

Important separators and parameterized algorithms

Dániel Marx

Institut für Informatik, Humboldt-Universität zu Berlin, Germany
dmarx@cs.bme.hu

Abstract. The notion of “important separators” and bounding the number of such separators turned out to be a very useful technique in the design of fixed-parameter tractable algorithms for multi(way) cut problems. For example, the recent breakthrough result of Chen et al. [3] on the DIRECTED FEEDBACK VERTEX SET problem can be also explained using this notion. In my talk, I will overview combinatorial and algorithmic results that can be obtained by studying such separators.

1 Introduction

Problems related to cutting a graph into parts satisfying certain properties or separating different parts of the graph from each other form a classical area of graph theory and combinatorial optimization, with strong motivation coming from applications. The study of these problems revealed deep mathematical structures, such as connections to linear programming and semidefinite programming. In this talk, we explore an aspect of these problems that has been investigated and exploited only recently. It seems that understanding the extremal properties of small separators can be used to obtain combinatorial results and fixed-parameter tractability results. In particular, the notion of “important separators” has been used (implicitly or explicitly) in recent results on parameterized algorithms for separation and related problems [11, 15, 2, 12, 1, 10, 9].

An (X, Y) -separator is a set S of edges that separate X and Y for each other, that is, $G \setminus S$ has no component containing vertices from both X and Y (most of what we discuss here can be extended to vertex cutsets, but for simplicity we stick to edge cuts now). An (X, Y) -separator S is *inclusionwise minimal* if no subset $S' \subset S$ is an (X, Y) -separator. The main definition of the talk is the following:

Definition 1. *Let $X, Y \subseteq V(G)$ be vertices, $S \subseteq E(G)$ be an (X, Y) -separator, and let R be the set of vertices reachable from X in $G \setminus S$. We say that S is an important (X, Y) -separator if it is inclusionwise minimal and there is no (X, Y) -separator S' with $|S'| \leq |S|$ such that $R \subset R'$, where R' is the set of vertices reachable from X in $G \setminus S'$.*

Note that an important (X, Y) -separator is not necessarily an important (Y, X) -separator. Intuitively, we want to minimize the size of the (X, Y) -separator

and at the same time we want to maximize the set of vertices that remain reachable from X after removing the separator. The important separators are the separators that are Pareto-optimal with respect to these two objectives. Note that we do not want the number of vertices reachable from X to be maximal, we just want that this set of vertices is *inclusionwise maximal* (i.e., we have $R \subset R'$ and *not* $|R| < |R'|$ in the definition). The main observation of [11] is that the number of important (X, Y) -separators of size at most k can be bounded by a function of k ; a better bound is implicit in [2].

Theorem 2. [11, 2] *Let $X, Y \subseteq V(G)$ be two sets of vertices in graph G , let $k \geq 0$ be an integer, and let \mathcal{S}_k be the set of all (X, Y) -important separators of size at most k . Then $|\mathcal{S}_k| \leq 4^k$ and \mathcal{S}_k can be constructed in time $|\mathcal{S}_k| \cdot n^{O(1)}$.*

The following lemma clearly proves the bound in Theorem 2: if the sum is at most 1, then there cannot be more than 4^k important (X, Y) -separators of size at most k .

Lemma 3. [10] *Let $X, Y \subseteq V(G)$. If \mathcal{S} is the set of all important (X, Y) -separators, then $\sum_{S \in \mathcal{S}} 4^{-|S|} \leq 1$.*

As an application, we can prove the following surprisingly simple, but still non-trivial combinatorial result:

Lemma 4. *Let $X, Y \subseteq V(G)$. The union of all inclusionwise minimal (X, Y) -separators of size at most k contains at most $k \cdot 4^k$ edges incident to Y .*

2 Multiway Cut

Let G be a graph and T be a set of terminals. A *multiway cut* is a set S of edges such that every component of $G \setminus S$ contains at most one vertex of T .

<p>MULTIWAY CUT</p> <p>Input: Graph G, set T of vertices, integer k</p> <p>Find: A multiway cut S of size at most k</p>

The MULTIWAY CUT problem is known to be NP-hard already for $|T| = 3$ terminals [6] (for two terminals, it is the classical minimum $s - t$ cut problem, hence it is polynomial-time solvable). For every fixed k , the problem is polynomial-time solvable: using brute force, we can try all possible subsets of k edges in time $n^{O(k)}$. Of course, for moderately large values of n , such a solution seems to be practically useless already for very small values of k , say for $k = 10$. Can we do anything significantly smarter than complete enumeration of these subsets? The main goal of parameterized complexity is to design algorithms where the combinatorial explosion is restricted to a well-defined parameter (such as the size k of the solution we are looking for). Recall that a problem with a

parameter k is *fixed-parameter tractable* if it can be solved in time $f(k) \cdot n^{O(1)}$ for some function f depending only on the parameter k [7, 8, 13]. If $f(k)$ is “nice,” say, $f(k)$ is c^k for some small constant c , then such an algorithm can be useful for small values of k even for large n .

