a factor of k could be saved (from cubic kernel to quadratic kernel) possibly follows from the fact that we switched from reduction rules based on counting arguments to reduction rules based on min/max duality.

For the sake of readability, we first avoid some technicalities to obtain a kernel of size $8k^2 + k$, and then show in the last part how to reduce this bound to $5k^2 + k$. A kernelization in $4k^2 + k$, to the cost of a new reduction rule, is then briefly sketched.

2 Matching based tools

Our kernalization of FEEDBACK VERTEX SET makes use of two results both admitting a proof based on maximum matching in graphs (nonbipartite and bipartite). Given a vertex x of G, an *x-flower* of order k is a set of k cycles pairwise intersecting exactly on x. Given a set of vertices A, an *A-path* is a path with length at least one, having its endvertices in A, and its internal vertices outside of A.

THEOREM 2.1. (Gallai [13]) *Let A be a subset of vertices of a graph G. If the maximum number of vertex disjoint A-paths is strictly less than $k + 1$, there exists a set of vertices $X \subseteq V$ of size at most $2k$ intersecting every A-path.*

This result follows from a stronger min/max statement, also presented in the original paper of Gallai. For a short proof and a generalization to digraphs, one can read Kriesell [16]. Schrijver [21] showed how Gallai's result can be expressed into a maximum matching problem in nonbipartite graphs. Hence, both conclusions of Theorem 2.1, the set of $k + 1$ disjoint A-paths or the set X are computable in polynomial time in $|V|$. A straightforward consequence of this result is:

COROLLARY 2.1. *Let x be a vertex of a graph G which is not a loop. If there is no x-flower of order $k + 1$, there exists a set of vertices $X \subseteq V \setminus x$ of size at most $2k$ intersecting every cycle containing x.*

Proof. If x is only incident to simple edges, there are no $k+1$ disjoint $N(x)$-paths in the graph $G\setminus x$, where $N(x)$ denotes the neighborhood of x. Thus Theorem 2.1 gives our conclusion. If x is joined to l vertices by double edges, we first select in X these l vertices, and then intersect every other cycle containing x with $2(k - l)$ vertices.

The following result is a straightforward consequence of Hall's theorem. We will need it to apply our key reduction rule. Here $N(Z)$ denotes the set of neighbors of the vertices in Z.

THEOREM 2.2. *Let G be a bipartite graph on bipartition (X, Y). There exists a polynomial algorithm which computes a subset Z of X such that $|N(Z)| < 2|Z|$ if such a subset Z exists.*

Proof. We construct an auxiliary graph H obtained from G by splitting every vertex x of X into two twins x_1 and x_2 with the same neighborhood in Y as x. We denote by X_1 and X_2 these two copies of X. Hall's Theorem gives:

- either there exists a matching in H which covers $X_1 \cup X_2$, in which case every subset Z of X has a set of neighbors in Y with size at least $2|Z|$.
- or there exists a subset of vertices $Z_1 \cup Z_2$, where $Z_1 \subseteq X_1$ and $Z_2 \subseteq X_2$, which neighborhood in Y has size strictly less than $|Z_1 \cup Z_2|$. Let Z be the subset of vertices x of X such that $x_1 \in Z_1$ or $x_2 \in Z_2$. Note that the size of Z is at least half of the size of $Z_1 \cup Z_2$. Moreover, the neighborhood $N(Z)$ of Z in Y is exactly the neighborhood of $Z_1 \cup Z_2$. Hence $|N(Z)| < 2|Z|$.

Since a maximum matching, and dually a contracting set, can be computed in polynomial time, the subset Z can be calculated when it exists.

We now make iterative use of Theorem 2.2 in our next proof.

