

On the (In)Approximability of the Monitoring Edge Geodetic Set Problem

Davide Bilò 

Department of Information Engineering, Computer Science, and Mathematics,
University of L'Aquila, Italy

Giordano Colli 

Department of Information Engineering, Computer Science, and Mathematics,
University of L'Aquila, Italy

Luca Forlizzi 

Department of Information Engineering, Computer Science, and Mathematics,
University of L'Aquila, Italy

Stefano Leucci 

Department of Information Engineering, Computer Science, and Mathematics,
University of L'Aquila, Italy

Abstract

We study the minimum *Monitoring Edge Geodetic Set* (MEG-SET) problem introduced in [Foucaud et al., CALDAM'23]: given a graph G , we say that an edge is monitored by a pair u, v of vertices if *all* shortest paths between u and v traverse e ; the goal is to find a subset M of vertices of G such that each edge of G is monitored by at least one pair of vertices in M , and $|M|$ is minimized.

In this paper, we prove that all polynomial-time approximation algorithms for the minimum MEG-SET problem must have an approximation ratio of $\Omega(\log n)$, unless $P = NP$. To the best of our knowledge, this is the first non-constant inapproximability result known for this problem. We also strengthen the known NP-hardness of the problem on 2-apex graphs by showing that the same result holds for 1-apex graphs. This leaves open the question of determining whether the problem remains NP-hard on planar (i.e., 0-apex) graphs.

On the positive side, we design an algorithm that computes good approximate solutions for hereditary graph classes that admit efficiently computable balanced separators of truly sublinear size. This immediately yields polynomial-time approximation algorithms achieving an approximation ratio of $O(n^{\frac{1}{4}} \sqrt{\log n})$ on planar graphs, graphs with bounded genus, and k -apex graphs with $k = O(n^{\frac{1}{4}})$. On graphs with bounded treewidth, we obtain an approximation ratio of $O(\log^{3/2} n)$. This compares favorably with the best-known approximation algorithm for general graphs, which achieves an approximation ratio of $O(\sqrt{n \log n})$ via a simple reduction to the SET COVER problem.

2012 ACM Subject Classification Theory of computation $\rightarrow$ Design and analysis of algorithms

Keywords and phrases Monitoring Edge Geodetic Set, Inapproximability, Approximation Algorithms

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2025.14

Funding This work was partially supported by the MONET (Mutual visibility On NETworks) project funded by the University of L'Aquila.

1 Introduction

Consider a communication network that might suffer link failures. Such a network can be modeled as an undirected and connected graph G , where vertices represent hosts and edges correspond to links between hosts. In order to quickly detect failure events, the designer wishes to equip the hosts of the network with additional *probes*: probes can communicate with one another and they can monitor the distance between their corresponding nodes in the network (e.g., via a traceroute mechanism). Whenever the distance between a pair of



© Davide Bilò, Giordano Colli, Luca Forlizzi, and Stefano Leucci;
licensed under Creative Commons License CC-BY 4.0

36th International Symposium on Algorithms and Computation (ISAAC 2025).

Editors: Ho-Lin Chen, Wing-Kai Hon, and Meng-Tsung Tsai; Article No. 14; pp. 14:1–14:19

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

vertices increases, this indicates that some communication link is inoperative, and an alarm can be raised. Observe that, in the above scenario, it may be possible for two probes to be connected by two (or more) edge-disjoint shortest paths, in which case they will not be able to detect an edge failure.

This motivates the following problem: given a network G , find the smallest possible set M of vertices to equip with probes while ensuring that each edge e of G has at least one pair of vertices $u, v \in M$ for which all shortest paths between u and v in G traverse e . The above problem is known as the minimum *Monitoring Edge Geodetic Set* (MEG-SET) problem and was introduced in [9], where the authors focus on providing upper and lower bounds on the size of minimum MEG-SETS for both general graphs and special graph classes (trees, cycles, unicyclic graphs, complete graphs, grids, hypercubes, ...).

Further bounds on the size of MEG-SETS have been given for the Cartesian and strong products of two graphs [12], for other graph products including join and direct products [21], as a function of the graph's girth and chromatic number [6], for ladder, butterfly, circulant and Benes networks, for convex polytopes [20], and for radix triangular mesh networks and Sierpiński graphs [17]. Moreover, the minimum MEG-SET problem was proven to be NP-hard on general graphs [12], NP-hard on 3-degenerate 2-apex graphs [8], and APX-hard on 4-degenerate graphs [7]. If the Exponential Time Hypothesis [13] holds, then the problem cannot be solved in subexponential time on 3-degenerate graphs [7]. In the same paper, the authors also provide an exact polynomial-time algorithm for interval graphs and two FPT algorithms: one for general graphs parameterized by the sum of cliquewidth and diameter, and the other for chordal graphs parameterized by treewidth. Polynomial-time algorithms for computing optimal MEG-SETS are also known for distance-hereditary graphs, P_4 -sparse graphs, bipartite permutation graphs, and strongly chordal graphs [10].

As far as approximation algorithms are concerned, it has been observed [2, 1, 7] that a simple reduction to the SET COVER problem yields a polynomial-time algorithm returning solutions of size $O(k^2 \log n)$, where k is the size of a MEG-SET of minimum cardinality, thus achieving an approximation ratio of $O(\min\{k \log n, n/k\}) = O(\sqrt{n \log n})$.¹

1.1 Our results

In this paper, we investigate the approximability and inapproximability of the minimum MEG-SET problem.

We prove that, if $P \neq NP$, the problem admits no polynomial-time approximation algorithm with an approximation factor of $o(\log n)$. Moreover, the problem admits no $(\alpha \log n)$ -approximation algorithm for any constant $\alpha < \frac{1}{2}$ unless $NP \subseteq DTIME(n^{O(\log \log n)})$. To the best of our knowledge, these are the first non-constant inapproximability results for the minimum MEG-SET problem.

We also extend the aforementioned $O(\sqrt{n \log n})$ -approximation algorithm for minimum MEG-SET to a generalized version of the problem in which edges have non-negative *lengths*, vertices have binary *costs*, and a minimum-cost subset of vertices that monitors a *given subset* of edges needs to be selected. We use this result as a building block to design a more involved approximation algorithm for minimum MEG-SET that relies on recursively computing vertex-separators and provides good approximations for graph families that admit sparse balanced vertex-separators. More precisely, we achieve an approximation ratio of

¹ To the best of our knowledge, this algorithm was first given and analyzed in a Bachelor's thesis [2], and the result was subsequently claimed in a short communication based thereon [1]. The same algorithm, and its analysis, also appear in a later unpublished work [7] by a different group of authors.

$O(n^{\frac{1}{4}}\sqrt{\log n})$ on planar graphs, on graphs with bounded genus, and on k -apex graphs when $k = O(n^{\frac{1}{4}})$. We also obtain an approximation ratio of $O(\log^{3/2} n)$ on graphs with bounded treewidth.

Finally, we show that the problem remains NP-hard even on 1-apex graphs (while the reduction of [6] uses 2-apex graphs). It is an interesting open problem to settle the complexity status of the problem for planar graphs.

1.2 Organization of the paper

We start by giving some preliminary definitions and notation in Section 2. Our inapproximability result for general graphs is presented in Section 4, while the NP-Hardness result for 1-apex graphs is given in Section 3. Finally, Section 5 is devoted to our approximation algorithms.

2 Preliminaries and notation

Throughout the paper, we consider simple, connected, and undirected graphs unless otherwise specified. Given a graph $G = (V, E)$, we may use $V(G)$ or $E(G)$ to refer to the vertex-set V or the edge-set E , respectively. We denote the open neighborhood of a vertex $v \in V(G)$ in a graph G as $N_G(v) = \{u \in V(G) : \{u, v\} \in E(G)\}$; in the same way, the closed neighborhood of a vertex $v \in V(G)$ in G is defined as $N_G[v] = N_G(v) \cup \{v\}$. When the graph is clear from the context, we refer to the open (resp. closed) neighborhood of a vertex $v \in V(G)$ as $N(v)$ (resp. $N[v]$). We also use n to refer to $|V(G)|$. For a set $C \subseteq V$, we define $N(C) = \bigcup_{v \in C} N(v)$ and $N[C] = \bigcup_{v \in C} N[v]$.

We say that an edge $(u, v) \in E(G)$ is *monitored* by a pair of distinct vertices $x, y \in V(G)$ if (u, v) lies on *all* the *shortest* paths between u and v . Similarly, a set of edges E' is monitored by $M \subseteq V$ if each edge $e \in E'$ is monitored by at least one pair of vertices in M . A *monitoring edge-geodetic set* MEG-SET of $G = (V, E)$ is a subset of vertices that monitors E .

A walk π between two vertices u and v in an edge-weighted graph G with weight function w is an ordered list of alternating (and not necessarily distinct) vertices and edges $\langle v_0, e_1, v_1, e_2, v_2, \dots, e_k, v_k \rangle$ with $v_0 = u$, $v_k = v$, and such that $e_i = (v_{i-1}, v_i)$ for all $i = 1, \dots, k$. The weight $w(\pi)$ of π is $\sum_{i=1}^k w(e_i)$.

We conclude this section by stating two well-known structural properties of MEG-SETS.

► **Lemma 1** ([2, 9]). *All vertices of degree 1 in G belong to all MEG-SETS of G .*

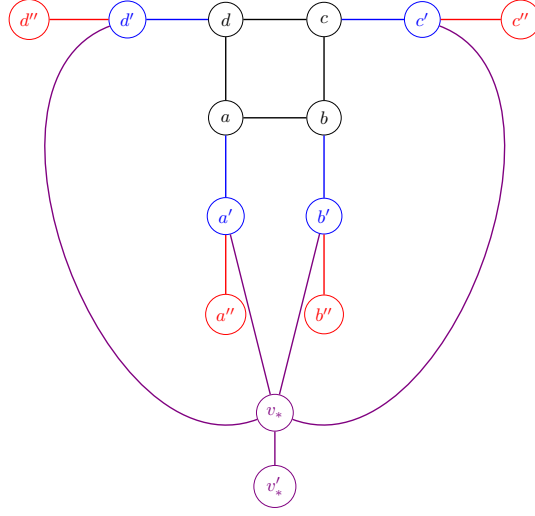
► **Lemma 2** ([2, 6]). *Let u be a vertex of degree 1 in G and let v be its sole neighbor. If $|V(G)| \geq 3$ and M is a MEG-SET of G , then $M \setminus \{v\}$ is a MEG-SET of G .*

3 NP-Hardness of minimum MEG-Set on graphs with apex 1

In this section, we show a reduction from the minimum PLANAR DOMINATING SET problem on graphs with girth at least seven (which is known to be NP-Hard [22]) to the minimum MEG-SET problem on graphs with apex 1.

