Approximately Counting Locally-Optimal Structures

Leslie Ann Goldberg[1,⋆], Rob Gysel[2], and John Lapinskas[1,⋆]

[1] Department of Computer Science, University of Oxford, UK.
[2] Department of Computer Science, University of California, Davis, US.

Abstract. A *locally-optimal* structure is a combinatorial structure that cannot be improved by certain (greedy) local moves, even though it may not be globally optimal. An example is a maximal independent set in a graph. It is trivial to construct an independent set in a graph. It is easy to (greedily) construct a maximal independent set. However, it is NP-hard to construct a globally-optimal (maximum) independent set. This situation is typical. Constructing a locally-optimal structure is somewhat more difficult than constructing an arbitrary structure, and constructing a globally-optimal structure is more difficult than constructing a locally-optimal structure. The same situation arises with listing. The differences between the problems become obscured when we move from listing to counting because nearly everything is #P-complete. However, we highlight an interesting phenomenon that arises in approximate counting, where approximately counting locally-optimal structures is apparently more difficult than approximately counting globally-optimal structures. Specifically, we show that counting maximal independent sets is complete for #P with respect to approximation-preserving reductions, whereas counting all independent sets, or counting maximum independent sets is complete for an apparently smaller class, $\#\mathrm{RH}\Pi_1$ which has a prominent role in the complexity of approximate counting. Motivated by the difficulty of approximately counting maximal independent sets in bipartite graphs, we also study counting problems involving minimal separators and minimal edge separators (which are also locally-optimal structures). Minimal separators have applications via fixed-parameter-tractable algorithms for constructing triangulations and phylogenetic trees. Although exact (exponential-time) algorithms exist for listing these structures, we show that the counting problems are as hard as they could possibly be. All of the exact counting problems are #P-complete, and all of the approximation problems are complete for #P with respect to approximation-preserving reductions. The full version from `http://arxiv.org/abs/1411.6829` is attached as an appendix. Theorem-numbering here matches the full version.

⋆ The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007–2013) ERC grant agreement no. 334828. The paper reflects only the authors' views and not the views of the ERC or the European Commission. The European Union is not liable for any use that may be made of the information contained therein.

1 Introduction

A *locally-optimal* structure is a combinatorial structure that cannot be improved by certain (greedy) local moves, even though it may not be globally optimal. An example is a maximal independent set in a graph. It is trivial to construct an independent set in a graph (for example, the singleton set containing any vertex is an independent set). It is easy to construct a maximal independent set (the greedy algorithm can do this). However, it is NP-hard to construct a globally-optimal independent set, which in this case means a maximum independent set. In the setting in which we work, this situation is typical. Constructing a locally-optimal structure is somewhat more difficult than constructing an arbitrary structure, and constructing a globally-optimal structure is more difficult than constructing a locally-optimal structure. For example, in bipartite graphs, it is trivial to construct an independent set, easy to (greedily) construct a maximal independent set, and more difficult to construct a maximum independent set (even though this can be done in polynomial time). This general phenomenon has been well-studied. In 1987, Johnson, Papadimitriou and Yannakakis [18] defined the complexity class PLS (for "polynomial-time local search") that captures local optimisation problems where one iteration of the local search algorithm takes polynomial time. As the authors point out, practically all empirical evidence leads to the conclusion that finding locally-optimal solutions is much easier than solving NP-hard problems, and this is supported by complexity-theoretic evidence, since a problem in PLS cannot be NP-hard unless NP=co-NP. An example that illustrates this point is the graph partitioning problem. For this problem it is trivial to find a valid partition, and it is NP-hard to find a globally-optimal (minimum weight) partition but Schäffer and Yannakakis [22] showed that finding a locally-optimal solution (with respect to a particular swapping-dynamics) is PLS-complete, so is presumably of intermediate complexity.

For listing combinatorial structures, a similar pattern emerges. Self-reducibility gives a nearly-trivial polynomial-space polynomial-delay algorithm for listing the independent sets of a graph [13]. A polynomial-space polynomial-delay algorithm for listing the *maximal* independent sets exists, due to Tsukiyama et al. [25], but it is more complicated. On the other hand, there is no polynomial-space polynomial-delay algorithm for listing the *maximum* independent sets unless P=NP. There is a polynomial-space polynomial-delay algorithm for listing the maximum independent sets of a bipartite graph [19], but this is substantially more complicated than any of the previous algorithms.