THEOREM 2.3. *Let G be a nonempty bipartite graph on bipartition (X, Y) with $|Y| \geq 2|X|$ and such that every vertex of Y has at least one neighbor in X. Then there exists nonempty subsets $X' \subseteq X$ and $Y' \subseteq Y$ such that the set of neighbors of Y' in G is exactly X', and such that every subset $Z \subseteq X'$ has at least $2|Z|$ neighbors in Y'. In addition, such a pair of subsets X', Y' can be computed in polynomial time in the size of G.*

Proof. We apply Theorem 2.2 to the graph G. If there is no subset Z, we just set $X' := X$ and $Y' := Y$. If a subset Z exists, we simply delete in G the vertices Z from X, and the vertices $N(Z)$ from Y. By our hypothesis, we have that $Z \neq X$ since $|N(X)| = |Y| \geq 2|X|$. Moreover, we still have $|Y \setminus N(Z)| \geq 2|X \setminus Z|$, and by construction, every vertex of $Y \setminus N(Z)$ have its (nonempty) neighborhood contained in $X \setminus Z$. Thus $G \setminus (Z \cup N(Z))$ satisfies the hypothesis of Theorem 2.3. Hence we can iterate our procedure which will terminate on some subsets X', Y' with the required properties.

3 The reduction rules

Let us list some basic reduction rules for the kernelization of the Feedback Vertex Set problem. We assume here that the input is a couple G, k. The first rule, **Rule 0**, simply says that if we can certify that the size of a minimum feedback vertex set of G is more than k, we reduce G, k to some trivial false instance G', k', for example G' is the loop and $k' = 0$.

- **Rule 1** If there is a loop on some vertex x, we reduce to $G' := G \setminus x$ and $k' := k - 1$.
- **Rule 2** If there is a vertex x with degree 0 or 1, we reduce to $G' := G \setminus x$ and $k' := k$.
- **Rule 3** If a vertex x is incident to exactly two edges xy and xz (possibly with $y = z$), we reduce to $G' := (G \setminus x) \cup yz$ and $k' := k$.
- **Rule 4** If there exists an x-flower of order $k + 1$, we reduce to $G' := G \setminus x$ and $k' := k - 1$.

Note that these reduction rules described so far are *safe*, i.e. G has a feedback vertex set of size at most k iff G' has a feedback vertex set of size at most k'. We now introduce our key-rule.

Rule 5 If there is a set of vertices X, a vertex $x \in V \setminus X$ and a set of connected components $\mathcal{C}$ of $G \setminus (X \cup x)$ (not necessarily all the connected components) such that:

- There is exactly one edge between x and every $C \in \mathcal{C}$.
- Every $C \in \mathcal{C}$ induces a tree.
- For every subset $Z \subseteq X$, the number of components of $\mathcal{C}$ having some neighbor in Z is at least $2|Z|$.

Then one can form a graph G' by joining x to every vertex of X by double edges, and removing the edges between x and the components of $\mathcal{C}$. We then reduce to G' and $k' := k$.

Theorem 3.1. *Rule 5 is safe.*

Proof. We more strongly show that the size of a minimum feedback vertex set in G' is equal to the size of a minimum feedback vertex set in G.

Let S' be a feedback vertex set of G'. Since the double edges incident to x force that $x \in S'$ or $X \subseteq S'$, we observe that S' is also a feedback vertex set of G. Indeed, assume for contradiction that C is a cycle of $G \setminus S'$. Then C contains x since we only modified edges incident to x when reducing G to G'. This means that S' does not contain x, thus it contains X. Hence every edge between x and some component of $\mathcal{C}$ is a bridge in $G \setminus S'$, and consequently cannot belong to C. Finally every edge of C belongs to $G' \setminus S'$, contradicting the fact that S' is a feedback vertex set for G'.

