Parameterized graph separation problems*

Dániel Marx

Department of Computer Science and Information Theory,
Budapest University of Technology and Economics
Budapest, H-1521, Hungary,
dmarx@cs.bme.hu

Abstract. We consider parameterized problems where some separation property has to be achieved by deleting as few vertices as possible. The following five problems are studied: delete k vertices such that (a) each of the given ℓ terminals is separated from the others, (b) each of the given ℓ pairs of terminals are separated, (c) exactly ℓ vertices are cut away from the graph, (d) exactly ℓ connected vertices are cut away from the graph, (e) the graph is separated into ℓ components, We show that if both k and ℓ are parameters, then (a), (b) and (d) are fixed-parameter tractable, while (c) and (e) are W[1]-hard.

1 Introduction

In this paper we study five problems where we have to delete vertices from a graph to achieve a certain goal. In all four cases, the goal is related to making the graph disconnected by deleting as few vertices as possible.

Classical flow theory gives us a way of deciding in polynomial time whether two vertices t_1 and t_2 can be disconnected by deleting at most k vertices. However, for every $\ell \geq 3$, if we have ℓ terminals t_1, t_2, ..., t_ℓ, then it is NP-hard to find k vertices such that no two terminals are in the same component after deleting these vertices [3]. In [8] a $(2 - 2/\ell)$-approximation algorithm was presented for the problem. Here we give an algorithm that is efficient if k is small: in Section 2 it is shown that the Minimum terminal separation problem is fixed-parameter tractable with parameter k. We also consider the more general Minimum terminal pair separation problem where ℓ pairs (s_1, t_1), ..., (s_ℓ, t_ℓ) are given, and it has to be decided whether there is a set of k vertices whose deletion separates each of the ℓ pairs. We show that this problem is fixed-parameter tractable if both k and ℓ are parameters. Our results can be used in the edge deletion versions of these problems as well.

In Section 3 we consider two separation problems without terminals. In the Separating ℓ Vertices problem exactly ℓ vertices have to be separated from the rest of the graph by deleting at most k vertices. In Separating into ℓ Components problem k vertices have to be deleted such that the remaining

* Research is supported in part by grants OTKA 44733, 42559 and 42706 of the Hungarian National Science Fund.

Problem	Parameter(s)		
	k	ℓ	k and ℓ
Minimum Terminal Separation	FPT (Theorem 1)	NP-hard for $\ell \geq 3$ [3]	FPT (Theorem 1)
Minimum Terminal Pair Separation	Open	NP-hard for $\ell \geq 3$ [3]	FPT (Theorem 2)
Separating ℓ Vertices	W[1]-hard (Theorem 4)	W[1]-hard (Theorem 4)	W[1]-hard (Theorem 4)
Separating ℓ Connected Vertices	W[1]-hard (Theorem 8)	W[1]-hard (Theorem 7)	FPT (Theorem 6)
Separating into ℓ Components	W[1]-hard (Theorem 9)	W[1]-hard (Theorem 9)	W[1]-hard (Theorem 9)

Table 1. Complexity of the problems with different parameterizations.

graph has at least ℓ connected components. The edge deletion variants of these problems were considered in [5], where it is shown that both problems are W[1]-hard with parameter ℓ. Here we show that the vertex deletion versions of both problems are W[1]-hard even if both k and ℓ are parameters. However, in the case of Separating ℓ Vertices if we restrict the problem to bounded degree graphs, then it becomes fixed-parameter tractable if both k and ℓ are parameters. Moreover, we also consider the variant Separating ℓ Connected Vertices, where it is also required that the separated vertices form a connected subgraph. It turns out that this problems is fixed-parameter tractable if both k and ℓ are parameters, but W[1]-hard if only one of them is parameter.

The results of the paper are summarized on Table 1.

2 Separating Terminals

The parameterized terminal separation problem studied in this section is formally defined as follows:

Minimum Terminal Separation
Input: A graph $G(V, E)$, a set of terminals $T \subseteq V$, and an integer k.
Parameter 1: k
Parameter 2: $\ell = |T|$
Question: Is there a set of vertices $S \subseteq V$ of size at most k such that no two vertices of T belong to the same connected component of $G \setminus S$?

Note that S and T do not have to be disjoint, which means that it is allowed to delete terminals. A deleted terminal is considered to be separated from all the other terminals (later we will argue that our results remain valid for the slightly different problem where the terminals cannot be deleted).

