Eliminating more than vertices in graphs

Thomas Delépine, under the supervision of Hoang La and François Pirot*

GALAC, LISN, Université Paris-Saclay, CNRS, Gif-sur-Yvette, France.

1 Extended abstract

1.1 General context

A central area of research in theoretical computer sciences is the study of NP-complete problems, ones that take an exponential amount of time to solve (Satisfiability, Travelling Salesman Problem [15] to name a few). During this internship, the initial problem that we were interested in was the Feedback Vertex Set problem [15], a problem on graphs. The problem asks whether it is possible to make a graph acyclic by removing a fixed amount of vertices. While these problems are hard in the general case, they might become easier for practical graphs since practical graphs usually have special structures. For example, the social graph is known to have a small diameter [36]. This is the premise behind the study of parameterized complexity, a field that aims at developing fast algorithms from problems that are hard in general, but easier when we restrict ourselves to graphs with some structural constraints. A well-studied parameter for which we have FPT algorithms for many hard problems (Feedback Vertex Set included [13, 14]) is treedepth. The treedepth of a graph G can be defined as the number of rounds needed to delete every vertex from G, where one can delete one vertex from each connected component of the current graph in a single round. From a structural point of view, one can bound the minimum number of vertices needed in a feedback vertex set depending on the treedepth of the graph. However, for Feedback Vertex Set in particular, deleting trees (acyclic graphs) instead of single vertices would yield a much better bound. This motivates the introduction of a new parameter, generalizing treedepth, that we call elimination depth.

1.2 Studied problem

Let $\mathcal{D}$ be a set of connected graphs. The elimination depth of a graph G by deleting graphs in $\mathcal{D}$ is the number of rounds needed to eliminate the whole graph, where one can delete an induced subgraph contained in $\mathcal{D}$ from each connected component of the current graph in a single round. Elimination depth generalizes treedepth as it is exactly treedepth when $\mathcal{D} = \{K_1\}$. A natural question in the study of graph parameters is to compute the extremal values of this parameter for a fixed class of graphs, i.e. how large can elimination depth be if we have additional structures on the graph to eliminate?

1.3 Summary of the proposed contributions

Let $\mathcal{C}$ be the class of graphs that we want to eliminate, we denote by $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ the extremal values of elimination depth in $\mathcal{C}$ when we delete graphs in $\mathcal{D}$. In this report, we provide general results for lower bounds and upper bounds of the extremal values of many elimination depths. More concretely, we provide a certificate for graphs of large elimination depth, called $\mathcal{D}$-shelters. We also provide sufficient conditions on $\mathcal{D}$ and $\mathcal{C}$ for $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ to be at least logarithmic. Intuitively, when $\mathcal{D}$ is "really close" to $\mathcal{C}$, we expect $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ to be constant. However, our general result implies that for many well-known classes of graphs for $\mathcal{C}$ and $\mathcal{D}$ (paths, trees, cycles, planar graphs, graphs with bounded maximum average degree, and so on), unless we can eliminate any graph of $\mathcal{C}$ in one round, then there always exists a graph G in $\mathcal{C}$ for which $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ is at least logarithmic (in the number of vertices of the input graph from $\mathcal{C}$).

Moreover, this logarithmic threshold is the best that we can hope for in some cases. To prove that this bound is tight for some $\mathcal{C}$ and $\mathcal{D}$, we devise some methods to obtain logarithmic upper bounds on $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ by studying the structures of balanced separators in block decompositions of graphs in $\mathcal{C}$. Since graphs with bounded treewidth are known for having tree decompositions with a bag being a balanced separator, we

*thomas.delepine@live.fr, hoang.la@universite-paris-saclay.fr, francois.pirot@universite-paris-saclay.fr

are also interested in variants of tree decompositions where the graphs induced by the bags have structural constraints (for example being connected).

Finally, unlike treedepth, elimination depth is (in general) not monotone under taking induced subgraph. We provide a characterization of the classes of graphs $\mathcal{D}$ for which $\mathrm{ed}^{\mathcal{D}}$ is monotone under taking induced subgraph.

1.4 Arguments supporting their validity

We provide proofs and concrete examples for every result given in this report. Below is a list of classes of graphs standing in for $\mathcal{D}$ and $\mathcal{C}$ for which our general result implies a logarithmic lower bound on $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$.

$\mathcal{D}$: K_1; girth $\geqslant k+1$; treewidth $\leqslant k$; degeneracy $\leqslant k$; chromatic number $\leqslant k$; trees; cycles; cliques; C_k-minor-free; mad $\leqslant r$; eulerian graphs.

$\mathcal{C}$: girth $\geqslant k$; treewidth $\leqslant k+1$; degeneracy $\leqslant k+1$; chromatic number $\leqslant k+1$; planar graphs; chordal graphs; K_t-minor-free; C_{k+1}-minor free; mad $\leqslant r+\varepsilon$.

For example, when $\mathcal{D}$ is the class of connected graphs of maximum average degree at most r with $r > 2$ and $\mathcal{C}$ is the class of graphs of maximum average degree at most $r+\varepsilon$ for some $\varepsilon > 0$, $\mathrm{ed}^{\mathcal{D}}(\mathcal{C}) = \Omega(\log(n))$ no matter how close ε is to 0. For some cases, we also have tight upper bounds. For example, $\mathrm{ed}^{\text{cycles}}(\text{planar graphs}) = \Theta(\log(n))$ or $\mathrm{ed}^{C_k\text{-minor-free}}(C_{k+1}\text{-minor-free}) = \Theta(\log(n))$.

1.5 Future work

While we have tight upper bounds for some classes as mentioned above, there are still many other cases to be studied. Our general result provides sufficient conditions on $\mathcal{C}$ and $\mathcal{D}$ for $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ to be at least logarithmic. It is thus interesting to study the extremal values of elimination depths for classes of graphs that do not respect these conditions. To study upper bounds, we took a look at connected treewidth (where the bags of the tree decomposition induce connected graphs) but we can also study other variants where other structural constraints are required depending on the graphs that we can delete.

We saw that elimination depths are monotone under taking induced subgraphs if and only if we are allowed to delete all graphs or only cliques. Non-monotonicity is the root of many difficulties when studying this parameter. One can also ask for a given $\mathcal{C}$, which graphs are we allowed to delete ($\mathcal{D}$) so that $\mathrm{ed}^{\mathcal{D}}$ is monotone on the graphs of $\mathcal{C}$.

For the reasons listed above, the study of elimination depth by deleting cliques seems compelling. The class of cliques (for $\mathcal{D}$) has the required constraints to provide logarithmic lower bounds for most class $\mathcal{C}$ but it is possible to build graphs for which this bound is not tight. For example, $\mathrm{ed}^{\text{cliques}}(\text{planar graphs}) = \Theta(\sqrt{n})$ (reached for square grids). Moreover, $\mathrm{ed}^{\text{cliques}}$ is also monotone under taking induced subgraphs for any graph.

From a more algorithmic perspective, one can use the decomposition provided by elimination depths to solve hard problems efficiently (as it was proven to be the case with treedepth), the questions now are to compute (an approximation of) such decompositions and to find the right $\mathcal{D}$ for the right problem.

Treedepth has also been generalized in another manner where the stopping condition in the elimination process is when the current graph is contained in a target class $\mathcal{T}$: this is called elimination distance to $\mathcal{T}$. Elimination distance was well-studied from an algorithmic point of view [5, 18]. We can also generalize this notion and elimination depth by allowing the deletions of graphs in $\mathcal{D}$ and stop when the current graph reaches $\mathcal{T}$.

1.6 Acknowledgments

I would like to express my gratitude to my supervisor, Hoang La, for the insightful discussions and productive work sessions throughout my internship. His guidance, thoughtful ideas, and feedbacks greatly contributed to both my learning experience and the successful completion of this report. I also want to thank François Pirot for his constructive suggestions and for providing interesting directions to continue my research.

I'm also grateful to the entire GALaC team—permanent members, PhD students, and fellow interns—for their friendship and support. My time there was both engaging and enjoyable.