We show that MULTIWAY CUT is FPT parameterized by k . The following observation connects MULTIWAY CUT and the concept of important separators:

Lemma 5 (Pushing Lemma). *Let $t \in T$ be a terminal that is not separated from $T \setminus t$ in G . If G has a multiway cut S , then it also has a multiway cut S' with $|S'| \leq |S|$ that contains an important $(t, T \setminus t)$ -separator.*

Using this observation, we can solve the problem by branching on the choice of an important separator and including it into the solution:

Theorem 6. [2] MULTIWAY CUT can be solved in time $4^k \cdot n^{O(1)}$.

Proof. We solve the problem by a recursive branching algorithm. If all the terminals are separated from each other, then we are done. Otherwise, let $t \in T$ be a terminal not separated from the rest of the terminals. Let us use the algorithm of Theorem 2 to construct the set \mathcal{S}_k consisting of every important $(t, T \setminus t)$ -separator of size at most k . By Lemma 5, there is a solution that contains one of these separators. Therefore, we branch on the choice of one of these separators, and for every important separator $S' \in \mathcal{S}_k$, we recursively solve the MULTIWAY CUT instance $(G \setminus S', k - |S'|)$. If one of these branches returns a solution S , then clearly $S \cup S'$ is a multiway cut of size at most k in G .

The correctness of the algorithm is clear from Lemma 5. We claim that the search tree explored by the algorithm has at most 4^k leaves. We prove this by induction on k , thus let us assume that the statement is true for every value less than k . This means that we know that the recursive call $(G \setminus S', k - |S'|)$ explores a search tree with at most $4^{k - |S'|}$ leaves. Using Lemma 3, we can bound the number of leaves of the search tree by

$$\sum_{S' \in \mathcal{S}_k} 4^{k - |S'|} \leq 4^k \cdot \sum_{S' \in \mathcal{S}_k} 4^{-|S'|} \leq 4^k.$$

□

The running time can be improved from $4^k \cdot n^{O(1)}$ to $2^k \cdot n^{O(1)}$ with somewhat different techniques [15, 5].

A natural generalization of MULTIWAY CUT can be obtained if, instead of requiring that all the terminals are separated from each other, we require that a specified set of pairs are separated from each other:

MULTICUT

Input: Graph G , pairs $(s_1, t_1), \dots, (s_\ell, t_\ell)$, integer k

Find: A set S of at most k edges such that $G \setminus S$ has no $s_i - t_i$ path for any i

Theorem 6 implies that MULTICUT is FPT jointly parameterized by k and ℓ , that is, can be solved in time $f(k, \ell) \cdot n^{O(1)}$. We can guess how the solution S partitions the 2ℓ vertices s_i, t_i ($1 \leq i \leq \ell$), identify those vertices that are supposed to be in the same component of $G \setminus S$, and solve the resulting MULTIWAY CUT instance. It is a more challenging question whether the problem is FPT parameterized by k (the size of the solution) only. Very recently, a positive answer was given to this question:

Theorem 7. [1, 12] MULTICUT is FPT parameterized by k .

The proof in [12] introduces a new way of using important separators: with the “random sampling of important separators” technique we can significantly simplify the problem instance. This technique has found applications for other problems [4, 10] and it is very likely that it will be of use in the future.

3 Directed Graphs

Problems on directed graphs are notoriously more difficult than problems on undirected graphs. This is phenomenon has been observed equally often in the area of polynomial-time algorithms, approximability, and fixed-parameter tractability. Let us see if the techniques based on important separators survive the generalization to directed graphs. First, important separators can be defined analogously for directed graphs and the bound of 4^k of Lemma 2 still holds. This gives us some hope that we would be able to use the technique on directed graphs and in particular to show that DIRECTED MULTIWAY CUT (Delete a set of at most k edges such that there is no $t_1 \rightarrow t_2$ path in $G \setminus S$ for any two distinct $t_1, t_2 \in T$) is fixed-parameter tractable. However, the Pushing Lemma (Lemma 5) is not true on directed graphs. This means that a straightforward generalization of Theorem 6 to directed graphs is not possible. Nevertheless, Chitnis et al. [4] showed, using the random sampling technique of [12], that the problem is FPT:

Theorem 8. [4] DIRECTED MULTIWAY CUT is FPT.

What about the more general DIRECTED MULTICUT problem? In contrast to the undirected version, the directed problem is W[1]-hard parameterized by k [12]. But the problem can be interesting even for small values of ℓ . The case $\ell = 2$ can be reduced to DIRECTED MULTIWAY CUT in a simple way, thus Theorem 8 implies that DIRECTED MULTICUT for $\ell = 2$ is FPT parameterized by k . The case of a fixed $\ell \geq 3$ and the case of jointly parameterizing with ℓ and k are still open.

