Constant Ratio Fixed-Parameter Approximation of the Edge Multicut Problem

Dániel Marx[1,⋆] and Igor Razgon[2,⋆⋆]

[1] Department of Computer Science and Information Theory, Budapest University of Technology and Economics
dmarx@cs.bme.hu
[2] Cork Constraint Computation Centre, University College Cork
i.razgon@cs.ucc.ie

Abstract. The input of the Edge Multicut problem consists of an undirected graph G and pairs of terminals $\{s_1, t_1\}, \ldots, \{s_m, t_m\}$; the task is to remove a minimum set of edges such that s_i and t_i are disconnected for every $1 \leq i \leq m$. The parameterized complexity of the problem, parameterized by the maximum number k of edges that are allowed to be removed, is currently open. The main result of the paper is a parameterized 2-approximation algorithm: in time $f(k) \cdot n^{O(1)}$, we can either find a solution of size $2k$ or correctly conclude that there is no solution of size k.

The proposed algorithm is based on a transformation of the Edge Multicut problem into a variant of parameterized Max-2-SAT problem, where the parameter is related to the number of clauses that are *not* satisfied. It follows from previous results that the latter problem can be 2-approximated in a fixed-parameter time; on the other hand, we show here that it is W[1]-hard. Thus the additional contribution of the present paper is introducing the first natural W[1]-hard problem that is constant-ratio fixed-parameter approximable.

1 Introduction

The minimum cut problem and its variants are among the most well-studied combinatorial optimization problems. The focus of the paper is Edge Multicut: given a graph G and pairs of vertices $\{s_1, t_1\}, \ldots, \{s_m, t_m\}$, remove a minimum set of edges such that s_i and t_i are disconnected for every $1 \leq i \leq m$. Edge Multicut generalizes the classical $s - t$ cut problem (disconnect s and t) and the Multiway Cut problem (disconnect all the terminals from each other). Edge Multicut can be approximated within a factor of $O(\log m)$ in polynomial time [13] (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 [17], no constant factor approximation is possible for Edge Multicut [5].

⋆ Supported by the Magyary Zoltán Felsőoktatási Közalapítvány and the Hungarian National Research Fund (OTKA grant 67651).

⋆⋆ Supported by Science Foundation Ireland (Grant Number 05/IN/I886).

A. Fiat and P. Sanders (Eds.): ESA 2009, LNCS 5757, pp. 647–658, 2009.

Parameterized complexity approaches hard computational problems through a multivariate analysis of the running time. Instead of expressing the running time as a function of the input size n only, the running time is expressed as a function of n and k, where k is a well-defined parameter of the input instance. We say that a problem (with a particular parameter k) is *fixed-parameter tractable (FPT)* if it can be solved in time $f(k) \cdot n^{O(1)}$, where f is an arbitrary function depending only on k. Thus we relax polynomial time by allowing exponential (or worse!) dependence on the parameter k. For more background on parameterized complexity, the reader is referred to the monographs [10,12,22].

Edge Multicut on trees is FPT, parameterized by the maximum number k of edges that can be deleted [3,15]. The problem and its vertex-cut version were studied in [14] for other classes of graphs. For general graphs, Edge Multicut is FPT if both k and m are chosen as parameters (i.e, the problem can be solved in time $f(k,m) \cdot n^{O(1)}$) [19,26]. However, it is an open question whether Edge Multicut is FPT in general graphs parameterized by k only. Besides the fundamental nature of the problem, there are other reasons why this question is important. It has been observed that Edge Multicut is equivalent to Fuzzy Cluster Editing, a correlation clustering problem [2,8,1]. Furthermore, it seems that cut problems are important ingredients in the solution of certain parameterized problems. For example, the fixed-parameter tractability of Directed Feedback Vertex Set [6] was a longstanding open question and solving a variant of directed multicut was an important step in its solution.

Recently, it has been proposed that the notion of approximability can be investigated in the framework of fixed-parameter tractability as well [4,7,9,21]. Here we follow this approach and present a parameterized 2-approximation for Edge Multicut: the main result of the paper is an algorithm with running time $f(k) \cdot n^{O(1)}$ that, given an instance of the Edge Multicut problem and an integer k, either finds a solution of size $2k$ or correctly concludes that no solution of size k exists. As surveyed in [21], so far there are very few natural problems where a parameterized approximation is possible, but the problem is not known to be fixed-parameter tractable.

The main idea of our approximation algorithm is to reduce Edge Multicut to a variant of Almost 2SAT (delete k clauses to make a 2-CNF formula satisfiable). The reduction is nontrivial: it consists of several steps and requires the use of iterative compression. Almost 2SAT is known to be fixed-parameter tractable [24] and this immediately implies a parameterized 2-approximation for the variant we use here. Proving that this variant is FPT would seem an obvious approach for proving that Edge Multicut is FPT. However, we rule out this possibility by showing the W[1]-hardness of the Almost 2SAT variant. This might be of independent interest, as it is the first natural W[1]-hard problem having a constant-ratio parameterized approximation.