Let $G = (V, E)$ be a planar input graph with $|V(G)| \geq 2$ and with girth $g(G) \geq 7$. We construct an instance of MEG-SET, namely H , starting from G and augmenting it as follows: for each vertex $v \in V(G)$, we add 2 vertices v' and v'' to G , then connect v to v' and v' to v'' . We denote the set containing all the vertices v' as L' and the set containing all the vertices v'' as L'' . Notice that:

- i) Each $v'' \in L''$ belongs to every MEG-SET of G due to Lemma 1;
- ii) Each $v' \in L'$ does not belong to any optimal MEG-SET of G due to Lemma 2.



■ **Figure 1** An example of the construction of the graph H from the input graph $G = (\{a, b, c, d\}, \{\{a, b\}, \{b, c\}, \{c, d\}, \{d, a\}\})$, which is shown in black.

The basic idea is to monitor each edge that is incident to a vertex $v \in V(G)$ that belongs to a *selected* MEG-SET in H . We denote the set of edges (v, v') as $E_{L'}$ and the set of edges (v', v'') as $E_{L''}$. Note that at this point of the construction, all edges are monitored by at least one pair of leaves in L'' . To avoid this, we shortcut all the shortest paths from one vertex in L'' to another by inserting two vertices v_* and v'_* and connecting them together. Then, we connect v_* to all vertices $v' \in L'$. We denote the set containing edges (v_*, v') with $v' \in L'$ as E_* . The vertex v_* is used to create shortcuts between each shortest path from a vertex in $L' \cup L''$ to a vertex in $L' \cup L''$. Notice that, since v'_* is a leaf, it belongs to every MEG-SET of G due to Lemma 1. The vertex v'_* is used to monitor the edges in E_* . Formally, let $H = (V(G) \cup L' \cup L'' \cup \{v_*, v'_*\}, E(G) \cup E_{L'} \cup E_{L''} \cup E_* \cup \{\{v_*, v'_*\}\})$, where:

- $L' = \{v' : v \in V(G)\};$
- $L'' = \{v'' : v \in V(G)\};$
- $E_{L'} = \{(v, v') : v \in V(G)\};$
- $E_{L''} = \{(v', v'') : v \in V(G)\};$
- $E_* = \{(v_*, v') : v \in V(G)\}.$

We prove the NP-Completeness of minimum MEG-SET on graphs with apex 1 via two lemmas. The first lemma shows that if there is a MEG-SET of the graph H having size at most $k + |V(G)| + 1$, there exists a DOMINATING SET of G having size at most k . Conversely, the second lemma shows that if there is a DOMINATING SET of the graph G having size at most k , there exists a MEG-SET of H having size $k + |V(G)| + 1$.

► **Lemma 3.** *Let D be a DOMINATING SET of G of size at most k . Then $M = D \cup L'' \cup \{v'_*\}$ is a MEG-SET of H and $|M| \leq k + |V(G)| + 1$.*

Proof. We prove that each edge $e \in E(H)$ is monitored by M . We distinguish four cases:

- Case 1: $e \in E(G)$. Let $e = (u, v)$. Since D is a DOMINATING SET of G , there are two subcases:
 - At least one of u and v must belong to D , therefore we assume w.l.o.g. that $u \in D$. Each path from v'' to u must either pass through the vertex v or pass through the vertex v_* . The unique shortest path including v is trivially $\langle (v'', v'), (v', v), (v, u) \rangle$ and has length 3. Each path passing through v_* must contain at least four edges. Therefore, (u, v) is monitored by $\{v'', u\}$, where both v'' and u are in M since $v'' \in L''$ and $u \in D$.

- Neither u nor v is in D . First, we observe that u and v are dominated by two distinct vertices in D ; otherwise, there would be an induced triangle in G , which is not allowed due to $g(G) \geq 7$. Let these two distinct vertices be $\{x, y\} \subseteq D$. They have distance $d(x, y)$ equal to 3 since, if this were not true, the edge (u, v) would not be monitored by the pair $\{x, y\}$ due to the existence of a shortest path of length 4 that uses only edges in $E(H) \setminus E(G)$. Finally, since $g(G) \geq 7$, it can be shown that there is no alternative shortest path of length 3 from x to y .
- Case 2: $e \in E_{L''} \cup E_*$. In this case, there is *exactly* one endpoint u' of e that belongs to L' . It follows that either $e = (u'', u')$ with $u'' \in L''$ or $e = (v_*, u')$. Notice that there is a unique shortest path from v'_* to u'' including the edges (v'_*, v_*) , (v_*, u') , (u', u'') . This implies that the edge e is monitored by $\{v'_*, u''\}$, where both v'_* and $u'' \in L''$ are in M .
- Case 3: $e \in E_{L'}$. Let $e = (u, u')$ where u is the unique endpoint of e in $V(G)$. Since D dominates all the vertices in $V(G)$, either $u \in D$, and we let $x = u$, or $u \notin D$ and there exists some vertex $x \in D \cap N(u)$. There is a single shortest path from u'' to x , and it traverses the edge (u, u') .
- Case 4: $e = (v'_*, v_*)$. All shortest paths from v'_* to any other vertex in H must traverse e , and hence e is monitored, e.g., by all pairs $\{v'_*, u\}$ with $u \in D$. Note that there exists at least one vertex $u \neq v'_*$ in M since $|M| \geq 2$.

The cardinality $|M|$ is equal to the sum of the cardinality of $M \setminus V(G)$ and the cardinality of $M \cap V(G)$, with $(M \setminus V(G)) \cap (M \cap V(G)) = \emptyset$. Since $|M \cap V(G)|$ is equal to the size of the DOMINATING SET of G , namely k , and $M \setminus V(G) = L'' \cup \{v'_*\}$, we can conclude that:

$$|M| = |M \cap V(G)| + |M \setminus V(G)| = k + |L''| + |\{v'_*\}| = k + |V(G)| + 1. \quad \blacktriangleleft$$

► **Lemma 4.** *Let M be a MEG-SET of H with size at most $k + |V(G)| + 1$ (for a suitable value of k). Then, $M \cap V(G)$ is a DOMINATING SET of G with size at most k .*

Proof. Let $M' = M \setminus (L' \cup \{v_*\})$ be the set obtained by removing the vertex v_* and all the vertices of L' from M . Due to Lemma 2, M' is a MEG-SET of G . Moreover, since $(L' \cup \{v_*\}) \cap V(G) = \emptyset$, we can observe that $M' \cap V(G) = M \cap V(G)$. Let $D = M' \setminus (L'' \cup \{v'_*\})$ be the set of vertices in the MEG-SET M' that are in $V(G)$. We prove that if $v \in V(G)$, then either $v \in D$ or $N(v) \cap D \neq \emptyset$. Towards a contradiction, suppose that $v \notin D$ and no vertices in $N(v)$ are in D . Since M' is a MEG-SET of H , it monitors every edge in $E(G) \subseteq E(H)$. Consider an edge (u, v) with $v \in N(u)$. (u, v) must be monitored by a pair of vertices x, y not in $\{u\} \cup N(u)$. It follows that the distance between x and y is at least four, but there is a path between x and y of length four that only uses edges in $E(H) \setminus E(G)$; thus, the edge (u, v) cannot be monitored by the pair $\{x, y\}$, which is a contradiction. The above discussion shows that $M' \cap V(G)$ is a DOMINATING SET of G . The size of such a DOMINATING SET is $|M' \cap V(G)| \leq |M \cap V(G)| = |M| - |M \setminus V(G)| \leq |M| - (|L''| + 1) \leq |M| - (|V(G)| + 1) \leq k$, where we used $(L'' \cup \{v'_*\}) \subseteq M \setminus V(G)$, as ensured by Lemma 1. $\blacktriangleleft$

► **Theorem 5.** *The MEG-SET decision problem is NP-Complete on graphs with apex 1.*

Proof. For each planar graph G with girth at least seven, it is possible to construct in polynomial time a graph H with apex 1, as discussed in this section. From Lemmas 3 and 4, there exists a DOMINATING SET D of G such that $|D| \leq k$ if and only if there exists a MEG-SET M of H such that $|M| \leq k + |V(G)| + 1$. Since the DOMINATING SET problem is NP-Hard even for planar graphs with girth at least seven, this implies that the MEG-SET decision problem is NP-Hard on graphs with apex 1. $\blacktriangleleft$

4 Inapproximability of MEG-Set

We reduce from the SET COVER problem. A SET COVER instance $\mathcal{I} = \langle X, \mathcal{S} \rangle$ is described as a set of η items $X = \{x_1, \dots, x_\eta\}$ and a collection $\mathcal{S} = \{S_1, \dots, S_h\}$ of $h \geq 2$ distinct subsets of X , such that each subset contains at least two items and each item appears in at least two subsets.² The goal is that of computing a collection $\mathcal{S}^* \subseteq \mathcal{S}$ of minimum size such that $\cup_{S_i \in \mathcal{S}^*} S_i = X$.³ It is known that, unless $P = NP$, all polynomial-time approximation algorithms for the SET COVER must have an approximation ratio of $(1 - o(1)) \ln \eta$ when $h = O(\text{poly}(\eta))$ [18, 4]. Moreover, unless $NP \subseteq \text{DTIME}(n^{O(\log \log n)})$, the same inapproximability result holds even for $h \leq \eta$ [5].

Given an instance $\mathcal{I} = \langle X, \mathcal{S} \rangle$ of SET COVER, we build an associated bipartite graph H whose vertex set $V(H)$ is $X \cup \mathcal{S}$ and such that H contains the edge (x_i, S_j) if and only if $x_i \in S_j$. We define $N = h + \eta$.