2 Introduction

A *graph* G is a pair $(V(G), E(G))$ of sets such that $E(G) \subseteq V(G) \times V(G)$. The elements of $V(G)$ are called the *vertices* of G and the elements of $E(G)$ are called the *edges* of G. In the following, we will consider simple graphs with no loops, meaning that for all $\{u, v\} \in E$, $u \neq v$. From now on, we denote an edge $\{u, v\} \in E(G)$ by uv for conciseness. A graph can be drawn using points to represent vertices and lines to represent edges.

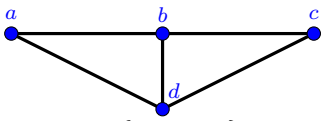


Figure 1: A graph with vertex set $\{a, b, c, d\}$ and edge set $\{ab, ad, bc, bd, cd\}$.

Graphs are data structures that represent a collection of objects with binary relations between pairs of these objects. For instance, a graph can represent road networks, by considering intersections as vertices and roads between two intersections as edges. Another example is a social network graph, with vertices representing people, and edges representing the friendship relation. The ability of a graph to model a wide range of problems motivates the study of graphs as a purely mathematical object. Most definitions and basic properties can be found in Diestel's *Graph Theory* book [7].

2.1 Some graph notations

Let $G \coloneqq (V(G), E(G))$ be a graph. The size of $V(G)$ is also called the *order* of G. The graph of order 0 is called *the null graph*. Let $e = uv \in E(G)$ be an edge of G. We say that e is *incident* to u and to v, and that u and v are *neighbors*. The *degree of a vertex* u, denoted by $\deg(u)$, is the number of its neighbors. We define the *deletion of an edge* e in G the operation resulting in $G - e \coloneqq (V(G), E(G) - e)$. We define the *deletion of a vertex* v in G, which results in $G - v$, as the removal of v from $V(G)$ and the deletion of all edges that are incident to v from $E(G)$. Let S be a set of vertices of G. We define the *deletion of a set* S in G, which results in $G - S$, by iteratively deleting the vertices from S in G. We say that a graph H is an *induced subgraph* of G, denoted by $H \subseteq_i G$, if H can be obtained from G with vertex deletions. We say that H is a *subgraph* of G, denoted by $H \subseteq G$ if H can be obtained from G with vertex and edge deletions. We say that H is a *proper* (induced) subgraph of G when it results from at least one vertex (or edge) deletion. Let $S \subseteq V(G)$, we define the subgraph of G induced by S, denoted by $G[S]$ as $G - (V(G) \setminus S)$. As a convention, we consider that the one and only (induced) subgraph of the null graph is itself.

Let k be positive integer, a *path* P_k *of length* $k - 1$ is a graph on k vertices $v_1, \ldots, v_k$ such that for all $0 \leqslant i < k$, $v_i v_{i+1} \in E(P_k)$. We say that v_1 and v_k are the two *endpoints* of P_k and $v_2, \ldots, v_{k-1}$ are the *internal vertices* of P_k. Let k be an integer with $k \geqslant 3$, a *cycle* C_k *of length* k is a path on k-vertices to which we add the edge $v_1 v_k$. A graph that contains no cycles (as a subgraph) is *acyclic*. A graph G is *connected* if and only if there exists a path (as a subgraph) between any two vertices of G. As a convention, we consider that the null graph is connected. If a graph is not connected, then we call *connected components* the maximal induced subgraphs of G that are connected.

2.2 Hard problems and sparse graphs

One of the many problems with important applications that can be modeled using graphs is the Feedback Vertex Set problem. A *feedback vertex set* of a graph G is a set $S \subseteq V(G)$ such that $G - S$ is acyclic. An application of the FVS problem is deadlock recovery in operating systems [35], in which a deadlock is a cycle in a system resource-allocation graph G. To recover from deadlocks, we need to abort a set of processes in the system, i.e. to delete a set S of vertices in G, so that $G - S$ contains no deadlocks, i.e. $G - S$ is acyclic. The purpose of the Feedback Vertex Set problem is to find the minimum number of processes to abort, i.e. the minimum size of a feedback vertex set in G.

The problem of deciding whether a graph contains a feedback vertex set of size at most k is widely studied. Indeed, it is one of Karp's famous 21 NP-complete problems, see [15]. A central and open question in theoretical computer science is whether a polynomial time algorithm exists to solve NP-complete problems or not, and the working hypothesis is that the answer is no. Hence, there are no (known) efficient algorithms to solve the feedback vertex set problem in general. The field of parameterized complexity aims

at understanding what is the minimum required information for a problem to become simple. For instance, we can consider the restriction of the feedback vertex set problem to sparse graphs. There exist many notions of sparse graphs, but intuitively they are graphs with a small (subquadratic) number of edges. Having fewer edges in the graph can make computing a feedback vertex set easier. In this report, our focus will be on the structural properties of sparse graphs instead of algorithms and hardness. Thus, we refer the readers to Cormen's *Introduction to Algorithms* book [6] for additional information.

A *graph parameter* or *graph invariant* is a function from the set of graphs to the set of real numbers. For instance, a well-known measure of sparsity of a graph G is the *maximum degree*, denoted by $\Delta(G)$, the maximum of the degree of vertices in G. Another well-studied measure is the *maximum average degree*, denoted by $\mathrm{mad}(G)$, which is the maximum of $\sum_{v\in V(H)} \deg(v)/|V(H)|$ over all subgraphs H of G. Graphs with bounded maximum degree or maximum average degree are indeed sparse: $|E(G)| \leqslant \frac{1}{2}\,\mathrm{mad}(G)|V(G)| \leqslant \frac{1}{2}\Delta(G)|V(G)|$. In 2006, Nešetřil and Ossona de Mendez introduced treedepth in [27], a graph parameter that is often used as a good measure for sparsity. More can be found about treedepth in their book about sparsity [26]. Treedepth turned out to be a very useful parameter with applications in structural graph theory [9] and algorithmic graph theory. Indeed, for NP-complete problems such as Maximum Weighted Perfect Matching [14] and Feedback Vertex Set [13], there exist polynomial time algorithms to solve these problems for graphs of bounded treedepth.

2.3 Treedepth

To introduce the notion of treedepth, consider the following one-player game. Let G be a graph. In each round, the player is allowed to delete one vertex per connected component of the current graph. The goal is to eliminate G, i.e. delete every vertex of G, in as few rounds as possible. See Figure 2 for an example.

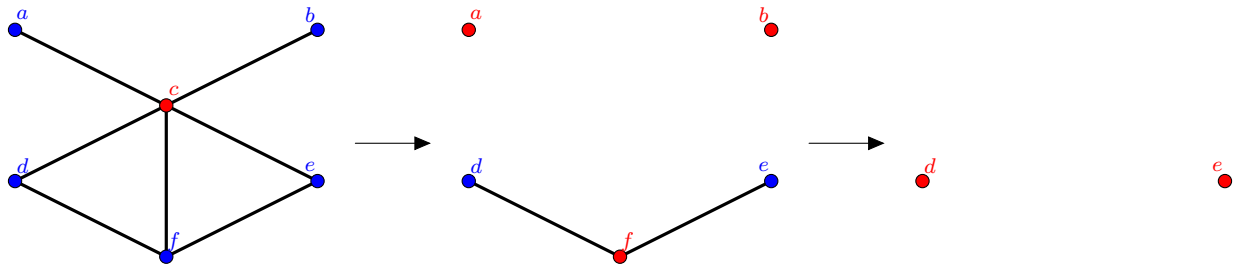


Figure 2: A graph that can be eliminated in 3 rounds by eliminating red vertices each round.

The *treedepth* of a graph G, denoted by $\mathrm{td}(G)$, is the smallest integer k such that we can delete every vertex of G in k rounds. More formally, $\mathrm{td}(G)$ is defined as follows.