When we move from constructing and listing to counting, these differences become obscured because nearly everything is #P-complete. For example, counting independent sets, maximal independent sets, and maximum independent sets of a graph are all #P-complete problems, even if the graph is bipartite [26]. Furthermore, even *approximately* counting independent sets, maximal independent sets, and maximum independent sets of a graph are all #P-complete with respect to approximation-preserving reductions [8].

The purpose of this paper is to highlight an interesting situation that arises in approximate counting where, contrary to the situations that we have just discussed, approximately counting locally-optimal structures is apparently more difficult than counting globally-optimal structures.

In order to explain the result, we first briefly summarise what is known about the complexity of approximate counting within #P. This will be explained in more detail in Sect. 2. There are three relevant complexity classes — the class containing problems which admit a fully-polynomial randomised approximation scheme (FPRAS), the class $\#\mathrm{RH}\Pi_1$, and #P itself. Dyer et al. [8] showed that #BIS, the problem of counting independent sets in a bipartite graph, is complete for $\#\mathrm{RH}\Pi_1$ with respect to approximation-preserving (AP) reductions and that #IS, the problem of counting independent sets in a (general) graph is #P-complete with respect to AP-reductions. It is generally believed that the $\#\mathrm{RH}\Pi_1$-complete problems are not FPRASable, but that they are of intermediate complexity, and are not as difficult to approximate as the problems which are #P-complete with respect to AP-reductions. Many problems have subsequently been shown to be $\#\mathrm{RH}\Pi_1$-complete and #P-complete with respect to AP-reductions. More examples will be given in Sect. 2.

We can now describe the interesting situation which emerges with respect to independent sets in bipartite graphs. Dyer et al. [8] showed that approximately counting independent sets and approximately counting *maximum* independent sets are both $\#\mathrm{RH}\Pi_1$-complete with respect to AP-reductions. Thus, the pattern outlined above would suggest that approximately counting *maximal* independent sets in bipartite graphs ought to also be $\#\mathrm{RH}\Pi_1$-complete. However, we show (Theorem 1, below) that approximately counting *maximal* independent sets in bipartite graphs is actually #P-complete with respect to AP-reductions. Thus, either $\#\mathrm{RH}\Pi_1$ and #P are equivalent in approximation complexity (contrary to the picture that has been emerging in earlier papers), or this is a scenario where approximately counting locally-optimal structures is actually more difficult than approximately counting globally-optimal ones.

Motivated by the difficulty of approximately counting maximal independent sets in bipartite graphs, we also study the problem of approximately counting other locally-optimal structures that arise in algorithmic applications. The problem of counting the *minimal separators* of a graph arises in diverse applications from triangulation theory to phylogeny construction in computational biology. A minimal separator is a particular type of vertex separator. Definitions are given in Sect. 1.1. Algorithmic applications arise because fixed-parameter-tractable algorithms are known whose running time is polynomial in the number of minimal separators of a graph. These algorithms were originally developed by Bouchitté and Todinca [5, 6] (and improved in [9]) to exactly solve the *treewidth* and *minimum-fill* problems; the former is widely studied due to its applicability to a number of other NP-complete problems [4]. The technique has recently been generalized [12] to cover problems including *treecost* [2] and *treelength* [21]. The algorithm can also be used to find a minimum-width *tree-decomposition* of a graph, a key data structure that is used to solve a variety of NP-complete problems in polynomial time when the width of the tree-decomposition is fixed [4]. In recent years, much research has been dedicated to exact-exponential algorithms for treewidth [3], the fastest of which [10] has running time closely connected to the number of minimal separators in the graph. Indeed, if the graph has M minimal separators, then the running time is bounded from above by a polynomial multiple of M^2 and from below by a polynomial multiple of M.

Bouchitté and Todinca's approach has also recently been applied to solve the *perfect phylogeny problem* and two of its variants [17]. In this problem, the input is a set of phylogenetic characters, each of which may be viewed as a partition of a subset of *species*. The goal is to find a phylogenetic tree such that every character is *convex* on that tree — that is, the parts of each partition form connected subtrees that do not overlap. Such a tree is called a *perfect phylogeny*.

In all of these applications, it would be useful to count the minimal separators of a graph, since this would give an a priori bound on the algorithms' running times. Thus, we consider the difficulty of this problem, whose complexity was previously unresolved, even in terms of exact computation. Theorem 2 shows that counting minimal separators is #P-complete, both with respect to Turing reductions (for exact computation) and with respect to AP-reductions. Thus, this problem is as difficult to approximate as any problem in #P.