Let k be an integer parameter, whose exact value will be chosen later, that satisfies $2 \leq k = O(\text{poly}(N))$. We construct a graph G that contains k copies $H_1, \dots, H_k$ of H as induced subgraphs. In the following, for any $\ell = 1, \dots, k$, we denote by $x_{i,\ell}$ and $S_{j,\ell}$ the vertices of H_ℓ corresponding to the vertices x_i and S_j of H , respectively. More precisely, we build G by starting with a graph that contains exactly the k copies $H_1, \dots, H_k$ of H and augmenting it as follows (see Figure 2):

- For each item $x_i \in X$, we add two new vertices y_i, y'_i along with the edge (y_i, y'_i) and all the edges in $\{(x_{i,\ell}, y_i) \mid \ell = 1, \dots, k\}$;
- We add all the edges $(y_i, y_{i'})$ for all $1 \leq i < i' \leq \eta$, so that the subgraph induced by $y_1, \dots, y_\eta$ is complete.
- For each set $S_j \in \mathcal{S}$, we add two new vertices z_j, z'_j along with the edge (z_j, z'_j) , and all the edges in $\{(S_{j,\ell}, z_j) \mid \ell = 1, \dots, k\}$.

Observe that the number n of vertices of G satisfies $n = 2h + 2\eta + k(\eta + h) = (k + 2)(\eta + h) = (k + 2)N$.

Let $Y = \{y_i \mid i = 1, \dots, \eta\}$, $Y' = \{y'_i \mid i = 1, \dots, \eta\}$, $Z = \{z_j \mid j = 1, \dots, h\}$, and $Z' = \{z'_j \mid j = 1, \dots, h\}$. Moreover, define L as the set of all vertices of degree 1 in G , i.e., $L = Y' \cup Z'$. By Lemma 1, the vertices in L belong to all MEG-SETS of G .

► **Lemma 6.** *L monitors all edges having both endvertices in $Y \cup Y' \cup Z \cup Z'$.*

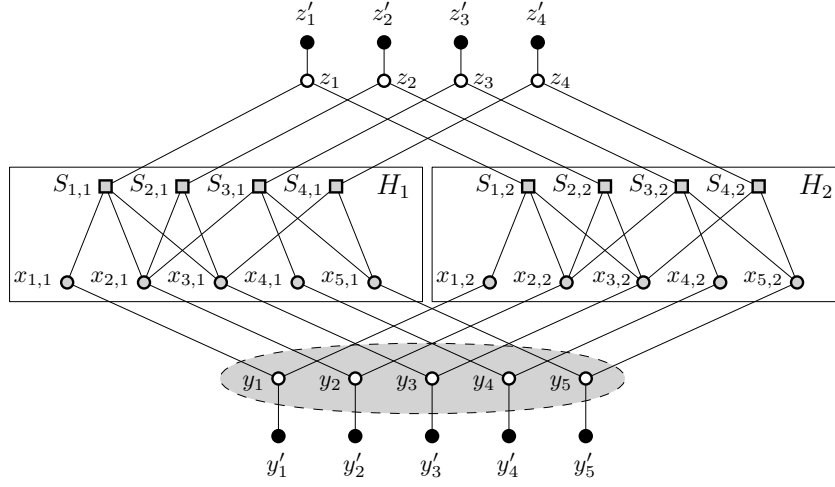
Proof. Observe that all shortest paths from $y'_i \in L$ (resp. $z'_j \in L$) to any other vertex $v \in L \setminus \{y'_i\}$ (resp. $v \in L \setminus \{z'_j\}$) must traverse the sole edge incident to y'_i (resp. z'_j), namely (y'_i, y_i) (resp. (z'_j, z_j)). Since $|L| \geq 2$, such a v always exists, and all edges incident to L are monitored by L .

The only remaining edges are those with both endpoints in Y . Let $(y_i, y_{i'})$ be such an edge. Since the only shortest path between y'_i and $y'_{i'}$ in G is $\langle y'_i, y_i, y_{i'}, y'_{i'} \rangle$, the pair $\{y'_i, y'_{i'}\}$ monitors $(y_i, y_{i'})$. ◀

► **Lemma 7.** *Let $S_1, \dots, S_k$ be k (not necessarily distinct) set covers of $\mathcal{I}$. The set $M = L \cup \{S_{j,\ell} \mid S_j \in \mathcal{S}_\ell, 1 \leq \ell \leq k\}$ is a MEG-SET of G .*

² This can be guaranteed w.l.o.g. by repeatedly reducing the instance by applying the first applicable of the following two reduction rules. Rule 1: if there exists an item x_i that is contained in a single subset S_j , then S_j belongs to all feasible solutions, and we reduce to the instance in which both S_j and x_i have been removed. Rule 2: if there exists a subset S_j that contains a single element, then (due to Rule 1) there is an optimal solution that does not contain S_j , and we reduce to the instance in which S_j has been removed. Notice that this process can only decrease the values of η and h .

³ We assume w.l.o.g. that $\cup_{i=1}^h S_i = X$, i.e., that a solution exists.



■ **Figure 2** The graph G obtained by applying our reduction with $k = 2$ to the SET COVER instance $\mathcal{I} = \langle X, \mathcal{S} \rangle$ with $\eta = 5$, $h = 4$, $S_1 = \{x_1, x_2, x_3\}$, $S_2 = \{x_2, x_3\}$, $S_3 = \{x_2, x_4, x_5\}$, and $S_4 = \{x_3, x_5\}$. To reduce clutter, the edges of the clique induced by the vertices y_i (in the gray area) are not shown.

Proof. Since $L \subseteq M$, by Lemma 6, we only need to argue about edges with at least one endvertex in some H_ℓ , with $1 \leq \ell \leq k$. Let $S_j \in \mathcal{S}_\ell$, and consider any $x_i \in S_j$. Edge $(S_{j,\ell}, z_j)$ is monitored by $\{z'_j, S_{j,\ell}\}$. Edges $(S_{j,\ell}, x_{i,\ell})$ and $(x_{i,\ell}, y_i)$ are monitored by $\{S_{j,\ell}, y'_i\}$.

The only remaining edges with at least one endvertex in H_ℓ are those incident to vertices $S_{j,\ell}$ with $S_j \in \mathcal{S} \setminus \mathcal{S}_\ell$. Consider any such S_j , let x_i be an item in S_j , and let $S_{k'} \in \mathcal{S}_\ell$ be any set such that $x_i \in S_{k'}$ (notice that both x_i and $S_{k'}$ exist since sets are non-empty and $\mathcal{S}_\ell$ is a set cover). Edge $(S_{j,\ell}, x_{i,\ell})$ is monitored by $\{S_{k',\ell}, z'_j\}$, which also monitors $(S_{j,\ell}, z_j)$. ◀

We say that a MEG-SET M is *minimal* if, for every $v \in M$, $M \setminus \{v\}$ is not a MEG-SET. Lemma 2 ensures that any minimal MEG-SET M does not contain any of the vertices y_i , for $i = 1, \dots, \eta$, or z_j for $j = 1, \dots, h$. Hence, $M \setminus L$ contains only vertices in $\bigcup_{\ell=1}^k V(H_\ell)$. The following lemma characterizes the structure of minimal MEG-SETS of G .

► **Lemma 8.** *Let M be a minimal MEG-SET of G . For every $i = 1, \dots, \eta$ and every $\ell = 1, \dots, k$, at least one of the following conditions is true: (i) $x_{i,\ell} \in M$; or (ii) there exists an index j such that $x_i \in S_j$ and the set M contains $S_{j,\ell}$.*

Proof. Let $S(x_{i,\ell})$ be the set of all $S_{j,\ell}$ such that $x_i \in S_j$. We show that no pair $\{u, v\}$ with $u, v \in M \setminus (\{x_{i,\ell}\} \cup S(x_{i,\ell}))$ can monitor the edge $(x_{i,\ell}, y_i)$.

Notice that, since M is minimal, M does not contain any vertex $y_{i'}$, for $i' = 1, \dots, \eta$, nor any vertex $z_{j'}$ for $j' = 1, \dots, h$.

Let $\lambda \in \{1, \dots, h\} \setminus \{\ell\}$ (such a λ always exists since $h \geq 2$), and notice that any path P in G that has both endvertices in $V(G) \setminus V(H_\ell)$ can always be transformed into a walk P' with as many edges as P by replacing any occurrence of a vertex from copy H_ℓ of H with the occurrence of the corresponding vertex from copy H_λ . Since P' does not contain $x_{i,\ell}$, this implies that no pair $\{u, v\}$ with $u, v \in V(G) \setminus V(H_\ell)$ can monitor $(x_{i,\ell}, y_i)$.

The above discussion allows us to restrict ourselves to the situation in which at least one vertex of $\{u, v\}$ belongs to $V(H_\ell)$. We say that a vertex of G is an *item-vertex* if it is some $x_{i',\ell'}$ with $i' = 1, \dots, \eta$ and $\ell' = 1, \dots, k$. Analogously, a vertex of G is a *set-vertex* if it is some $S_{j',\ell'}$ with $j' = 1, \dots, h$ and $\ell' = 1, \dots, k$. We consider the following cases and, for each of them, we exhibit a shortest path $\tilde{P}$ from u to v in G that does not traverse $(x_{i,\ell}, y_i)$:

Case 1: u is an item-vertex in H_ℓ and v is a set-vertex. In this case, $u = x_{i',\ell}$ for some $i' \neq i$, and we distinguish the following sub-cases:

- If $v = S_{j',\ell}$ and $x_{i'} \in S_{j'}$, then $\tilde{P}$ consists of the edge $(x_{i',\ell}, S_{j',\ell})$.
- If $v = S_{j',\ell}$, $x_{i'} \notin S_{j'}$, and there exists some $S_{j''}$ such that $x_{i'} \in S_{j''}$ and $S_{j'} \cap S_{j''} \neq \emptyset$, then let $x_{i''} \in S_{j'} \cap S_{j''}$. We choose $\tilde{P} = \langle x_{i',\ell}, S_{j'',\ell}, x_{i'',\ell}, S_{j',\ell} \rangle$.
- If $v = S_{j',\ell}$, $x_{i'} \notin S_{j'}$, and there exists no $S_{j''}$ such that $x_{i'} \in S_{j''}$ and $S_{j'} \cap S_{j''} \neq \emptyset$, then let $x_{i''} \in S_{j'}$ (notice that $x_i \notin S_{j'}$), and choose $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell}, S_{j',\ell} \rangle$.
- If $v = S_{j',\ell'}$, $x_{i'} \in S_{j'}$, and $\ell' \neq \ell$, then choose $\tilde{P} = \langle x_{i',\ell}, S_{j',\ell}, z_{j'}, S_{j',\ell'} \rangle$.
- If $v = S_{j',\ell'}$, $x_{i'} \notin S_{j'}$, and $\ell' \neq \ell$, then let $x_{i''} \in S_{j'}$ (notice that $x_i \notin S_{j'}$) and choose $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell'}, S_{j',\ell'} \rangle$.