Besides giving an algorithm for a particular problem, the paper has a conceptual contribution as well by introducing a new technique: we demonstrate that reduction to Almost 2SAT can be a useful approach in the design of fixed-parameter algorithms. We believe that this technique will find uses for other problems in the future. However, it is not obvious what type of problems can be handled this way: for example, it was not apparent that Multicut has any connections with 2SAT.

2 Preliminaries

The objects considered in the present paper are (simple undirected) graphs and 2-CNF formulas. We define the related notation that will be used further in the paper. For a graph G, we denote by $V(G)$ and $E(G)$ its set of vertices and edges, respectively. For C such that either $C \subseteq V(G)$ or $C \subseteq E(G)$, $G \setminus C$ is the graph obtained from G by removal of the elements of C (if $C \subseteq V(G)$ then the edges incident to C are removed from G as well). For $E^* \subseteq E(G)$, $G[E^*]$ is a graph whose set of edges is E^* and the set of vertices is the set of end points of E^*.

Let us specify two sets $\{s_1, \dots, s_\ell\}$ and $\{t_1, \dots, t_\ell\}$ of vertices of G and call their union the set of *terminal* vertices. Let $\mathbf{T} = \{\{s_1, t_1\}, \dots, \{s_l, t_\ell\}\}$ and let C be either a set of *non-terminal* vertices or a set of edges of G such that $G \setminus C$ has no path between s_i and t_i for each i from 1 to l. In this case, we say that C *separates* $\mathbf{T}$ in G. If C is a set of edges, we also say that C is an edge multicut (EMC) of $(G, \mathbf{T})$. Let $Y \subseteq V(G)$. We say that C separates $\mathbf{T}$ and Y in G if C separates $\mathbf{T}$ in G and $G \setminus C$ has no path between any two distinct vertices of Y. If C is a set of edges, we also say that C is an EMC of $(G, \mathbf{T}, Y)$. Note that in the Edge Multicut problem we have to find a set of edges that separates a set $\mathbf{T}$ of terminal pairs.

Now we define the central problem considered in the present paper.

The emc problem
Input: A graph G, an integer k, and a set $\mathbf{T}$ of pairs of terminal vertices of G
Parameter: k
Output: An EMC of $(G, \mathbf{T})$ of size at most k or 'NO' if no such EMC exists.

We will also need the auxiliary problems defined below and referred as AEMC1 and AEMC2.

The aemc1 problem
Instance: A graph G, an integer k, a set $\mathbf{T}$ of pairs of terminal vertices of G, a set Y of at most $2k + 1$ non-terminal vertices separating $\mathbf{T}$ in G
Parameter: k
Output: An EMC of $(G, \mathbf{T}, Y)$ of size at most k or 'NO' if no such EMC exists.

The aemc2 problem
Instance: The same as in the AEMC1 problem
Parameter: k
Output: An EMC of $(G, \mathbf{T})$ of size at most k or 'NO' if no such EMC exists.

Finally, we use the following modification of the Almost 2-SAT problem [24]. Let F be a 2-CNF formula and let $C = (\ell_1 \vee \ell_2)$ be a clause of F. A literal l *satisfies* C if $\ell = \ell_1$ or $\ell = \ell_2$. A set L of literals satisfies F or, in other words, L is a *satisfying assignment* of F if L does not contain a literal together with its negation and each clause of F is satisfied by at least one literal of L. Let $L = \{\ell_1, \ldots, \ell_r\}$. We denote the 2-CNF formula $\bigwedge_{i=1}^{r}(\ell_i \vee \ell_i)$ by $\bigwedge L$. The clause $(\ell_1 \rightarrow \ell_2)$ is shorthand for $(\neg \ell_1 \vee \ell_2)$.

Almost 2-sat problem with blocks and fixed literals (2-asat-bfl)
Instance: (F, P, L, k) where

- F is a *satisfiable* 2-CNF formula with possible repeated occurrences of clauses;
- P is a family of (not necessarily disjoint) subsets (*blocks*) of at most 2 clauses covering all the clauses of F;
- L is a set of literals;
- k is a non-negative integer.

Parameter: k.
Output: A set B of at most k blocks of P such that $F' \wedge \bigwedge L$ is satisfiable (F' is the formula obtained from F as a result of removal of the clauses of B) or 'NO' if no such set of blocks exists.

Let P be a parameterized problem where the parameter k is an integer appearing in the input and the task is to find some object of size at most k or report 'NO' if no such object exists. Following [7,21], we say that problem P is *fixed-parameter approximable* (FPA) with ratio c if there is an $f(k) \cdot n^{O(1)}$ time algorithm that either returns an object satisfying all output specifications except that its size is at most ck, or 'NO' and in the latter case it is guaranteed that there is no object of size at most k satisfying the output specifications.

