Fixed-Parameter Tractability of Multicut Parameterized by the Size of the Cutset

Dániel Marx[*]
Institut für Informatik
Humboldt-Universität zu Berlin, Germany
dmarx@cs.bme.hu

Igor Razgon[†]
Department of Computer Science
University of Leicester, UK
ir45@mcs.le.ac.uk

ABSTRACT

Given an undirected graph G, a collection $\{(s_1,t_1),\ldots,(s_k,t_k)\}$ of pairs of vertices, and an integer p, the EDGE MULTICUT problem ask if there is a set S of at most p edges such that the removal of S disconnects every s_i from the corresponding t_i. VERTEX MULTICUT is the analogous problem where S is a set of at most p vertices. Our main result is that both problems can be solved in time $2^{O(p^3)} \cdot n^{O(1)}$, i.e., fixed-parameter tractable parameterized by the size p of the cutset in the solution. By contrast, it is unlikely that an algorithm with running time of the form $f(p) \cdot n^{O(1)}$ exists for the directed version of the problem, as we show it to be W[1]-hard parameterized by the size of the cutset.

Categories and Subject Descriptors

F.2 [**Theory of Computing**]: Analysis of Algorithms and Problem Complexity

General Terms

Algorithms

Keywords

multicut, fixed-parameter tractability

1. INTRODUCTION

From the classical results of Ford and Fulkerson on minimum $s-t$ cuts [16] to the more recent $O(\sqrt{\log n})$-approximation algorithms for sparsest cut problems [35, 1, 14], the study of cut and separation problems have a deep and rich theory. One well-studied problem in this area is the EDGE MULTICUT problem: given a

---

graph G and pairs of vertices (s_1,t_1), …, (s_k,t_k), remove a minimum set of edges such that every s_i is disconnected from its corresponding t_i for every $1 \leq i \leq k$. For $k = 1$, EDGE MULTICUT is the classical $s - t$ cut problem and can be solved in polynomial time. For $k = 2$, EDGE MULTICUT remains polynomial-time solvable [37], but it becomes NP-hard for every fixed $k \geq 3$ [11]. EDGE MULTICUT can be approximated within a factor of $O(\log k)$ in polynomial time [17] (even in the weighted case where the goal is to minimize the total weight of the removed edges). However, under the Unique Games Conjecture of Khot [24], no constant factor approximation is possible [7]. One can analogously define the VERTEX MULTICUT problem, where the task is to remove a minimum set of vertices. An easy reduction shows that the vertex version is more general than the edge version.

Using brute force, one can decide in time $n^{O(p)}$ if a solution of size at most p exists. Our main result is a more efficient exact algorithm for small values of p (the O^* notation hides factors that are polynomial in the input size):

THEOREM 1.1. *Given an instance of* VERTEX MULTICUT *or* EDGE MULTICUT *and an integer p, one can find in time $O^*(2^{O(p^3)})$ a solution of size p, if such a solution exists.*

That is, we prove that VERTEX MULTICUT and EDGE MULTICUT are fixed-parameter tractable parameterized by the size p of the solution, resolving a very challenging open question in the area of parameterized complexity[1]. (Recall that a problem is *fixed-parameter tractable* (FPT) with a particular parameter p if it can be solved in time $f(p) \cdot n^{O(1)}$, where f is an arbitrary function depending only on p; see [13, 15, 31] for more background). The question was first asked explicitly perhaps in [26]; it has been restated more recently as an open problem in e.g., [20, 8]. Our result shows in particular that multicut is polynomial-time solvable if the size of the optimum solution is $O(\sqrt[3]{\log n})$ (where n is the input size).

One reason why multicut is a fundamental problem is that it is able to express several other problems. It has been observed that a correlation clustering problem called FUZZY CLUSTER EDITING can be reduced to (and in fact, equivalent with) EDGE MULTICUT [3, 12, 2]. Our results show that FUZZY CLUSTER EDITING is FPT parameterized by the editing cost, settling this open problem discussed e.g., in [3].

Related results. The fixed-parameter tractability of multicut and related problems has been thoroughly investigated in the literature. EDGE MULTICUT is NP-hard on trees, but it is known to be FPT, parameterized by the maximum number p of edges that can be deleted, and admits a polynomial kernel [5, 21]. Multi-

---

[1]Independently of our work and using very different techniques Bousquet et al. [4] (see also their paper in this volume) also proved the fixed-parameter tractability of multicut.

cut problems were studied in [20] for certain restricted classes of graphs. For general graphs, VERTEX MULTICUT is FPT if both p and and the number of terminal pairs k are chosen as parameters (i.e, the problem can be solved in time $f(p,k) \cdot n^{O(1)}$ [27, 36, 19] for some function f). The algorithm of Theorem 1.1 is superior to these result in the sense that the running time depends polynomially on the number of terminals, and the exponential dependence is restricted to the parameter p. For the special case of MULTIWAY CUT (where terminals in a set T have to be pairwise separated form each other), algorithms with running time of the form $f(p) \cdot n^{O(1)}$ were already known [27, 8, 19], but apparently these algorithms do not generalize in an easy way to multicut. An FPT 2-approximation algorithm was given in [28] for EDGE MULTICUT: in time $O^*(2^{O(p \log p)})$, one can find a solution of size $2p$ if a solution of size p exists. There is no obvious FPT algorithm for the problem even on bounded-treewidth graphs, although one can obtain linear-time algorithms if the treewidth remains bounded after adding an edge $s_i t_i$ for each terminal pair [18, 32]. A PTAS is known for bounded-degree graphs of bounded treewidth [6].

Our techniques. The first two steps of our algorithm follows [28]. We start by an opening step that is standard in the design of FPT algorithms. Instead of solving the original VERTEX MULTICUT problem, we solve the compression version of the problem, where the input contains a solution W of size $p+1$, and the task is to find a solution of size p (if exists). A standard argument called *iterative compression* [34, 23] shows that if the compression problem is FPT, then the original problem is FPT. Alternatively, we can use the polynomial-time approximation algorithm of Gupta [22], which produces a solution W of size p^2 if a solution of size p exists. In this case, $O(p^2)$ iterations of the compression algorithm gives a solution of size p.

Next, as in [28], we try to reduce the compression problem to ALMOST 2SAT (delete k clauses to make a 2-CNF formula satisfiable; also known as 2CNF DELETION), which is known to be FPT [33]. However, our 2SAT formulation is very different from the one in [28]: we introduce a single variable x_v only for each vertex of G, while in [28] there is a variable $x_{v,w}$ for every $v \in V(G)$ and $w \in W$. This simpler reduction to ALMOST 2SAT is correct only if the instance satisfies two quite special properties:

(1) every component of $G \setminus W$ is adjacent to at most two vertices of W ("has at most two legs"), and

(2) there is a solution S such that every component of $G \setminus S$ contains a vertex of W ("no vertex is isolated from W after removing the solution").

The main part of the paper is devoted to achieving these properties. In order to achieve property (1), we show by an analysis of cuts and performing appropriate branchings that the set W can be extended in such a way that every component has at most two legs (Section 5). To achieve property (2), we describe a nontrivial way of sampling random subset of vertices such that if we remove this subset by a certain contraction operation (taking the torso of the graph), then without changing the solution, we get rid of the parts not reachable from W with some positive probability (Section 4). This random sampling uses the concept of "important separators," which was introduced in [27], and has been implicitly used in [9, 33, 8] in the design of parameterized algorithms. We consider the random selection of important separators the main new technical idea of the paper. Subsequently to the current paper, the technique was applied in the very different context of clustering [25].

Directed graphs. Having resolved the fixed-parameter tractability of VERTEX MULTICUT, the next obvious question is what happens on directed graphs. Note that for directed graphs, the edge

and vertex versions are equivalent. In directed graphs, multicut becomes much harder to approximate: there is no polynomial-time $2^{\log^{1-\varepsilon} n}$-approximation for any $\varepsilon > 0$, unless NP $\subseteq$ ZPP [10]. From the fixed-parameter tractability point of view, the directed version of the problem received particular attention because DIRECTED FEEDBACK VERTEX SET or DFVS (delete p vertices to make the graph acyclic) can be reduced to DIRECTED MULTICUT. The fixed-parameter tractability of DFVS had been a longstanding open question in the area of parameterized complexity until it was solved by Chen et al. [9] recently. The main idea that led to the solution is that DFVS can be reduced to a variant (in fact, special case) of DIRECTED MULTICUT called SKEW MULTICUT, where the task is to break every path from s_i to t_j for every $i > j$. By showing that SKEW MULTICUT if FPT parameterized by the size of the solution, Chen et al. [9] proved the fixed-parameter tractability of DFVS. We show that, unlike SKEW MULTICUT, the general DIRECTED MULTICUT problem is unlikely to be FPT (see the full version [29]).

THEOREM 1.2. DIRECTED MULTICUT *is* W[1]-*hard parameterized by the size p of the solution.*

Theorem 1.2 leaves open a number of interesting questions. Is DIRECTED MULTICUT FPT for a fixed (say $k = 2$ or $k = 3$) number of terminal pairs? Is it perhaps FPT parameterized by *both* the solution size and the number terminals (i.e., is there an $f(p,k) \cdot n^{O(1)}$ time algorithm)? Is the problem easier on acyclic graphs? Is the special case DIRECTED MULTIWAY CUT easier? The fixed-parameter tractability of SKEW MULTICUT suggests that it is not unreasonable to expect a positive answer to at least some of these questions. The study of approximation algorithms for cut problems uncovered deep mathematical connections. It is possible that the study of these problems from these problems from the viewpoint of parameterized complexity and understanding the extremal combinatorics of small cuts can will lead to further surprising connections.