Case 2: u is an item-vertex in H_ℓ and v is an item-vertex. In this case, $u = x_{i',\ell}$ for some $i' \neq i$, and we distinguish the following sub-cases:

- If $v = x_{i'',\ell}$ with $i'' \neq i$ and there exists some set $S_{j'}$ such that $x_{i'}, x_{i''} \in S_{j'}$, then choose $\tilde{P} = \langle x_{i',\ell}, S_{j',\ell}, x_{i'',\ell} \rangle$.
- If $v = x_{i'',\ell}$ with $i'' \neq i$ and there is no set $S_{j'}$ such that $x_{i'}, x_{i''} \in S_{j'}$, then choose $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell} \rangle$.
- If $v = x_{i',\ell'}$ with $\ell' \neq \ell$, then choose $\tilde{P} = \langle x_{i',\ell}, y_{i'}, x_{i',\ell'} \rangle$.
- If $v = x_{i'',\ell'}$ with $i'' \neq i'$ and $\ell' \neq \ell$, then choose $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell'} \rangle$.

Case 3: u is a set-vertex in H_ℓ and v is a set-vertex. In this case, $u = S_{j',\ell}$ for some j' such that $x_{i,\ell} \notin S_{j'}$, and we distinguish the following sub-cases:

- If $v = S_{j'',\ell}$ and there exists some $x_{i'}$ such that $x_{i'} \in S_{j'} \cap S_{j''}$, then $i' \neq i$, and we pick $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, S_{j'',\ell} \rangle$.
- If $v = S_{j'',\ell}$, $S_{j'} \cap S_{j''} = \emptyset$, and there exists some $S_{j'''}$ such that $S_{j'} \cap S_{j'''} \neq \emptyset$ and $S_{j''} \cap S_{j'''} \neq \emptyset$, then choose any $x_{i'} \in S_{j'} \cap S_{j'''}$ and any $x_{i''} \in S_{j''} \cap S_{j'''}$. We pick $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, S_{j''',\ell}, x_{i'',\ell}, S_{j'',\ell} \rangle$.
- If $v = S_{j'',\ell}$, $S_{j'} \cap S_{j''} = \emptyset$, and there is no $S_{j'''}$ such that $S_{j'} \cap S_{j'''} \neq \emptyset$ and $S_{j''} \cap S_{j'''} \neq \emptyset$, then choose $x_{i'} \in S_{j'} \setminus \{x_i\}$, $x_{i''} \in S_{j''} \setminus \{x_i\}$, and pick $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell}, S_{j'',\ell} \rangle$.
- If $v = S_{j',\ell'}$ with $\ell' \neq \ell$, then we pick $\tilde{P} = \langle S_{j',\ell}, z_{j'}, S_{j',\ell'} \rangle$.
- If $v = S_{j'',\ell'}$ with $j'' \neq j'$ and $\ell' \neq \ell$, and there exists some item $x_{i'} \in S_{j'} \cap S_{j''}$, then we choose $\tilde{P} = \langle S_{j',\ell}, z_{j'}, S_{j',\ell'}, x_{i',\ell'}, S_{j'',\ell'} \rangle$.
- If $v = S_{j'',\ell'}$ with $j'' \neq j'$ and $\ell' \neq \ell$, and $S_{j'} \cap S_{j''} = \emptyset$, then let $x_{i'} \in S_{j'} \setminus \{x_i\}$, $x_{i''} \in S_{j''} \setminus \{x_i\}$. We pick $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell'}, S_{j'',\ell'} \rangle$.

Case 4: u is an item-vertex in H_ℓ and $v \in L$. In this case, $u = x_{i',\ell}$ for some $i' \neq i$, and we distinguish the following sub-cases:

- If $v = y'_{i'}$, then $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y'_{i'} \rangle$.
- If $v = y'_{i''}$ with $i'' \neq i'$, then $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y_{i''}, y'_{i''} \rangle$.
- If $v = z'_{j'}$ and $x_{i'} \in S_{j'}$, then pick $\tilde{P} = \langle x_{i',\ell}, S_{j',\ell}, z_{j'}, z'_{j'} \rangle$.
- If $v = z'_{j'}$, $x_{i'} \notin S_{j'}$, and there exists some $S_{j''}$ for which $S_{j'} \cap S_{j''} \neq \emptyset$, then let $x_{i''} \in S_{j'} \cap S_{j''}$ and choose $\tilde{P} = \langle x_{i',\ell}, S_{j'',\ell}, x_{i'',\ell}, S_{j',\ell}, z_{j'}, z'_{j'} \rangle$.
- If $v = z'_{j'}$, $x_{i'} \notin S_{j'}$, and there is no $S_{j''}$ for which $S_{j'} \cap S_{j''} \neq \emptyset$, then let $x_{i''} \in S_{j'} \setminus \{x_i\}$. We choose $\tilde{P} = \langle x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell}, S_{j',\ell}, z_{j'}, z'_{j'} \rangle$.

Case 5: u is a set-vertex in H_ℓ and $v \in L$. In this case, $u = S_{j',\ell}$ for some j' such that $x_{i,\ell} \notin S_{j'}$, and we distinguish the following sub-cases:

- If $v = y'_{i'}$ with $x_{i'} \in S_{j'}$, then $i' \neq i$, and we choose $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, y_{i'}, y'_{i'} \rangle$.
- If $v = y'_{i'}$ with $x_{i'} \notin S_{j'}$, then let $x_{i''} \in S_{j'}$ and choose $\tilde{P} = \langle S_{j',\ell}, x_{i'',\ell}, y_{i''}, y_{i'}, y'_{i'} \rangle$.
- If $v = z'_{j'}$, then $\tilde{P} = \langle S_{j',\ell}, z_{j'}, z'_{j'} \rangle$.

- If $v = z'_{j''}$ with $j'' \neq j'$ and $S_{j'} \cap S_{j''} \neq \emptyset$, then let $x_{i'} \in S_{j'} \cap S_{j''}$ and choose $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, S_{j'',\ell}, z'_{j''}, z'_{j''} \rangle$.
- If $v = z'_{j''}$ with $j'' \neq j'$, $S_{j'} \cap S_{j''} = \emptyset$, and there exists some $S_{j'''}$ such that $S_{j'} \cap S_{j'''} \neq \emptyset$ and $S_{j''} \cap S_{j'''} \neq \emptyset$, then let $x_{i'} \in S_{j'} \cap S_{j'''}$ and $x_{i''} \in S_{j''} \cap S_{j'''}$. We choose $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, S_{j''',\ell}, x_{i'',\ell}, S_{j'',\ell}, z'_{j''}, z'_{j''} \rangle$.
- If $v = z'_{j''}$ with $j'' \neq j'$, $S_{j'} \cap S_{j''} = \emptyset$, and there is no $S_{j'''}$ such that $S_{j'} \cap S_{j'''} \neq \emptyset$ and $S_{j''} \cap S_{j'''} \neq \emptyset$, then let $x_{i'} \in S_{j'}$ and $x_{i''} \in S_{j''} \setminus \{x_i\}$, and choose $\tilde{P} = \langle S_{j',\ell}, x_{i',\ell}, y_{i'}, y_{i''}, x_{i'',\ell}, S_{j'',\ell}, z'_{j''}, z'_{j''} \rangle$. ◀

► **Lemma 9.** *Given a MEG-SET M' of G , we can compute in polynomial time a MEG-SET M of G such that $|M| \leq |M'|$ and, for every $\ell = 1, \dots, k$, the set $\mathcal{S}_\ell = \{S_j \in \mathcal{S} \mid S_{j,\ell} \in M\}$ is a set cover of $\mathcal{I}$.*

Proof. Let M'' be a minimal MEG-SET of G that is obtained from M' by possibly discarding some of the vertices. Clearly, $|M''| \leq |M'|$ and M'' can be computed in polynomial time. Moreover, by Lemma 8, for every $i = 1, \dots, \eta$ and every $\ell = 1, \dots, k$, M'' contains $x_{i,\ell}$ or some $S_{j,\ell}$ such that S_j covers x_i . We compute M from M'' by replacing each $x_{i,\ell} \in M''$ with $S_{j,\ell}$, where $S_j \in \mathcal{S}$ is any set that covers x_i . As a consequence, for every $\ell = 1, \dots, k$, the set $\mathcal{S}_\ell = \{S_j \in \mathcal{S} \mid S_{j,\ell} \in M\}$ is a set cover of $\mathcal{I}$. Moreover, since M'' contains all vertices in L by Lemma 1, so does M . Then, Lemma 7 implies that M is a MEG-SET of G . ◀

► **Lemma 10.** *Let α , c , and ε be constants of choice satisfying $\alpha > 0$, $c \geq 1$, and $0 < \varepsilon \leq \frac{1}{4c^2(\alpha+1)^2}$. Any polynomial-time $(\alpha \ln n)$ -approximation algorithm for the minimum MEG-SET problem implies the existence of a polynomial-time $((2\alpha c + \sqrt{\varepsilon}) \ln \eta)$ -approximation algorithm for SET COVER instances with η items and $h \leq \eta^c$ sets.*

Proof. Consider an instance $\mathcal{I} = \langle X, \mathcal{S} \rangle$ of SET COVER with $|X| = \eta$ and $|\mathcal{S}| = h \leq \eta^c$. Let h^* be the size of an optimal set cover of $\mathcal{I}$.

In the rest of the proof, we assume w.l.o.g. that $N \geq 4$, $\eta \geq 2^{1/\varepsilon}$, and $h^* \geq \frac{4\alpha}{\varepsilon}$. Indeed, if any of the above three conditions do not hold, we can solve $\mathcal{I}$ in polynomial time using a brute-force search.