Proposition 1. *The* 2-ASAT-BFL *problem is FPA with ratio 2 and the approximation can be achieved in time $O(25^k km^2)$ where m is the number of clauses of F.*

Proof. Claim 8 in [23] (this is the full version of [24]) states that in time $O(5^k km^2)$ it is possible either to compute a set S of at most k clauses so that $F' \wedge \bigwedge L$ is satisfiable, where F' is the formula obtained from F as a result of the removal of S, or to conclude that no such set of clauses exists. Run this algorithm with parameter $2k$ (thus raising the exponential part to 25^k). Assume that the algorithm returns a set S of clauses of size at most $2k$. Then return an arbitrary minimal set B of blocks covering these clauses. Otherwise return 'NO'. Clearly, if a set of blocks B is returned and F' is the formula resulting from removal of the clauses of these blocks from F then $F' \wedge \bigwedge L$ is satisfiable: in particular, all clauses of S are removed. If the resulting algorithm returns 'NO', it follows that removal of $2k$ clauses cannot make F satisfiable. Since each block consists of at most 2 clauses, it follows that removal of k blocks cannot make F satisfiable, implying that the 'NO' answer is correct. $\square$

3 Reduction to Almost 2-Sat

Let $(G, \mathbf{T}, Y, k)$ be an instance of the AEMC1 problem. We define the instance (F, P, L, k) of the 2-ASAT-BFL problem corresponding to $(G, \mathbf{T}, Y, k)$. Then we show that $(G, \mathbf{T}, Y, k)$ is a 'YES' instance of the AEMC1 problem[1] if and only if (F, P, L, k) is a 'YES' instance of the 2-ASAT-BFL problem. The fixed-parameter approximability of the AEMC1 will then follow from Proposition 1.

The set of variables of F is $\{z_{u,v} | u \in V(G), v \in Y\}$. The variable $z_{u,v}$ represents the truth of the ground statement "u and v belong to the same connected component". The clauses of F can be partitioned into the following 3 groups.

Group 1. For each $\{s_i, t_i\} \in \mathbf{T}$ and for each $v \in Y$, the group contains $2k + 1$ copies of clause $(\neg z_{s_i,v} \vee \neg z_{t_i,v})$. The purpose of these clauses is to forbid two terminals to be separated to belong to the same connected component.

Group 2. For each pair $\{v_1, v_2\}$ of vertices of Y such that $v_1 \neq v_2$, and for each $u \in V(G)$ the group contains $2k + 1$ copies of clause $(\neg z_{u,v_1} \vee \neg z_{u,v_2})$. The purpose of these clauses is to forbid two different vertices of Y to belong to the same connected component.

Group 3. For each $\{u, v\} \in E(G)$ and for each $w \in Y$, the group contains clause $(z_{u,w} \to z_{v,w})$ and clause $(z_{v,w} \to z_{u,w})$. These clauses show that vertices u and v belong to the same connected component.

Observe that F is satisfiable, for instance, by an assignment including the negative literals of all the variables. The set P of blocks is defined as follows. For each clause of F there is a block containing this clause only. Also for all possible pairs $\{(u, w_1), (v, w_2)\}$ where $\{u, v\} \in E(G)$, $w_1 \in Y$, $w_2 \in Y$, there is a block $\{(z_{u,w_1} \to z_{v,w_1}), (z_{v,w_2} \to z_{u,w_2})\}$. Finally, $L = \{z_{v,v} | v \in Y\}$.

Lemma 2. *If $(G, \mathbf{T}, Y, k)$ is a 'YES' instance of the* AEMC1 *problem then (F, P, L, k) is a 'YES' instance of the* 2-ASAT-BFL *problem.*

Proof. Let C be an EMC of $(G, \mathbf{T}, Y, k)$ of size at most k. We associate with each $\{u, v\} \in C$ the block $B(\{u, v\})$ corresponding to the 'location' of u and v. In particular, if in $G \setminus C$ there are two different vertices w_1 and w_2 of Y such that u belongs to the component of w_1 while v belongs to the component of w_2 then $B(\{u, v\}) = \{(z_{u,w_1} \to z_{v,w_1}), (z_{v,w_2} \to z_{u,w_2})\}$. Otherwise if exactly one of $\{u, v\}$, say, u belongs to the connected component of some vertex $w \in Y$ while the connected component of v contains no vertex of Y then $B(\{u, v\}) = \{(z_{u,w} \to z_{v,w})\}$. Finally, if neither u nor v belong to the same component with a vertex of Y then $B(\{u, v\})$ can be an arbitrarily chosen block.

Let F' be a 2-CNF formula obtained from F by removal of the union of all $B(\{u, v\})$. We claim that F' is satisfiable by an assignment including L as a subset. In particular, let L^* be the set of literals of the variables of V created as follows: $z_{u,w} \in L^*$ whenever u belongs to the same component with w in $G \setminus C$, otherwise $\neg z_{u,w} \in L^*$. Clearly, $L \subseteq L^*$. We claim that L^* satisfies F'. Clauses of

[1] A 'YES' instance is one whose output is not 'NO'.