Motivated by applications to treewidth [9] and phylogeny [16, 17], we also consider various heuristic approximations to the minimal separator problem. The number of inclusion-minimal separators is a natural choice for a lower bound on the number of minimal separators. Conversely, the number of (s, t)-minimal separators, taken over all vertices s and t, is a natural choice for an upper bound on the number of minimal separators. Theorem 2 shows that both of these bounds are difficult to compute, either exactly or approximately. Finally, the number and structure of 2-component minimal separators is important in computational biology. These separators arise naturally in the problem of determining whether a subset of "quartet phylogenies" can be assembled uniquely [16]. Thus, we study the problem of counting such minimal separators. Theorem 2 shows that they are complete for #P with respect to exact and approximate computation.

Our new results about counting minimal vertex separators are obtained by first considering the problem of counting minimal edge separators. These locally-optimal structures are also known as *bonds* or *minimal cuts*, and are well-studied in other contexts — see e.g. Diestel [7]. Theorem 3 gives the first hardness result for counting these structures, either exactly or approximately.

1.1 Detailed Results

We now give formal definitions of the problems that we study, and state our results precisely. Our first result is that counting maximal independent sets in a bipartite graph is #P-complete with respect to Approximation-Preserving (AP) reductions (even though counting maximum independent sets in bipartite graphs is only $\#\mathrm{RH}\Pi_1$-complete with respect to these reductions). (AP-reductions are discussed in Sect. 2.)

Definition 1 *Let G be a graph. We say that an independent set $X \subseteq V(G)$ of G is* maximal *if no proper superset of X is an independent set of G.*

Problem 1 #MaximalBIS.
Input: *A bipartite graph* G*.*
Output: *The number of maximal independent sets of* G*.*

The following theorem is proved in Sect. 3.

Theorem 1 #MaximalBIS $\equiv_{\mathrm{AP}}$ #SAT.

Next we state our results relating to counting minimal separators.

Definition 2 *Let* $G = (V, E)$ *be a graph, and let* $X \subseteq V$*. For distinct* $s, t \in V$*, we say* X *is an* (s, t)-separator *of* G *if* s *and* t *lie in different components of* $G - X$*. If, in addition, no proper subset of* X *is an* (s, t)*-separator of* G*, then we say that* X *is a* minimal (s, t)-separator *of* G*. We say* X *is a* minimal separator *of* G *if* X *is a minimal* (a, b)*-separator of* G *for some* $a, b \in V$*.*

For example, let $V = \{1, 2, 3, 4, 5\}$, let $E = \{\{1, 2\}, \{2, 3\}, \{3, 4\}, \{4, 1\}, \{1, 5\}\}$, and let G be the graph (V, E). G is a four-edge cycle with a pendant vertex. Then $\{1, 3\}$ is a minimal separator of G since it is a minimal $(2, 4)$-separator.

We have already seen that algorithms for counting and approximately counting minimal separators are useful in algorithmic applications. There is also lots of existing work on listing minimal separators. Given a graph G, let n be the number of vertices and let m be the number of edges. Kloks and Kratsch, and independently, Sheng and Liang, showed how to compute all (s, t)-minimal separators in $O(n^3)$ time per (s, t)-minimal separator [20,23]. Computing all minimal separators by computing (s, t)-minimal separators for each possible vertex pair in this way leads to an $O(n^5)$ time per minimal separator listing algorithm. Berry, Bordat, and Cogis [1] improved this approach, computing all minimal separators in $O(n^3)$ time per minimal separator. Each of these algorithms require storing minimal separators in an adequate data structure. Takata's algorithm [24] generates the set of minimal separators in $O(n^3 m)$ time per minimal separator but linear space. A graph has at most $O(1.6181^n)$ minimal separators [11]. We study the following computational problems, based on our desire to count and to approximately count minimal separators.

Problem 2 #(s, t)-BiMinimalSeps.
Input: *A bipartite graph* G *and two vertices* $s, t \in V(G)$*.*
Output: *The number of minimal* (s, t)*-separators of* G*, denoted by* $\mathrm{MS}(G, s, t)$*.*

Problem 3 #BiMinimalSeps.
Input: *A bipartite graph* G*.*
Output: *The number of minimal separators of* G*, denoted by* $\mathrm{MS}(G)$*.*

Theorem 2 below shows that both problems are #P-complete to solve exactly and are complete for #P with respect to approximation-preserving reductions.

Motivated by considerations in phylogeny [16] we also consider various heuristic approximations to the minimal separator problem. We start by defining the notion of an inclusion-minimal separator, since the number of these is a natural lower bound for the number of minimal separators.