It follows from the graph minor theory of Robertson and Seymour that Minimum Terminal Separation is fixed-parameter tractable. The celebrated result of Robertson and Seymour states that graphs are well-quasi ordered with respect to the minor relation. Moreover, the same holds for graphs where the edges are colored with a fixed number of colors. For every terminal $v \in T$, we add a new vertex v' and a red edge vv' (the original edges have color black). Now separating the terminals and separating the red edges are the same problem. Consider the set $\mathscr{G}_k$ that contains those red-black graphs where the red edges can be separated by deleting at most k vertices. It is easy to see that $\mathscr{G}_k$ is closed with respect to taking minors. Therefore by the Graph Minor Theorem, $\mathscr{G}_k$ has a finite set of forbidden minors. Another result of Roberson and Seymour states that for every graph H there is an $O(|V|^3)$ algorithm for finding an H-minor, therefore membership in $\mathscr{G}_k$ can be tested in $O(|V|^3)$ time. This means that for every k, Minimum Terminal Separation can be solved in $O(|V|^3)$ time, thus the problem is (non-uniformly) fixed-parameter tractable. However, the constants given by this non-constructive method are incredibly large. In this section we give a direct combinatorial algorithm for the problem, which is more efficient.

The notion of important separator is the most important definition in this section:

Definition 1. *Let $G(V,E)$ be a graph. For subsets $X, S \subseteq V$, the set of vertices reachable from $X \setminus S$ in $G \setminus S$ is denoted by $R(S,X)$. For $X, Y \subseteq V$, the set S is called an* (X,Y)-separator *if $Y \cap R(S,X) = \emptyset$. An (X,Y)-separator is* minimal *if none of its proper subsets are (X,Y)-separators. An (X,Y)-separator S'* dominates *an (X,Y)-separator S, if $|S'| \leq |S|$ and $R(S,X) \subset R(S',X)$. A subset S is an* important (X,Y)-separator *if it is minimal, and there is no (X,Y)-separator S' that dominates S.*

Abusing notations, the one element set $\{v\}$ is denoted by v. We note that X and Y can have non-empty intersection, but in this case every (X,Y)-separator has to contain $X \cap Y$.

We use Figure 1 to demonstrate the notion of important separator. Let $X = \{x\}$ and $Y = \{y_1, y_2, y_3, y_4\}$, we want separate these two sets. X and Y can be separated by deleting x, this is the only separator of size 1. There are several size 2 separators, for example $\{a, f\}$, $\{b, g\}$, $\{b, j\}$, $\{c, j\}$. However, only $\{c, j\}$ is an important separator: $R(\{c, j\}, x) = \{x, a, b, f, g, h, i\}$ and the set of vertices reachable from x is smaller for the other size 2 separators. There are two size 3 important separators: $\{c, k, \ell\}$ and $\{j, d, e\}$. Separator $\{c, h, i\}$ is not important, since it is dominated both by $\{c, j\}$ and by $\{c, k, \ell\}$. Finally, there is only one important size 4 separator, Y itself.

Testing whether a given (X,Y)-separator S is important can be done as follows. First, minimality can be easily checked by testing for each vertex $s \in S$ whether $S \setminus s$ remains separating. If it is minimal, then for every vertex $s \in S$, we test whether there is an $(R(S,X) \cup s, Y)$-separator S' of size at most $|S|$. This separator can be found in $O(|V|^3)$ time using network flow techniques. If there is such a separator, then S is not important. Notice that if S is not important, then this method can be used to find an important separator that dominates S. The test can be repeated for S', and if it is not important, then we get another separator S'' that dominates S'. We repeat this as many times as necessary.

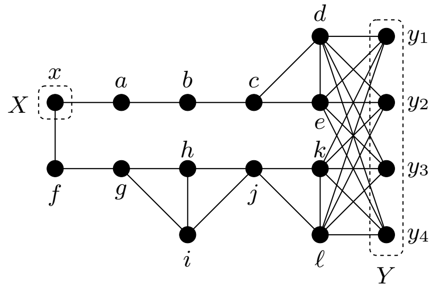

Fig. 1.

Since the set of vertices reachable from X increases at each step, eventually we arrive to an important separator.