Now we assume that S is a feedback vertex set of G, and show that there exists a feedback vertex set S' of G' with size at most $|S|$. If S contains x, it is indeed a feedback vertex set of G' since G and G' only differ on edges incident to x. So we now assume that S does not contain x. Let us denote by Y the set $X \setminus S$. Let us also denote by Z the set of vertices of S which belong to some component of $\mathcal{C}$. The crucial fact is that $S' = (S \cup Y) \setminus Z$ is a feedback vertex set of G', since every C in $\mathcal{C}$ is a component of $G' \setminus S'$. We just have to show that the size of S' is at most the size of S. Assume for contradiction that $|Z| < |Y|$. Observe that every vertex y of Y has a neighbor in at most one component C in $\mathcal{C}$ which does not intersect Z, otherwise the vertices x, y together with two components $C, C' \in \mathcal{C}$ disjoint from Z and joined to y would contain a cycle. This means that the total number of C in $\mathcal{C}$ which have a neighbor in Y is at most $|Y| + |Z| < 2|Y|$, contradicting the definition of X.

Observe that any application of a reduction rule strictly decreases the value $n + s$, where n is the number of vertices of the graph G and s is its number of simple edges. Thus the total number of reduction rules one can apply starting from a graph G is linear in the size of G. Therefore, to get a kernel, we just have to prove that if G is large enough, one can efficiently find a reduction rule to apply. This is the aim of the next section.

4 Applying the rules

We now show how to construct a polynomial kernel for the graph G.

Theorem 4.1. *If G is a graph on n vertices with $n > 8k^2 + k$, one can find a reduction rule to apply to G in polynomial time in n.*

Proof. The application of rules 1 up to 3 is routine, and hence we can assume that G is a loopless graph, with minimum degree 3, and such that the only multiple edges are double edges. Observe now that if there exists a feedback vertex set S of G with at most k vertices, the number of edges between S and $V \setminus S$ is at least $8k^2$. Indeed, $V \setminus S$ spans a forest and hence induces at most $|V \setminus S| - 1$ edges. Moreover every vertex of $V \setminus S$ has degree at least 3, so the total number of edges leaving $V \setminus S$ is at least $3|V \setminus S| - 2(|V \setminus S| - 1) > 8k^2$.

Hence, there must be a vertex of S with degree at least $8k^2/|S| \geq 8k$. In particular, if G has maximum degree less than $8k$, we simply give a negative answer by applying Rule 0. Therefore, we now assume that there exists a vertex x with degree at least $8k$. If there exists an x-flower of order $k+1$, we apply Rule 4. Otherwise, we apply Theorem 2.1 in order to find a set of vertices $X \subseteq V \setminus x$ with $|X| \leq 2k$ such that there is no cycle containing x in $G \setminus X$.

We denote by $\mathcal{C}$ the set of components of $G \setminus (X \cup x)$. Note that the total degree of x inside X is at most $3k$ since the worst case is when x is linked to k vertices of X by double edges (no more than k since there is no x-flower of order $k+1$) and to the other by simple edges. Hence x is incident to at least $5k$ edges which are not incident to X. Since X meets all the cycles containing x, each component of $\mathcal{C}$ is joined to x with at most one edge. Hence at least $5k$ components of $\mathcal{C}$ are joined to x. At most k of these components contain a cycle, otherwise we would reduce G by Rule 0. Consequently, there exists a set Y of at least $4k$ components of $\mathcal{C}$ which induces trees and are joined to x with a single edge.

We now form a bipartite simple graph B on vertex set X, Y where vC is an edge of B, for $v \in X$ and $C \in Y$, if and only if there exists an edge between v and C in G. Since every component C in Y spans a tree, G has minimum degree three and C is only linked to x by an edge, there are edges leaving C which are not incident to x, and thus are incident to X. This means that every element of Y is joined to a vertex of X. Moreover we have $|Y| \geq 2|X|$, hence we can apply Theorem 2.3, to find nonempty subsets $X' \subseteq X$ and $Y' \subseteq Y$ such that the set of neighbors of Y' in B is exactly X', and such that every subset $Z \subseteq X'$ has at least $2|Z|$ neighbors in Y'. Observe now that:

- every element C of Y' is a component of the graph $G \setminus (X' \cup x)$,
- every element C of Y' is linked to x with a single edge,
- every element C of Y' spans a tree.
- every subset Z of X' is linked to at least $2|Z|$ elements of Y'

Consequently, we can apply Rule 5 to G, X', x.