Definition 4 *Let G be a graph. A minimal separator X of G is said to be an* inclusion-minimal separator *if no proper subset of X is a minimal separator.*

In the five-vertex example above, the minimal separator $\{1, 3\}$ is not an inclusion-minimal separator since $\{1\} \subset \{1, 3\}$ is a minimal $(5, 4)$-separator. However $\{1\}$ is an inclusion-minimal separator. We consider the following computational problem.

Problem 4 #BiInclusionMinimalSeps.
Input: *A bipartite graph G.*
Output: *The number of inclusion-minimal separators of G, denoted by* $\mathrm{IMS}(G)$.

We also consider the problem of counting 2-component minimal separators since these arise in phylogenetic assembly.

Problem 5 #(s, t)-BiConnMinimalSeps.
Input: *A bipartite graph G and two vertices $s, t \in V(G)$.*
Output: *The number of minimal (s, t)-separators X of G such that $G - X$ has exactly two connected components.*

Problem 6 #BiConnMinimalSeps.
Input: *A bipartite graph G.*
Output: *The number of minimal separators X of G such that $G - X$ has exactly two connected components.*

Our main theorem about minimal separators shows that all of these problems are #P-complete and are also complete for #P with respect to AP-reductions.

Theorem 2 *The problems* #(s, t)-BiMinimalSeps, #BiMinimalSeps, #(s, t)-BiConnMinimalSeps, #BiConnMinimalSeps *and* #BiInclusionMinimalSeps *are* #P*-complete and are equivalent to* #SAT *under AP-reduction.*

In order to prove Theorem 2, we first study algorithmic problems related to other natural locally-optimal structures, namely minimal edge-separators. These problems are also interesting for their own sake.

Definition 5 *Let $G = (V, E)$ be a graph, and let $F \subseteq E$. For distinct $s, t \in V$, we say F is an* (s, t)-edge separator *of G if s and t lie in different components of $G - F$. If in addition no proper subset of F is an (s, t)-edge separator of G then we say that F is a* minimal (s, t)-edge separator *of G. We say F is a* minimal edge separator *of G if it is a minimal (a, b)-edge separator for some $a, b \in V$.*

There is no need to define inclusion-minimal edge separators, since these turn out to be the same as minimal edge separators (unlike the situation for vertex separators). We show that both of the following problems are #P-complete with respect to AP-reductions, and that both are #P-complete to compute exactly.

Problem 7 #(s,t)-BiMinimalEdgeSeps.
Input: *A bipartite graph G and two vertices $s, t \in V(G)$.*
Output: *The number of minimal (s,t)-edge separators of G, denoted by* $\mathrm{MES}(G, s, t)$.

Problem 8 #BiMinimalEdgeSeps.
Input: *A bipartite graph G.*
Output: *The number of minimal edge separators of G, denoted* $\mathrm{MES}(G)$.

Theorem 3 *The problems* #BiMinimalEdgeSeps *and* #(s,t)-BiMinimalEdgeSeps *are* #P*-complete and are equivalent to* #SAT *under AP-reduction.*

In the full version we also study two other locally-optimal structures related to maximal independent sets in bipartite graphs. Theorem 4 shows that counting *dominating sets* in bipartite graphs is also #P-hard with respect to AP-reductions. Also, maximal independent sets in bipartite graphs can be represented as unions of sets, so (Theorems 5 and 6) a set union problem is also #P-hard with respect to AP-reductions, and so is its inverse.

2 Preliminaries

Most of our notation is standard, and we therefore defer it to Sect. 2 of the full version. The notions of a *fully polynomial randomised approximation scheme* (or *FPRAS*) and an *approximation-preserving reduction* (or *AP-reduction*) are standard in the field. If there is an AP-reduction from f to g, we write $f \leq_{\mathrm{AP}} g$.

Dyer et al. [8] studied counting problems in #P and identified three classes of counting problems that are interreducible under AP-reductions. The first class, containing the problems that have an FPRAS, are trivially equivalent under AP-reduction since all the work can be embedded into the reduction (which declines to use the oracle). The second class is the equivalence class of #SAT, the problem of counting satisfying assignments to a Boolean formula in CNF, under AP-reduction. These problems are complete for #P with respect to AP-reductions. Zuckerman [28] has shown that #SAT cannot have an FPRAS unless RP = NP, so the same is true of any problem to which #SAT is AP-reducible.