Group 1 are satisfied because otherwise there is a pair $\{s_i, t_i\}$ of terminals that belong to the same component (with some $w \in Y$) in $G \setminus C$ in contradiction to being C an EMC of $(G, \mathbf{T}, Y)$. Clauses of Group 2 are satisfied because otherwise there is a vertex of $G \setminus C$ that belongs to two distinct connected components, which is absurd. Finally, assume that a clause $c = (z_{u,w} \rightarrow z_{v,w})$ is not satisfied by L^*. It can happen only if u belongs to the same component with w, while v does not. By description of Group 3, $\{u, v\} \in E(G)$ and, since u and v belong to different connected components of $G \setminus C$, $\{u, v\} \in C$. Observe that the first or the second condition of creation of $B(\{u, v\})$ is satisfied and hence $c \in B(\{u, v\})$, i.e. c is not a clause of F'. The proof is now complete. □

Lemma 3. *If* (F, P, L, k) *is a 'YES' instance of the* 2-ASAT-BFL *problem then* $(G, \mathbf{T}, Y, k)$ *is a 'YES' instance of the* AEMC1 *problem.*

Proof. Let B ($|B| \leq k$) be a set of blocks whose removal from F makes the resulting 2-CNF formula F' satisfiable by an assignment L^* such that $L \subseteq L^*$. Observe that it makes no sense to include in B any of the blocks containing only a single clause from Group 1 or 2: those clauses are present in $2k + 1$ copies in F, hence they have to be satisfied in L^* as well. Thus we can safely assume that every block in B contains one or two clauses from Group 3.

By construction, each block of B corresponds to exactly one edge of G. Let C be the set of all such edges. We claim that C is an EMC of $(G, \mathbf{T}, Y)$. Assume first that C does not separate Y, i.e., there are vertices w_1 and w_2 of Y such that $G \setminus C$ has a path p from w_1 to w_2. If the length of p is 1 then let $S = \{(z_{w_1,w_1} \rightarrow z_{w_2,w_1})\}$. Otherwise, let $u_1, \ldots, u_q$ be the intermediate vertices of p listed in the order of their occurrence when p is traversed from w_1 to w_2 and let $S = \{(z_{w_1,w_1} \rightarrow z_{u_1,w_1}), (z_{u_1,w_1} \rightarrow z_{u_2,w_1}), \ldots, (z_{u_{q-1},w_1} \rightarrow z_{u_q,w_1}), (z_{u_q,w_1} \rightarrow z_{w_2,w_1})\}$. Since $L \subseteq L^*$, $z_{w_1,w_1} \in L^*$ as well as $z_{w_2,w_2} \in L^*$. If all the clauses of S are contained in F', then $z_{w_2,w_1} \in L^*$ would follow from this chain of implications, contradicting $(\neg z_{w_2,w_1} \vee \neg z_{w_2,w_2})$ (that necessarily belongs to F'). Hence at least one clause of S belongs to a block of B, implying that at least one edge of p belongs to C.

Thus, if C is not an EMC of $(G, \mathbf{T}, Y)$, it remains to assume that C does not separate $\mathbf{T}$, i.e., there is $\{s, t\} \in \mathbf{T}$ such that $G \setminus C$ has a path p between s and t. By definition of Y, p contains at least one vertex $w \in Y$.[2] If s and w are adjacent in p then let $S = \{(z_{w,w} \rightarrow z_{s,w})\}$. Otherwise, let $u_1, \ldots u_q$ be the intermediate vertices of p occurring in p between w and s listed in the order they occur if p is traversed from w to s. Then $S = \{(z_{w,w} \rightarrow z_{u_1,w}), (z_{u_1,w} \rightarrow z_{u_2,w}), \ldots, (z_{u_{q-1},w} \rightarrow z_{u_q,w}), (z_{u_q,w} \rightarrow z_{s,w})\}$. Arguing as in the previous case, we derive that either $z_{s,w} \in L^*$ or one of the edges corresponding to S belongs to C. Arguing analogously regarding the subpath of p between w and t we derive that either $z_{t,w} \in L^*$ or at least one edge of this subpath belongs to C. It follows that if no edge of p belongs to C then both $z_{s,w} \in L^*$ and $z_{t,w} \in L^*$ hold. But

[2] Notice that this is the only place where it is essential that Y separates all the pairs of terminals of $\mathbf{T}$.

this is a contradiction since in this case the clause $(\neg z_{s,w} \vee \neg z_{t,w})$ is not satisfied. We conclude that C is an EMC of $(G, \mathbf{T}, Y)$. □

The following theorem is an immediate consequence of Proposition 1, Lemma 2, and Lemma 3.