Let X and Y be two sets of vertices, then there is at most one important (X, Y)-separator of size 1. A size 1 separator has to be a cut vertex (here we ignore the special cases where $|X| = 1$ or $|Y| = 1$). If there are multiple cut vertices that separate X and Y, then there is a unique cut vertex that is farthest from X and closest to Y. This vertex will be the only important (X, Y)-separator.

However, for larger sizes, there can be many important (X, Y)-separators of a given size. For an example, see Figure 2. To separate the two large cliques X and Y, for each $1 \le i \le t$, either a_i, or both b_i and c_i have to be deleted. If we choose to delete both b_i and c_i, then we have to delete two vertices instead of one, but the set of vertices reachable from X increases, it includes a_i. Therefore there are $\binom{t}{t/2}$ important (X, Y)-separators of size $3t/2$: for $t/2$ of the i's we delete a_i, and for the remaining $t/2$ we delete b_i and c_i. All these separators are important, since $R(S', X)$ and $R(S'', X)$ are pairwise incomparable for two such separators S' and S''. Thus the number of important separators of a given size k can be exponential in k. However, we show that this number is independent of the size of the graph:

Lemma 1. *For sets of vertices X, Y, there are at most 4^{k^2} important (X, Y)-separators of size k. Moreover, these separators can be enumerated in polynomial time per separator.*

Proof. The proof is by induction on k. We have seen above that the statement holds for $k = 1$. Let S be an important (X, Y)-separator of size k in G. We count how many other important separators can be in G. If H is another important (X, Y)-separator of size k, then we consider two cases depending on whether $Z = S \cap H$ is empty or not. If Z is not empty, then it is easy to see that $H \setminus Z$ is an important $(X \setminus Z, Y \setminus Z)$-separator in $G \setminus Z$. Since $|H \setminus Z| < k$, thus by the induction hypotheses the number of such separators is at most $4^{(k-1)^2}$. There are not more than 2^k possibilities for the set Z, and for each set Z there are at most $4^{(k-1)^2}$ possibilities for the set H, hence the total number of different H that intersect S is at most $2^k 4^{(k-1)^2}$.

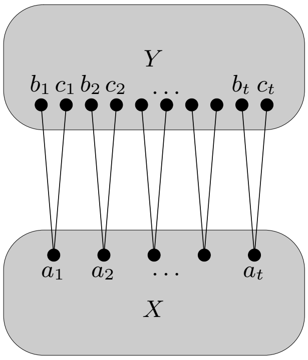

Fig. 2. A graph where there is an exponential number of important separators that separate the large cliques X and Y.

Next we count those separators that do not intersect S. Such a separator H contains ℓ vertices from $R(S, X)$ and $k-\ell$ vertices from $R(S, Y)$. It is not possible that $\ell = 0$: that would imply that $R(S, X) \cup S \subseteq R(H, X)$ and S would not be an important separator. Here we used the minimality of S: if none of $R(S, X)$ and S is deleted, then every vertex of S can be reached from X. Similarly, it is not possible that $\ell = k$ because H would not be an important separator in that case. To see this, notice that by the minimality of S, from every vertex of S a vertex of Y can be reached using only the vertices in $R(S, Y)$. Therefore no vertex of S can be reached from X in $G \setminus H$, otherwise H would not be an (X, Y)-separator. Since S is an (X, Y)-separator, thus this also means that no vertex of $R(S, Y)$ can be reached. Therefore $R(H, X)$ is contained in $R(S, X)$, and since $\ell > 0$, the containment is proper.

We divide H into two parts: let $H_1 = H \cap R(S, X)$ and $H_2 = H \cap R(S, Y)$ (see Figure 3). The separator S is also divided into two parts: $S_1 = S \cap R(H, X)$ contains those vertices that can be reached from X in $G \setminus H$, while $S_2 = S \setminus S_1$ contains those that cannot be reached. Let G_1 be the subgraph of G induced by $R(S, X) \cup S$, and G_2 be the subgraph induced by $R(S, Y) \cup S$. Now it is clear that H_1 is an $(X \cup S_1, S_2)$-separator in G_1, and H_2 is a $(S_1, Y \cup S_2)$-separator in G_2. Moreover, we claim that they are important separators. First, if H_1 is not minimal, i.e., it remains an $(X \cup S_1, S_2)$-separator without $v \in H_1$, then H would be an (X, Y)-separator without v as well. Assume therefore that an $(X \cup S_1, S_2)$-separator H_1^* in G_1 dominates H_1. In this case $H_1^* \cup H_2$ is an (X, Y)-separator in G with $R(H, X) \subset R(H_1^* \cup H_2, X)$, contradicting the assumption that H is an important separator. A similar argument shows that H_2 is an important $(S_1, Y \cup S_2)$-separator in G_2. By the induction hypotheses, we have a bound on the possible number of such separators. For a given division (S_1, S_2) and ℓ, there can be at most $4^{\ell^2} 4^{(k-\ell)^2}$ possibilities. There are at most 2^k possibilities for (S_1, S_2), and the value of ℓ is between 1 and $k-1$. Therefore the total number of different separators (including S itself and the at most $2^k 4^{(k-1)^2}$ sets in the