The third class appears to be of intermediate complexity. It contains all of the counting problems expressible in a certain logically-defined complexity class, #RHΠ_1. Typical complete problems include counting the downsets in a partially ordered set [8], computing the partition function of the ferromagnetic Ising model with local external magnetic fields [14], and counting the independent sets in a bipartite graph, which is formally defined as follows.

Problem 12 #BIS.
Input: *A bipartite graph G.*
Output: *The number of independent sets in G, denoted by* $\mathrm{IS}(G)$.

In [8] it was shown that #BIS is complete for the logically-defined complexity class #RHΠ_1 with respect to AP-reductions. Goldberg and Jerrum [15] have conjectured that there is no FPRAS for #BIS. Early indications point to the fact that it may be of intermediate complexity, between the FPRASable problems and those that are complete for #P with respect to AP-reductions.

3 Hardness of #MaximalBIS

We first prove that #MaximalBIS is complete for #P with respect to AP-reductions. We reduce from the well-known problem of counting independent sets in an arbitrary graph.

Problem 13 #IS.
Input: *A graph G.*
Output: *The number of independent sets in G.*

Dyer et al. [8, Theorem 3] shows that #IS is complete for #P with respect to AP-reductions. Using this we can now prove Theorem 1.

Proof. Since #MaximalBIS is in #P, #MaximalBIS $\leq_{\mathrm{AP}}$ #SAT follows from [8]. We will show #IS $\leq_{\mathrm{AP}}$ #MaximalBIS. Let $\mathrm{MIS}(G)$ denote the number of maximal independent sets in a graph G. Let $G = (V, E)$ be an instance of #IS. Without loss of generality let $V = [n]$ for some $n \in \mathbb{N}$, let $m = |E|$, and let $t = n+2$. We shall construct an instance G' of #MaximalBIS with the property that $\mathrm{IS}(G) \leq \mathrm{MIS}(G')/2^{tm} \leq \mathrm{IS}(G)+\frac{1}{4}$, which will be sufficient for the reduction. See Fig. 1 for an example.

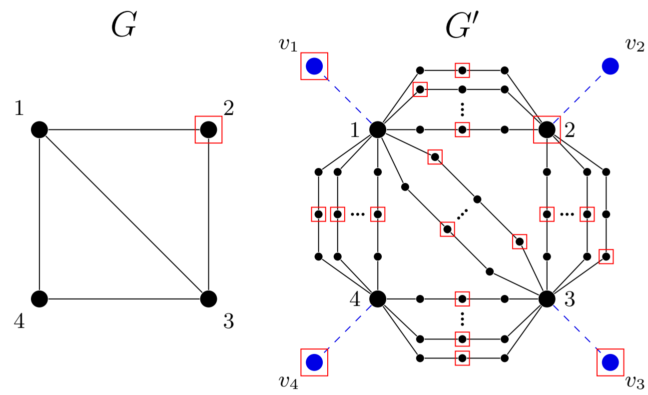

Fig. 1. An example of the reduction from an instance G of #IS to an instance G' of #MaximalBIS used in the proof of Theorem 1. The boxes around vertices indicate a non-maximal independent set in G and one of its maximal counterparts in G'. Note that the presence of v_4 ensures that vertex 4 has an occupied neighbour in G'.

Informally, we obtain a bipartite graph G' (an instance of #MaximalBIS) from G by first t-thickening and then 4-stretching each of G's edges and by also adding a bristle to each of G's vertices. Formally, we define G' as follows. For each $e \in E$ let X_e, Y_e and Z_e be sets of t vertices. We require all of these sets to be disjoint from each other and from $[n]$. Write $X_e = \{x_e^k \mid k \in [t]\}$, $Y_e = \{y_e^k \mid k \in [t]\}$, and $Z_e = \{z_e^k \mid k \in [t]\}$. Also, let $W = \bigcup_{e \in E} X_e \cup Y_e \cup Z_e$. Let $V^* = \{v_1, \ldots, v_n\}$ be a set of distinct vertices which is disjoint from $[n] \cup W$. Then we define $V(G') = [n] \cup V^* \cup W$ and

$$E(G') = \{\{i, v_i\} \mid i \in [n]\} \cup \bigcup_{\substack{e=\{i,j\}\in E \\ i<j \\ k\in[t]}} \{\{i, x_e^k\}, \{x_e^k, y_e^k\}, \{y_e^k, z_e^k\}, \{z_e^k, j\}\} \ .$$

Let $S \subseteq [n]$ be arbitrary. We shall determine the number $\mathrm{MIS}_S(G')$ of maximal independent sets $T \subseteq V(G')$ with $T \cap [n] = S$, and thereby bound $\mathrm{MIS}(G')$.