We now construct the graph G with $n = (k+2)N \leq N^2$ vertices by making $k = N - 2$ copies of H . Next, we run the $(\alpha \ln n)$ -approximation algorithm to compute a MEG-SET M' of G , and we use Lemma 9 to find a MEG-SET M with $|M| \leq |M'|$ that contains k set covers $\mathcal{S}_1, \dots, \mathcal{S}_k$ in polynomial time. Among these k set covers, we output one $\mathcal{S}'$ of minimum size.

To analyze the approximation ratio of the above algorithm, let M^* be an optimal MEG-SET of G . Lemma 7 ensures that $|M^*| \leq |L| + kh^* = N + kh^*$, and hence:

$$|M| \leq |M'| \leq (\alpha \ln n)|M^*| \leq \alpha(N + kh^*) \ln n = \alpha(N + kh^*) \ln N^2 = 2\alpha kh^* \ln N + 2\alpha N \ln N.$$

Therefore, we have:

$$\begin{aligned} |\mathcal{S}'| &\leq \frac{|M|}{k} \leq 2\alpha h^* \ln N + \frac{2\alpha N \ln N}{k} \leq 2\alpha h^* \ln N + 4\alpha \ln N = \left(2\alpha + \frac{4\alpha}{h^*}\right) h^* \ln N \\ &\leq (2\alpha + \varepsilon) h^* \ln N \leq (2\alpha + \varepsilon) h^* \ln(h + \eta) \leq (2\alpha + \varepsilon) h^* \ln(2\eta^c) \\ &\leq (2\alpha + \varepsilon) h^* (1 + \varepsilon) c \ln \eta = (2\alpha c + \varepsilon c(2\alpha + \varepsilon + 1)) h^* \ln \eta \leq (2\alpha c + \sqrt{\varepsilon}) h^* \ln \eta, \end{aligned}$$

where we used $2\alpha + \varepsilon + 1 \leq 2(\alpha + 1) \leq \frac{1}{c\sqrt{\varepsilon}}$, which follows from our hypothesis $\varepsilon \leq \frac{1}{4c^2(\alpha+1)^2}$. ◀

It is known that, unless $P = NP$, there exists a constant $c \geq 1$ such that all polynomial-time approximation algorithms for instances of SET COVER having η items and $h \leq \eta^c$ sets must have an approximation factor of $(1 - o(1)) \ln \eta$ [18, 4]. Then, Lemma 10 with any constant $\alpha < \frac{1}{2c}$ and $\varepsilon = (\frac{1}{2} - \alpha c)^2$ implies that no polynomial-time $(\alpha \ln n)$ -approximation algorithm for MEG-SET can exist, unless $P = NP$.

Similarly, we can choose $c = 1$ [5] to rule out any polynomial-time algorithm with an approximation of $\alpha \ln n$ for MEG-SET, where α is a constant strictly smaller than $\frac{1}{2}$, unless $NP \subseteq DTIME(n^{O(\log \log n)})$.

► **Theorem 11.** *All polynomial-time approximation algorithms for MEG-SET must have an approximation factor of $\Omega(\log n)$, unless $P = NP$. Moreover, unless $NP \subseteq DTIME(n^{O(\log \log n)})$, the approximation factor must be at least $(\frac{1}{2} - \varepsilon) \ln n$ for any constant $\varepsilon > 0$.*

5 Approximation algorithms

This section is devoted to designing our approximation algorithms. It is convenient to work on a generalization of the MEG-SET problem, in which each edge e of G has an associated non-negative *weight* $w(e)$ and shortest paths are computed w.r.t. these weights. Moreover, each vertex v of G has a binary cost $c(v) \in \{0, 1\}$, and the input additionally specifies a subset E' of the edges in E . The goal is to find a set of vertices M that monitors all edges in E' and has minimum cost, defined as $c(M) = \sum_{v \in M} c(v)$. We name this generalization GMEG-SET.

Notice that the case in which all vertex costs are 1, all edges weights are 1, and $E' = E$ matches the standard definition of MEG-SET, hence any approximation algorithm for GMEG-SET (whose approximation ratio is computed w.r.t. the cost of the returned set) also provides an approximation algorithm for MEG-SET.

One can observe that a GMEG-SET exists if and only if, for every edge $(u, v) \in E$, the shortest path between u and v is unique and consists of the sole edge (u, v) . Indeed, if this condition is met, the set V is a trivial solution; otherwise, no subset of vertices can monitor edge (u, v) . As a consequence, in the rest of this section, we restrict ourselves to instances of GMEG-SET that satisfy the above property, which can be checked in polynomial time.

In the rest of this section, we first show a simple generalization of the $O(\sqrt{n \log n})$ -approximation algorithm for MEG-SET [2, 1, 7] to GMEG-SET, and then we use it as a building block to develop a more involved algorithm that provides good approximate solutions for GMEG-SET on hereditary graph classes that admit small balanced vertex-separators.

5.1 A simple approximation algorithm for GMEG-Set

We start with the description of an $\sqrt{2c(M^*) \ln n}$ -approximation algorithm for GMEG-SET, where $c(M^*)$ denotes the cost of an optimal GMEG-SET, that will be useful in the sequel.

For technical convenience, we assume that all optimal solutions to the GMEG-SET instance select at least two vertices with non-zero cost.⁴

Let Z be the set of all vertices of cost 0. Observing that it is possible to check in polynomial time whether an edge of E' is monitored by Z , we construct an instance of SET COVER in which the items are exactly the edges in E' that are not monitored by Z . The

⁴ Otherwise, there is at most one vertex with non-zero cost in an optimal solution, and such a vertex can be guessed in polynomial time.

instance contains a set S_p for each (unordered) pair $p = \{u, v\}$ of distinct vertices u, v in $V \setminus Z$. The set S_p contains all items e that are monitored by $Z \cup p$ (notice that an item in S_p is not necessarily monitored by p).

Next, we compute a $(\ln |E'|)$ -approximate solution $\mathcal{S}'$ for such an instance in polynomial time. This can be done by solving instances up to a constant size by brute force, while larger instances can be handled using the well-known greedy algorithm for SET COVER [15], which has an approximation ratio of $\ln \eta - \ln \ln \eta + O(1)$, where η is the number of items [19]. We return the GMEG-SET M obtained by selecting all vertices in Z together with all vertices involved in at least one pair p such that $S_p \in \mathcal{S}'$, i.e., $M = Z \cup \bigcup_{S_p \in \mathcal{S}'} p$.

Observe that, if M^* is an optimal GMEG-SET, then the collection containing all sets S_p for all unordered pairs p of vertices in $M^* \setminus Z$ is a feasible solution $\mathcal{S}''$ to the SET COVER instance. Since $c(v) \in \{0, 1\}$ for each $v \in V(G)$, if $\mathcal{S}^*$ is a minimum-cost set cover, we have $|\mathcal{S}^*| \leq \binom{|M^* \setminus Z|}{2} < \frac{|M^* \setminus Z|^2}{2} = \frac{c(M^*)^2}{2}$. Then:

$$c(M) \leq 2 \cdot |\mathcal{S}'| \leq 2 \cdot |\mathcal{S}^*| \cdot \ln |E'| < c(M^*)^2 \ln |E'| \leq 2c(M^*)^2 \ln n.$$

We have proved the following lemma.

► **Lemma 12.** *There exists a polynomial-time $(2c(M^*) \ln n)$ -approximation algorithm for GMEG-SET.*

By observing that a trivial upper bound on the approximation ratio of M is $\frac{c(V)}{c(M^*)}$, we have that the overall approximation ratio is upper bounded by $\min \left\{ 2c(M^*) \ln n, \frac{c(V)}{c(M^*)} \right\} \leq \sqrt{2c(V) \ln n}$, since the above minimum is maximized when $c(M^*) = \sqrt{\frac{c(V)}{2 \ln n}}$.

► **Lemma 13.** *There exists a polynomial-time $\sqrt{2c(V) \ln n}$ -approximation algorithm for GMEG-SET.*

Lemma 13 implies the existence of a polynomial-time $O(\sqrt{n \log n})$ -approximation algorithm for MEG-SET because $c(V) = n$.

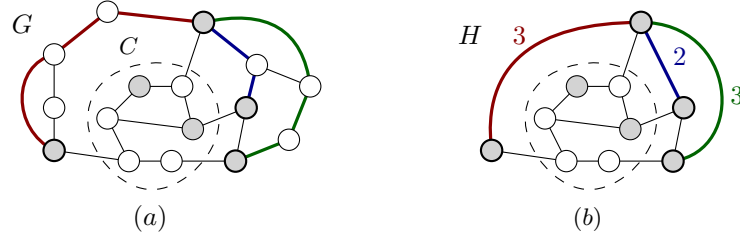
5.2 An approximation algorithm based on balanced graph separators

In this section, we present our main positive result. We start by giving some preliminary definitions and proving some useful structural properties of GMEG-SETS.

5.2.1 Some structural properties

Fix some instance of GMEG-SET with graph G , vertex costs $c(\cdot)$, edge weights $w(\cdot)$, and set of edges to be monitored E' . We refer to a subset of vertices of G as a *cluster* and, given a cluster C , we define the *boundary* δC of C as $N_G(C) \setminus C$. We say that a path π between two vertices u, v in G is a *C-bypass* if all the following conditions are satisfied: (i) $u, v \in \delta C$, (ii) π is a shortest path in G , (iii) π consists of at least two edges, and (iv) all internal vertices of π belong to $V \setminus (C \cup \delta C)$. Notice that the existence of a *C-bypass* between two vertices u, v in G implies that $(u, v) \notin E$.

Given a cluster C , we define the *projection* of C as the following instance of GMEG-SET. The instance's graph H has a vertex set consisting of all vertices in $C \cup \delta C$, where vertices in C retain their original cost in G , while vertices in δC have a cost of 0. The edge set of H consists of all edges in $G[C \cup \delta C]$ with their original weight in G , plus some additional *projection edges*. More precisely, for each pair of distinct vertices $u, v \in \delta C$ with at least



■ **Figure 3** (a) The graph G of an instance of GMEG-SET and a cluster C . Vertices in the boundary δC are shown in bold. Three C -bypasses are highlighted in red, blue, and green, respectively. All edge weights are unitary and are not shown. (b) The graph of the projection H of C . There are three projection edges, highlighted in red, blue, and green, corresponding to the C -bypass shown with the same color in (a). The number next to a projection edge represents its weight, while non-projection edges have unit weight (not shown). Gray vertices form a GMEG-SET of H . The vertices in the boundary are shown in bold and have a cost of 0. Notice how the gray vertices also monitor all non-projection edges in G .

one C -bypass between them, we add a *projection edge* $e = (u, v)$ of weight $d_G(u, v)$, and we associate e with an arbitrarily selected C -bypass b_e between u and v . We refer to H as the *projection graph* of C , and we use E_C to denote the set of all non-projection edges of H . The edges of H to be monitored are all those in $E(H) \cap E'$. See Figure 3 for an example.