2. PRELIMINARIES

Let G be an undirected graph and let $\mathbf{T} = \{(s_1,t_1), \ldots, (s_k,t_k)\}$ be a set of terminal pairs. We say that a set $S \subseteq V(G)$ of vertices is a *multicut* of $(G, \mathbf{T})$ if there is no component of $G \setminus S$ that contains both s_i and t_i for some $1 \le i \le k$ (note that it is allowed that S contains s_i or t_i). The central problem of the paper is the following:

VERTEX MULTICUT
Input: A graph G, an integer p, and a set $\mathbf{T}$ of pairs of vertices of G
Output: A multicut of $(G, \mathbf{T})$ of size at most p or "NO" if no such multicut exists.

2.1 Compression

The first step in the proof of Theorem 1.1 is a standard technique in the design of parameterized algorithm: we define and solve the *compression problem*, where it is assumed that the input contains a feasible solution of size larger than p. As this technique is standard (and in particular, we follow the approach of [28] for EDGE MULTICUT), we keep this section short and informal.

MULTICUT COMPRESSION
Input: A graph G, an integer p, a set $\mathbf{T}$ of pairs of vertices of G, and a multicut W of $(G, \mathbf{T})$
Output: A multicut of $(G, \mathbf{T})$ of size at most p, or "NO" if no such set S exists.

Our main technical contribution is showing that MULTICUT COMPRESSION is FPT parameterized by p and $|W|$.

LEMMA 2.1. MULTICUT COMPRESSION *can be solved in time* $O^*(2^{O((p+\log|W|)^3+|W|\log|W|)})$.

Intuitively, it is clear that proving Lemma 2.1 could be easier than solving VERTEX MULTICUT: the extra input W can give us useful structural information about the graph (and as $|W|$ appears in a running time, a large W is also helpful). What's not obvious is how solving MULTICUT COMPRESSION gives us any help in the solution of the original VERTEX MULTICUT problem. We sketch two methods.

Method 1. Let us use the polynomial-time approximation algorithm of Gupta [22] to find a multicut W of size at most $c \cdot \text{OPT}^2$, where c is a universal constant and OPT is the minimum size of a multicut. If $|W| \geq c \cdot p^2$, then we can safely answer "NO", as there is no multicut of size at most p. Otherwise, we run the algorithm of Lemma 2.1 for this set W to obtain a solution in time $O^*(2^{O((p+\log|W|)^3)}) = O^*(2^{O(p^3)})$.

Method 2. The standard technique of *iterative compression* [34, 23] allows us to reduce VERTEX MULTICUT to at most $|V(G)|$ instances of MULTICUT COMPRESSION with $|W| = p+1$. This technique was used for the 2-approximation of EDGE MULTICUT in [28] and its application is analogous in our case. Let $(G, \mathbf{T}, p)$ be an instance of VERTEX MULTICUT. Suppose that $V(G) = \{v_1, \ldots, v_n\}$ and let $G_i = G[\{v_1, \ldots, v_i\}]$. One by one, we consider the instances $(G_i, \mathbf{T}, p)$ in ascending order of i, and for each instance we find a solution S_i of size at most p. We start with $S_0 = \emptyset$. For some $i > 0$, we compute S_i provided that S_{i-1} is already known. Observe that $S_{i-1} \cup \{v_i\}$ is a multicut of size $p+1$ for $(G_i, \mathbf{T})$. Thus we can use the algorithm for MULTICUT COMPRESSION, which either returns a multicut S_i of $(G_i, \mathbf{T})$ having size at most p or returns "NO". In the first case, we can continue the iteration with $i+1$. In the second case, there is no multicut of size p for $(G_i, \mathbf{T})$ (as there is no such multicut even for $(G_i, \mathbf{T})$), and hence we can return "NO".

Both methods result in $O^*(2^{O(p^3)})$ time algorithms. However, we feel it important to mention both, as improvements in Lemma 2.1 might have different effects on the two methods.

It will be convenient to work with a slightly modified version of the compression problem. We say that a set $S \subseteq V(G)$ is a *multiway cut* of $W \subseteq V(G)$ if every component of $G \setminus S$ contains at most one vertex of W.

MULTICUT COMPRESSION*
Input: A graph G, an integer p, a set $\mathbf{T}$ of pairs of vertices of G, and a multicut W of $(G, \mathbf{T})$
Output: A set S of size at most p such that $S \cap W = \emptyset$, S is multicut of $(G, \mathbf{T})$ and a multiway cut of W or "NO" if no such set S exists.

In Sections 3–5, we prove the this problem is FPT:

LEMMA 2.2. MULTICUT COMPRESSION* *can be solved in time* $O^*(2^{O((p+\log|W|)^3)})$.

It is not difficult to reduce MULTICUT COMPRESSION to MULTICUT COMPRESSION* (an analogous reduction was done in [28] for the the edge case). We briefly sketch such a reduction. In order to solve an instance $(G, \mathbf{T}, W, p)$ of MULTICUT COMPRESSION, we first guess the intersection X of the multicut W given in the input and the solution S we are looking for. This guess results in at most $\sum_{i=1}^{p} \binom{|W|}{i}$ branches; in each branch, we remove the vertices of X from G and decrease p by $|X|$. Thus in the following, we can restrict our attention to solutions disjoint from W. Next, we branch on all possible partitions $(W_1, \ldots, W_t)$ of W, contract each W_i into a single vertex, and solve MULTICUT COMPRESSION* on the resulting instance $(G', \mathbf{T}', W', p')$. One of the partitions $(W_1, \ldots, W_t)$
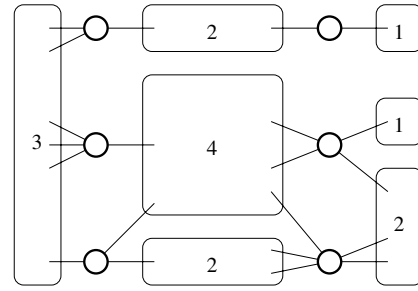

Figure 1: An instance with 7 components. The strong circles are the vertices of W, the numbers show the number of legs for each component.

corresponds to the way the solution S partitions W into connected components, and in this case S is a multiway cut of W' in G'. Thus if the original MULTICUT COMPRESSION instance has a solution S, then it is a solution of one of the constructed MULTICUT COMPRESSION* instances. Conversely, any solution of the constructed instances is a solution of the original instance. As the number of partitions of W can be bounded by $|W|^{O(|W|)}$, the running time claimed in Lemma 2.1 follows from Lemma 2.2. Thus proving Lemma 2.2 implies the main result Theorem 1.1.

2.2 Components and legs

Given an instance $(G, \mathbf{T}, W, p)$ of MULTICUT COMPRESSION*, we say that a component C of $G \setminus W$ has ℓ-*legs* if C is adjacent with ℓ vertices of W. We say that a component is *bipedal* if it has two legs. In Sections 3–4, we solve MULTICUT COMPRESSION* in the special case where every component has only one or two legs (we will call this special case BIPEDAL MULTICUT COMPRESSION*).

LEMMA 2.3. *The* BIPEDAL MULTICUT COMPRESSION* *problem can be solved in time* $O^*(2^{O((p+\log|W|)^3)})$.

Let $I = (G, \mathbf{T}, W, p)$ be an instance of the BIPEDAL MULTICUT COMPRESSION* problem, and let S be a solution for I. The *isolated part* of the solution is the set of vertices not reachable from any vertex of W in $G \setminus S$. We say that the solution S is *nonisolating* if the isolated part is empty, i.e., $G \setminus S$ has exactly $|W|$ components. In Section 3, we show that if the BIPEDAL MULTICUT COMPRESSION* instance has a nonisolating solution, then it can be found by a quite intuitive reduction to an FPT prolem ALMOST 2SAT. Next in Section 4, we present a randomized algorithm that modifies the instance such that if a solution exists, then it makes the solution nonisolating with positive probability. The algorithm is based on a randomized contraction of sets defined by "important separators"; we review this concept in Section 2.3. We complete the proof of Lemma 2.3 by derandomizing this algorithm.

Finally, in Section 5, we show how the general problem can be reduced to the bipedal case. The reduction is achieved by choosing an appropriate set B in a component with more than two legs and guessing, for each vertex $v \in B$, which vertex of W is reachable from v after removing the solution . Based on these guesses, we can identify each vertex of B with a vertex of W. We prove that if the set B is chosen appropriately, then after a bounded number of branchings, every component has one or two legs.

2.3 Important separators

The concept of important separators were introduced in [27] to deal with the multiway cut problem. If X is a set of vertices in

graph G, then we denote by $N_G(X)$ the neighborhood of X in G and define $\gamma_G(X) := |N_G(X)|$. We drop the subscript G if it is clear from the context.

Let G be an undirected graph and let $X, Y \subseteq V(G)$ be two disjoint sets. A set $S \subseteq V(G)$ of vertices is an $X - Y$ *separator* if S is disjoint from $X \cup Y$ and there is no component[2] K of $G \setminus S$ with $K \cap X \neq \emptyset$ and $K \cap Y \neq \emptyset$. To improve readability, we write $s - Y$ separator instead of $\{s\} - Y$ separator if s is a single vertex.