Definition 1. Let G be a graph,

$$\mathrm{td}(G) \coloneqq \begin{cases} 0 & \text{if } G \text{ is the null graph,} \\ \max\{\mathrm{td}(H)\colon H \text{ is a connected component of } G\} & \text{if } G \text{ is disconnected,} \\ 1+\min\{\mathrm{td}(G-v)\colon v\in V(G)\} & \text{if } G \text{ is nonnull and connected.} \end{cases}$$

The vertex that realizes the minimum in $1+\min\{\mathrm{td}(G-v)\colon v\in V(G)\}$ when G is connected corresponds to the next vertex to be deleted in an optimal move for the player. When G is disconnected, the cost of deleting every vertex of G is the same as the one for its "most expensive" connected component.

Another equivalent definition of treedepth uses the closure of forests. Let F be a *rooted forest*, that is an acyclic graph with a special vertex in each connected component called a *root*. If F contains a unique connected component, then F is a *tree*. Let t, u be two vertices of F, in the same connected component with root r. We say that t is an *ancestor* of u and that u is a *descendant* of t if t lies on the unique path from u to r in F. A vertex of a rooted forest that has no descendant is called a *leaf*. The *depth of a vertex u in a rooted tree* is the number of vertices in the unique path between u and the root. The *depth of a rooted tree* is the maximum depth of its vertices. The *depth of a rooted forest* is the maximum depth of its trees. The *closure* of F, denoted by $\mathrm{clos}(F)$ is the graph with vertex set $V(F)$ and edge set $\{tu\colon t \text{ is an ancestor of } u \text{ in } F\}$. An *elimination forest* of G is a rooted forest F such that $G \subseteq \mathrm{clos}(F)$. The treedepth of a graph G is the smallest integer k such that G has an elimination forest F of depth k;

such a forest is called an *optimal elimination forest* of G. The only elimination forest of the null graph is the null graph with no bags. See Figure 3 for an example.

Let G be a graph with an elimination forest of depth k. For every integer $1 \leqslant i \leqslant k$, the vertices at depth i correspond to the ones that we delete on round i in the game. Since the edges of the graph follow an ancestor-descendant relationship, vertices on the same depth will be in different connected components of the current graph (where all vertices on the previous depths have been deleted).

There are many other definitions of treedepth: in Section 3.1, we introduce another definition of treedepth using a cops-and-robbers game (see [28] for many more). The problem of computing the treedepth or an optimal elimination forest is NP-complete [30]. The question of computing optimal elimination trees in the general case has been studied in [11], and for graphs of bounded treedepth in [31] and [24].

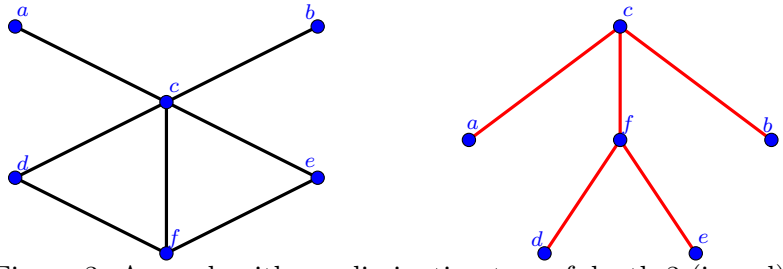


Figure 3: A graph with an elimination tree of depth 3 (in red).

An important property of treedepth is its monotonicity with respect to the subgraph relation.

Observation 2. *Let G be a graph. For every subgraph $H \subseteq G$, $\mathrm{td}(H) \leqslant \mathrm{td}(G)$.*

This observation stems from the fact that if we delete vertices in $V(G) - V(H)$ from an elimination forest of G, then we obtain an elimination forest of H.

To better understand treedepth, we can start by studying the treedepth of trees. Counter-intuitively, the tree itself is not always an optimal elimination tree. For example, a strategy to eliminate a path is to delete the "middle" vertex of each of the smaller paths every round to obtain a logarithmic (in terms of the number of vertices) number of rounds. In general, our strategy is based on separating the tree into two equal parts.

Lemma 3. *Let T be a tree of order $n \geqslant 1$. There exists a vertex v such that each connected component of $T - v$ contains at most $\frac{1}{2}n$ vertices.*

Proof. Let $C(v) \coloneqq \max\{|V(T')| \colon T' \text{ is a connected component of } T - v\}$. Assume for the sake of contradiction that for every $v \in V(T), C(v) > \frac{1}{2}n$. Let v_0 be such that $C(v_0)$ is minimized. Let $(T_i)_{i \geqslant 1}$ be the connected components of $T - v_0$. For every $i \geqslant 1$, there is $w_i \in V(T_i)$ such that v_0 and w_i are neighbors. Without loss of generality, suppose that $|V(T_1)| > \frac{1}{2}n$. Then $|V(T) - V(T_1)| < \frac{1}{2}n$. However, each connected component of $T - w_1$ is either a proper induced subgraph of T_1, or $T - T_1$. So, $C(w_1) < C(v_0)$ which contradicts the minimality of v_0. □

Proposition 4. *If T is a tree of order $n \geqslant 1$, then $\mathrm{td}(T) \leqslant 1 + \log_2(n)$.*

Proof. We prove this lemma by induction on n. If $n = 1$, then $\mathrm{td}(T) = 1$. Let T be a tree of order $n > 1$. By Lemma 3, there exists v such that every component of $T - v$ contains at most $\frac{1}{2}n$ vertices. Let $T_{\max}$ be the connected component of $T - v$ with the largest treedepth. By induction hypothesis, $\mathrm{td}(T) \leqslant 1 + \mathrm{td}(T_{max}) \leqslant 1 + \left(1 + \log_2\left(\frac{n}{2}\right)\right) = 1 + \log_2(n)$. □

2.4 Minimum feedback vertex sets in graphs with bounded treedepth

Going back to the study of hard problems in sparse graphs, we can take a look at feedback vertex sets of sparse graphs. When considering feedback vertex sets of a graph, we want to know the minimum number of vertices to delete so that the remaining graph is acyclic. In other words, we want to upper bound $\mathrm{fvs}(G)$, *the size of a minimum feedback vertex set* of a graph G. For instance, in [10], bounds on fvs were given for graphs of small maximum degree and for *planar graphs* (graphs that can be drawn in the plane such that no two edges cross each other). Similar works were done for graphs of bounded degeneracy and

bounded treewidth [16] (see Section 4.5 for the definitions) as well as graphs with bounded maximum average degree [25]. The first result of this report is a bound on fvs for graphs of fixed treedepth.

Theorem 5. *For every graph G of order $n \geqslant 1$ such that* $\mathrm{td}(G) \geqslant 2$,

$$\mathrm{td}(G) - (1 + \log_2(n)) \leqslant \mathrm{fvs}(G) \leqslant \frac{\mathrm{td}(G) - 2}{\mathrm{td}(G)} n.$$

Proof. For the first inequality, we will prove that $\mathrm{td}(G) \leqslant \mathrm{fvs}(G) + \log_2(n) + 1$. To do so, it suffices to give an elimination forest of G of depth $\mathrm{fvs}(G) + \log_2(n) + 1$. The strategy is to eliminate vertices in at most $\mathrm{fvs}(G)$ rounds to reach an acyclic graph, i.e. a forest. Then, since each tree of the forest can be eliminated in at most $\log_2(n) + 1$ rounds by Proposition 4, we obtain an elimination forest of the corresponding depth.