Chen et al. [3] considered the following (slightly unnatural) variant of DIRECTED MULTICUT:

SKEW MULTICUT

Input: Graph G , pairs $(s_1, t_1), \dots, (s_\ell, t_\ell)$, integer k

Find: A set S of at most k edges such that $G \setminus S$ has no $s_i \rightarrow t_j$ path for any $i \leq j$

For this problem, the Pushing Lemma can be made to work: there is a solution that contains an important $(s_1, \{t_1, \dots, t_\ell\})$ -separator. Therefore, arguments analogous to Theorem 6 give:

Theorem 9. [3] *SKEW MULTICUT is FPT parameterized by k .*

The reason why Chen et al. [3] considered this problem is that it formed an important ingredient in their proof showing that DIRECTED FEEDBACK VERTEX SET is FPT.

DIRECTED FEEDBACK VERTEX SET

Input: A directed graph G , integer k

Find: A set S of at most k vertices such that $G \setminus S$ has no directed cycle

Using the technique of iterative compression (introduced by Reed et al. [14]), Chen et al. [3] gave a nice reduction from DIRECTED FEEDBACK VERTEX SET to SKEW MULTICUT. Together with Theorem 9, this reduction established the fixed-parameter tractability of DIRECTED FEEDBACK VERTEX SET, resolving a longstanding open problem.

Theorem 10. [3] *DIRECTED FEEDBACK VERTEX SET is FPT parameterized by k .*

4 Conclusions

The notion of important separators seems to be useful for a wide range of combinatorial and algorithmic problems. In a particular application, first we need to observe that important separators are relevant (an example of this is the Pushing Lemma for MULTIWAY CUT) and then we can try to apply the upper bound of Theorem 2. The random sampling technique of [12] raises the applicability of important separators to a new level. After the initial application for MULTICUT, randomized sampling turned out to be useful for DIRECTED MULTIWAY CUT [4], and, in a very different context, for a clustering problem [10]. Based on the recent surge of results using important separators, one can safely expect that it will find further uses.

References

1. Bousquet, N., Daligault, J., Thomassé, S.: Multicut is FPT. In: Proceedings of the 43rd ACM Symposium on Theory of Computing. pp. 459–468 (2011)
2. Chen, J., Liu, Y., Lu, S.: An improved parameterized algorithm for the minimum node multiway cut problem. In: WADS. pp. 495–506 (2007)
3. Chen, J., Liu, Y., Lu, S., O’Sullivan, B., Razgon, I.: A fixed-parameter algorithm for the directed feedback vertex set problem. J. ACM 55(5) (2008)

4. Chitnis, R., Hajiaghayi, M., Marx, D.: Fixed-parameter tractability of directed multiway cut parameterized by the size of the cutset (2011), manuscript.
5. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.: On multiway cut parameterized above lower bounds, accepted to IPEC 2010.
6. Dahlhaus, E., Johnson, D.S., Papadimitriou, C.H., Seymour, P.D., Yannakakis, M.: The complexity of multiterminal cuts. *SIAM J. Comput.* 23(4), 864–894 (1994)
7. Downey, R.G., Fellows, M.R.: *Parameterized Complexity*. Monographs in Computer Science, Springer, New York (1999)
8. Flum, J., Grohe, M.: *Parameterized Complexity Theory*. Springer, Berlin (2006)
9. Heggenes, P., van 't Hof, P., Lokshtanov, D., Paul, C.: Obtaining a bipartite graph by contracting few edges. *CoRR* abs/1102.5441 (2011)
10. Lokshtanov, D., Marx, D.: Clustering with local restrictions. In: 38th International Colloquium on Automata, Languages and Programming, *Lecture Notes in Computer Science*, vol. 6755, pp. 785–797. Springer (2011)
11. Marx, D.: Parameterized graph separation problems. *Theoret. Comput. Sci.* 351(3), 394–406 (2006)
12. Marx, D., Razgon, I.: Fixed-parameter tractability of multicut parameterized by the size of the cutset. In: *Proceedings of the 43rd ACM Symposium on Theory of Computing*. pp. 469–478 (2011)
13. Niedermeier, R.: *Invitation to fixed-parameter algorithms*, *Oxford Lecture Series in Mathematics and its Applications*, vol. 31. Oxford University Press, Oxford (2006)
14. Reed, B., Smith, K., Vetta, A.: Finding odd cycle transversals. *Operations Research Letters* 32(4), 299–301 (2004)
15. Xiao, M.: Algorithms for multiterminal cuts. In: *CSR*. pp. 314–325 (2008)