DEFINITION 2.4. *Let* $X, Y \subset V(G)$ *be disjoint sets of vertices,* $S \subseteq V(G)$ *be an* $X - Y$ *separator, and let* K *be the union of every component of* $G \setminus S$ *intersecting* X. *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* $K' \supset K$, *where* K' *is the union of every component of* $G \setminus S'$ *intersecting* X.

Note that the order of X and Y matters: an important $X - Y$ separator is not necessarily an important $Y - X$ separator.

We state without proof some properties of Definition 2.4 that are easy to see:

PROPOSITION 2.5. *Let* G *be a graph,* $X, Y \subseteq V(G)$ *be two disjoint sets of vertices, and* S *be an important* $X - Y$ *separator.*

1. *For every* $v \in S$, *the set* $S \setminus \{v\}$ *is an important* $X - Y$ *separator in* $G \setminus v$.

2. *If* S *is an* $X' - Y$ *separator for some* $X' \supset X$, *then* S *is an important* $X' - Y$ *separator.*

3. *If* $G[X]$ *is connected, then* S *is an important* $X' - Y$ *separator for every* $\emptyset \neq X' \subset X$.

4. *If* S *is an* $X - Y$ *separator in* G' *for some supergraph* G' *of* G, *then* S *is an important* $X - Y$ *separator in* G'.

The number of important separators were bound in [27] (although the notation there is slightly different). A better bound is implicit in [8].

LEMMA 2.6. *Let* $X, Y \subseteq V(G)$ *be disjoint sets of vertices in graph* G. *For every* $p \geq 0$, *there are at most* 4^p *important* $X - Y$ *separators of size at most* p. *Furthermore, we can enumerate all these separators in time* $O^*(4^p)$.

3. FINDING A NONISOLATING SOLUTION BY REDUCTION TO ALMOST 2SAT

The goal of this section is to show that we can solve BIPEDAL MULTICUT COMPRESSION* if there is at least one nonisolating solution. In Section 4, we show that it is sufficient to solve the problem under this assumption.

Let $x_1, \ldots, x_n$ be a set of variables; a literal is either a variable x_i or its negation $\bar{x}_i$. Recall that a 2CNF formula is a conjunction of clauses with at most two literals in each clause, e.g., $(\bar{x}_1 \vee x_2) \wedge (\bar{x}_3) \wedge (x_1 \vee \bar{x}_4)$. It is well-known that a satisfying assignment for a 2CNF formula can be found in linear time (if exists). However, it is NP-hard to find an assignment that maximizes the number of satisfied clauses, or equivalently, to find a minimum set of clauses whose removal makes the formula satisfiable. Razgon and O'Sullivan [33] gave an $O^*(15^k)$ time algorithm for the problem of deciding if a 2CNF formula can be made satisfiable by the deletion of at most k clauses; they call this problem ALMOST

---

[2]Throughout this paper, when we refer to a component K of a graph, we consider the set of vertices of this component. We omit saying "the set of vertices of" for the sake of brevity.

2SAT. We need a variant of the result here, where instead of deleting at most k clauses, we are allowed to delete at most k variables. An easy reduction gives an algorithm for this variant. If ϕ is a 2CNF formula and X is a set of variables, then we denote by $\phi \setminus X$ the formula obtained by removing every clause containing a literal of a variable in X.

THEOREM 3.1. *Given a* 2CNF *formula* ϕ *and an integer* k, *in time* $O^*(15^k)$ *we can either find a set* X *of at most* k *variables such that* $\phi \setminus X$ *is satisfiable, or correctly state that no such set* X *exists.*

It is not difficult to reduce finding a nonisolating solution to the problem solved by Theorem 3.1. For each vertex v of $G \setminus W$, we introduce a variable whose value expresses which leg of the component containing v is reachable from v. This formulation cannot express that a vertex is separated from both legs. However, as we assume that there is a nonisolating solution, we do not have to worry about such vertices.

LEMMA 3.2. *Let* $I = (G, \mathbf{T}, W, p)$ *be an instance of* BIPEDAL MULTICUT COMPRESSION* *that has a nonisolating solution of size at most* p. *In time* $O^*(15^p)$, *we can find a (not necessarily nonisolating) solution.*

PROOF. We encode the BIPEDAL MULTICUT COMPRESSION* instance $I = (G, \mathbf{T}, W, p)$ as a 2CNF formula ϕ the following way. For each component C of $G \setminus W$ having two legs, let $\ell_0(C)$ and $\ell_1(C)$ be the two legs. If component C has only one leg, then let $\ell_0(C)$ be this leg, and let $\ell_1(C)$ be undefined. For every vertex $v \in C$, let $\ell_0(v) = \ell_0(C)$ and $\ell_1(v) = \ell_1(C)$. We construct a formula ϕ whose variables correspond to $V(G) \setminus W$. The intended meaning of the variables is that v has value $b \in \{0, 1\}$ if v is in the same component as $\ell_b(v)$ after removing the solution. To enforce this interpretation, ϕ contains the following clauses:

- Group 1: $(u \to v)$, $(v \to u)$ for every adjacent $u, v \in V(G) \setminus W$.

- Group 2: If u is a neighbor of $\ell_b(u)$ for some $b \in \{0, 1\}$, then there is a clause $(u = b)$.

- Group 3: If $(u, v) \in \mathbf{T}$, $u, v \notin W$, and $\ell_{b_u}(u) = \ell_{b_v}(v)$ for some $b_u, b_v \in \{0, 1\}$, then there is a clause $(u \neq b_u \vee v \neq b_v)$ (e.g., if $\ell_0(u) = \ell_1(v)$, then the clause is $(u \vee \bar{v})$).

- Group 4: If $(u, v) \in \mathbf{T}$, $u \in W$, $v \notin W$, and $\ell_b(v) = u$ for some $b \in \{0, 1\}$, then there is a clause $(v \neq b)$.

This completes the description of ϕ. Note that no clause is introduced for pairs $(u, v) \in \mathbf{T}$ with $u, v \in W$, but these pairs are automatically separated by a solution that is a multiway cut of W. Furthermore, we can assume that W induces a independent set, otherwise there is no solution.

We show first that if I has a nonisolating solution S, then removing the corresponding variables of ϕ makes it satisfiable. As S is nonisolating and it is a multiway cut of W, every vertex of $G \setminus S$ is in the same component as exactly one of $\ell_0(v)$ and $\ell_1(v)$; let the value of variable v be b if vertex v is in the same component as $\ell_b(v)$. It is clear that this assignment satisfies the clauses in the first two groups. Consider a clause $(u \neq b_u \vee v \neq b_v)$ from the third group. This means that $(u, v) \in \mathbf{T}$ and $\ell_{b_u}(u) = \ell_{b_v}(v) = w \in W$. If this clause is not satisfied, then $u = b_u$ and $v = b_v$. By the way the assignment was defined, this is only possible if u is in the same component of $G \setminus S$ as $\ell_{b_u}(u) = w$ and v is in the same component of $G \setminus S$ as $\ell_{b_v}(v) = w$. Therefore, u and v are in the same component of $G \setminus S$, contradicting the assumption that S is a solution of I. Clauses in Group 4 can be checked similarly.

We have shown that ϕ can be made satisfiable by the deletion of p variables. By Theorem 3.1, we can find such a set S' of variables time $O^*(15^p)$. To complete the proof, we show that such a set S' corresponds to a (not necessarily nonisolating) solution of I. Let us show first that S' is a multiway cut of W. Suppose that there is a path P connecting $w_0, w_1 \in W$ in $G \setminus S'$. We can assume that the internal vertices of P are disjoint from W, i.e., they are in one component C of $G \setminus W$ with two legs. Thus there is a path P' from a neighbor v_0 of w_0 to a neighbor v_1 of w_1 in $C \setminus S'$. Suppose without loss of generality that $\ell_0(C) = w_0$ and $\ell_1(C) = w_1$. As the clauses in Group 1 are satisfied, every variable of P' has the same value. However, because of the clauses in Group 2, we have $x_{v_0} = 0$ and $x_{v_1} = 1$, a contradiction. Therefore, we can assume that S' is a multiway cut of W.

Suppose now that there is some $(u, v) \in \mathbf{T}$ such that $u, v \notin W$ are in the same component of $G \setminus S'$; let P be a $u - v$ path in $G \setminus S'$. As W is a multicut of $\mathbf{T}$, it is clear that P goes through at least one vertex of W. We have seen that S' is a multiway cut of W, thus P goes through exactly one vertex of W. Let $P = P_1 w P_2$ for some path P_1 that is fully contained in the component of $G \setminus W$ containing u and path P_2 fully contained in the component containing v. Let $b_u, b_v \in \{0, 1\}$ be such that $\ell_{b_u}(u) = \ell_{b_v}(v) = w$. Group 1 ensures that every variable of P_1 has the same value and Group 2 ensures that the last variable of P_1 has value b_u, thus $u = b_u$. A similar argument shows that $v = b_v$. However, this means that clause $(u \neq b_u \vee v \neq v_u)$ of Group 3 is not satisfied, a contradiction. Finally, a similar argument shows that the clauses in Group 4 ensure that pairs $(u, v) \in \mathbf{T}$ with $u \in W$, $v \notin W$ are separated. $\quad\square$

4. MAKING THE SOLUTION NONISOLATING

In this section, we present a randomized transformation that, given an instances of BIPEDAL MULTICUT COMPRESSION* having a solution, it modifies the instance in such a way that the new instance has a *nonisolating* solution with probability $2^{-O(p^3)}$. The main result of this section is a derandomized version of this transformation:

LEMMA 4.1. *Given an instance I of the* BIPEDAL MULTICUT COMPRESSION* *problem, we can construct in time $O^*(2^{O(p^3)})$ a set of $t = 2^{O(p^3)} \log n$ instances $I_1, \ldots, I_t$ such that*

1. *If I has no solution, then I_i has no solution for any $1 \leq i \leq t$.*

2. *If I has a solution, then I_i has a* nonisolating *solution for at least one $1 \leq i \leq t$.*

Thus we can solve BIPEDAL MULTICUT COMPRESSION* by constructing the instances $I_1, \ldots, I_t$ of Lemma 4.1 and applying the algorithm of Lemma 3.2 to each instance. This will prove Lemma 2.3.