We provide different structural properties of walks in G and the projection graph H of a cluster C . To distinguish between the weights of such walks, we will add the graph as a subscript to the weight function.

► **Lemma 14.** *Let C be a cluster, and let H be the projection graph of C . The following claims hold for any $u, v \in C \cup \delta C$.*

- (a) *For every walk π between u and v in H , there exists a walk π' between u and v in G such that $w_H(\pi) = w_G(\pi')$ and $E(\pi) \cap E_C = E(\pi') \cap E_C$;*
- (b) *For every shortest path π between u and v in G , there exists a walk π' between u and v in H such that $w_G(\pi) = w_H(\pi')$ and $E(\pi) \cap E_C = E(\pi') \cap E_C$.*

Proof. We start by proving (a). Given a walk π between u and v in H , we build the walk π' from u to v in G from π by replacing each projection edge $e \in \pi$ with its associated C -bypass b_e . By construction, $E(\pi) \cap E_C = E(\pi') \cap E_C$. Moreover, for each edge $e \in E_C$, $w_H(e) = w_G(e)$ by construction, while for each projection edge e , $w_H(e) = w_G(b_e)$. Therefore, $w_H(\pi) = w_G(\pi')$.

We now prove (b). Given a shortest path π between u and v in G , we decompose π into an alternating sequence of walks $\pi_1, \hat{\pi}_1, \pi_2, \hat{\pi}_2, \dots, \pi_k$, where π_i is a maximal subpath of π that contains only edges in E_C , and $\hat{\pi}_i$ is a maximal subpath of π that contains only edges not in E_C . Let u_i, v_i be the endvertices of $\hat{\pi}_i$. We build a walk π' from u_i to v_i in H from π by replacing each $\hat{\pi}_i$ with the projection edge $e_i = (u_i, v_i)$ associated with a C -bypass b_{e_i} between u_i and v_i . By construction, $E(\pi) \cap E_C = E(\pi') \cap E_C$. Moreover, for each edge $e \in E_C$, $w_H(e) = w_G(e)$ by construction. Now, since π is a shortest path from u to v in G , $\hat{\pi}_i$ is indeed a C -bypass in G , which implies $w_H(e_i) = w_G(\hat{\pi}_i)$. Therefore, $w_G(\pi) = w_H(\pi')$. ◀

► **Corollary 15.** *Let C be a cluster, and let H be the projection graph of C . The following claims hold for any $u, v \in C \cup \delta C$.*

- (a) *For every shortest path π between u and v in H , there exists a shortest path π' between u and v in G such that $w_H(\pi) = w_G(\pi')$ and $E(\pi) \cap E_C = E(\pi') \cap E_C$;*
- (b) *For every shortest path π between u and v in G , there exists a shortest path π' between u and v in H such that $w_G(\pi) = w_H(\pi')$ and $E(\pi) \cap E_C = E(\pi') \cap E_C$.*

Proof. We start by proving (a). Let π be a shortest path between u and v in H . By Lemma 14 (a), there exists a walk π' between u and v in G such that $w_H(\pi) = w_G(\pi')$ and $E(\pi) \cap E_C = E(\pi') \cap E_C$. We now argue that π' is actually a shortest path in G . Suppose, towards a contradiction, that there exists some path π'' between u and v in G such that $w_G(\pi'') < w_G(\pi')$. By Lemma 14 (b), this implies the existence of a walk π''' between u and v in H such that $w_H(\pi''') = w_G(\pi'')$, yielding the contradiction $w_H(\pi) = w_G(\pi') > w_G(\pi'') = w_H(\pi''')$.

We now prove (b). Let π be a shortest path between u and v in G . By Lemma 14 (b), there exists a walk π' between u and v in H such that $w_G(\pi) = w_H(\pi')$ and $E(\pi) \cap E_C = E(\pi') \cap E_C$. We now argue that π' is actually a shortest path in H . Suppose, towards a contradiction, that there exists some path π'' between u and v in H such that $w_H(\pi'') < w_H(\pi')$. By Lemma 14 (a), this implies the existence of a walk π''' between u and v in G such that $w_G(\pi''') = w_H(\pi'')$, yielding the contradiction $w_G(\pi) = w_H(\pi') > w_H(\pi'') = w_G(\pi''')$. ◀

► **Corollary 16.** *Let C be a cluster, and let H be the projection graph of C . Let $e \in E_C$ be a non-projection edge in H , and let u, v be a pair of vertices in H . Edge e is monitored by the pair u, v in G if and only if it is monitored by u, v in H .*

Proof. Assume that e is monitored by the pair u, v in G , and let π be a shortest path between u and v in G . By Corollary 15 (b), there exists a shortest path π' between u and v in H that traverses e . We now argue that *all* shortest paths between u and v in H contain e . Indeed, if H contained a shortest path π'' between u and v avoiding e , then Corollary 15 (a) would imply the existence of a shortest path between u and v avoiding e in G , i.e., u, v would not monitor e in G .

Assume now that e is monitored by the pair u, v in H , and let π be a shortest path between u and v in H . By Corollary 15 (a), there exists a shortest path π' between u and v in G that traverses e . We now argue that *all* shortest paths between u and v in G contain e . Indeed, if G contained a shortest path π'' between u and v avoiding e , then Corollary 15 (b) would imply the existence of a shortest path between u and v avoiding e in H , i.e., u, v would not monitor e in H . ◀

► **Lemma 17.** *Let C be a cluster, and let M be a GMEG-SET of G . Then, $(M \cap C) \cup \delta C$ is a GMEG-SET for the projection of C .*

Proof. Consider any edge $e \in E_C$ that belongs to E' , let u', v' be a pair of vertices in M that monitors e in G , and let π be a shortest path from u' to v' in G . Let u (resp. v) be the last vertex in $(M \cap C) \cup \delta C$ encountered during a traversal of the subpath π_u (resp. π_v) of π from u' (resp. v') to the closest endvertex of e . Observe that u (resp. v) exists since either $u' \in C \cup \delta C$, or π_u (resp. π_v) has the endvertex u' (resp. v') that is not in $C \cup \delta C$ and the other endvertex in $C \cup \delta C$, hence it must traverse at least one vertex in δC .

Since the subpath of π from u to v contains edge e , the pair u, v monitors e in G . Hence, by Corollary 16, u, v monitors e in the projection graph of C . ◀

► **Lemma 18.** *Let C be a cluster. Any GMEG-SET for the projection of C monitors all edges in $E_C \cap E'$ in G .*

Proof. Let M be a GMEG-SET for the projection of C , and let H be the corresponding graph. We consider a generic edge $e \in E_C \cap E'$ and we prove that e is also monitored by M in G . Let $u, v \in M$ be a pair of vertices that monitors e in H . By Corollary 16, the pair u, v monitors e in G . ◀

5.2.2 Algorithm description

In this section, we describe our main approximation algorithm that, as we will show, computes better than the $(\sqrt{2c(V)} \ln n)$ -approximate solutions for the class of hereditary graphs that admit small balanced vertex-separators. Before delving into the details of the algorithm, we provide some basic definitions.

► **Definition 19.** *Given $\alpha \in (0, \frac{1}{2})$, an α -balanced vertex-separator of a graph $G = (V, E)$ is a partition of V into three non-empty sets S, A, B , such that (i) each path from a vertex in A to a vertex in B traverses at least one vertex in S , and (ii) $\min\{|A|, |B|\} \geq \alpha|V|$. The size of the separator is $|S|$.*

We say that a class $\mathcal{G}$ of graphs has *efficiently computable α -balanced separators* of size $O(n^\beta)$ if (i) $\mathcal{G}$ is hereditary, i.e., given a graph $G \in \mathcal{G}$, all induced subgraphs of G are in $\mathcal{G}$, (ii) there are constants $n_0, \alpha > 0, \beta_0 \geq 1$, and $\beta \geq 0$ such that any graph $G \in \mathcal{G}$ with $n \geq n_0$ vertices admits an α -balanced vertex-separator of size at most $\beta_0 n^\beta$, and (iii) such a separator can be found in polynomial time. We say that a graph is α -non-separable if it admits no α -balanced vertex-separator. Similarly, we say that a cluster C of G is α -non-separable if $G[C]$ is α -non-separable. Notice that if a class of graphs $\mathcal{G}$ has efficiently computable α -balanced separators, $G \in \mathcal{G}$, and C is an α -non-separable cluster of G , then $|C| = O(1)$.

We consider a class $\mathcal{G}$ with efficiently computable α -balanced separators of size $\beta_0|V(G)|^\beta$ for some constants $\beta_0 \geq 1$ and $\beta \geq 0$, and we restrict our attention (w.l.o.g.) to instances of GMEG-SET whose input graph $G = (V, E)$ belongs to $\mathcal{G}$ and satisfies $c(V) \geq 2$.⁵ Our algorithm works in two phases.

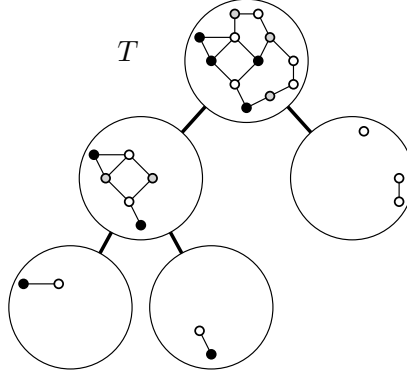
First phase. In the first phase, we compute a hierarchical decomposition of G into clusters that are pairwise vertex-disjoint. We note that such a decomposition will not result in a partition of the vertices of G , i.e., some vertices will not be assigned to any of the clusters, and we refer to the vertices that are left unassigned as *crossing vertices*. Our decomposition relies on recursively computing α -balanced vertex-separators of G and guarantees that any path between vertices of different clusters traverses at least one crossing vertex. It follows that all the vertices in the boundaries of the returned clusters are crossing vertices.