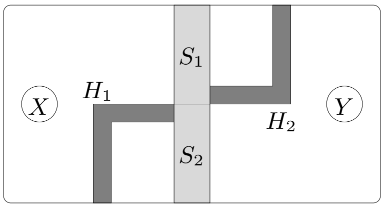

Fig. 3. Separators in the proof of Lemma 1.

first case) is at most

$$1 + 2^k 4^{(k-1)^2} + \sum_{\ell=1}^{k-1} 2^k 4^{\ell^2} 4^{(k-\ell)^2} \le 1 + 2^k 4^{(k-1)^2} + (k-1)2^k 4^{(k-1)^2+1}$$
$$\le k 2^k 4^{(k-1)^2+1} \le 4^k 4^{(k-1)^2+1} = 4^{k+k^2-2k+2} \le 4^{k^2},$$

what we had to show (in the first inequality we used $\ell^2 + (k-\ell)^2 \le (k-1)^2 + 1$, which holds since $1 \le \ell \le k-1$). The proof also gives an algorithm for finding all the important separators. To handle the first case, we take every subset Z of S, and recursively find all the important size $k - |Z|$ separators in $G \setminus S$. In the second case, we consider every $1 \le \ell \le k-1$ and every division (S_1, S_2) of S. We enumerate every important $(X \cup S_1, S_2)$-separator S_1 in G_1 and every important $(S_1, Y \cup S_2)$-separator in G_2. For each S_1, S_2, it has to be checked whether $S_1 \cup S_2$ is an important (X, Y)-separator. As it was shown above, every important separator can be obtained in such a form. Our algorithm makes a constant number of recursive calls with smaller k, therefore the running time is uniformly polynomial. □

What makes important separators important is that a separator in a solution can be always replaced by an important separator:

Lemma 2. *If there is a set S of vertices that separates the terminals $t_1, \ldots, t_r$, then there is a set H with $|H| \le |S|$ that also separates the terminals and contains an important $(\{t_1\}, \{t_2, t_3, \ldots, t_r\})$-separator.*

Proof. Let $S_0 \subseteq S$ be those vertices of S that can be reached from t_1 without going through other vertices of S. Clearly, S_0 is a $(\{t_1\}, \{t_2, t_3, \ldots, t_r\})$-separator, and it contains a minimal separator S_1. If S_1 is important, then we are ready, otherwise there is an important $(\{t_1\}, \{t_2, t_3, \ldots, t_r\})$-separator S_1' that dominates S_1. We claim that $S' = (S \setminus S_1) \cup S_1'$ also separates the terminals. If this is true, then $|S_1'| \le |S_1|$ implies $|S'| \le |S|$, proving the lemma.

Since S_1' is a $(\{t_1\}, \{t_2, t_3, \ldots, t_r\})$-separator, thus S' separates t_1 from all the other vertices. Assume therefore that there is a path P in $G \setminus S'$ connecting terminals t_i and t_j. Since S separates t_i and t_j, thus this is only possible if P goes through a vertex v of S_1. Every vertex of $S_1 \subseteq S_0$ has a neighbor in $R(S, t_1)$, let w this neighbor of v. Since $R(S, t_1) \subseteq R(S', t_1)$, vertex w can be reached from t_1 in $G \setminus S'$. Therefore t_i can be reached from t_1 via w and v, which is a contradiction, since S' is a $(\{t_1\}, \{t_2, t_3, \ldots, t_r\})$-separator. □