Theorem 4. *The* AEMC1 *problem is FPA with ratio 2.*

4 Fixed-Parameter Approximability of the Emc Problem

We prove the main result of the paper in this section: EMC is fixed-parameter approximable with ratio 2. First, we reduce the AEMC2 problem to the AEMC1 problem. The only difference between the two problems is that in the instance $(G, \mathbf{T}, Z, k)$ of the AEMC2 problem, the solution does not have to separate Z. However, the algorithm can be extended by trying all possible ways in which the solution partitions the set Z.

Lemma 5. *The* AEMC2 *problem is FPA with ratio 2.*

Proof. Apply the following algorithm. Explore all possible partitions of vertices of Y into subsets. For the given partition $Z = Z_1 \cup \cdots \cup Z_q$, let G^* be the graph obtained from G by *contracting* each Z_i into a vertex y_i (loops produced by the contraction are removed, multiple edges are subdivided). Let $Y = \{y_1, \ldots, y_q\}$. Using Theorem 4, we can obtain a 2-approximation for the instance $(G^*, \mathbf{T}, Y, k)$ of the AEMC1. If for at least one such instance an EMC S of $(G^*, \mathbf{T}, Y)$ is returned then return S. Otherwise, return 'NO'.

Since the number of partitions of Z depends on $|Z| \leq 2k + 1$, the above algorithm is an FPT algorithm with parameter k. It is easy to see that if the algorithm returns an EMC S of $(G^*, \mathbf{T}, Y, k)$, then S is an EMC of $(G, \mathbf{T})$ as well. Conversely, assume that $(G, \mathbf{T})$ has an EMC C of size at most k. Let $Z_1, \ldots, Z_q$ be the partition of Z so that two vertices get into the same partition class if and only if they belong to the same connected component of $G \setminus C$. According to Theorem 4, being applied to the tuple $(G^*, \mathbf{T}, Y, k)$ resulting from this partition, the above algorithm necessarily produces an EMC S of $(G^*, \mathbf{T})$ having size at most $2k$. Consequently, if the above algorithm returns 'NO' an EMC of $(G, \mathbf{T})$ of size at most k cannot exist and the answer 'NO' is valid. □

The problem AEMC2 is easier than EMC, since the input contains more information, namely the set Z separating $\mathbf{T}$. We apply a methodology known under the name 'iterative compression' which essentially gives us such a set Z 'for free.' Iterative compression was first used by Reed et al. [25] and has become a very useful technique in the design of parameterized algorithms [6,18,16,20,24].

Theorem 6. *The* EMC *problem is FPA with ratio 2.*

Proof. Let $(G, \mathbf{T}, k)$ be an instance of the EMC problem. Let $e_1, \ldots, e_m$ be the edges of G. Let $G_0, \ldots, G_m$ be the graphs defined as follows. For each G_i,

$V(G_i) = V(G)$. $E(G_0) = \emptyset$ and for each $i > 0$, $E(G_i) = \{e_1, \dots, e_i\}$. One by one, we consider the $(G_i, \mathbf{T}, k)$ instances of the EMC problem in ascending order of i, and for each instance we find a 2-approximation. The approximation for each $(G_i, \mathbf{T}, k)$ results in output S_i, where S_i is either a set of edges or 'NO'. In particular, $S_0 = \emptyset$. Consider computing of S_i, $i > 0$ provided that S_{i-1} is already known. If $S_{i-1} =$ 'NO' then $S_i =$ 'NO', as S_i is a supergraph of S_{i-1}. Otherwise, S_{i-1} is an EMC of size at most $2k$ for $\mathbf{T}$ in G_{i-1}, hence $S_{i-1} \cup \{e_i\}$ is an EMC of size at most $2k + 1$ for $\mathbf{T}$ in G_i. Subdivide each edge of $S_{i-1} \cup \{e_i\}$ with a new vertex; clearly, subdivisions does not change the existence of an EMC. Let G^* be the graph obtained this way and let Z be the set of new vertices. It follows that Z has size at most $2k + 1$ and separates $\mathbf{T}$ in G^*. Thus we can use the algorithm for AEMC2 on the instance $(G^*, \mathbf{T}, Z, k)$. It either returns an EMC of $\mathbf{T}$ in G^* of size at most $2k$ (which can be modified to obtain a EMC S_i of $\mathbf{T}$ in G by replacing each subdivided edge by the corresponding edge of $S_{i-1} \cup \{e_i\}$) or returns 'NO', in which case we can set $S_i =$ 'NO'. The validity of the algorithm is easy to verify by induction on i combined with Lemma 5. □

We conclude the section with computing the runtime of the algorithm achieving the ratio 2 approximation of the EMC problem. Denote $|V(G)|$ by n, $|E(G)|$ by m and $|\mathbf{T}|$ by ℓ. The iterative compression process described in the proof of Theorem 6 takes $O(m)$ iterations of solving the AEMC2 problem. The algorithm for the AEMC2 problem takes $P(2k + 1, k)$ iterations of solving the AEMC1 problem, where $P(2k + 1, k)$ is the number of partitions of a $2k + 1$-element set into at most k classes. Finally, in order to solve the AEMC1 problem the graph is transformed into a 2-CNF formula. The number of clauses of this formula is $m_1 = O(\ell k^2 + nk^3 + mk) = O(nk^3 + mk)$ (the term ℓk^2 corresponding to the number of clauses of Group 1 is absorbed by nk^3). Then the 2-ASAT-BFL problem is solved for the obtained formula,which takes $O(25^k k m_1^2) = O(25^k k^3 (n^2 k^4 + m^2))$. Thus the overall complexity is $O(25^k P(2k + 1, k) \cdot k^3 (n^2 k^4 + m^2))$.