4.1 Torsos and nonisolating solutions

The randomized transformation can be conveniently described using the operation of taking the *torso* of a graph.

DEFINITION 4.2. *Let G be a graph and $C \subseteq V(G)$. The graph* $\mathrm{torso}(G, C)$ *has vertex set C and two vertices $a, b \in C$ are adjacent if $\{a, b\} \in E(G)$ or there is a path P in G connecting a and b whose internal vertices are not in C.*

It is easy to show that this operation preserves separation inside C:

PROPOSITION 4.3. *Let $C \subseteq V(G)$ be a set of vertices in G and let $a, b \in C$ two vertices. A set $S \subseteq C$ separates vertices a and b in* $\mathrm{torso}(G, C)$ *if and only if S separates these vertices in G.*

PROOF. Let P be a path connecting a and b in G and suppose that P is disjoint from the set S. The path P contains vertices from C and from $V(G) \setminus C$. If $u, v \in C$ are two vertices such that every vertex of P between u and v is from $V(G) \setminus C$, then by definition there is an edge uv in $\mathrm{torso}(G, C)$. Using these edges, we can modify P to obtain a path P' that connects a and b in $\mathrm{torso}(G, C)$ and avoids S.

Conversely, suppose that P is a path connecting a and b in the graph $\mathrm{torso}(G, C)$ and it avoids $S \subseteq C$. If P uses an edge uv that is not present in G, then this means that there is a path connecting u and v whose internal vertices are not in C. Using these paths, we can modify P to obtain a path P' that uses only the edges of G. Since $S \subseteq C$, the new vertices on the path are not in S, i.e., P' avoids S as well. $\quad\square$

Let $I = (G, W, \mathbf{T}, p)$ be an arbitrary instance of BIPEDAL MULTICUT COMPRESSION*. Given a set $Z \subseteq V(G) \setminus W$ of vertices, the *reduced instance $I/Z = (G', W, \mathbf{T}', p)$* is defined the following way:

1. The graph G' is $\mathrm{torso}(G, V(G) \setminus Z)$.

2. For every $v \in V(G)$, let $\phi(v) = N_G(C)$ if v belongs to component C of $G[Z]$, and let $\phi(v) = \{v\}$ if $v \notin Z$. The set $\mathbf{T}'$ is obtained by by replacing every pair $(x, y) \in \mathbf{T}$ with the set of pairs $\{(x', y') \mid x' \in \phi(x), y' \in \phi(y)\}$.

The main observation is that if we perform this torso operation for a Z that is sufficiently large to cover the isolated part of a hypothetical solution S and sufficiently small to be disjoint from S, then S becomes an nonisolating solution of I/Z. Furthermore, the torso operation is "safe" in the sense that it does not create new solutions.

LEMMA 4.4. *Let $I = (G, \mathbf{T}, W, p)$ be an instance of* BIPEDAL MULTICUT COMPRESSION* *and let $Z \subseteq V(G) \setminus W$ be a set of vertices. If I has no solution, then I/Z has no solution either. Furthermore, if I has a solution S such that Z covers the isolated part and $Z \cap S = \emptyset$, then S is a nonisolating solution of I/Z.*

PROOF. Let G and G' be the graphs in instances I and I/Z, respectively. To prove the first statement, we show that if $S' \subseteq V(G')$ is a solution of I/Z, then S' is a solution of I as well. Suppose that some pair (x, y) of I is not separated by S'. Let P be a path in $G \setminus S'$ going from x to y. Let x' and y' be the first and last vertex of P not in Z, respectively, and let P' be the subpath of P from x' to y'. (Note that P cannot be fully contained in Z, as it contains at least one vertex of W.) By the way I/Z is defined, (x', y') is a pair in I/Z, hence S' separates x' and y' in $G' = \mathrm{torso}(G, C)$. Using Prop. 4.3 with $C = V(G) \setminus Z$, we get that S' separates x' and y' in G, which is in contradiction with the existence of the path P. A similar argument shows that there is no path in $G \setminus S'$ that connects two vertices of W.

For the second statement, suppose that S is a solution of I with $S \cap Z = \emptyset$. Let us show that S is a solution of I/Z as well. Suppose that S does not separate x' and y' in G' for some pair (x', y') of I/Z. Using Prop. 4.3 with $C = V(G) \setminus Z$, we get that S does not separate x' and y' in G, i.e., there is a $x' - y'$ path P in $G \setminus S$. By the way the pairs in I/Z were defined, there is a pair (x, y) of I and there is an $x - x'$ path P_1 such that x' is the only vertex of P_1 not in Z, and there is a $y - y'$ path P_2 such that y' is the only vertex of P_2 not in Z. Clearly, these paths are disjoint form S. Therefore, the concatenation of P_1, P, P_2 is an $x - y$ path in $G \setminus S$, contradicting that S is a solution of I.

To see that S is nonisolating in G', consider a vertex v of $G' \setminus S$. As $v \notin Z$ is not in the isolated part of the solution S of I, there is a path P in $G \setminus S$ going from v to a vertex $w \in W$. Again by Prop. 4.3, this means that there is a $v - w$ path in $G' \setminus S$ as well, which means that v is not in the isolated part of the solution S of I'. $\quad\square$
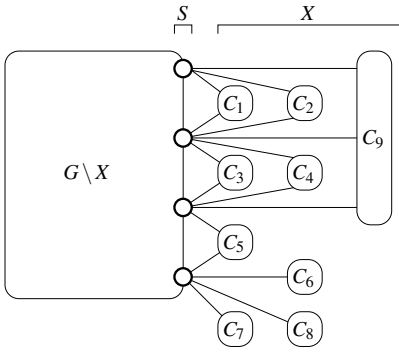
Figure 2: A solution S where the isolated part X consists of 9 important components (the components of $G \setminus X$ and the set W are not shown in the figure). The isolated part is the disjoint union of 5 important clusters: $C_1 \cup C_2, C_3 \cup C_4, C_5, C_6 \cup C_7 \cup C_8,$ **and** $C_9.$

4.2 Important components and clusters

In light of Lemma 4.4, what we need to do is to guess a set Z that covers the isolated part of a solution. Lemma 4.7 below allows us to restrict our attention to very special sets Z whose boundary is formed from important separators.

DEFINITION 4.5. *A set $C \subseteq V(G)$ is an* important component *if $G[C]$ is connected and $N_G(C)$ is an important $C - W$ separator of size at most p. For every $S \subseteq V(G) \setminus W$, the* important cluster L_S *is the (disjoint) union of every important component with $N(C) = S$.*

Observe that every important component is contained in exactly one important cluster, i.e, the important clusters form a partition of the important components. Every important $C - W$ separator is an important $v - W$ separator for every $v \in C$ (Prop. 2.5(3)). Thus Lemma 2.6 gives a bound on the number of important components that can contain a vertex v.

PROPOSITION 4.6. *Every vertex $v \in V(G) \setminus W$ is contained in at most 4^p important components and important clusters. Furthermore, all the important components and clusters can be enumerated in time $O^*(4^p)$.*

Note that the total number of important components and clusters cannot be bounded by a function of p: for example, it may happen that almost every vertex of $V(G) \setminus W$ forms an important component of size 1.

In the following lemma, we show that there is a solution where every component of the isolated part is an important component. We cannot bound the number of these important components by a function of p, but they can be partitioned into at most 2^p isolated clusters (see Figure 2). This is the reason why we are selecting important clusters instead of important components in Section 4.3.

LEMMA 4.7. *If instance $I = (G,W,\mathbf{T},p)$ of* BIPEDAL MULTI-CUT COMPRESSION* *has a solution, then it has an inclusionwise minimal solution where the isolated part is the disjoint union of at most 2^p important clusters.*

PROOF. Let S be a solution of I with isolated part R. Recall that $\gamma(R)$ is defined as $|N(R)|$. Let us choose S such that $\gamma(R)$ is minimum possible, and among such solutions, $|R|$ (or equivalently, $|R| + \gamma(R)$) is maximum possible. Let $C_1, \dots, C_s$ be the components

of $G[R]$. We show that every C_i is an important component. If this is true, then these components can be classified into at most $2^{|S|} \leq 2^p$ disjoint groups according to the neighborhood $N(C_i) \subseteq S$. Therefore, each group can be covered by an important cluster. Observe that if C and C' are two important components with $N(C) = N(C')$ and C is in the isolated part, then C' has to be in the isolated part as well (in particular, the minimality of S implies that C' cannot contain a vertex of S). Therefore, the union of the at most 2^p important clusters covering the groups is exactly the isolated part, and we are done.