Lemma 1 and Lemma 2 allows us to use the method of bounded search trees to solve the Minimum Terminal Separation problem:

Theorem 1. Minimum Terminal Separation *is fixed-parameter tractable with parameter k.*

Proof. We select an arbitrary terminal t that is not already separated from every other terminal. By Lemma 2, there is a solution that contains an important $(t, T \setminus t)$-separator. Using Lemma 1, we enumerate all the at most $k4^{k^2}$ important separators of size at most k, and select a separator S from this list. We delete S from G, and recursively solve the problem for $G \setminus S$ with problem parameter $k - |S|$. At each step we can branch into at most $k4^{k^2}$ directions, and the problem parameter is decreased by at least one, hence the search tree has height at most k and has at most $k^k 4^{k^3}$ leaves. The work to be done is polynomial at each step, hence the algorithm is uniformly polynomial. □

A natural way to generalize Minimum Terminal Separation is to have a more complicated restriction on which terminals should be separated. Instead of a set of terminals where every terminal has to be separated from every other terminal, in the following problem there are pairs of terminals, and every terminal has to be separated only from its pair:

Minimum Terminal Pair Separation
Input: A graph $G(V, E)$, pairs of vertices (s_1, t_1), (s_2, t_2), $\ldots$, (s_ℓ, t_ℓ), and an integer k.
Parameter 1: k
Parameter 2: ℓ
Question: Is there a set of vertices $S \subseteq V$ of size at most k such that for every $1 \le i \le \ell$, vertices s_i and t_i are in different components of $G \setminus S$?

Let $T = \bigcup_{i=1}^{\ell} \{s_i, t_i\}$ be the set of terminals. We can prove an analog of Lemma 2: there is an optimal solution containing an important separator.

Lemma 3. *If there is a set S of vertices that separates every pair, then there is a set S' with $|S'| \le |S|$ that also separates the pairs and S' contains an important $(\{s_1\}, T')$-separator for some subset $T' \subseteq T$.*

Proof. We proceed similarly as in the proof of Lemma 2. Let T' be the set of those terminals that are separated from s_1 in $G \setminus S$. Let $S_0 \subseteq S$ be the vertices reachable from s_1 without going through other vertices of S. Clearly, S_0 is an (s_1, T')-separator, and it contains a minimal (s_1, T')-separator S_1. If S_1 is not important, then there is an important (s_1, T')-separator S_1' that dominates S_1. We claim that $S' = (S \setminus S_1) \cup S_1'$ also separates the pairs. Clearly, $t_1 \in T'$, hence s_1 and t_1 are separated in S'. Assume therefore that s_i and t_i are connected by a path P in $G \setminus S'$. As in Lemma 2, path P goes through a vertex of S_1, and it follows that both s_i and t_i are connected to s_1 in $G \setminus S'$. Therefore $s_i, t_i \neq T'$. However, this implies that s_1 is connected to s_i and t_i in $G \setminus S$, hence S does not separate s_i from t_i, a contradiction. □

To find k vertices that separate the pairs, we use the same method as in Theorem 1. In Lemma 3, there are 2^ℓ different possibilities for the set T', and by Lemma 1, for each T' there are at most $k4^{k^2}$ different separators of size at most k. Therefore we can generate $2^\ell \cdot k2^{k^2}$ separators such that one of them is contained in an optimum solution. This results in a search tree with at most $2^{k\ell} \cdot k^k 4^{k^3}$ leaves.

Theorem 2. *The* Minimum Terminal Pair Separation *problem is fixed-parameter tractable with parameters k and ℓ.* □

Separating the terminals in T can be expressed as separating $\binom{|T|}{2}$ pairs, hence Minimum Terminal Separation is a special case of Minimum Terminal Pair Separation. However, Theorem 2 does not imply Theorem 1. In Theorem 2 the number of pairs is a parameter, while the size of T can be unbounded in Theorem 1. We do not know the complexity of Minimum Terminal Pair Separation if only k is the parameter.

As noted above, in the separation problems we assume that any vertex can be deleted, even the terminals themselves. However, we can consider the slightly more general problem, when the input contains a set V^* of distinguished vertices, and these vertices cannot be deleted. All the results in this section hold for this variant of the problem as well. In all of the proofs, when a new separator is constructed, then it is constructed from vertices that were contained in some other separator.