5 Hardness of the 2-ASAT-BFL Problem

It is easy to see from the above discussion that the fixed-parameter tractability of the 2-ASAT-BFL problem would imply the fixed-parameter tractability of the EMC problem. In this section we show that the latter is very unlikely to be derived in this way because the 2-ASAT-BFL problem turns out to be W[1]-hard. To the best of our knowledge, this is the first problem known to be both W[1]-hard and FPA with a constant ratio.

Theorem 7. 2-ASAT-BFL *problem is* W[1]*-hard even if the blocks are disjoint.*

Proof. The proof is by reduction from MULTICOLORED CLIQUE, where given a graph G, an integer k, and a proper k-coloring of the vertices of G, the task is to decide whether there is a k-clique in G. (Proper k-coloring is a mapping from $V(G)$ to $\{1, \dots, k\}$ such that adjacent vertices have different colors.) MULTICOLORED CLIQUE is known to be W[1]-hard [11]. We can assume that every

c-colored vertex has at least one neighbor from every color class except c: otherwise the vertex cannot be part of a k-clique and can be safely deleted. Let n_c be the number of vertices of color c. Let $v_{c,i}$ $(1 \leq i \leq n_c)$ be the vertices with color c. Let $d(c,i,c') \geq 1$ be the number of neighbors of $v_{c,i}$ having color c'. Let us fix an ordering of these neighbors and let $n(c,i,c',j)$ be the j-th neighbor of $v_{c,i}$ having color c' in this ordering.

Set $k' := \binom{k}{2}$. We construct a satisfiable 2-CNF formula F and a set of literals L such that deletion of k' blocks makes F satisfiable by an assignment including L if and only if G has a multicolored clique of size k. For every $1 \leq c \leq k$ and $0 \leq i \leq n_c$, we introduce a variable $x_{c,i}$. For every $1 \leq c, c' \leq k$, $c \neq c'$, $1 \leq i \leq n_c$, $0 < j < d(c,i,c')$, we introduce a variable $y_{c,i,c',j}$. For ease of notation, we define $y_{c,i,c',0} := x_{c,i-1}$ and $y_{c,i,c',d(c,i,c')} := x_{c,i}$ (note that the second index of $x_{c,i}$ can be 0, while it is at least 1 for $y_{c,i,c',j}$).

The clauses of F are the union of the disjoint blocks B_e for each edge e of G. Suppose that edge e connects $v_{c,i}$ and $v_{c',i'}$ and $n(c,i,c',j) = v_{c',i'}$ as well as $n(c',i',c,j') = v_{c,i}$ hold for some j, j'. Block B_e consists of the clauses $(y_{c,i,c',j-1} \rightarrow y_{c,i,c',j})$ and $(y_{c',i',c,j'-1} \rightarrow y_{c',i',c,j'})$. It is easy to see that F is satisfiable by setting all the variables to 0. The set L of literals is defined as follows: $L = \{x_{c,0} | 1 \leq c \leq k\} \cup \{\neg x_{c,n_c} | 1 \leq c \leq k\}$.

Before introducing the formal proof, we give an intuitive explanation. Formula F can be considered as containing k *components*, one for each color. The component corresponding to a color c consists of n_c *fragments*, one for each vertex colored in c. The fragment corresponding to vertex $v_{c,i}$ consists of $k-1$ sets of implications one for each $c' \neq c$, and it is convenient to imagine that each such set is a *sequence of implications* of the form $y_{c,i,c',0} \rightarrow y_{c,i,c',1} \rightarrow \cdots \rightarrow y_{c,i,c',d_{c,i,c'}}$. Due to the settings $y_{c,i,c',0} := x_{c,i-1}$ and $y_{c,i,c',d(c,i,c')} := x_{c,i}$ and the literals of L, F can be made satisfiable if and only if for each component of color c we identify a fragment corresponding to vertex $v_{c,i}$ and remove a clause from each sequence of implications of this fragment. That is, to make the formula satisfiable, it is necessary and sufficient to remove $k(k-1)$ clauses. Since we want to remove only $k' = k(k-1)/2$ blocks, we have to find out such fragments whose sequences of implications can be partitioned into pairs so that for each pair there is a block 'covering' both sequences of this pair. The blocks are designed in such a way that two fragments can be 'connected' by at most one block and even this can happen only in the case when the vertices corresponding to these fragments are adjacent. It follows that removal of k' blocks can make F' satisfiable if and only if the considered fragments correspond to a set of mutually adjacent vertices, one for each color, i.e., a multicolored clique.