Let G' be the induced subgraph of G that contains all (and only) the vertices having non-zero cost. We build a rooted tree T in which each vertex u of T is associated with some subgraph G_u of G' and with an instance of GMEG-SET H_u . The root r of T is associated with $G_r = G'$ and with the projection H_r of $V(G_r)$ (w.r.t. the input instance of GMEG-SET). A generic vertex u of T is a leaf if H_u admits a solution of cost 0, or if G_u is α -non-separable. Otherwise, u has exactly two children a, b in T corresponding to the subgraphs $G_a = G_u[A]$ and $G_b = G_u[B]$ induced by the sets A, B of an α -balanced vertex-separator S_u, A, B of G_u . The instances of GMEG-SET H_a and H_b are the projections of $V(G_a)$ and $V(G_b)$ w.r.t. the instances H_u , respectively. See Figure 4 for an example.

Second phase. In the second phase of our algorithm, we visit the tree in a bottom-up fashion and, for each examined vertex u , we compute a $\rho(h_u, n_u)$ -approximate solution for the projection H_u of $V(G_u)$. Here, $\rho(h_u, n_u)$ is a function that depends on the number $n_u = |V(G_u)|$ of vertices in G_u and, possibly, on the height h_u of u in T .⁶ Observe that all vertices in G_u have cost 1 w.r.t. the cost function c_u of H_u , i.e., $n_u = c_u(V(G_u))$.

⁵ Otherwise, when $c(V) \leq 1$, an optimal solution consists either of all vertices of cost 0, or of all vertices of cost 0 plus the sole vertex of cost 1.

⁶ The height of vertex u in T is the height of the subtree of T rooted in u , i.e., the maximum number of edges in the longest path from u to a (not necessarily proper) descendant of u .



■ **Figure 4** The tree T representing a possible hierarchical decomposition of the graph shown in the root vertex for the parameters $\beta_0 = 1$, $\beta = \frac{1}{2}$, $\alpha = \frac{1}{4}$. Vertices in a separator are shown in gray in the corresponding cluster. Black vertices in a leaf cluster u of T form an optimal solution to instance H_u (not shown) when combined with the vertices of cost 0 resulting from the projection. Observe that the instance corresponding to the right child of the root admits a solution of cost 0. Gray and black vertices in an internal cluster u form a feasible GMEG-SET of the corresponding instance H_u (not shown) when combined with the vertices of cost 0.

In detail, if u is a leaf in T , then either there exists an optimal GMEG-SET of H_u with cost 0, or $|V(G_u)| = O(1)$, and we can find an optimal GMEG-SET of H_u in constant time by brute force (possibly both). Regardless of the case, we have an optimal solution for H_u , i.e., a $\rho(0, n_u)$ -approximation for $\rho(0, n_u) = 1$.

When the examined vertex u of T is a generic internal vertex at height h having a and b as children, we compute a GMEG-SET M_u for H_u by choosing the set of smallest cost between (i) a GMEG-SET M'_u obtained as the union of a $\rho(h_u - 1, n_a)$ -approximate GMEG-SET M_a for H_a with a $\rho(h_u - 1, n_b)$ -approximate GMEG-SET M_b for H_b , and (ii) a $(2c_u(M^*) \ln |V(G_u)|)$ -approximate solution computed using the algorithm of Section 5.1 (see Lemma 12), where M^* denotes an optimal solution for H_u and c_u is the cost function of H_u . The next lemma provides an upper bound on the cost of $c_u(M'_u)$.

► **Lemma 20.** *Let u be an internal vertex at height h_u with children a and b . If $\rho(h_u - 1, x)$ is monotonically non-decreasing w.r.t. x , M_a is a $\rho(h_u - 1, n_a)$ -approximate GMEG-SET for H_a , and M_b is a $\rho(h_u - 1, n_b)$ -approximate GMEG-SET for H_b , then M'_u is a GMEG-SET for H_u with $c_u(M'_u) \leq |S_u| + c_u(M_u^*)\rho(h_u - 1, (1 - \alpha)n_u)$, where M_u^* is an optimal solution for H_u .*

Proof. Let $M_a^* = (M^* \cap V(G_a)) \cup \delta V(G_a)$ (resp. $M_b^* = (M^* \cap V(G_b)) \cup \delta V(G_b)$), where the boundary is w.r.t. the instance H_u . By Lemma 17, M_a^* and M_b^* are GMEG-SET for H_a and H_b , respectively. Then, $c_a(M_a^*) = c_a(M^* \cap V(G_a)) + c_a(\delta V(G_a)) = c_a(M^* \cap V(G_a)) = c_u(M^* \cap V(G_a))$, and a similar derivation shows that $c_b(M_b^*) = c_u(M^* \cap V(G_b))$.

Since M_a and M_b are a $\rho(h_u - 1, n_a)$ - and a $\rho(h_u - 1, n_b)$ -approximation of H_a and H_b , respectively, we have $c_a(M_a) \leq c_u(M^* \cap V(G_a))\rho(h_u - 1, n_a)$ and $c_b(M_b) \leq c_u(M^* \cap V(G_b))\rho(h_u - 1, n_b)$.

Let E'_u be the set of edges to be monitored in H_u . By Lemma 18, M_a monitors all edges of E'_u in $G_u[V(G_a) \cup \delta V(G_a)]$, and M_b monitors all edges of E'_u in $G_u[V(G_b) \cup \delta V(G_b)]$; therefore, $M_a \cup M_b$ is a GMEG-SET for H_u . Since $M_a \cup M_b \subseteq M'_u$, we have that M'_u is a GMEG-SET of H_u of cost:

$$\begin{aligned}
c_u(M') &\leq c_u(M_a \cup M_b) \leq c_u(S_u) + c_u(M_a \setminus S) + c_u(M_b \setminus S) \leq |S_u| + c_a(M_a) + c_b(M_b) \\
&\leq |S_u| + c_u(M_a^* \cap V(G_a))\rho(h_u - 1, n_a) + c_u(M_b^* \cap V(G_b))\rho(h_u - 1, n_b) \\
&\leq |S_u| + (c_u(M_a^* \cap V(G_a)) + c_u(M_b^* \cap V(G_b)))\rho(h_u - 1, (1 - \alpha)n_u) \\
&\leq |S_u| + c_u(M^*)\rho(h_u - 1, (1 - \alpha)n_u). \quad \blacktriangleleft
\end{aligned}$$

The following two lemmas upper bound the approximation ratio achieved by M_u for the instance H_u .

► **Lemma 21.** *Let $\rho(h, x) = x^{\beta/2} \cdot \frac{\sqrt{2\beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}}$. If $\beta > 0$, then M_u is a $\rho(h_u, n_u)$ -approximate GMEG-SET for H_u .*

Proof. The proof is by induction on h_u . The claim is trivially true for $h_u = 0$ since $\rho(0, n_u) = 1 \leq \sqrt{2 \ln 2} \leq \frac{\sqrt{n_u \beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}}$. Therefore, we assume that the claim holds for $h_u - 1 \geq 0$ and we prove that it holds for h_u . Consider a vertex u at height h_u , and let a and b be its two children, and let S_u be the corresponding separator of G_u . Let M^* be an optimal GMEG-SET for H_u . By Lemma 20, we have:

$$c(M) \leq |S_u| + c(M^*)\rho(h_u - 1, (1 - \alpha)n_u) \leq \beta_0 n_u^\beta + c(M^*)(1 - \alpha)^{\beta/2} n_u^{\beta/2} \cdot \frac{\sqrt{2\beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}}.$$

Since the returned solution is the one of minimum cost between M'_u and a solution with approximation ratio $2c(M^*) \ln |V(G_u)| \leq 2c(M^*) \ln n$, the resulting approximation ratio is:

$$\begin{aligned}
&\min \left\{ \frac{\beta_0 n_u^\beta}{c(M^*)} + (1 - \alpha)^{\beta/2} n_u^{\beta/2} \frac{\sqrt{2\beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}}, 2c(M^*) \ln n \right\} \\
&\leq \min \left\{ \frac{\beta_0 n_u^\beta}{c(M^*)}, 2c(M^*) \ln n \right\} + (1 - \alpha)^{\beta/2} n_u^{\beta/2} \frac{\sqrt{2\beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}} \\
&\leq \sqrt{2\beta_0 \ln n} \cdot n_u^{\beta/2} + (1 - \alpha)^{\beta/2} n_u^{\beta/2} \frac{\sqrt{2\beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}} \\
&= \sqrt{2\beta_0 \ln n} \cdot n_u^{\beta/2} \left(1 + \frac{(1 - \alpha)^{\beta/2}}{1 - (1 - \alpha)^{\beta/2}} \right) = n_u^{\beta/2} \cdot \frac{\sqrt{2\beta_0 \ln n}}{1 - (1 - \alpha)^{\beta/2}}
\end{aligned}$$

where, in the second inequality, we use the fact that $\min \left\{ \frac{\beta_0 n_u^\beta}{c(M^*)}, 2c(M^*) \ln n \right\}$ is maximized for $\frac{\beta_0 n_u^\beta}{c(M^*)} = 2c(M^*) \ln n$, i.e., when $c(M^*) = \sqrt{\frac{\beta_0 n_u^\beta}{2 \ln n}}$. ◀

► **Lemma 22.** *Let $\rho(h, x) = (h + 1)\sqrt{2\beta_0 \ln n}$. If $\beta = 0$, then M_u is a $\rho(h_u, n_u)$ -approximate GMEG-SET for H_u .*

Proof. The proof is by induction on h . The claim is trivially true for $h = 0$ since $\rho(0, n_u) = 1 \leq \sqrt{2 \ln 2} \leq (h_u + 1)\sqrt{n_u \beta_0 \ln n}$. Therefore, we assume that the claim holds for $h_u - 1 \geq 0$ and we prove that it holds for h_u . Consider a vertex u at height h_u , let a and b be its two children, and let S_u be the corresponding separator of G_u . Let M^* be an optimal solution for H_u . By Lemma 20, we have:

$$c(M) \leq |S_u| + c(M^*)\rho(h_u - 1, (1 - \alpha)n) \leq \beta_0 n_u^\beta + c(M^*)h_u \sqrt{2\beta_0 \ln n}.$$

Since the returned solution is the one of minimum cost between M'_u and a solution with approximation ratio $2c(M^*) \ln |V(G_u)| \leq 2c(M^*) \ln n$, the resulting approximation ratio is:

$$\begin{aligned} \min \left\{ \frac{\beta_0}{c(M^*)} + h_u \sqrt{2\beta_0 \ln n}, 2c(M^*) \ln n \right\} &\leq \min \left\{ \frac{\beta_0}{c(M^*)}, 2c(M^*) \ln n \right\} + h_u \sqrt{2\beta_0 \ln n} \\ &\leq \sqrt{2\beta_0 \ln n} + h_u \sqrt{2\beta_0 \ln n} = (h_u + 1) \sqrt{2\beta_0 \ln n}, \end{aligned}$$

where we used the fact that $\min \left\{ \frac{\beta_0}{c(M^*)}, 2c(M^*) \ln n \right\}$ is maximized for $c(M^*) = \sqrt{\frac{\beta_0}{2 \ln n}}$. ◀

We can finally state the main results of this section.