We can consider the variants of Minimum Terminal Separation and Minimum Terminal Pair Separation where the terminals have to be separated by deleting at most k edges. The edge deletion problems received more attention in the literature: they were consider in e.g. [4,3,7] under the names multiway cut, multiterminal cut, and multicut. As noted in [8], it is easy to reduce the edge deletion problem to vertex deletion, therefore our algorithms can be used for these edge deletion problems as well. For completeness, we briefly describe a possible reduction. The edge deletion problem can be solved by considering the line graph (in the line graph $L(G)$ of G the vertices correspond to the edges of G, and two vertices are connected if the corresponding two edges have a common vertex.) However, we have to do some tinkering before we can define the terminals in the line graph. For each terminal v_i of G, add a new vertex v_i' and a new edge v_iv_i'. Let v_i' be the terminal instead of v_i. If edge v_iv_i' is marked as unremovable, then this modification does not change the solvability of the instance. Now the problem can be solved by using the vertex separation algorithms (Thereom 1 and 2) on the line graph $L(G)$. The terminals in the line graph are the vertices corresponding to the edges v_iv_i'. These edges were marked as unremovable, hence these vertices are contained in the set V^* of distinguished vertices in the line graph.

Theorem 3. *The edge deletion versions of* Minimum Terminal Separation *(with parameter k) and* Minimum Terminal Pair Separation *(with parameters k and ℓ) are fixed-parameter tractable.* □

3 Cutting up a Graph

Finding a good separator that splits a graph into two parts of approximately equal size is a useful algorithmic technique (see [9,10] for classic examples). This motivates the study of the following problem, where a given number of vertices has to be separated from the rest of the graph:

> Separating $\boldsymbol{\ell}$ Vertices
> *Input:* A graph $G(V, E)$, integers k and ℓ.
> *Parameter 1:* k
> *Parameter 2:* ℓ
> *Question:* Is there a partition $V = X \cup S \cup Y$ such that $|X| = \ell$, $|S| \leq k$ and there is no edge between X and Y?

It follows from [2] that the problem is NP-hard in general. Moreover, it is not difficult to show that the parameterized version of the problem is hard as well, even with both parameters:

Theorem 4. Separating ℓ Vertices *is* W[1]*-hard with parameters k and ℓ.*

Proof. The proof is by reduction from Maximum Clique. Let G be a graph with n vertices and m edges, it has to be determined whether G has a clique of size k. We construct G' as follows. In G' there are n vertices $v_1, \ldots, v_n$ that correspond to the vertices of G, these vertices form a clique in G'. Furthermore, G' has m vertices $e_1, \ldots, e_m$ that correspond to the edges of G. If the end points of edge e_j in G are vertices v_{j_1} and v_{j_2}, then connect vertex e_j with vertices v_{j_1} and v_{j_2} in G'. We set $\ell' = \binom{k}{2}$ and $k' = k$.

If there is a clique of size k, then we can cut ℓ' vertices by removing k' vertices. From $v_1, \ldots, v_n$ remove those k vertices that correspond to the clique. Now the $\binom{k}{2}$ vertices of G' that correspond to the edges of the clique are isolated vertices. On the other hand, assume that ℓ' vertices can be cut by deleting k' vertices. The remaining vertices of $v_1, \ldots, v_n$ form a clique of size greater than ℓ' (assuming $n > \binom{k}{2} + k$), hence the ℓ' separated vertices correspond to ℓ' edges of G. These vertices have to be isolated, since they cannot be connected to the large clique formed by the remaining v_i's. This means that the end vertices of the corresponding edges were all deleted. Therefore these $\ell' = \binom{k}{2}$ edges can have at most $k' = k$ end points, which is only possible if the end points induce a clique of size k in G. □

If we consider only bounded degree graphs, then Separating ℓ Vertices becomes fixed-parameter tractable:

Theorem 5. Separating ℓ Vertices *is fixed-parameter tractable with parameters k, ℓ, and d, where d is the maximum degree of the graph.*

Proof. Consider a solution $V = X \cup S \cup Y$, and consider the subgraph induced by $X \cup S$. This subgraph consists of some number of connected components, let $X_i \cup S_i$ be the vertex set of the ith component. For each i, the pair (S_i, X_i) has the following two properties:

(1) in graph G the set S_i separates X_i from the rest of the graph, and
(2) $X_i \cup S_i$ induces a connected graph.