First, note that for every $S \subseteq [n]$, the set $S \cup \{v_i \in V^* \mid i \notin S\} \cup \bigcup_e Y_e$ is a maximal independent set of G', so $\mathrm{MIS}_S(G')$ is non-zero. Also, if T is a maximal independent set of G' and $T \cap [n] = S$ then $T \cap V^* = \{v_i \in V^* \mid i \notin S\}$. In particular, this implies that every unoccupied vertex in $[n]$ has an occupied neighbour in V^*.

Consider an edge $e = \{i, j\} \in E$, where $i < j$, and a value $k \in [t]$. If T is a maximal independent set of G' containing both i and j then $T \cap \{x_e^k, y_e^k, z_e^k\} = \{y_e^k\}$. However, if T is a maximal independent set of G' containing i but not j then $T \cap \{x_e^k, y_e^k, z_e^k\}$ can either be $\{y_e^k\}$ or $\{z_e^k\}$. This choice can be made independently for each $k \in [t]$. Similarly, if T is a maximal independent set of G' containing neither i nor j then $T \cap \{x_e^k, y_e^k, z_e^k\}$ can either be $\{x_e^k, z_e^k\}$, or $\{y_e^k\}$.

Given $S \subseteq [n]$, let $\mu(S)$ be the number of edges of G with both endpoints in S. We conclude from the previous observations that $\mathrm{MIS}_S(G') = 2^{(m-\mu(S))t}$ so $\mathrm{MIS}(G') = \sum_{S \subseteq [n]} 2^{(m-\mu(S))t}$. Since each independent set S of G has $\mu(S) = 0$, $\mathrm{MIS}(G') \geq \mathrm{IS}(G) 2^{mt}$. Furthermore, since there are at most 2^n sets $S \subseteq [n]$ that are not independent sets of G, and each of these has $\mu(S) \geq 1$, we have

$$\mathrm{IS}(G) \leq \frac{\mathrm{MIS}(G')}{2^{tm}} \leq \mathrm{IS}(G) + 2^n 2^{-t} = \mathrm{IS}(G) + \frac{1}{4} \ . \tag{1}$$

Equation (1) implies that there is an AP-reduction from #IS to #MaximalBIS. The details of the reduction showing how to tune the accuracy parameter in the oracle call for approximating $\mathrm{MIS}(G')$ in order to get a sufficiently good approximation to $\mathrm{IS}(G)$ are exactly as in the proof of Theorem 3 of [8]. □

4 Minimal separator problems

The definition of minimal edge separator generalises naturally to multigraphs (see the full version). In order to prove Theorems 2 and 3 we consider two intermediate problems related to counting *maximum* minimal edge separators.

Problem 14 #LargeMinimalEdgeSeps.
Input: *A multigraph G and the maximum cardinality x of any minimal edge separator in G.*
Output: *The number of minimal edge separators of G with maximum cardinality, denoted by* LMES(G).

Problem 15 #(s,t)-LargeMinimalEdgeSeps.
Input: *A multigraph G, two distinct vertices $s, t \in V$, and the maximum cardinality y of any minimal (s,t)-edge separator in G.*
Output: *The number of minimal (s,t)-edge separators of G with maximum cardinality, which we denote by* LMES(G, s, t).

The following proposition, due to Whitney [27], implies that minimal edge separators can be expressed in terms of vertex cuts.

Proposition 12 *Let $G = (V, E)$ be a connected multigraph. Then a multiset $F \subseteq E$ is a minimal edge separator of G if and only if $G - F$ has exactly two non-empty components, and F is the multiset of edges between them.* ☐

Since MAX-CUT is an intractable optimisation problem, we can show (see Lemma 16 of the full version) that the problems #LargeMinimalEdgeSeps and #(s,t)-LargeMinimalEdgeSeps are #SAT-hard to approximate and are #P-complete. In order to prove Theorem 3 it is then necessary to relate these problems to #BiMinimalEdgeSeps and #(s,t)-BiMinimalEdgeSeps. This is achieved by the following technical lemma, which is the heart of the proof.