Suppose that some C_i is not important: in this case, there is an important component $C'_i \supset C_i$ such that $\gamma(C'_i) \leq \gamma(C_i)$. Let $S' := (S \setminus N(C_i)) \cup N(C'_i)$; it is clear that $|S'| \leq |S|$. We claim that S' is also a solution of instance I. For this purpose, we first show that every path P connecting a vertex $v \in R \cup S$ with a vertex of W has to go through S'. Indeed, path P has to go through a vertex of S by the definition of R. Thus P can be disjoint from S' and go through S only if it contains a vertex from $N(C_i) \setminus N(C'_i) \subseteq C'_i$. However, $N(C'_i) \subseteq S'$ separates every vertex of C'_i from W, contradicting the assumption that P is disjoint from S'.

To show that S' is a solution of I', suppose that some $x,y \in W$ or some pair (x,y) is not separated by S'; let P be a $x - y$ path in $G \setminus S'$. Path P has to go through S and a vertex of W, thus by the previous claim, P goes through S'. This means that S' is a solution of I with $|S'| \leq |S|$ and therefore by the minimality of the solution S, we have $|S'| = |S|$. Note that $N(C_i) \neq N(C'_i)$, hence $|S'| = |S|$ is only possible if $S \neq S'$.

Let R' be the isolated part in solution S'. Again by the previous claim, every vertex of $R \cup S$ is either in S' or separated from W in $G \setminus S'$, thus $R \cup S \subseteq R' \cup S'$. Suppose first that $R \cup S = R' \cup S'$. The set S contains exactly those vertices of $R \cup S$ that have a neighbor outside $R \cup S$ (by the minimalty of S). The set S' has to contain every such vertex (to separate R' and W), thus $S \subseteq S'$. Howerver, we have seen that $|S| = |S'|$ and $S \neq S'$, a contradiction. Therefore, $R \cup S \subset R' \cup S'$, and $|S'| = |S|$ implies $|R'| > |R|$. This contradicts the maximality of S with respect to the size of the isolated part. ☐

Important separators that induce cliques are nested, hence we can get a bound of p instead of 4^p for the number of such separators. Lemma 4.10 uses this result to improve the probabilities in the randomized selection.

LEMMA 4.8. *Every vertex $v \in V(G) \setminus W$ is contained in at most p important clusters X where $N(X)$ is a clique.*

PROOF. Assume the opposite. We first show that if X_1 and X_2 are important components containing v such that $N(X_1)$ and $N(X_2)$ are cliques, then either $X_1 \subseteq X_2$ or $X_2 \subseteq X_1$. If $X_1 \setminus X_2 \neq \emptyset$ and X_1 is connected, then there is a vertex $x_1 \in X_1 \cap N(X_2)$. As $N(X_2)$ is a clique, every vertex of $N(X_2)$ is adjacent with x_1, implying that $N(X_2) \subseteq X_1 \cup N(X_1)$. If $X_2 \setminus X_1 \neq \emptyset$, then a symmetrical argument shows that $N(X_1) \subseteq X_2 \cup N(X_2)$. We claim that $N(X_1 \cup X_2) \subseteq N(X_1) \cap N(X_2)$ and hence $\gamma(X_1 \cup X_2) \leq \gamma(X_1), \gamma(X_2)$; as $X_1 \cup X_2 \supset X_1, X_2$, this would contradict the assumption that X_1 and X_2 are important components. Consider a vertex $u \in N(X_1 \cup X_2)$, which must have a neighbor $w \in X_1 \cup X_2$. If $w \in X_1 \cap X_2$, then $u \in N(X_1) \cap N(X_2)$ and we are done. Suppose without loss of generality that $w \in X_1 \setminus X_2$. Then $u \in N(X_1) \subseteq X_2 \cup N(X_2)$, but $u \notin X_2$ by definition, hence u has to be in $N(X_2)$ as well.

We have shown that the important components containing v whose boundaries are cliques form a chain. This means that there are at most p of them, as the boundary sizes must be different. Using that every important component is contained in exactly one important cluster, we get the bound on the number of important clusters. ☐

4.3 Randomized selection of sets

By Lemmas 4.4 and 4.7, we need to construct a set Z that is the union of at most 2^p important clusters. As the number of important clusters cannot be bounded by a function of p, we cannot try all possibilities. Instead of complete enumeration, we randomly select important clusters (possibly much more than 2^p) and let Z be their union. What we need is that Z is sufficiently large to cover the at most 2^p important clusters of the isolated part given by Lemma 4.7, but Z is sufficiently small to be disjoint from S. We want to set the probabilities such that the probability of this event can be bounded from below by a function of p. First we present a simpler version of the proof, where the probability of success is double exponentially small in p (Lemma 4.9). This simpler proof highlights the main idea of the randomized reduction. In Lemma 4.10, we improve the probability to $2^{-O(p^3)}$.

LEMMA 4.9. *There is a randomized algorithm that, given an instance $I = (G, W, \mathbf{T}, p)$ of* BIPEDAL MULTICUT COMPRESSION*, in time $O^*(4^p)$ constructs a set $Z \subseteq V(G) \setminus W$ such that if I has a solution, then I/Z has a nonisolating solution with probability at least $2^{-2^{O(p)}}$.*

PROOF. By Lemma 4.7, there is a solution such that the isolated part is the disjoint union of at most 2^p important clusters. Let us fix such a hypothetical solution S, and suppose that the isolated part is the disjoint union of important clusters $L_1, \ldots, L_s$. Let $\mathcal{X}$ be the set of all important clusters in G. Let $\mathcal{X}'$ be a subset of $\mathcal{X}$ where every $X \in \mathcal{X}$ appears with probability 4^{-p} independently at random. Let Z be the union of the sets in $\mathcal{X}'$. If the events

(E1) $Z \cap S = \emptyset$, and

(E2) $L_j \subseteq Z$ for every $1 \le j \le s$

hold, then by Lemma 4.4, I/Z has a nonisolating solution, and we are done.

Let us estimate the probability that both (E1) and (E2) hold. Let $A = \{L_1, \ldots, L_s\}$ and let $B = \{X \in \mathcal{X} \mid X \cap S \ne \emptyset\}$; we have $|A| \le 2^p$ and $|B| \le p \cdot 4^p$ (by Lemma 4.8, every vertex of S is contained in at most 4^p important clusters). If no member of B is selected, then no set of $\mathcal{X}'$ contains a vertex of S, and hence $Z \cap S = \emptyset$. If every member of A is selected, then every L_j is a subset of Z. Therefore, the probability that (E1) and (E2) hold can be bounded from below by the probability of the event that every member of A is selected and no member of B is selected. As A and B are disjoint, this probability is at least

$$(4^{-p})^{2^p} \cdot (1 - 4^{-p})^{p \cdot 4^p} \ge 4^{-p \cdot 2^p} \cdot e^{-2p} = 2^{-2^{O(p)}}$$

(in the inequality, we use that $1 + x \ge \exp(x/(1+x))$ for every $x > -1$ and $1 - 4^{-p} \ge 1/2$). □

In order to optimize the success probability, we do the randomized selection of important components in two phases: first we select some important components and add new edges to the graph and in the second phase we restrict our attention to important clusters whose boundaries are cliques.

LEMMA 4.10. *There is a randomized algorithm that, given an instance $I = (G, W, \mathbf{T}, p)$ of* BIPEDAL MULTICUT COMPRESSION*, in time $O^*(4^p)$ constructs a set $Z \subseteq V(G) \setminus W$ such that if I has a solution, then I/Z has a nonisolating solution with probability at least $2^{-O(p^3)}$.*

PROOF. The randomized algorithm consists of two phases. By Lemma 4.7, there is a solution S such that the isolated part is the

disjoint union of at most 2^p important clusters; let us fix such a hypothetical solution S and let R be the isolated part.

Let us consider those pairs $x, y \in S$ for which there is an $x - y$ path with internal vertices in the isolated part R. For every such pair $x, y \in S$, select a component C of $G[R]$ with $x, y \in N_G(C)$; let $C_1, \ldots, C_q$ with $q \le \binom{|S|}{2} \le \binom{p}{2}$ be the selected components. Note that every C_j is an important component.

Phase 1. Let $\mathcal{C}$ be the set of all important components; Prop. 4.6 states that these sets can be enumerated. In the first phase, we select a subset $\mathcal{C}' \subseteq \mathcal{C}$ by putting every $C \in \mathcal{C}$ into $\mathcal{C}'$ with probability $p_1 = 4^{-p}$ independently at random. Then for every component $C \in \mathcal{C}'$, we make $N_G(C)$ a clique; let G' be the graph obtained this way. Let us estimate the probability that the events

(E1) every $C \in \mathcal{C}'$ is disjoint from S,

(E2) $G[R]$ and $G'[R]$ have the same connected components (as vertex sets), and

(E3) $G'[N_{G'}(C_i)]$ is a clique for every $1 \le i \le q$

hold. Let $A_1 = \{C_1, \ldots, C_q\}$ and let $B_1 = \{C \in \mathcal{C} \mid S \cap C \ne \emptyset\}$. Clearly, $|A_1| \le p^2$ and $|B_1| \le |S| \cdot 4^p \le p \cdot 4^p$ (by Prop. 4.6, every vertex of S can be contained in at most 4^p members of $\mathcal{C}$). If no set of B_1 is selected, then (E1) holds. If (E1) holds, then we have $N_G(K) = N_{G'}(K)$ for every component K of $G[R]$. Indeed, the boundary of K can change only if some $x \in K$ and $y \notin K \cup N_G[K]$ both appear in $N_G(C)$ for some $C \in \mathcal{C}'$, but such a connected C would have to contain a vertex of $N_G(K) \subseteq S$ as well, contradicting (E1). Therefore, if (E1) holds, then (E2) holds as well and in particular $N_{G'}(C_i) = N_G(C_i)$ for every i. Therefore, assuming that (E1) holds and every member of A_1 is in $\mathcal{C}'$, $N_G(C_i) = N_{G'}(C_i)$ becomes a clique in G' for every $1 \le i \le s$. Thus the probability that (E1–E3) hold can be bounded from below by the probability of the event that every set in A_1 is selected and no set from B_1 is selected. As the sets A_1 and B_1 are disjoint, this probability is at least $(1 - 4^{-p})^{p \cdot 4^p} \cdot (4^{-p})^{p^2} \ge e^{-2p} \cdot 4^{-p^3}$.