On the other hand, assume that the pairs (X_1, S_1), …, (X_t, S_t) satisfy (1), (2), and the sets X_1, …, X_t, S_1, …, S_t are pairwise disjoint. In this case if $X = X_1 \cup \cdots \cup X_t$ has size exactly ℓ and $S = S_1 \cup \cdots \cup S_t$ has size at most k, then they form a solution. Therefore we generate all the pairs that satisfy these requirements, and use color coding to decide whether there are disjoint pairs with the required total size. If there is a solution, then this method will find one.

By requirement (2) a pair (X_i, S_i) induces a connected subgraph of size at most $k + \ell$. We enumerate each such connected subgraph. If a vertex v is contained in a connected subgraph of size at most $k + \ell$, then all the vertices of the subgraph are at a distance of less than $k + \ell$ from v. The maximum degree of the graph is d, thus there are at most $d^{k+\ell}$ vertices at distance less than $k+\ell$ from v. Therefore the number of connected subgraphs that contain v and have size at most $k + \ell$ is a constant, which means that there is a linear number of such subgraphs in the whole graph. We can enumerate these subgraphs in linear time. Each subgraph can be divided into a pair (X_i, S_i) in at most $2^{k+\ell}$ different ways. From these pairs we retain only those that satisfy requirement (1).

Having generated all the possible pairs (X_1, S_1), …, (X_p, S_p), a solution can be found as follows. We consider a random coloring of the vertices with $c := k + \ell$ colors. Using dynamic programming, we try to find a solution where every vertex of $X \cup S$ has a distinct color. Subproblem (C', j, k', ℓ') asks whether it is possible to select some pairs from the first j pairs such that (a) they are pairwise disjoint, (b) they use only vertices with color C', (c) the union of the S_i's has size k', and (d) the union of the X_i's has size ℓ'. For $j = 0$, the subproblems are trivial. If the subproblems for $j - 1$ are solved, then the problem can be solved for j using the following two recurrence relations. First, if subproblem $(C', j-1, k', \ell')$ is true, then clearly (C', j, k', ℓ') is true as well. Moreover, if every vertex of $X_j \cup S_j$ has distinct color (denote by C_j these colors), and subproblem $(C' \setminus C_j, j-1, k' - |S_j|, \ell' - |X_j|)$ is true, then a solution for this subproblem can be extended by the pair (X_j, S_j) to obtain a solution for (C', j, k', ℓ'). Using these two rules, all the subproblems can be solved.

If there is a solution $X \cup S$, then by probability at least $c!/c^c$ (where $c = k+\ell$ is the number of colors) these vertices receive distinct colors, and the algorithm described above finds a solution. Therefore if there is a solution, then on average we have to repeat the method $c^c/c!$ (constant) times to find a solution. The algorithm can be derandomized using the standard method of k-perfect hash functions, see [6, Section 8.3] and [1]. □

A variant of Separating ℓ Vertices is the Separating ℓ Connected Vertices problem where we also require that X induces a connected subgraph of G. This problem is fixed-parameter tractable:

Theorem 6. *The* Separating ℓ Connected Vertices *problems is fixed-parameter tractable with parameters* k *and* ℓ*.*

Proof. A vertex with degree at most $k+\ell$ will be called a *low degree* vertex, let G_0 be the subgraph induced by these vertices. A vertex v with degree more than $k+\ell$ cannot be part of X: at most k neighbors of v can be in S, hence v would have more than ℓ neighbors in X, which is impossible if $|X| = \ell$. Therefore X is a connected subgraph of G_0. As in the proof of Theorem 5, a bounded degree graph has a linear number of connected subgraphs of size ℓ. For each such subgraph, it has to be checked whether it can be separated from the rest of the graph by deleting at most k vertices. □

However, if only k is parameter, then the problem is W[1]-hard. This follows from the proof of Theorem 4. We construct the $n+m$ vertex graph as before, but instead of asking whether it is possible to separate $\binom{k}{2}$ vertices by deleting k vertices, we ask whether it is possible to separate $n+m-\binom{k}{2}-k$ connected vertices by deleting k vertices. The two questions have the same answer, thus

Theorem 7. Separating ℓ Connected Vertices *is* W[1]*-hard with parameter* k*.* □

Similarly, the problem is W[1]-hard if only ℓ is the parameter.