Lemma 17 *Let $G = (V, E)$ be a connected multigraph, writing $n = |V|$ and $m = |E|$. Suppose (G, x) is an instance of* #LargeMinimalEdgeSeps, *and (G, s, t, y) is an instance of* #(s,t)-LargeMinimalEdgeSeps. *Let $k = \lceil m + \log_2(m) + 10 \rceil$. Then there exists a graph G' such that the following properties hold.*

(i) *G' is bipartite, $V \subseteq V(G')$, and $|V(G')| \leq |E|k + |V|$.*
(ii) $\mathrm{LMES}(G) \leq \mathrm{MES}(G')/2^{kx} \leq \mathrm{LMES}(G) + \frac{1}{4}$.
(iii) $\mathrm{LMES}(G, s, t) \leq \mathrm{MES}(G', s, t)/2^{ky} \leq \mathrm{LMES}(G, s, t) + \frac{1}{4}$.

The bipartite graph G' is constructed from G by first k-thickening and then 2-stretching each edge of G. The construction works because almost every minimal edge separator of G' has the following properties: (a) it contains at most one of the edges of a 2-path corresponding to an edge of G, and (b) if there is an intersection, then it intersects every such 2-path (so can be viewed as cutting the edge of G). It turns out that any minimal edge separator F of G corresponds to precisely $2^{k|F|}$ such minimal edge separators of G', and there aren't too many other minimal edge separators of G', so the construction goes through.

In order to prove Theorem 2, which is about vertex separators and not about edge separators, we need a similar, but more difficult, lemma.

Lemma 18 *Let $G = (V, E)$ be a connected multigraph, writing $n = |V|$ and $m = |E|$. Suppose (G, x) is an instance of* #LargeMinimalEdgeSeps, *and (G, s, t, y) is an instance of* #(s,t)-LargeMinimalEdgeSeps. *Let $k = \lceil m + n + \log_3(n^2) + 16 \rceil$. Then there exists a graph G' such that the following properties hold.*

(i) G' *is bipartite,* $V \subseteq V(G')$*, and* $|V(G')| \leq 3|E|k + |V|$*.*
(ii) $\mathrm{LMES}(G) \leq \mathrm{MS}(G')/3^{kx} \leq \mathrm{LMES}(G) + \frac{1}{4}$.
(iii) $\mathrm{LMES}(G, s, t) \leq \mathrm{MS}(G', s, t)/3^{ky} \leq \mathrm{LMES}(G, s, t) + \frac{1}{4}$.
(iv) $\mathrm{LMES}(G) \leq \mathrm{IMS}(G')/3^{kx} \leq \mathrm{LMES}(G) + \frac{1}{4}$.

The construction of the bipartite graph G' is similar to the earlier one — G' is constructed from G by first k-thickening and then 4-stretching each edge of G. We are able to associate minimal (vertex) separators of G' with minimal edge separators of G in a similar way to the proof of Lemma 17, but the correspondence is significantly messier since a minimal separator of G' may contain vertices of V. Indeed, there may be exponentially many such separators (as a function of k)!

We define our correspondence as follows. If X is a minimal (vertex) separator of G' we define $\pi(X)$, the corresponding edge-separator of G, to be the set of edges e of G such that X contains some new vertex in the stretched thickening of e. The point is to identify a set of "good" minimal separators of G' so that every minimal edge separator F of G corresponds to exactly $3^{k|F|}$ good minimal separators of G', and not too many minimal separators of G' are not good. The details are somewhat complicated, and the notion of "good" needs to be refined. For $z \in \mathbb{N}$, we say that X is *z-good*, if it satisfies the following conditions: (a) it intersects at most one of the edges of a 4-path corresponding to an edge of G, (b) if there is an intersection, then it intersects every such 4-path (so it cuts the edge of G), (c) X contains no vertices of G, (d) $|\pi(X)| = z$. The key to the proof is showing that all but at most $3^{kx}/4$ minimal separators of G' are x-good, and all but at most $3^{ky}/4$ minimal (s, t)-separators of G' are y-good. The argument involves several steps. First, we show that there are at most $2^5 mk$ minimal separators in G' which are not minimal (b, c)-separators for some vertices b, c of G. Then we consider $a \in \mathbb{N}$ and distinct vertices b and c of G. We show that there are at most $2^{m+n}3^{k(a-1)}$ minimal (b, c)-separators X of G' with $|\pi(X)| < a$. Finally, we consider distinct vertices b and c of G and let z be the maximum cardinality of any minimal (b, c)-edge separator G. We show that, if X is a minimal (b, c)-separator of G' with $|\pi(X)| \geq z$, then X is z-good. This is the most difficult step. The details are included in the full version.

Acknowledgements We thank Luca Manzoni and Yuri Pirola.