In the following, we assume that (E1–E3) hold. By assumption, the isolated part of S is the disjoint union of important clusters $L_1, \ldots, L_s$ of G with $s \le 2^p$. We claim that every L_j is an important cluster of G'. Consider a component K of $G[L_j]$. By (E2), K is a component of $G'[R]$. Furthermore, K is an important component of G': by Prop. 2.5(4), $N_G(K) = N_{G'}(K)$ remains an important $K - W$ separator in G'. Thus every component of L_j is an important component of G', which means that G' has an important cluster $L'_j \supseteq L_j$. This cluster L'_j is fully contained in the isolated part R: $N_G(L'_j) \subseteq S$ and the minimality of S implies that L'_j cannot contain a vertex of S. By (E2), $G[R]$ and $G'[R']$ have the same components, thus $L'_j \subseteq R$ contains the same components as L_j.

Phase 2. Let $\mathcal{X}$ be the set of all important clusters X in G' for which $G'[N_{G'}(X)]$ is a clique. We have seen that (E1–E3) implies that $\mathcal{X}$ contains every L_j. Let $\mathcal{X}'$ be a subset of $\mathcal{X}$ where every $X \in \mathcal{X}$ appears with probability $p_2 = 1 - 2^{-p}$ independently at random. Let Z be the union of the sets in $\mathcal{X}'$. If the events

(E4) $Z \cap S = \emptyset$, and

(E5) $L_j \subseteq Z$ for every $1 \le j \le s$

hold, then by Lemma 4.4, I/Z has a nonisolating solution, and we are done.

Let us estimate the probability that both (E4) and (E5) hold. Let $A_2 = \{L_1, \ldots, L_s\}$ and let $B_2 = \{X \in \mathcal{X} \mid X \cap S \ne \emptyset\}$; we have $|A_2| \le 2^p$ and $|B_2| \le p^2$ (by Lemma 4.8, every vertex of S is contained in at most p important clusters whose boundary is a clique). If no

member of B_2 is selected, then no set of $\mathcal{X}'$ contains a vertex of S, and hence $Z \cap S = \emptyset$. If every member of A_2 is selected, then every L_j is a subset of Z. Therefore, the probability that (E4) and (E5) hold can be bounded from below by the probability of the event that every member of A_2 is selected and no member of B_2 is selected, which is at least $(2^{-p})^{p^2} \cdot (1 - 2^{-p})^{2^p} \geq 2^{-p^3} \cdot e^{-2}$.

Taking into account the probability of success in both phases, we get that I/Z has a nonisolating solution with probability $2^{-O(p^3)}$. As the number of important components is at most $4^p |V(G)|$, the running time is $O^*(4^p)$. $\quad\square$

4.4 Derandomization

By running $2^{O(p^3)}$ times the algorithm of Lemma 4.10, we get a collection of instances that satisfy the requirements of Lemma 4.1 with arbitrary large constant probability. We can derandomize the algorithm of Lemma 4.10 using the standard technique of splitters. Recall that an (n, r, r^2)-*splitter* is a family of functions from $[n]$ to $[r^2]$ such that for any subset $X \subseteq [n]$ with $|X| = r$, one of the functions in the family is injective on X. Naor, Schulman, and Srinivasan [30] gave an explicit construction of an (n, r, r^2)-splitter of size $O(r^6 \log r \log n)$.

Observe that in the first phase of the algorithm of Lemma 4.10, a random subset of a universe $\mathcal{C}$ of size $n_1 = |\mathcal{C}| \leq 4^p \cdot n$ is selected. There is a collection $A_1 \subseteq \mathcal{C}$ of $a_1 \leq p^2$ sets and a collection $B_1 \subseteq \mathcal{C}$ of $b_1 \leq p \cdot 4^p$ sets such that if every set in A_1 is selected and no set in B_1 is selected, then (E1–E3) hold. Instead of the selecting a random subset, we try every function f in an $(n_1, a_1 + b_1, (a_1 + b_1)^2)$-splitter family and every subset $F \subseteq [(a_1 + b_1)^2]$ of size a_1 (there are $\binom{(a_1+b_1)^2}{a_1} = 2^{O(p^3)}$ such sets F). For a particular choice of f and F, we select those sets $C \in \mathcal{C}$ for which $f(C) \in F$. By the definition of the splitter, there will be a function f that is injective on $A_1 \cup B_1$, and there is a subset F such that $f(C) \in F$ for every A_1 and $f(C) \notin F$ for every B_1. For such an f and F, the selection will ensure that (E1–E3) hold.

In the second phase, we select a random subset of universe $\mathcal{X}$ of size $n_2 \leq pn$, and there is a collection $A_2 \subseteq \mathcal{X}$ of size $a_2 \leq 2^p$ and a collection $B_2 \subseteq \mathcal{X}$ of size $b_2 \leq p^2$ such that if every set in A_2 is selected and no set in B_2 is selected, then (E4) and (E5) hold. As in the first phase, we can replace this random choice by enumerating the functions of an $(n_2, a_2 + b_2, (a_2 + b_2)^2)$-splitter and every subset $\bar{F} \subseteq [(a_2 + b_2)^2]$ of size b_2 (there are $\binom{(a_2+b_2)^2}{b_2} = 2^{O(p^3)}$ such sets $\bar{F}$). This time, we select a set $X \in \mathcal{X}$ if $f(X)$ is *not* in $\bar{F}$ and it is clear that there is an f and $\bar{F}$ for which (E4) and (E5) hold.

Let us bound the number of branches of the algorithm. In both phases, the size of the splitter family is $2^{O(p)} \cdot \log n$ and the there are $2^{O(p^3)}$ possible F. (Note that the splitter family can be constructed in time polynomial in the size of the family.) Thus the algorithm produces $2^{O(p^3)} \cdot \log n$ instances, proving Lemma 4.1.

5. REDUCTION TO THE BIPEDAL CASE

Let $(G, \mathbf{T}, W, p)$ be an instance of the MULTICUT COMPRESSION* problem. Let us call a component of $G \setminus W$ having at least two legs a *non-trivial component* of G w.r.t. W (when the context is clear, we will just refer to a non-trivial component). As the solution of MULTICUT COMPRESSION* has to be a set S that is disjoint from W and a multiway cut of W, the number of non-trivial components is a lower bound on the size of the solution.

We present an algorithm that solves the given instance of the MULTICUT COMPRESSION* problem either by applying the algorithm for the BIPEDAL MULTICUT COMPRESSION* problem (in case every component has at most two legs) or by recursive appli-

cation to a set of instances whose number is bounded by a function of p and such that in each instance either the parameter is decreased or the number of non-trivial components is increased.

The main idea for the branching is the following. Let B be a set of vertices in $G \setminus W$ and let S be a hypothetical solution for MULTICUT COMPRESSION*. We try to guess what happens to each vertex of B in the solution S. It is possible that a vertex $v \in B$ is in S; in this case, we delete v from the instance and reduce the parameter. If v is separated away from W, we argue that it can be assumed that the set separating v from W is an important separator. We guess this important separator, remove it from the graph, and decrease the parameter appropriately. Finally, if v is not in S and is not separated away from W, then it is in the same component as precisely one $w \in W$ (as S is a multiway cut of W). In this case, identifying v and w does not change the solution.

The following lemma formalizes these observations. Given a set B of vertices in $G \setminus W$ and a function $f : B \to W$, we denote by G_f the graph obtained by replacing each set $\{w\} \cup f^{-1}(w)$ with a single vertex (with removal of loops and multiple occurrences of edges). To simplify the presentation, we will assume that this new vertex is also named w. We denote by $\mathbf{T}_f$ the set of terminal pairs where each vertex $v \in B$ is replaced by $f(v)$, and we denote by $\mathbf{T} \setminus B$ the set where every pair involving a vertex in B is removed.

LEMMA 5.1. *Let K be a non-trivial component of $G \setminus W$ with set of legs $\widehat{W}$ and let $B \subseteq K$. Then $(G, \mathbf{T}, W, p)$ has a solution if and only if one of the following statements is true.*

- *There is a $v \in B$ such that the instance $(G \setminus v, \mathbf{T} \setminus v, W, p-1)$ has a solution.*

- *There is a $v \in B$ and an important $v - \widehat{W}$ separator S' of size at most p in $G[K \cup \widehat{W}]$ such that the instance $(G \setminus S', \mathbf{T} \setminus S', W, p - |S'|)$ has a solution.*

- *There is a function $f : B \to \widehat{W}$ such that instance $(G_f, \mathbf{T}_f, W, p)$ has a solution.*

Lemma 5.1 determines a set of recursive calls to be applied in order to solve the given instance $(G, \mathbf{T}, W, p)$ of the MULTICUT COMPRESSION* problem. It is clear that in each step, the number of direction we branch into is bounded by a function of p, $|B|$, and $|W|$ (recall that the number of $v - \widehat{W}$ important separators of size at most p is at most 4^p and the number of functions $f : B \to \widehat{W}$ can be bounded by $|\widehat{W}|^{|B|}$). However, in order to ensure that the size of the search tree is bounded, we need to ensure that the *height* of the search tree is bounded as well. This is obvious for the first two type of branches, as p decreases. The following property ensures that in every branch of the third type, either the number of nontrivial components increases or we get an instance that trivially has no solution.