Now we introduce the formal proof. Suppose that G has a multicolored clique K of size k; let v_{c,i_c} be the vertex of K having color c. For every $1 \leq c, c' \leq k$, $c \neq c'$, there is an integer $j_{c,c'}$ such that $n(c,i_c,c',j_{c,c'}) = v_{c',i_{c'}}$. Let F' be the formula obtained from F by deleting the blocks corresponding to the edges of K.

Consider a set L^* of literals of variables $y_{c,i,c',j}$ (for every $1 \leq c, c' \leq k$, $c \neq c'$, $1 \leq i \leq n_c$, $0 \leq j \leq d(c,i,c')$) created as follows: $y_{c,i,c',j} \in L^*$ if $i < i_c$

(independently of the values of c' and j), or $i = i_c$, provided that $j < j_{c,c'}$. Otherwise $\neg y_{c,i,c',j} \in L^*$.

Since L^* contains literals of all variables $y_{c,i,c',0}$ and $y_{c,i,c',d(c,i,c')}$, it in fact contains the literals of all variables $x_{c,i}$. Let us verify that all variables $x_{c,i}$ are consistently assigned. In addition, to ensure that $L \subseteq L^*$, we check that L^* contains $x_{c,0}$ and $\neg x_{c,n_c}$ for $1 \leq c \leq k$. Consider first a variable $x_{c,0}$. By definition its value equals the value of $y_{c,1,c',0}$ for all possible values of c'. If $i_c > 1$ then $y_{c,1,c',0} \in L^*$. Otherwise, $i_c = 1$ and in this case, as $j_{c,c'} \geq 1$, it follows again that $y_{c,1,c',0} \in L^*$. Thus we have verified the validity of assigning $x_{c,0}$. Now, consider x_{c,n_c}. By definition, $x_{c,n_c} = y_{c,n_c,c',d(c,n_c,c')}$ for all possible values of c'. Clearly $n_c \geq i_c$. If $n_c > i_c$ then $\neg y_{c,n_c,c',d(c,n_c,c')} \in L^*$. Otherwise, $\neg y_{c,n_c,c',d(c,n_c,c')} \in L^*$ because $n_c \geq j_{c,c'}$, implying the validity of assigning x_{c,n_c}. Finally, consider $x_{c,i}$ when $0 < i < n_c$. The value of $x_{c,i}$ is equal to the value of $y_{c,i,c',d(c,i,c')}$ and the value of $y_{c,i+1,c',0}$ for all the values of c'. Using the description of L^*, it is not hard to verify the consistency of instantiation of $x_{c,i}$ by considering first $i < i_c$ then $i = i_c$ and finally $i > i_c$. It remains to verify that each clause of F' is satisfied by L^*. Assume that a clause $(y_{c,i,c',j-1} \to y_{c,i,c',j})$ is not satisfied. This is only possible if $i = i_c$ and $j - 1 < j_{c,c'}$ and $j \geq j_{c,c'}$, i.e., $j = j_{c,c'}$, but in that case the clause was deleted from F', a contradiction.

For the other direction of the proof, suppose that it is possible to obtain, by the deletion of at most k' blocks, a formula F' that has a satisfying assignment L^* such that $L \subseteq L^*$. In particular this means that $x_{c,0} \in L^*$ and $\neg x_{c,n_c} \in L^*$ for every $1 \leq c \leq k$. Thus for every $1 \leq c \leq k$, there is a smallest $1 \leq i_c \leq n_c$ such that $x_{c,i_c-1} \in L^*$ and $\neg x_{c,i_c} \in L^*$. We claim that $K := \{v_{c,i_c} : 1 \leq c \leq k\}$ is a clique of size k. Let E^* be the set of edges corresponding to the deleted blocks. We show that for every $1 \leq c, c' \leq k$ and $c \neq c'$, v_{c,i_c} is adjacent to a c'-colored vertex in $G[E^*]$. It follows that $G[E^*]$ has k vertices of degree $k-1$. On the other hand $|E^*| = \binom{k}{2}$. It only possible if $G[E^*]$ is a complete graph and $V(G[E^*]) = K$. In other words, K is a clique of size k in G.