Theorem 8. Separating ℓ Connected Vertices *is* W[1]*-hard with parameter* ℓ*.*

Proof. The reduction is from Maximum Clique. It is not difficult to show that Maximum Clique remains W[1]-hard for regular graphs. Assume that we are given an r-regular graph G, and it has to be decided whether there is a clique of size k. If $r \le k^4$, then the problem is fixed parameter tractable: for every vertex v, we select $k-1$ neighbors of v in at most $\binom{k^4}{k-1}$ possible ways, and test whether these k vertices form a clique. Thus it will be assumed that $r > k^4$.

Consider the line graph $L(G)$ of G, i.e., the vertices of $L(G)$ correspond to the edges of G. Set $\ell = \binom{k}{2}$ and $k' = k(r-k+1)$. If G has a size k clique then the ℓ edges induced by the clique can be separated from the rest of the line graph: for each vertex of the clique, we have to delete the $r-k+1$ edges leaving the clique. On the other hand, assume that ℓ vertices of G' can be separated by deleting k vertices. The corresponding ℓ edges in G span a set T of vertices of size $t \le 2\ell$. We show that $t = k$, thus T is a clique of size k in G. Assume that $t > k$. Each vertex of T has at least $r-t+1$ edges that leave T. The corresponding $t(r-t+1)$ vertices have to be deleted from the line graph of G, hence $k' \ge t(r-t+1)$. However, this is not possible since

$$t(r-t+1)-k' = (t-k)r - t(t-1) + k(k-1) \ge (t-k)r - 4\ell^2 \ge r - k^4 > 0$$

(in the first inequality we use $4\ell^2 \ge t^2$, in the second $t > k$ and $\ell < k^2/2$). □

The vertex connectivity is the minimum number of vertices that has to be deleted to make the graph disconnected. Using network flow techniques, vertex connectivity can be determined in polynomial time. By essentially the same proof as in Theorem 4, we can show hardness for this problem as well:

> SEPARATING INTO ℓ COMPONENTS
> *Input:* A graph $G(V, E)$, integers k and ℓ
> *Parameter 1:* k
> *Parameter 2:* ℓ
> *Question:* Is there a set S of k vertices such that $G \setminus S$ has at least ℓ connected components?

Theorem 9. SEPARATING INTO ℓ COMPONENTS *is* W[1]*-hard with parameters k and ℓ.*

Proof. The construction is the same as in Theorem 4, but this time we set $\ell' = \binom{k}{2} + 1$ and $k' = k$. By deleting the vertices corresponding to a clique of size k the graph is separated into ℓ' components. The converse is also easy to see, the argument is the same as in Theorem 4. □

References

1. N. Alon, R. Yuster, and U. Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.
2. T. N. Bui and C. Jones. Finding good approximate vertex and edge partitions is NP-hard. *Inform. Process. Lett.*, 42(3):153–159, 1992.
3. W. H. Cunningham. The optimal multiterminal cut problem. In *Reliability of computer and communication networks (New Brunswick, NJ, 1989)*, volume 5 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 105–120. Amer. Math. Soc., Providence, RI, 1991.
4. E. Dahlhaus, D. S. Johnson, C. H. Papadimitriou, P. D. Seymour, and M. Yannakakis. The complexity of multiterminal cuts. *SIAM J. Comput.*, 23(4):864–894, 1994.
5. R. Downey, V. Estivill-Castro, M. Fellows, E. Prieto, and F. Rosamund. Cutting up is hard to do. In J. Harland, editor, *Electronic Notes in Theoretical Computer Science*, volume 78. Elsevier, 2003.
6. R. G. Downey and M. R. Fellows. *Parameterized complexity*. Monographs in Computer Science. Springer-Verlag, New York, 1999.
7. N. Garg, V. V. Vazirani, and M. Yannakakis. Primal-dual approximation algorithms for integral flow and multicut in trees. *Algorithmica*, 18(1):3–20, 1997.
8. N. Garg, V. V. Vazirani, and M. Yannakakis. Multiway cuts in node weighted graphs. *J. Algorithms*, 50(1):49–61, 2004.
9. R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM J. Appl. Math.*, 36(2):177–189, 1979.
10. R. J. Lipton and R. E. Tarjan. Applications of a planar separator theorem. *SIAM J. Comput.*, 9(3):615–627, 1980.