DEFINITION 5.2. *Let K be a non-trivial component and let $\widehat{W} \subseteq W$ be its set of legs. Let B be a subset of K. We say that B is a shattering set if for any function $f : B \to \widehat{W}$ one of the following statements is true regarding the instance $(G_f, \mathbf{T}_f, W, p)$ of the MULTICUT COMPRESSION*.*

- *There is a $w \in \widehat{W}$ such that there is no $w - (\widehat{W} \setminus \{w\})$ separator of size at most p in in $G_f[K \cup \widehat{W}]$.*

- *The number of non-trivial components is strictly greater in $G_f \setminus W$ than in $G \setminus W$.*

Note that the first possibility includes the case when $G_f[\widehat{W}]$ is not an independent set. In Section 5.1, we present a polynomial-time algorithm for finding a shattering set. Together with Lemma 5.1, it is sufficient to prove the fixed-parameter tractability of the MULTICUT COMPRESSION* problem, i.e. to prove Lemma 2.2.

LEMMA 5.3. *Given an instance* $(G, \mathbf{T}, W, p)$ *of the* MULTICUT COMPRESSION* *problem and a component* K *of* $G \setminus W$ *with more than two legs, we can find a shattering set* $B \subseteq K$ *of size at most* $3p$ *in polynomial time.*

5.1 Finding a shattering set

We start with two simple lemmas.

LEMMA 5.4. *Let* K *be a non-trivial component with a set* $\widehat{W}$ *of at least 3 legs. If* $G[M_1]$ *and* $G[M_2]$ *are both connected for two disjoint sets* $M_1, M_2 \subseteq K$, *then at most one of* M_1 *and* M_2 *can be a multiway cut of* $(G[K \cup \widehat{W}], \widehat{W})$.

PROOF. Assume the opposite. Since no two vertices of $\widehat{W}$ belong to the same component of $G[K \cup \widehat{W}] \setminus M_1$ and $|\widehat{W}| \geq 3$, we can specify two vertices w' and w'' of $\widehat{W}$ whose respective components C' and C'' in $G[K \cup \widehat{W}] \setminus M_1$ are disjoint from the connected set M_2. Then there is a $w' - w''$ path in $G[K \cup \widehat{W}]$ that uses vertices from C', then vertices from (the connected set) M_1, then vertices from C''. This path is disjoint from M_2, contradicting the assumption that M_2 is a multiway cut. $\square$

LEMMA 5.5. *Let* K *be a non-trivial component with a set* $\widehat{W}$ *of at least 3 legs. Let* $B \subseteq K$ *be a non-shattering set. Then there is exactly one connected component of* $G[K \setminus B]$ *which is a multiway cut of* $(G[K \cup \widehat{W}], \widehat{W})$.

PROOF. Let $f : B \to \widehat{W}$ be the mapping witnessing that B is not a shattering set. Let $K' \subseteq K \setminus B$ be the unique non-trivial component of $G_f \setminus W$ which is a subset of K. It is easy to see that K' a component of $G[K \setminus B]$ as well. Furthermore, K' is a multiway cut of $(G[K \cup \widehat{W}], \widehat{W})$. Otherwise, a path between vertices of $\widehat{W}$ in $G[K \cup \widehat{W}] \setminus K'$ would correspond to a walk of G_f between the same vertices which belong to a non-trivial component that is a subset of K but different from K', in contradiction to the definition of f. Finally, Lemma 5.4 implies that K' is the unique connected component of $G[K \setminus B]$ being a multiway cut of $G[K \cup \widehat{W}]$. $\square$

Let K be a non-trivial component with a set of legs $\widehat{W}$. Let $M \subseteq K$ be a multiway cut of $(G[K \cup \widehat{W}], \widehat{W})$. We call $N(M)$ (i.e the open neighborhood of M) the *boundary* of M. For each $w \in \widehat{W}$, the *image* $I(w)$ of w is the set of vertices of $N(M)$ reachable from w in $G[K \cup \widehat{W}] \setminus M$ (the image may include vertex w itself). Note that $I(w)$ is nonempty for any $w \in \widehat{W}$: consider the first vertex of $N(M)$ on a path from w to some other leg in $\widehat{W}$. For $X \subseteq \widehat{W}$, we let $I(X) = \bigcup_{w \in X} I(w)$. Let us select a distinguished leg $w^* \in \widehat{W}$. We say that M is *good* if all of the following conditions are true.

- $G[M]$ is connected.
- $N(M) = I(\widehat{W})$ or, in other words, each vertex of $N(M)$ is reachable in $G[K \cup \widehat{W}] \setminus M$ from some vertex of $\widehat{W}$.
- $|I(w^*)| \leq p$ and $|I(\widehat{W} \setminus \{w^*\})| \leq p$ holds (and hence we have $|N(M)| \leq 2p$).

LEMMA 5.6. *Let* K *be a non-trivial component with a set* $\widehat{W}$ *of at least 3 legs and a distinguished leg* w^*. *Let* M *be a good multiway cut of* $(G[K \cup \widehat{W}], \widehat{W})$. *Then there is a polynomial-time algorithm that either returns a shattering set of size at most* $3p$ *or a good multiway cut* $M' \subset M$.

PROOF. The desired algorithm first computes a smallest $I(w^*) - I(\widehat{W} \setminus \{w^*\})$ separator S of $G[N(M) \cup M]$ (recall that the images are nonempty). Observe that S is an inclusionwise minimal $w^* - \widehat{W} \setminus \{w^*\}$ separator in $G[K \cup \widehat{W}]$ (and hence nonempty). We consider three cases:

1. If $|S| > p$, then the algorithm returns $B := N(M) \setminus \widehat{W}$ reporting it as a shattering set.

2. If $|S| \leq p$ and there is a unique connected component M' of $G[K \setminus (N(M) \cup S)]$ which is a multiway cut of $(G[K \cup \widehat{W}], \widehat{W})$, then the algorithm returns M' reporting it as a good multiway cut.

3. If $|S| \leq p$ and there is no such M', then the algorithm returns $B := (N(M) \cup S) \setminus \widehat{W}$ reporting it as a shattering set.

This algorithm clearly takes polynomial time. The remaining proof establishes correctness of the algorithm in each of these three cases.

Case 1. Suppose that $B := N(M) \setminus \widehat{W}$ is not a shattering set and let $f : B \to \widehat{W}$ be a function witnessing this. It is not hard to see that M is a connected component in $G_f \setminus W$ whose set of legs is a subset of $\widehat{W}$. We consider three subcases and arrive to a contradiction in each of them.

Case 1a. M is a trivial component of $G_f \setminus W$. Let w be the only leg of M. Let w_1 and w_2 be other two distinct legs of K in G that are different from w. It follows that f maps every vertex of $I(w_1) \cup I(w_2)$ to w implying that there is a $w - w_1$ and $w - w_2$ path in G_f whose internal vertices belong to two different components on w_1 and w_2 in $G[K \cup \widehat{W}] \setminus M$. Thus G_f has at least two non-trivial components that are subsets of K, in contradiction to the choice of f.

Case 1b. M is a nontrivial component of $G_f \setminus W$ and $f(v) = w$ for every $v \in I(w)$ and $w \in \widehat{W}$ (i.e., each vertex on the boundary is mapped to its preimage). As the smallest $I(w^*) - I(\widehat{W} \setminus \{w^*\})$ separator in $G[N(M) \cup M]$ is larger than p, $G[M \cup \widehat{W}]$ does not have a $w^* - \widehat{W} \setminus \{w^*\}$ separator of size at most p in contradiction to f being a witnessing function.

Case 1c. M is a nontrivial component of $G_f \setminus W$ and $f(v) = w_2$ for some $v \in I(w_1)$, $w_1, w_2 \in \widehat{W}$, $w_1 \neq w_2$. By definition of $I(w_1)$, there is a $v - w$ path in G whose internal vertices are fully contained in $K \setminus M$. Therefore, there is a $w_1 - w_2$ path in G_f whose internal vertices are disjoint from M, implying that G_f has a nontrivial component which is a subset of K, but distinct from the nontrivial component M. Thus the number of nontrivial components increases, a contradiction.

Case 2. We show first that $M' \subset M$ in this case. Clearly, $M' \neq M$, as M' is disjoint from (the nonempty) $S \subseteq M$. Thus $M' \not\subset M$ is only possible if M' is disjoint from M, but Lemma 5.4 implies that the two disjoint connected sets M and M' cannot be both multiway cuts.

For clarity, from now on we use $I_M(w)$ and $I_{M'}(w)$ for the image of w on the boundary of M and M', respectively. Observe that $I_M(w) \cap N(M') \subseteq I_{M'}(w)$ for every $w \in \widehat{W}$: for every $v \in I_M(w) \cap N(M')$, there is a $w - v$ path disjoint from M, which is obviously disjoint from $M' \subseteq M$ as well. This immediately implies that either $I_M(w^*)$ or $I_M(\widehat{W} \setminus \{w^*\})$ is disjoint from $N(M')$: otherwise, a vertex $v_1 \in I_M(w^*) \cap N(M')$ and a vertex $v_2 \in I_M(w^*) \cap N(M')$ can be connected by a path P whose internal vertices are in M' (hence disjoint from S), and path P can be extended to a $w^* - \widehat{W} \setminus \{w^*\}$ path disjoint from S, contradicting the definition of S. Therefore, either $N(M') \subseteq I_M(w^*) \cup S$ or $N(M') \subseteq I_M(\widehat{W} \setminus \{w^*\}) \cup S$ holds.