For the second inequality, we want to find a feedback vertex set of G of size $\frac{\mathrm{td}(G)-2}{\mathrm{td}(G)} n$. Let V_i be the set of vertices eliminated during the i-th round for $1 \leqslant i \leqslant \mathrm{td}(G)$. For all i and j with $i \neq j$, V_i and V_j are disjoint and $V_i \cup V_j$ induces a forest of star in G. Consider the sum $\sum_{i \neq j} |V_i| + |V_j|$. In this sum, we count each V_i $\mathrm{td}(G) - 1$ time. Therefore, $\sum_{i \neq j} |V_i| + |V_j| = (\mathrm{td}(G) - 1)n$. There are $\frac{\mathrm{td}(G)(\mathrm{td}(G)-1)}{2}$ pairs (i, j) with $i \neq j$ and $1 \leqslant i, j \leqslant \mathrm{td}(G)$, so $\max\{|V_i| + |V_j| : i \neq j \text{ and } \leqslant i, j \leqslant \mathrm{td}(G)\} \geqslant (\mathrm{td}(G) - 1)n \frac{2}{\mathrm{td}(G)(\mathrm{td}(G)-1)} = \frac{2}{\mathrm{td}(G)} n$. So $\mathrm{fvs}(G) \leqslant n - \frac{2}{\mathrm{td}(G)} n = \frac{\mathrm{td}(G)-2}{\mathrm{td}(G)} n$. □

Both inequalities are tight. The first one is reached for P_n as $\mathrm{td}(P_n) = 1 + \lfloor \log_2(n) \rfloor$ and $\mathrm{fvs}(P_n) = 0$. The second one is reached for *cliques* K_n on n vertices, graphs where every edge is present, as $\mathrm{td}(K_n) = n$ and $\mathrm{fvs}(K_n) = n - 2$.

Unfortunately, in most other cases, the upper bound in Theorem 5 is not very good. For example, consider the *square grid* $G_{k,k}$ where $V(G) = \{(i, j) : 1 \leqslant i \leqslant j \leqslant k\}$ and $E(G) = \{(i, j)(k, l) : |i - k| + |j - l| = 1\}$. It is known that $\mathrm{td}(G_{k,k}) = \Theta(k)$ (the lower bound can be obtained using treewidth [34] and the upper bound consists in eliminating the "middle" row or column at each step) and $\mathrm{fvs}(G_{k,k}) = (\frac{1}{3} + o(1))k^2$ [21]. Using Theorem 5, we only obtain that $\mathrm{fvs}(G_{k,k}) \leqslant k^2 - 2k$.

We managed to bound fvs by deleting only one vertex per connected component in each round. However, one can wonder what happens when we are allowed to delete an induced tree per connected component in each round. In other words, we are eliminating a forest from G in each round. To obtain a small feedback vertex set of G in such a setting, one can leave the biggest forest and delete all other vertices. This leads us to the following result. Let $\mathcal{T}$ be the set of all trees. Let us denote by $\mathrm{ed}^{\mathcal{T}}(G)$ the smallest integer k such that we can eliminate G in k rounds with these new rules.

Proposition 6. *For every graph G of order $n \geqslant 1$,* $\mathrm{ed}^{\mathcal{T}}(G) - 1 \leqslant \mathrm{fvs}(G) \leqslant \frac{\mathrm{ed}^{\mathcal{T}}(G)-1}{\mathrm{ed}^{\mathcal{T}}(G)} n$.

Proposition 6 still gives tight bounds for paths and cliques. Moreover, for square grids, we get that $\mathrm{ed}^{\mathcal{T}}(G_{k,k}) = 2$ for every k (by deleting the tree formed by one column and every other row which leaves a forest of paths) and $\mathrm{fvs}(G_{k,k}) \leqslant \frac{1}{2}k^2$, which is of the right order of magnitude.

2.5 A new parameter: elimination depth

More generally, depending on the problems we are studying, we might want to eliminate some adapted structures that could be more complex than just vertices or *independent sets*, a set of vertices with no edges between them. This motivates the introduction of a new parameter that we call elimination depth. Let $\mathcal{D}$ be a set of graphs. The *elimination depth* of a graph G by deleting graphs in $\mathcal{D}$, denoted $\mathrm{ed}^{\mathcal{D}}$, is defined as follows.

Definition 7. Let G be a graph and let $\mathcal{D}$ be a set of connected graphs containing K_1,

$$\mathrm{ed}^{\mathcal{D}}(G) := \begin{cases} 0 & \text{if } G \text{ is the null graph,} \\ \max\{\mathrm{ed}^{\mathcal{D}}(H) : H \text{ is a connected component of } G\} & \text{if } G \text{ is disconnected,} \\ 1 + \min\{\mathrm{ed}^{\mathcal{D}}(G - V(H)) : H \subseteq_i G \text{ and } H \in \mathcal{D}\} & \text{if } G \text{ is connected.} \end{cases}$$

To ensure that $\mathrm{ed}^{\mathcal{D}}$ is well-defined, we need to be able to partition G into induced subgraphs contained in $\mathcal{D}$. A sufficient condition is for $\mathcal{D}$ to contain K_1, the graph with only one vertex. In this way, elimination depth also generalizes treedepth as $\mathrm{ed}^{\mathcal{D}} = \mathrm{td}$ for $\mathcal{D} = \{K_1\}$. Here, we only allow the deletion of connected graphs because deleting disconnected graphs (with an unbounded number of connected components) removes

the tree-like structure of the decomposition as the corresponding elimination forest will always be a path. As for disconnected graphs with a bounded number of components, this can be approximated by deleting only connected graphs (see Proposition 9).

Moreover, elimination depth also provides upper bounds on other well-known parameters. For example, if $\mathcal{D}$ is the set of trees, then for every graph G, $\mathrm{ed}^{\mathcal{D}}(G)$ is at least the minimum number of parts in a partition of the vertices of G into induced forests (an undirected version of the *dichromatic number* [29]). When $\mathcal{D}$ is the set of cliques, for every graph G, $\mathrm{ed}^{\mathcal{D}}(G)$ is at least the minimum number of parts in a partition of the vertices of G into disjoint union of cliques (also known as the *subchromatic number* [2]).

Similarly to elimination forest for treedepth, we can define a *$\mathcal{D}$-elimination forest* as a pair $(F, (B_f)_{f \in V(F)})$ where F is a rooted forest and $(B_f)_{f \in V(F)}$ are *bags*, sets of vertices of G associated to vertices of F such that $G[B_f] \in \mathcal{D}$ for all $f \in V(F)$. The bags form a partition of $V(G)$ and for all $u \in B_f$, $u' \in B_{f'}$, if $uu' \in E(G)$, then f and f' are in an ancestor-descendant relationship in F. See Figure 4 for an example. The elimination depth of a graph G deleting graphs in $\mathcal{D}$ is the minimum integer k such that G has a $\mathcal{D}$-elimination forest of depth k; such a forest is called an *optimal $\mathcal{D}$-elimination forest* of G. The only $\mathcal{D}$-elimination forest of the null graph is the null graph with no bags. An *optimal $\mathcal{D}$-deletion* in a graph G with $\mathrm{ed}^{\mathcal{D}}(G) = k$ is $W \subseteq V(G)$ such that $G[W] \in \mathcal{D}$ and $\mathrm{ed}^{\mathcal{D}}(G - W) = k - 1$.

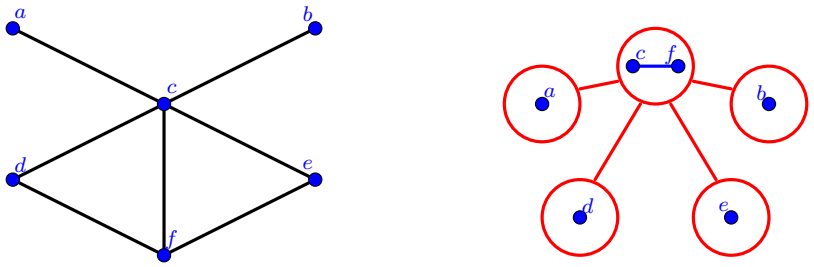


Figure 4: Let $\mathcal{D}$ be the set of paths. A graph with a $\mathcal{D}$-elimination forest of depth 2 (in red).

To better understand elimination depth, we exhibit some of its properties.