Suppose that v_{c,i_c} is not adjacent to a c'-colored vertex in $G[E^*]$. That is, E^* does not contain any of the edges $\{v_{c,i_c}, n(c, i_c, c', j)\}$ for $1 \leq j \leq d(c, i_c, c')$. This means that none of the clauses $(y_{c,i_c,c',j-1} \to y_{c,i_c,c',j})$ $(1 \leq j \leq d(c, i_c, c'))$ are deleted. Since $d(c, i_c, c') \geq 1$, these clauses ensure that if $y_{c,i_c,c',0} = 1$, then $y_{c,i_c,c',d(c,i_c,c')} = 1$ as well. However, by the definition of i_c, we have $y_{c,i_c,c',0} = x_{c,i_c-1} = 1$ and $y_{c,i_c,c',d(c,i_c,c')}) = x_{c,i_c} = 0$, which gives a contradiction. □

References

1. Bansal, N., Blum, A., Chawla, S.: Correlation clustering. Machine Learning 56(1-3), 89–113 (2004)
2. Bodlaender, H.L., Fellows, M.R., Heggernes, P., Mancini, F., Papadopoulos, C., Rosamond, F.: Clustering with partial information. In: MFCS 2008: Proceedings of the 33rd international symposium on Mathematical Foundations of Computer Science, pp. 144–155. Springer, Heidelberg (2008)

3. Bousquet, N., Daligault, J., Thomasse, S., Yeo, A.: A polynomial kernel for multicut in trees. In: Albers, S., Marion, J.-Y. (eds.) 26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009). Leibniz International Proceedings in Informatics, vol. 3, pp. 183–194. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany (2009)
4. Cai, L., Huang, X.: Fixed-parameter approximation: conceptual framework and approximability results. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 96–108. Springer, Heidelberg (2006)
5. Chawla, S., Krauthgamer, R., Kumar, R., Rabani, Y., Sivakumar, D.: On the hardness of approximating multicut and sparsest-cut. Comput. Complexity 15(2), 94–114 (2006)
6. 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)
7. Chen, Y., Grohe, M., Grüber, M.: On parameterized approximability. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 109–120. Springer, Heidelberg (2006)
8. Demaine, E.D., Emanuel, D., Fiat, A., Immorlica, N.: Correlation clustering in general weighted graphs. Theor. Comput. Sci. 361(2-3), 172–187 (2006)
9. Downey, R., Fellows, M., McCartin, C.: Parameterized approximation algorithms. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 121–129. Springer, Heidelberg (2006)
10. Downey, R.G., Fellows, M.R.: Parameterized Complexity. Monographs in Computer Science. Springer, New York (1999)
11. Fellows, M.R., Hermelin, D., Rosamond, F.A., Vialette, S.: On the parameterized complexity of multiple-interval graph problems. Theor. Comput. Sci. 410(1), 53–61 (2009)
12. Flum, J., Grohe, M.: Parameterized Complexity Theory. Springer, Berlin (2006)
13. Garg, N., Vazirani, V.V., Yannakakis, M.: Approximate max-flow min-(multi)cut theorems and their applications. SIAM J. Comput. 25(2), 235–251 (1996)
14. Guo, J., Hüffner, F., Kenar, E., Niedermeier, R., Uhlmann, J.: Complexity and exact algorithms for vertex multicut in interval and bounded treewidth graphs. European J. Oper. Res. 186(2), 542–553 (2008)
15. Guo, J., Niedermeier, R.: Fixed-parameter tractability and data reduction for multicut in trees. Networks 46(3), 124–135 (2005)
16. Hüffner, F., Niedermeier, R., Wernicke, S.: Techniques for practical fixed-parameter algorithms. The Computer Journal 51(1), 7–25 (2008)
17. Khot, S.: On the power of unique 2-prover 1-round games. In: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing, pp. 767–775. ACM, New York (2002) (electronic)
18. Liu, Y., Lu, S., Chen, J., Sze, S.-H.: Greedy localization and color-coding: improved matching and packing algorithms. In: Bodlaender, H.L., Langston, M.A. (eds.) IWPEC 2006. LNCS, vol. 4169, pp. 84–95. Springer, Heidelberg (2006)
19. Marx, D.: Parameterized graph separation problems. Theoretical Computer Science 351(3), 394–406 (2006)
20. Marx, D.: Chordal deletion is fixed-parameter tractable. To appear in Algorithmica (2008)
21. Marx, D.: Parameterized complexity and approximation algorithms. The Computer Journal 51(1), 60–78 (2008)

22. Niedermeier, R.: Invitation to Fixed-Parameter Algorithms. Oxford Lecture Series in Mathematics and Its Applications, vol. 31 (2006)
23. Razgon, I., O'Sullivan, B.: Almost 2-sat is fixed-parameter tractable. CoRR, abs/0801.1300 (2008)
24. Razgon, I., O'Sullivan, B.: Almost 2-sat is fixed-parameter tractable (extended abstract). In: Aceto, L., Damgård, I., Goldberg, L.A., Halldórsson, M.M., Ingólfsdóttir, A., Walukiewicz, I. (eds.) ICALP 2008, Part I. LNCS, vol. 5125, pp. 551–562. Springer, Heidelberg (2008)
25. Reed, B., Smith, K., Vetta, A.: Finding odd cycle transversals. Operations Research Letters 32(4), 299–301 (2004)
26. Xiao, M.: Algorithms for multiterminal cuts. In: Hirsch, E.A., Razborov, A.A., Semenov, A., Slissenko, A. (eds.) Computer Science – Theory and Applications. LNCS, vol. 5010, pp. 314–325. Springer, Heidelberg (2008)