To show that $|I_{M'}(w^*)|$ and $|I_{M'}(\widehat{W} \setminus \{w^*\})|$ are both at most p, we argue as follows. Suppose first that $N(M') \subseteq I_M(w^*) \cup S$.

We show that $I_{M'}(\widehat{W} \setminus \{w^*\}) = S \cap N(M')$ and therefore $I_{M'}(w^*) = I_M(w^*) \cap N(M')$, proving the bound on both $|I_{M'}(w^*)|$ and $|I_{M'}(\widehat{W} \setminus \{w^*\})|$. Note that this implies furthermore that $N(M') = I_{M'}(\widehat{W})$, i.e., every vertex of $N(M')$ is the image of some leg. From $I_M(w^*) \cap N(M') \subseteq I_{M'}(w^*)$ we already know that $I_{M'}(\widehat{W} \setminus \{w^*\}) \subseteq S \cap N(M')$. To show that $S \cap N(M') \subseteq I_{M'}(\widehat{W} \setminus \{w^*\})$ holds, consider a vertex $s \in S \cap N(M')$. By the minimality of S, for some $w \in \widehat{W} \setminus \{w^*\}$ there is a $w^* - w$ path P in $G[K \cup \widehat{W}]$ that intersects S exactly in s. As M' is a multiway cut, path P has to go through M'. Furthermore, we can assume that P contains exactly 2 vertices of $N(M')$. One of these two vertices is s, and the other has to be a vertex $v \in I_M(w^*)$ (as P contains only one vertex of S). It follows that P has a subpath P_1 connecting $\widehat{W}$ and v, and a subpath P_2 connecting $\widehat{W}$ and s, both of them disjoint from M'. As $v \in I_M(w^*)$, path P_1 connects w^* and v, thus path P_2 connects s and w. Therefore, s is in $I_{M'}(\widehat{W} \setminus \{w^*\})$, what we had to show.

A symmetrical argument (exchanging the role of w^* and $\widehat{W} \setminus \{w^*\}$) shows that if $N(M') \subseteq I_M(\widehat{W} \setminus \{w^*\}) \cup S$, then $I_{M'}(w^*) = S \cap N(M')$, implying the bounds on $|I_{M'}(w^*)|$ and $|I_{M'}(\widehat{W} \setminus \{w^*\})|$. Thus in both cases, we proved that $M' \subset M$ is a good multiway cut.

Case 3. Assume now that the algorithm returns $B := (S \cup N(M)) \setminus \widehat{W}$ as a shattering set. This happens because there is no unique component of $G[K \setminus (N(M) \cup S)]$ which is a multiway cut of $(G[K \cup \widehat{W}], \widehat{W})$. According to Lemma 5.5, $N(M) \cup S$ is indeed a shattering set in this case. Clearly, its size is at most $3p$. $\square$

Lemma 5.3 follows by iterative application of Lemma 5.6.

PROOF (OF LEMMA 5.3). It is not hard to see that K is a good multiway cut of $(G[K \cup \widehat{W}], \widehat{W})$. Let $M_0 = K$. Apply the algorithm of Lemma 5.6 to M_0. The algorithm either returns a shattering set of size at most $3p$ or a good multiway cut $M_1 \subset M_0$. In the former case just return the shattering set, in the latter case, apply the algorithm of Lemma 5.6 to M_1. Continuing this way, we obtain a sequence $M_0 \supset M_1 \supset \ldots$ of good multiway cuts of decreasing size. It follows that after at most $|V(G)|$ iterative applications of the algorithm of Lemma 5.6, a shattering set of size at most $3p$ will be returned. $\square$

6. REFERENCES

[1] S. Arora, S. Rao, and U. Vazirani. Expander flows, geometric embeddings and graph partitioning. *J. ACM*, 56(2):1–37, 2009.

[2] N. Bansal, A. Blum, and S. Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004.

[3] H. L. Bodlaender, M. R. Fellows, P. Heggernes, F. Mancini, C. Papadopoulos, and F. A. Rosamond. Clustering with partial information. *Theor. Comput. Sci.*, 411(7-9):1202–1211, 2010.

[4] N. Bousquet, J. Daligault, and S. Thomassé. Multicut is FPT. Manuscript, arXiv:1010.5197, 2010.

[5] N. Bousquet, J. Daligault, S. Thomasse, and A. Yeo. A polynomial kernel for multicut in trees. In *STACS*, pages 183–194, 2009.

[6] G. Calinescu, C. G. Fernandes, and B. Reed. Multicuts in unweighted graphs and digraphs with bounded degree and bounded tree-width. *Journal of Algorithms*, 48(2):333 – 359, 2003.

[7] S. Chawla, R. Krauthgamer, R. Kumar, Y. Rabani, and D. Sivakumar. On the hardness of approximating multicut and sparsest-cut. *Comput. Complexity*, 15(2):94–114, 2006.

[8] J. Chen, Y. Liu, and S. Lu. An improved parameterized algorithm for the minimum node multiway cut problem. *Algorithmica*, 55(1):1–13, 2009.

[9] J. Chen, Y. Liu, S. Lu, B. O'Sullivan, and I. Razgon. A fixed-parameter algorithm for the directed feedback vertex set problem. *J. ACM*, 55(5), 2008.

[10] J. Chuzhoy and S. Khanna. Polynomial flow-cut gaps and hardness of directed cut problems. *J. ACM*, 56(2):1–28, 2009.

[11] 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.

[12] E. D. Demaine, D. Emanuel, A. Fiat, and N. Immorlica. Correlation clustering in general weighted graphs. *Theor. Comput. Sci.*, 361(2-3):172–187, 2006.

[13] R. G. Downey and M. R. Fellows. *Parameterized Complexity*. Springer, New York, 1999.

[14] U. Feige, M. Hajiaghayi, and J. R. Lee. Improved approximation algorithms for minimum weight vertex separators. *SIAM J. Comput.*, 38(2):629–657, 2008.

[15] J. Flum and M. Grohe. *Parameterized Complexity Theory*. Springer, Berlin, 2006.

[16] L. R. Ford, Jr. and D. R. Fulkerson. Maximal flow through a network. *Canad. J. Math.*, 8:399–404, 1956.

[17] N. Garg, V. V. Vazirani, and M. Yannakakis. Approximate max-flow min-(multi)cut theorems and their applications. *SIAM J. Comput.*, 25(2):235–251, 1996.

[18] G. Gottlob and S. T. Lee. A logical approach to multicut problems. *Inf. Process. Lett.*, 103(4):136–141, 2007.

[19] S. Guillemot. Fpt algorithms for path-transversal and cycle-transversal problems. *Discrete Optimization*, 8(1):61 – 71, 2011. Parameterized Complexity of Discrete Optimization.

[20] J. Guo, F. Hüffner, E. Kenar, R. Niedermeier, and J. Uhlmann. Complexity and exact algorithms for vertex multicut in interval and bounded treewidth graphs. *European J. Oper. Res.*, 186(2):542–553, 2008.

[21] J. Guo and R. Niedermeier. Fixed-parameter tractability and data reduction for multicut in trees. *Networks*, 46(3):124–135, 2005.

[22] A. Gupta. Improved results for directed multicut. In *SODA*, pages 454–455, 2003.

[23] F. Hüffner, R. Niedermeier, and S. Wernicke. Techniques for practical fixed-parameter algorithms. *The Computer Journal*, 51(1):7–25, 2008.

[24] S. Khot. On the power of unique 2-prover 1-round games. In *STOC*, pages 767–775, 2002.

[25] D. Lokshtanov and D. Marx. Clustering with local restrictions, 2010.

[26] D. Marx. Parameterized graph separation problems. In *IWPEC*, pages 71–82. 2004.

[27] D. Marx. Parameterized graph separation problems. *Theoretical Computer Science*, 351(3):394–406, 2006.

[28] D. Marx and I. Razgon. Constant ratio fixed-parameter approximation of the edge multicut problem. *Inf. Process. Lett.*, 109(20):1161–1166, 2009.

[29] D. Marx and I. Razgon. Fixed-parameter tractability of multicut parameterized by the size of the cutset. *CoRR*, abs/1010.3633, 2010.

[30] M. Naor, L. J. Schulman, and A. Srinivasan. Splitters and near-optimal derandomization. In *FOCS*, pages 182–191, 1995.

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

[32] R. Pichler, S. Rümmele, and S. Woltran. Multicut algorithms via tree decompositions. In *CIAC*, pages 167–179. 2010.

[33] I. Razgon and B. O'Sullivan. Almost 2-SAT is fixed-parameter tractable. *J. Comput. Syst. Sci.*, 75(8):435–450, 2009.

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

[35] J. Sherman. Breaking the multicommodity flow barrier for $O(\sqrt{\log n})$-approximations to sparsest cut. In *FOCS*, pages 363–372, 2009.

[36] M. Xiao. Algorithms for multiterminal cuts. In *CSR*, pages 314–325, 2008.

[37] M. Yannakakis, P. C. Kanellakis, S. S. Cosmadakis, and C. H. Papadimitriou. Cutting and partitioning a graph after a fixed pattern. In *ICALP*, pages 712–722, 1983.