Observation 8. *Let $\mathcal{D}$, $\mathcal{D}'$ be two sets of connected graphs containing K_1 such that $\mathcal{D} \subseteq \mathcal{D}'$. For every graph G, $\mathrm{ed}^{\mathcal{D}}(G) \geqslant \mathrm{ed}^{\mathcal{D}'}(G)$.*

This observation comes from the fact that any $\mathcal{D}$-elimination forest of G is a $\mathcal{D}'$-elimination forest of G. In the other direction, if one can partition a graph in $\mathcal{D}'$ into a constant number of induced subgraphs in $\mathcal{D}$, then $\mathrm{ed}^{\mathcal{D}}$ and $\mathrm{ed}^{\mathcal{D}'}$ differ only by this constant factor. More formally, we have the following proposition.

Proposition 9. *Let $\mathcal{D}$, $\mathcal{D}'$ be two sets of connected graphs containing K_1. Let c be an integer with $c \geqslant 1$. If every graph in $\mathcal{D}'$ has a $\mathcal{D}$-elimination path on at most c vertices, then for every graph G, $\mathrm{ed}^{\mathcal{D}}(G) \leqslant c \cdot \mathrm{ed}^{\mathcal{D}'}(G)$.*

Proof. Let $(F, (B_f)_{f \in V(F)})$ be a $\mathcal{D}'$-elimination forest for a graph G. We can construct a $\mathcal{D}$-elimination forest for G from $(F, (B_f)_{f \in V(F)})$ by eliminating each bag in at most c rounds, because for every $f \in V(F)$, $G[B_f] \in \mathcal{D}'$ and so $G[B_f]$ has a $\mathcal{D}$-elimination path on at most c vertices by assumption. In other words, we replace each vertex of F by a path of length at most c and connect the endpoints accordingly (one to the parent, the other to the children) to obtain a $\mathcal{D}$-elimination forest of depth at most c times the depth of F. □

In Proposition 9, we cannot replace "$\mathcal{D}$-elimination path" by "$\mathcal{D}$-elimination forest" (or tree). Suppose that one can eliminate a graph D in $\mathcal{D}'$ by deleting graphs in $\mathcal{D}$ in c rounds using the fact that they were able to delete graphs from multiple components of D in some round. Using the same strategy to eliminate D in G will not work. Indeed, during a round, two vertices can be in separate connected components if we only consider an induced subgraph of D (in $\mathcal{D}'$) of the current graph H ($\subseteq_i G$) but are actually in the same connected component in H. Therefore, eliminating a graph in $\mathcal{D}'$ in c rounds by deleting graphs of $\mathcal{D}$ does not mean that we can eliminate it in c rounds when it is an induced subgraph of a bigger graph.

As a consequence of Proposition 9, if we had allowed the deletion of disconnected graphs with a bounded number of connected components, then a similar bound would hold, and it would only differ from the version with connected graphs by a constant factor. Moreover, if $\mathcal{D}$ contains graphs of order at most c

for some integer c, then $\mathrm{ed}^{\mathcal{D}}$ is a c-approximation of td. For this reason, we will only consider classes of connected graphs containing arbitrarily large graphs (like paths, trees, cycles, and so on).

The motivation of finding interesting bounds for hard problems leads us to study the elimination depths of sparse graphs. More precisely, given a class $\mathcal{C}$ of sparse graphs and a class $\mathcal{D}$ of graphs that we are allowed to delete, we want to know the "worst case scenario" for the corresponding elimination depth. This motivates the study of the extremal values of elimination depth when we fix $\mathcal{C}$ and $\mathcal{D}$.

Definition 10. Let $\mathcal{C}$ be a set of graphs. Let $\mathcal{D}$ be a set of connected graphs containing K_1. The *extremal value* of the elimination depth in $\mathcal{C}$ by deleting graphs from $\mathcal{D}$ is defined as

$$\mathrm{ed}^{\mathcal{D}}(\mathcal{C}) := (n \mapsto \max\{\mathrm{ed}^{\mathcal{D}}(G) \colon G \in \mathcal{C}, |V(G)| = n\}).$$

The same can be defined for treedepth since it corresponds to the case where $\mathcal{D} = \{K_1\}$. Let $\mathcal{T}$ be the class of trees. Clearly, $\mathrm{ed}^{\mathcal{T}}(\mathcal{T}) = 1$ whereas recall that $\mathrm{td}(\mathcal{T}) = \lceil \log_2(n+1) \rceil$. This shows that the extremal values of elimination depths can be in a much different order of magnitude than the extremal values of treedepth depending on the graphs that we are allowed to delete.

2.6 Our main contributions

In this report, we study methods to find lower and upper bounds on the extremal values of elimination depths parameterized by some classes of graphs.

Section 3 is dedicated to lower bounds. First, we give an equivalent definition of elimination depth using cops and robber games in Section 3.1. The game theory point of view lead to the notion of $\mathcal{D}$-shelter in Section 3.2, a certificate to attests of the large elimination depth of a graph. Then in Section 3.3, we provide sufficient conditions on $\mathcal{C}$ and $\mathcal{D}$ for $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ to be at least logarithmic. This result is very general since many usual graphs classes respect those conditions. When $\mathcal{C}$ and $\mathcal{D}$ are really close to each others, our intuition is that $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ is small and maybe even bounded by a constant. The theorem of Section 3.3 provides many examples for which this intuition is false. For example, this is the case when $\mathcal{D}$ is the class of connected graphs of maximum average degree at most r with $r > 2$ and $\mathcal{C}$ is the class of graphs of maximum average degree at most $r + \varepsilon$ for some $\varepsilon > 0$. More precisely, $\mathrm{ed}^{\mathcal{D}}(\mathcal{C}) = \Omega(\log_2(n))$ no matter how close ε is to 0.

Section 4 is dedicated to methods to compute upper bounds on elimination depths. Those methods are based on the notion of central blocks. Section 4.2 is dedicated to the study of central blocks, and to characterizing necessary and sufficient conditions for a graph to have a central block. Then, we derive two methods to bound the elimination depth, one in which we eliminate the central block (see *Section* 4.3), and one in which we separate the central block (see Section 4.4). Graphs of bounded treewidth are known to have logarithmic treedepth. In Section 4.5, we try to find more precise results for elimination depth of graphs of bounded treewidth.

Unlike treedepth, elimination depth is not monotone under taking induced subgraphs. Section 5 is a complete characterization of classes of graphs $\mathcal{D}$ for which $\mathrm{ed}^{\mathcal{D}}$ is monotone under taking induced subgraph. Notice that we introduced the notion of dumbbell operators in Section 3.3.1 to handle monotonicity.

3 Lower bounding extremal values of elimination depths

Compared to treedepth, elimination depth ($\mathrm{ed}^{\mathcal{D}}$) of a class of graphs $\mathcal{C}$ allows deleting many vertices per round. Especially, when $\mathcal{D}$ contains big induced subgraphs of $\mathcal{C}$, it is not always clear that there exist graphs in $\mathcal{C}$ of large elimination depths. In this section, we will explore this problem by first studying a certificate for graphs of large elimination depths, and then we will exhibit sufficient conditions on $\mathcal{C}$ and $\mathcal{D}$ for $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ to be at least logarithmic. We end the section by listing some examples of well-known classes of graphs that respect these conditions.

3.1 Elimination depths as a cops-and-robbers game

The one-player game defined for elimination depths can also be seen as a two-player game with a cop who is trying to catch a robber by eliminating hideouts (vertices) forming certain specific structures (a graph in $\mathcal{D}$), and the robber who is trying to survive by taking shelter in what remains (connected components of the remaining graph). We provide a formal definition of this game below. Upper bounds on elimination depths can be obtained from capturing strategies for the cop, whereas lower bounds are obtained from survival strategies for the robber.

Let t be a nonnegative integer and let G be a graph. Let $\mathcal{D}$ be a set of connected graphs containing K_1. The *t-rounds $\mathcal{D}$-selection-deletion game in G* is a two-player game with a cop and a robber defined as follows.