References

1. Berry, A., Bordat, J.P., Cogis, O.: Generating all the minimal separators of a graph. International Journal of Foundations of Computer Science 11(3), 397–403 (2000)
2. Bodlaender, H.L., Fomin, F.V.: Tree decompositions with small cost. Discrete Applied Mathematics 145(2), 143 – 154 (2005)
3. Bodlaender, H.L., Fomin, F.V., Koster, A.M., Kratsch, D., Thilikos, D.M.: On exact algorithms for treewidth. ACM Trans. Algorithms 9(1), 12:1–12:23 (2012)
4. Bodlaender, H.L., Koster, A.M.: Combinatorial optimization on graphs of bounded treewidth. The Computer Journal 51(3), 255–269 (2008)

5. Bouchitté, V., Todinca, I.: Treewidth and minimum fill-in: grouping the minimal separators. SIAM Journal on Computing 31(1), 212–232 (2001)
6. Bouchitté, V., Todinca, I.: Listing all potential maximal cliques of a graph. Theoretical Computer Science 276(1-2), 17–32 (2002)
7. Diestel, R.: Graph Theory, 4th Edition, Graduate texts in mathematics, vol. 173. Springer (2012)
8. Dyer, M., Goldberg, L.A., Greenhill, C., Jerrum, M.: The relative complexity of approximate counting problems. Algorithmica 38(3), 471–500 (2004)
9. Fomin, F.V., Kratsch, D., Todinca, I., Villanger, Y.: Exact algorithms for treewidth and minimum fill-in. SIAM Journal on Computing 38(3), 1058–1079 (2008)
10. Fomin, F.V., Villanger, Y.: Finding induced subgraphs via minimal triangulations. In: 27th STACS. pp. 383–394 (2010)
11. Fomin, F.V., Villanger, Y.: Treewidth computation and extremal combinatorics. Combinatorica 32(3), 289–308 (2012)
12. Furuse, M., Yamazaki, K.: A revisit of the scheme for computing treewidth and minimum fill-in. Theoretical Computer Science 531(0), 66 – 76 (2014)
13. Goldberg, L.A.: Efficient Algorithms for Listing Combinatorial Structures. Cambridge University Press (1993), Cambridge Books Online
14. Goldberg, L.A., Jerrum, M.: The complexity of ferromagnetic Ising with local fields. Combin. Probab. Comput. 16(1), 43–61 (2007)
15. Goldberg, L.A., Jerrum, M.: Approximating the partition function of the ferromagnetic Potts model. J. ACM 59(5), Art. 25, 31 (2012)
16. Gysel, R.: Unique perfect phylogeny characterizations via uniquely representable chordal graphs. CoRR abs/1305.1375 (2013)
17. Gysel, R.: Minimal triangulation algorithms for perfect phylogeny problems. In: 8th LATA. vol. 8370, pp. 421–432. Springer (2014)
18. Johnson, D.S., Papadimitriou, C.H., Yannakakis, M.: How easy is local search? J. Comput. Syst. Sci. 37(1), 79–100 (1988)
19. Kashiwabara, T., Masuda, S., Nakajima, K., Fujisawa, T.: Generation of maximum independent sets of a bipartite graph and maximum cliques of a circular-arc graph. J. Algorithms 13(1), 161–174 (1992)
20. Kloks, T., Kratsch, D.: Listing all minimal separators of a graph. SIAM Journal on Computing 27, 605–613 (1998)
21. Lokshtanov, D.: On the complexity of computing treelength. Discrete Applied Mathematics 158(7), 820 – 827 (2010)
22. Schäffer, A.A., Yannakakis, M.: Simple local search problems that are hard to solve. SIAM J. Comput. 20(1), 56–87 (1991)
23. Shen, H., Liang, W.: Efficient enumeration of all minimal separators in a graph. Theoretical Computer Science 180(1-2), 169–180 (1997)
24. Takata, K.: Space-optimal, backtracking algorithms to list the minimal vertex separators of a graph. Discrete Appl. Math. 158(15), 1660–1667 (2010)
25. Tsukiyama, S., Ide, M., Ariyoshi, H., Shirakawa, I.: A new algorithm for generating all the maximal independent sets. SIAM J. Comput. 6(3), 505–517 (1977)
26. Vadhan, S.P.: The complexity of counting in sparse, regular, and planar graphs. SIAM J. Comput. 31(2), 398–427 (2001)
27. Whitney, H.: Planar graphs. Fundamenta Mathematicae 21(1), 73–84 (1933)
28. Zuckerman, D.: On unapproximable versions of NP-complete problems. SIAM J. Comput. 25(6), 1293–1304 (1996)