► **Theorem 23.** *There exists a polynomial-time approximation algorithm for GMEG-SET with an approximation ratio of $O(c(V)^{\beta/2} \sqrt{\log n})$ if $\beta > 0$ and an approximation ratio of $O(\log c(V) \cdot \sqrt{\log n})$ if $\beta = 0$.*

Proof. We start by arguing that M_r achieves the claimed approximation ratios for H_r . When $\beta > 0$, this follows directly from Lemma 21 and from the fact that G_r has $n_r = c(V)$ vertices. Regarding the case $\beta = 0$, if u is the parent of v , then $n_v \leq (1 - \alpha)n_u$. Then, the tree T has height at most $\left\lceil \log_{\frac{1}{1-\alpha}} n_r \right\rceil = \left\lceil \frac{\log c(V)}{\log \frac{1}{1-\alpha}} \right\rceil$, and Lemma 22 implies that the achieved approximation is at most $\left(1 + \left\lceil \frac{\log c(V)}{\log \frac{1}{1-\alpha}} \right\rceil\right) \cdot \sqrt{2\beta_0 \ln n}$.

To obtain the claimed approximations for the input instance of GMEG-SET, let Z be the set of vertices having cost 0 in such instance, and consider $M = M_r \cup Z$. Since $c(M) = c(M_r)$, $c(V) = c(V(G'))$, and $n \geq n_r$, the claim on the approximation trivially holds, and we only need to argue that M is indeed a GMEG-SET.

By Lemma 18, M_r monitors, in G , all edges of E' that belong to $V(G') \cup \delta V(G')$. Hence, if $V(G') \cup \delta V(G') = V(G)$, we are done. Otherwise, let $A = V(G) \setminus (V(G') \cup \delta V(G'))$, and let H_A be the projection of A . Since $A \subseteq Z$, the set Z is a GMEG-SET for H_A , and by Lemma 18, Z monitors, in G , all edges of E' that belong to $V(A) \cup \delta V(A)$. Hence, $A \cup M_r \subseteq Z \cup M_r$ monitors all edges in E' , i.e., it is a GMEG-SET. ◀

The above theorem implies the following approximability results for GMEG-SET (and for MEG-SET, where $c(V) = n$):

► **Corollary 24.** *GMEG-SET is approximable in polynomial time on:*

- *planar graphs, with an approximation ratio of $O(c(V)^{\frac{1}{4}} \sqrt{\log n})$ using the balanced separators of size $O(\sqrt{n})$ shown in [16];*
- *graphs with bounded treewidth, with an approximation ratio of $O(\log c(V) \sqrt{\log n})$, using the separator induced by a centroid bag of the corresponding tree decomposition [3, Sections 12.3 and 12.4];*
- *graphs with constant genus, with an approximation ratio of $O(c(V)^{\frac{1}{4}} \sqrt{\log n})$ using the balanced separators of size $O(\sqrt{n})$ shown in [11];*
- *graphs with apex $k = O(n^{1/4})$, with an approximation ratio of $O(c(V)^{\frac{1}{4}} \sqrt{\log n})$, since it is possible to find, in polynomial time, a set S of $O(k)$ vertices such that $G[V(G) \setminus S]$ is planar [14].*

References

- 1 Davide Bilò, Giordano Colli, Luca Forlizzi, and Stefano Leucci. On the inapproximability of finding minimum monitoring edge-geodetic sets (short paper). In Ugo de'Liguoro, Matteo Palazzo, and Luca Roversi, editors, *Proceedings of the 25th Italian Conference on Theoretical Computer Science, Torino, Italy, September 11-13, 2024*, volume 3811 of *CEUR Workshop Proceedings*, pages 219–224. CEUR-WS.org, 2024. URL: <https://ceur-ws.org/Vol-3811/paper110.pdf>.
- 2 Giordano Colli. Monitoring graph edges via shortest paths: computational complexity and approximation algorithms. Bachelor's thesis, December 2023. Publicly accessible version: <https://arxiv.org/abs/2506.12021>. URL: <https://hdl.handle.net/20.500.12319/14489>.
- 3 Reinhard Diestel. *Graph Theory, 5th Edition*. Graduate texts in mathematics. Springer, 2018.
- 4 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In David B. Shmoys, editor, *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 - June 03, 2014*, pages 624–633. ACM, 2014. doi:10.1145/2591796.2591884.
- 5 Uriel Feige. A threshold of $\ln n$ for approximating set cover. *J. ACM*, 45(4):634–652, 1998. doi:10.1145/285055.285059.
- 6 Florent Foucaud, Clara Marcille, Zin Mar Myint, R. B. Sandeep, Sagnik Sen, and S. Taruni. Bounds and extremal graphs for monitoring edge-geodetic sets in graphs. *Discret. Appl. Math.*, 366:106–119, 2025. doi:10.1016/J.DAM.2024.12.032.
- 7 Florent Foucaud, Clara Marcille, R. B. Sandeep, Sagnik Sen, and S Taruni. Algorithms and complexity for monitoring edge-geodetic sets in graphs, 2024. doi:10.48550/arXiv.2409.19067.
- 8 Florent Foucaud, Pierre-Marie Marcille, Zin Mar Myint, R. B. Sandeep, Sagnik Sen, and S. Taruni. Monitoring edge-geodetic sets in graphs: Extremal graphs, bounds, complexity. In Subrahmanyam Kalyanasundaram and Anil Maheshwari, editors, *Algorithms and Discrete Applied Mathematics - 10th International Conference, CALDAM 2024, Bhilai, India, February 15-17, 2024, Proceedings*, volume 14508 of *Lecture Notes in Computer Science*, pages 29–43. Springer, 2024. doi:10.1007/978-3-031-52213-0_3.
- 9 Florent Foucaud, Krishna Narayanan, and Lekshmi Ramasubramony Sulochana. Monitoring edge-geodetic sets in graphs. In Amitabha Bagchi and Rahul Muthu, editors, *Algorithms and Discrete Applied Mathematics - 9th International Conference, CALDAM 2023, Gandhinagar, India, February 9-11, 2023, Proceedings*, volume 13947 of *Lecture Notes in Computer Science*, pages 245–256. Springer, 2023. doi:10.1007/978-3-031-25211-2_19.
- 10 Florent Foucaud, Arti Pandey, and Kaustav Paul. Characterizing optimal monitoring edge-geodetic sets for some structured graph classes, 2025. doi:10.48550/arXiv.2503.06086.
- 11 John R. Gilbert, Joan P. Hutchinson, and Robert Endre Tarjan. A separator theorem for graphs of bounded genus. *J. Algorithms*, 5(3):391–407, 1984. doi:10.1016/0196-6774(84)90019-1.
- 12 John Haslegrave. Monitoring edge-geodetic sets: Hardness and graph products. *Discret. Appl. Math.*, 340:79–84, 2023. doi:10.1016/j.dam.2023.06.033.
- 13 Russell Impagliazzo and Ramamohan Paturi. On the complexity of k-sat. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001. doi:10.1006/JCSS.2000.1727.
- 14 Bart M. P. Jansen and Michal Włodarczyk. Lossy planarization: a constant-factor approximate kernelization for planar vertex deletion. In Stefano Leonardi and Anupam Gupta, editors, *STOC '22: 54th Annual ACM SIGACT Symposium on Theory of Computing, Rome, Italy, June 20 - 24, 2022*, pages 900–913. ACM, 2022. doi:10.1145/3519935.3520021.
- 15 David S. Johnson. Approximation algorithms for combinatorial problems. In Alfred V. Aho, Allan Borodin, Robert L. Constable, Robert W. Floyd, Michael A. Harrison, Richard M. Karp, and H. Raymond Strong, editors, *Proceedings of the 5th Annual ACM Symposium on Theory of Computing, April 30 - May 2, 1973, Austin, Texas, USA*, pages 38–49. ACM, 1973. doi:10.1145/800125.804034.
- 16 Richard J. Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979. doi:10.1137/0136016.

- 17 Rongrong Ma, Zhen Ji, Yifan Yao, and Yalong Lei. Monitoring-edge-geodetic numbers of radix triangular mesh and sierpiński graphs. *Int. J. Parallel Emergent Distributed Syst.*, 39(3):353–361, 2024. doi:10.1080/17445760.2023.2294369.
- 18 Dana Moshkovitz. The projection games conjecture and the np-hardness of $\ln n$ -approximating set-cover. *Theory Comput.*, 11:221–235, 2015. doi:10.4086/TOC.2015.V011A007.
- 19 Peter Slavík. A tight analysis of the greedy algorithm for set cover. *J. Algorithms*, 25(2):237–254, 1997. doi:10.1006/JAGM.1997.0887.
- 20 Ao Tan, Wen Li, Xiumin Wang, and Xin Li. Monitoring edge-geodetic numbers of convex polytopes and four networks. *Int. J. Parallel Emergent Distributed Syst.*, 38(4):301–312, 2023. doi:10.1080/17445760.2023.2220143.
- 21 Xin Xu, Chenxu Yang, Gemaji Bao, Ayun Zhang, and Xuan Shao. Monitoring-edge-geodetic sets in product networks. *Int. J. Parallel Emergent Distributed Syst.*, 39(2):264–277, 2024. doi:10.1080/17445760.2024.2301929.
- 22 Igor E. Zverovich, Vadim E. Zverovich, Igor Zverovich, and Vadim Zverovich. An induced subgraph characterization of domination perfect graphs. *Journal of Graph Theory*, 20:375–395, 1995. doi:10.1002/jgt.3190200313.