5 Beyond $8k^2 + k$

In order to get a better bound than $8k^2$, one can use the full generality of Gallai's result which gives a certificate to the maximum order of an x-flower:

THEOREM 5.1. *Let G be a graph and x be a vertex of G which is not a loop. The maximum order of an x-flower is equal to the minimum of $|X| + \sum_{C \in \mathcal{C}} \lfloor \frac{e(x,C)}{2} \rfloor$, where X is a subset of vertices, $\mathcal{C}$ is the set of all components of $G \setminus (X \cup x)$, and $e(x, C)$ is the number of edges between x and C.*

Again this min/max result is polynomially computable. Using Theorem 5.1 instead of Theorem 2.1 in the proof of Theorem 4.1 gives a better bound of $5k^2 + k$. We sketch the proof of this, based on the lines of the proof of Theorem 4.1.

There exists a vertex x with minimum degree $5k$ for which there is no x-flower of order $k + 1$. Let X be a subset certifying the maximality of the order of an x-flower, as in Theorem 5.1. Let $\mathcal{C}'$ be the components of $\mathcal{C}$ which are linked to x with more than one edge. We denote by e' the total number of edges between x and the components of $\mathcal{C}'$. Note that $|X| + e'/3$ is at most k, the worst case being when x is linked with exactly three edges to each component of $\mathcal{C}'$. Hence $4k \geq 4|X| + e'$.

Furthermore, there are at most k components of $\mathcal{C} \setminus \mathcal{C}'$ which contain a cycle. In all, the number c of components of $\mathcal{C}$ which are trees and are linked to x with exactly one edge is at least $5k - 2|X| - e' - k$ since each vertex of X can be linked to x with two edges. We then have $c \geq 4k - 2|X| - e'$, thus $c \geq 2|X|$. Finally, we satisfy the hypothesis of Theorem 2.3, and we conclude as in Theorem 4.1.

As a conclusion of this paper, we remark that in the previous analysis, the k components containing a cycle we discarded and the $2|X|$ edges we assumed between x and X can be counted together to obtain a better bound than $5k^2 + k$. Indeed, one can add another rule asserting that if x belong to a p-flower F, and there are q disjoint cycles in $G \setminus F$, then if $p + q \geq k + 1$, we can reduce G since x is certainly in a feedback vertex set. Using this new rule, we can obtain a kernelization of FEEDBACK VERTEX SET of size $4k^2 + k$.

6 Conclusion

The challenge is now to go below $O(k^2)$. In the case of planar graphs, a linear kernel was proposed in [4]. A natural question would be to ask for the existence of a polynomial kernel for the directed feedback vertex set, since this problem was recently proved to be FPT in [7].

References

[1] A. Becker, R. Bar-Yehuda, and D. Geiger, *Randomized algorithms for the loop cutset problem*, J. Artificial Intelligence Research, 12 (2000), pp. 219–234.

[2] H. L. Bodlaender, *On disjoint cycles*, Int. J. Found. Computer Science, 5 (1994), pp. 59–68.

[3] H.L. Bodlaender, *A cubic kernel for feedback vertex set*, in W. Thomas and P. Well (Eds.), Proceedings 24th International Symposium on Theoretical Aspects of Computer Science, STACS 2007, pp. 320–331. Springer, Lecture Notes in Computer Science.

[4] H.L. Bodlaender and E. Penninkx, *A Linear Kernel for Planar Feedback Vertex Set*, IWPEC 2008, pp. 160–171.

[5] H.L. Bodlaender, L. Cai, J. Chen, M.R. Fellows, J.A. Telle and D. Marx, *Open problems in parameterized and exact computation*, Technical Report UU-CS-2006-052, Utrecht University, 2006.