- If $t = 0$, then the robber chooses a connected component G_0 of G.
- Let $t \geqslant 1$. For $1 \leqslant i \leqslant t$, at the beginning of round i, the cop chooses a set $W_{i-1} \subseteq V(G_{i-1})$ to delete such that $G_{i-1}[W_{i-1}] \in \mathcal{D}$. The robber chooses G_i, a connected component of $G_{i-1} - W_{i-1}$, to take shelter in.
- If G_t is the null graph, then cop wins. Otherwise, the robber wins.

Let G be a graph. A *strategy for the cop* (in a $\mathcal{D}$-selection-deletion game) in G is a (partial) function on certain connected induced subgraphs H of G that maps H to a subset W of $V(H)$ such that $H[W] \in \mathcal{D}$. We consider that the function that returns $\emptyset$ for the null graph is a strategy for the cop. Let C be a connected induced subgraph of G, we define the *restriction of a strategy σ in G to C* as $\sigma|_C$ such that $\sigma|_C(H) = \sigma(H)$ if and only if H is a connected induced subgraph of C and σ is defined for H. We can extend the definition of the restriction of a strategy to the null graph. Let t be a nonnegative integer. A strategy σ for the cop in a t-rounds $\mathcal{D}$-selection-deletion game in G is a *winning strategy* if either G is the null graph or $t \geqslant 1$ and for every connected component C of $G - \sigma(G)$, $\sigma|_C$ is a winning strategy in the $(t-1)$-rounds game in C. The *$\mathcal{D}$-cop number* of a graph G, denoted by $\mathrm{cop}^{\mathcal{D}}(G)$ is the minimum integer t such that the cop has a winning strategy for the t-rounds $\mathcal{D}$-selection-deletion game in G. The $\mathcal{D}$-cop number is well-defined because $\mathcal{D}$ contains K_1. Moreover, notice that $\mathrm{cop}^{\mathcal{D}}(G) = 0$ if and only if G is the null graph.

Theorem 11. *Let $\mathcal{D}$ be a set of connected graphs containing K_1. For every graph G, $\mathrm{ed}^{\mathcal{D}}(G) = \mathrm{cop}^{\mathcal{D}}(G)$.*

Proof. First, let $t = \mathrm{cop}^{\mathcal{D}}(G)$ and we prove that $\mathrm{ed}^{\mathcal{D}}(G) \leqslant t$ by induction on t. To do so, we build a $\mathcal{D}$-elimination forest of depth at most t from a winning strategy σ for the cop in the t-rounds $\mathcal{D}$-selection-deletion game in G. If $t = 0$, then G is the null graph and $\mathrm{ed}^{\mathcal{D}}(G) = 0$. Suppose that $t \geqslant 1$. By induction hypothesis, for every connected component C of $G - \sigma(G)$, $\mathrm{ed}^{\mathcal{D}}(C) \leqslant \mathrm{cop}^{\mathcal{D}}(C)$ because $\mathrm{cop}^{\mathcal{D}}(C) \leqslant t-1$. So there exists a $\mathcal{D}$-elimination forest of $G - \sigma(G)$ (consisting of $\mathcal{D}$-elimination trees of connected components of $G - \sigma(G)$) of depth at most $t - 1$. We add a new root whose bag is $\sigma(G)$ to this forest to obtain a $\mathcal{D}$-elimination forest of depth at most t for G.

Secondly, let $t = \mathrm{ed}^{\mathcal{D}}(G)$ and we prove that $\mathrm{cop}^{\mathcal{D}}(G) \leqslant t$ by induction on t. If $t = 0$, then G is the null graph so $\mathrm{cop}^{\mathcal{D}}(G) = 0$. Suppose that $t \geqslant 1$. Let W be an optimal $\mathcal{D}$-deletion in G. Since $\mathrm{ed}^{\mathcal{D}}(G-W) = t-1$, by the induction hypothesis, for every connected component C of $G - W$, there is a winning strategy σ_C for the $(t-1)$-rounds game in C. We can define a winning strategy σ such that $\sigma(G) = W$ and for every connected component C of $G - W$, for every induced subgraph H of C for which σ_C is defined, $\sigma(H) = \sigma_C(H)$. By definition, σ is a winning strategy for the t-rounds game in G. □

3.2 Certificates for large elimination depths: shelters

Elimination depths can be seen as the smallest number of rounds necessary for the cop to win the corresponding game, but we can also look at it as one more than the maximum number of rounds the robber is sure to survive.

Let G be a nonnull graph. A *strategy for the robber* τ (in a $\mathcal{D}$-selection-deletion game) in G is a (partial) function on certain connected induced subgraph H of G and for every subset W of $V(H)$ that induces a graph in $\mathcal{D}$ such that $\tau(H, W)$ is a connected component of $H - W$. Let C be a connected induced subgraph of G, the *restriction of a strategy τ in G to C* $\tau|_C$ is such that $\tau|_C(H, W) = \tau(H, W)$ if and only if τ is defined on H, a connected induced subgraph of G, and $W \subseteq V(H)$ induces a graph in $\mathcal{D}$. A strategy τ for the robber for the t-rounds $\mathcal{D}$-selection-deletion game in G is a *survival strategy* if either $t = 0$ and G is not the null graph, or $t \geqslant 1$ and for every $W \subseteq V(G)$ such that $G[W] \in \mathcal{D}$, $\tau|_{\tau(G,W)}$ is a survival strategy for the robber for the $(t-1)$-rounds $\mathcal{D}$-selection-deletion game in $\tau(G, W)$. If the robber has a survival strategy for the t-rounds game in G, then we say that the robber *survives* t rounds in G.

The $\mathcal{D}$-robber number of a graph G, denoted by $\mathrm{robber}^{\mathcal{D}}(G)$ is the maximum integer t such that the robber has a survival strategy for the t-rounds $\mathcal{D}$-selection-deletion game in G. By definition, the existence of a survival strategy for the robber in a t-rounds $\mathcal{D}$-selection-deletion game in G implies that there exist no winning strategies for the cop and vice versa. Along with Theorem 11, we have the following observation.

Observation 12. *Let $\mathcal{D}$ be a set of connected graphs G containing K_1. For every graph G,* $\text{robber}^{\mathcal{D}}(G)+1 = \text{cop}^{\mathcal{D}}(G) = \text{ed}^{\mathcal{D}}(G)$.

Similarly to a $\mathcal{D}$-elimination forest of depth t, a structure encoding a winning strategy for the cop in the t-rounds $\mathcal{D}$-selection-deletion game, we give a structure encoding a survival strategy for the robber called $\mathcal{D}$-shelter. Let G be a graph. Let $\mathcal{S}$ be a set of subsets of $V(G)$. For every $S \in \mathcal{S}$, $\mathcal{S}|_S$ denotes $\{S' \in \mathcal{S} \colon S' \subseteq S\}$.

Definition 13. Let G be a nonnull graph and let $\mathcal{D}$ be a set of connected graphs containing K_1. Let $\mathcal{S}$ be a nonempty set of nonempty subsets of $V(G)$ inducing connected graphs in G. We call $\mathcal{S}$ a *$\mathcal{D}$-shelter of G of thickness* 0 if and only if there exists $W \subseteq V(G)$ such that $G[W] \in \mathcal{D}$ and for every $S \in \mathcal{S}$, $W \cap S \neq \emptyset$. Let t be a positive integer. We call $\mathcal{S}$ a *$\mathcal{D}$-shelter of G of thickness t* if and only if for every $W \subseteq V(G)$ such that $G[W] \in \mathcal{D}$, there exists $S \in \mathcal{S}$ and C a connected component of $G - W$ such that $\mathcal{S}|_S$ is a $\mathcal{D}$-shelter of C of thickness at least $t - 1$.