[6] K. Burrage, V. Estivill-Castro, M. R. Fellows, M. A. Langston, S. Mac, and F. A. Rosamond, *The undirected feedback vertex set problem has a poly(k) kernel*, in H. L. Bodlaender and M. A. Langston, editors, Proceedings 2nd International Workshop on Parameterized and Exact Computation, IWPEC 2006, pp. 192–202. Springer Verlag, Lecture Notes in Computer Science, vol. 4169, 2006.

[7] J. Chen, Y. Liu, S. Lu, B. O'Sullivan and I. Razgon, *A fixed-parameter algorithm for the directed feedback vertex set problem*, STOC 2008, pp. 177-186.

[8] F. K. H. A. Dehne, M. R. Fellows, M. A. Langston, F. A. Rosamond, and K. Stevens. *An* $O(2^{O(k)}n^3)$ *fpt algorithm for the undirected feedback vertex set problem*, Theory Comput. Systems 41 (2007), pp. 479–492.

[9] R. G. Downey and M. R. Fellows, *Fixed-parameter tractability and completeness*, Congressus Numerantium, 87 (1992), pp. 161–178.

[10] R. G. Downey and M. R. Fellows, Parameterized Complexity. Springer, 1998.

[11] J. Flum and M. Grohe, Parameterized Complexity Theory. Springer, 2006.

[12] F. V. Fomin, S. Gaspers, and C. Knauer, *Finding a minimum feedback vertex set in time* $O(1.7548^n)$, in H. L. Bodlaender and M. A. Langston, editors, Proceedings 2nd International Workshop on Parameterized and Exact Computation, IWPEC 2006, pp. 183-191. Springer Verlag, Lecture Notes in Computer Science, vol. 4169, 2006.

[13] T. Gallai, *Maximum-minimum sätze und verallgemeinerte Faktoren von Graphen*, Acta. Math. Acad. Sci. Hungaricae, 12 (1961), pp. 131–173.

[14] J. Guo, J. Gramm, F. Huffner, R. Niedermeier, and S. Wernicke, *Improved fixed parameter algorithms for two feedback set problems*, in Proc. 9th Int. Workshop on Algorithms and Data Structures WADS 2004, pp. 158-168. Springer Verlag, Lecture Notes in Computer Science, vol. 3608, 2004.

[15] I. A. Kanj, M. J. Pelmajer, and M. Schaefer, *Parameterized algorithms for feedback vertex set*, in Proc. 1st Int. Workshop on Parameterized and Exact Computation, IWPEC 2004, pp. 235-248. Springer Verlag, Lecture Notes in Computer Science, vol. 3162, 2004.

[16] M. Kriesell, *Disjoint A-paths in digraphs*, J. Comb. Theory, Ser. B, 95 (2005), pp. 168–172.

[17] R. Niedermeier, Invitation to fixed-parameter algorithms. Oxford University Press, 2006.

[18] V. Raman, S. Saurabh, and C. R. Subramanian, *Faster fixed parameter tractable algorithms for undirected feedback vertex set*, in Proceedings 13th International Symposium on Algorithms and Computation, ISAAC 2002, pp. 241 - 248. Springer Verlag, Lecture Notes in Computer Science, vol. 2518, 2002.

[19] V. Raman, S. Saurabh, and C. R. Subramanian, *Faster algorithms for feedback vertex set*, in Proceedings 2nd Brazilian Symposium on Graphs, Algorithms, and Combinatorics, GRACO 2005, Electronic Notes in Discrete Mathematics, 19 (2005), pp. 273–279.

[20] I. Razgon, *Exact computation of maximum induced forest*, in L. Arge and R. Freivalds, editors, 10th ScandinavianWorkshop on Algorithm Theory, SWAT 2006, pp. 160–171. Springer Verlag, Lecture Notes in Computer Science, vol. 4059, 2006.

[21] A. Schrijver,*A short proof of Mader's* S*-paths theorem*, J. Comb. Theory, Ser. B, 82 (2001), pp. 319-321.