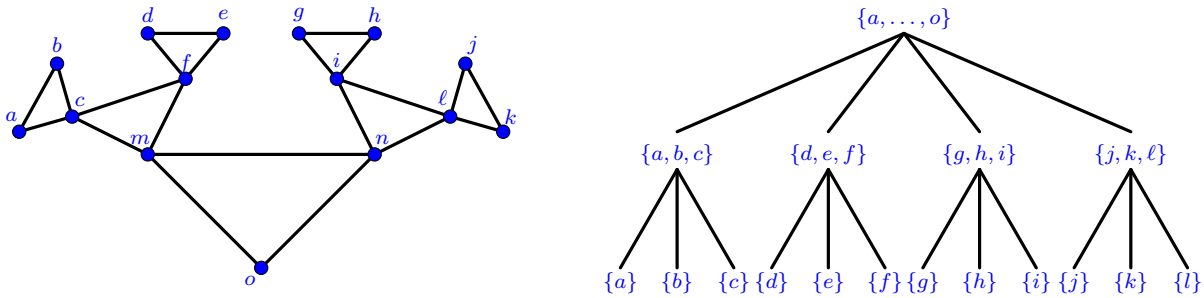


Figure 5: A trees-shelter of thickness 2 (on the right) for the graph on the left (T_3 as defined in Definition 16). The edges between the sets of vertices represent the subset relation (transitive edges are not drawn).

We show that the notions of $\mathcal{D}$-shelter and thickness as defined above correspond to a survival strategy for the robber.

Theorem 14. *Let G be a nonnull graph and let $\mathcal{D}$ be a set of connected graphs containing K_1. Let t be the largest integer such that there exists a $\mathcal{D}$-shelter of G of thickness t. Then* $\text{robber}^{\mathcal{D}}(G) = t$.

Proof. First, we will prove that if there exists $\mathcal{S}$, a $\mathcal{D}$-shelter of G of thickness t, then $\text{robber}^{\mathcal{D}}(G) \geqslant t$, i.e. the robber survives the t-rounds $\mathcal{D}$-selection-deletion game in G. We proceed by induction on t. If $t = 0$, then $\text{robber}^{\mathcal{D}}(G) \geqslant 0$ since G is not the null graph. Suppose that $t \geqslant 1$. Let $W \subseteq V(G)$ such that $G[W] \in \mathcal{D}$ (which always exists since $\mathcal{D}$ contains K_1 and G is nonnull). By definition of $\mathcal{S}$, there exists $S \in \mathcal{S}$ and C a connected component of $G - W$ such that $\mathcal{S}|_S$ is a $\mathcal{D}$-shelter of C of thickness at least $t - 1$. By the induction hypothesis, the robber survives the $(t - 1)$-rounds game in C. Therefore, for any W, if the cop chooses to delete W, then the robber survives $t - 1$ more rounds in C. This proves that the robber survives the t-rounds game in G.

Secondly, we will prove that there exists a $\mathcal{D}$-shelter of G of thickness at least $t = \text{robber}^{\mathcal{D}}(G)$. We proceed by induction on t. If $t = 0$, then $\{V(G)\}$ is a $\mathcal{D}$-shelter of G of thickness 0. Suppose $t \geqslant 1$. For every $W \in V(G)$ such that $G[W] \in \mathcal{D}$, there exists C a connected component of $G - W$ such that the robber survives the $(t - 1)$-rounds game in C. So by induction hypothesis, there exists $\mathcal{S}_W$, a $\mathcal{D}$-shelter of C of thickness at least $t - 1$. Consider $\mathcal{S}$, the shelter constructed by the union of all $\mathcal{S}_W$ and $V(G)$. By construction, $\mathcal{S}$ is a $\mathcal{D}$-shelter of G of thickness at least t. □

As a consequence of Observation 12 and Theorem 14, we have the following equivalent definition of elimination depth using shelters.

Observation 15. *Let G be a nonnull graph and let $\mathcal{D}$ be a set of connected graphs containing K_1. If t is the largest integer such that G has a $\mathcal{D}$-shelter of thickness t, then* $\text{ed}^{\mathcal{D}}(G) = t + 1$.

Our definition of $\mathcal{D}$-shelters and thickness generalize an existing definition of shelters and thickness for treedepth [12] (when $\mathcal{D} = \{K_1\}$).

3.3 Building graphs with large elimination depths

In this section, we will see a method to construct graphs of large elimination depth. First, we illustrate our ideas with a simple class of graphs with large elimination depth when we delete trees.

Definition 16. A *complete binary tree* is a rooted tree such that all its vertices either have 0 or 2 direct descendants (*children*), and all leaves are at the same depth. We define $(T_k)_{k\geqslant 1}$ as the sequence of graphs such that for all $k \geqslant 1$, T_k is the graphs constructed from the complete binary tree of depth $k+1$ by adding edges between pairs of vertices that share a common direct ancestor (*parent*). See Figure 6 for an example.

Another construction for $(T_k)_{k\geqslant 1}$ is by induction, with T_1 defined as K_3, and T_{k+1} being constructed from two copies of T_k glued on a triangle. See Figure 6 for an example of this construction.

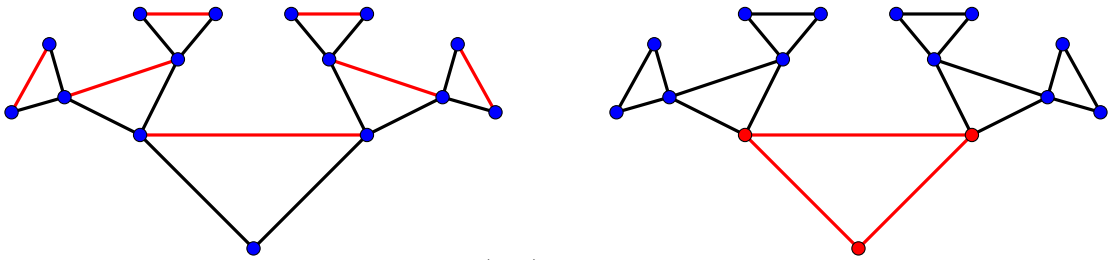


Figure 6: Two ways to build T_3: from adding (red) edges to a complete binary tree of depth 4 and from gluing two copies of T_2 on a new (red) triangle.

Proposition 17. *For every positive integer k,* $\mathrm{ed}^{\mathrm{trees}}(T_k) = 1 + \lceil \frac{k}{2} \rceil = 1 + \left\lceil \frac{\log_2(|V(T_k)|+1)-1}{2} \right\rceil$.

A strategy to reach this elimination depth for T_k, $k > 1$ is to always delete a leaf-to-leaf path that passes through both vertices at depth 2 in the underlying complete binary tree. After such deletion, we obtain connected components that are all elements of $\{K_1, T_1, \ldots, T_{k-2}\}$. Thus, by induction, we have $\mathrm{ed}^{\mathrm{trees}}(T_k) \leqslant 1 + \mathrm{ed}^{\mathrm{trees}}(T_{k-2})$, $\mathrm{ed}^{\mathrm{trees}}(T_1) = 2$, and $\mathrm{ed}^{\mathrm{trees}}(T_2) = 2$. This shows that $\mathrm{ed}^{\mathrm{trees}}(T_k) \leqslant 1 + \lceil \frac{k}{2} \rceil$. Conversely, since we cannot delete a triangle in one round no matter which tree we delete from the graph, there will be a remaining copy of T_{k-2}. We generalize this idea and provide a formal proof of this lower bound in the following subsections. Notice that Figure 5 provides a shelter of thickness 2 for T_3, thus proving that $ed^{\mathrm{trees}}(T_3) \geqslant 3$. The shelter provided for T_3 can be easily extended into a shelter of any T_k. This construction implies a logarithmic lower bound for $\mathrm{ed}^{\mathrm{trees}}$(outerplanar graphs) (where *outerplanar graphs* are planar graphs where every vertex can be drawn on a single *face*, a region bounded by vertices and edges in a planar drawing of the graph).

Unlike treedepth, the difficulty when analyzing the elimination depths of a graph is its non-monotonicity with respect to the induced subgraph relation. More formally, treedepth is *monotone under taking induced subgraph*, that is for all graphs G and H such that $H \subseteq_i G$, $\mathrm{td}(G) \geqslant \mathrm{td}(H)$, while this is not always the case for elimination depth. See Figure 7 for an example.

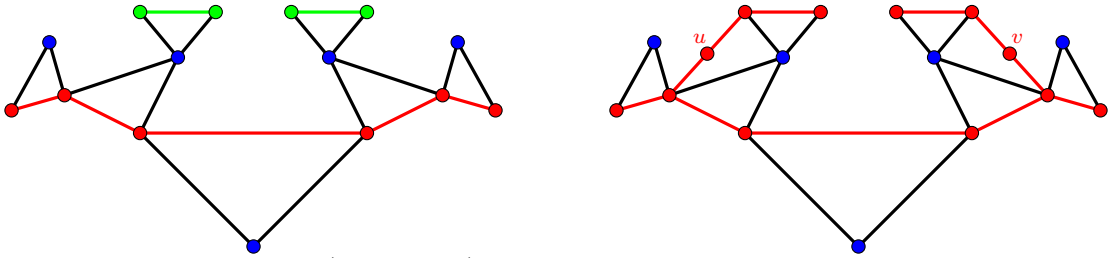


Figure 7: When deleting trees, T_3 (on the left) has elimination depth 3 but it is an induced subgraph (by deleting u and v) of a graph (on the right) with elimination depth 2. The order of deletion is red $\rightarrow$ blue $\rightarrow$ green.

In Section 5, we characterize $\mathcal{D}$ for which $\mathrm{ed}^{\mathcal{D}}$ is monotone under taking induced subgraphs. Next, we introduce a tool to build graphs while "preserving the monotonicity" of their elimination depths.

3.3.1 Dumbbell operators

Dumbbell operators, denoted by $\bullet_p$, are binary graph operators indexed by a positive integer p, that take two graphs and connect them by a path on p vertices. More formally,

Definition 18. Let p be a positive integer. Let G and H be two graphs and let $u \in V(G)$ and $v \in V(H)$. The *dumbbell operator* indexed by p of (G, u) and (H, v), denoted by $(G, u) \bullet_p (H, v)$ is the graph constructed from a copy of G, a copy of H, and the addition of a path connecting G and H, on p vertices, and whose endpoints are u and v. When $p = 1$, this operator is the same as *identifying* u and v, i.e. deleting u from G and v from H and replacing them with w with neighborhood $N_G(u) \cup N_H(v)$.

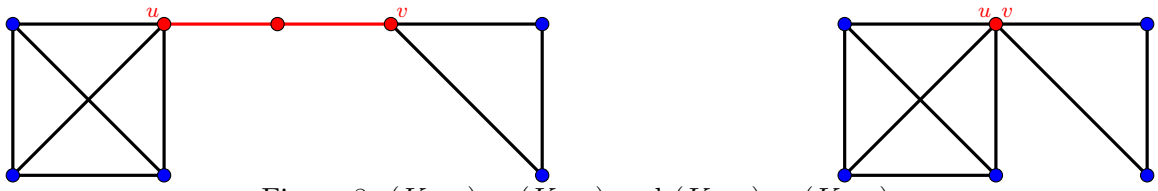


Figure 8: $(K_4, u) \bullet_3 (K_3, v)$ and $(K_4, u) \bullet_1 (K_3, v)$.

Definition 19. Let $\mathcal{D}$ be a set of graphs. We say that $\mathcal{D}$ respects *the interior property over* $\bullet_p$ if and only if for all graphs G and H, and for all vertices $u \in V(G)$, $v \in V(H)$, $(G, u) \bullet_p (H, v) \in \mathcal{D} \implies G \in \mathcal{D}$ and $H \in \mathcal{D}$.

Proposition 20. *Let $\mathcal{D}$ be a set of connected graphs containing K_1. If $\mathcal{D}$ respects the interior property over $\bullet_1$, then for all graphs G, H and for all vertices $u \in V(G)$, $v \in V(H)$, $\mathrm{ed}^{\mathcal{D}}((G, u) \bullet_1 (H, v)) \geqslant \max(\mathrm{ed}^{\mathcal{D}}(G), \mathrm{ed}^{\mathcal{D}}(H))$.*

Proof. Since $\bullet_1$ is a symmetric operator, it is enough to prove that $\mathrm{ed}^{\mathcal{D}}((G, u) \bullet_1 (H, v)) \geqslant \mathrm{ed}^{\mathcal{D}}(G)$. Let $(F, (B_f)_{f \in V(F)})$ be an optimal $\mathcal{D}$-elimination forest of depth d for $(G, u) \bullet_1 (H, v)$. We can now construct $(F', (B'_f)_{f \in V(F')})$, an elimination forest for G, from $(F, (B_f)_{f \in V(F)})$ by considering the restriction of every bag to $V(G)$ and removing all the empty bags. We preserve the ancestor-descendant relationship in each tree by making the parent of an empty bag the parent of its children if they exist. We claim that the resulting decomposition is a $\mathcal{D}$-elimination forest of depth at most d of G. By construction, the new bags form a partition of $V(G)$, for all $w \in B'_f$ and $x \in B'_{f'}$, if $wx \in E(G)$ then f and f' are in an ancestor-descendant relationship in F', and the depth of F' is at most d. It suffices to prove that for every $f \in V(F')$, $G[B'_f] \in \mathcal{D}$. Indeed, either $B'_f = B_f \in \mathcal{D}$ or $((G, u) \bullet_1 (H, V))[B_f]$ is isomorphic to $(G', u) \bullet_1 (H', v) \in \mathcal{D}$ for some connected induced subgraphs $G' \subseteq_i G$ and $H' \subseteq_i H$, in which case $G[B'_f]$ is isomorphic to $G' \in \mathcal{D}$ since $\mathcal{D}$ respects the interior property over $\bullet_1$. □

Corollary 21. *Let $\mathcal{D}$ be a class of connected graphs containing K_1. If $\mathcal{D}$ respects the interior property over $\bullet_1$, then for every positive integer p, for all graphs G, H and for all vertices $u \in V(G)$, $v \in V(H)$, $\mathrm{ed}^{\mathcal{D}}((G, u) \bullet_p (H, v)) \geqslant \max(\mathrm{ed}^{\mathcal{D}}(G), \mathrm{ed}^{\mathcal{D}}(H))$.*

Proof. Let P be a path on p vertices, and let p_1, p_2 be its two endpoints. By definition of $\bullet_p$ and by Proposition 20, $\mathrm{ed}^{\mathcal{D}}((G, u) \bullet_p (H, v)) = \mathrm{ed}^{\mathcal{D}}(((G, u) \bullet_1 (P, p_1), p_2) \bullet_1 (H, v)) \geqslant \max(\mathrm{ed}^{\mathcal{D}}((G, u) \bullet_1 (P, p_1)), \mathrm{ed}^{\mathcal{D}}(H)) \geqslant \max(\mathrm{ed}^{\mathcal{D}}(G), \mathrm{ed}^{\mathcal{D}}(H))$. □

Corollary 21 provides a notion of monotonicity of elimination depths with respect to dumbbell operators.

3.3.2 Building graphs with dumbbell operators

In this section, we will see how to use dumbbell operators to construct graphs of large elimination depths. Let G be a graph. A *cut-vertex* of G is a vertex v such that $G - v$ contains more connected components than G. We say that G is 2-*connected* if and only if G is connected and contains no cut-vertices. A *block* B of G is a maximal set of vertices such that $G[B]$ is 2-connected. Let A be the set of cut-vertices of G and let $\mathcal{B}$ be the set of blocks of G. The *block graph* of G is the graph F such that $V(F) = A \cup \mathcal{B}$ and $E(F) = \{aB \colon a \in A, B \in \mathcal{B}, \text{ and } a \in B\}$, see Figure 9 for an example. The block graph of a graph is a forest by the definition of blocks. The *block decomposition* of G is the pair $(F, \mathcal{B})$. By definition, a cut-vertex must be contained in at least two blocks, therefore, a leaf of a block graph is always a block. A block that is a leaf of F is a *leaf block*. We will study block decompositions in more detail in Section 4.2.

Lemma 22. *Every graph G contains a vertex that is not a cut-vertex.*

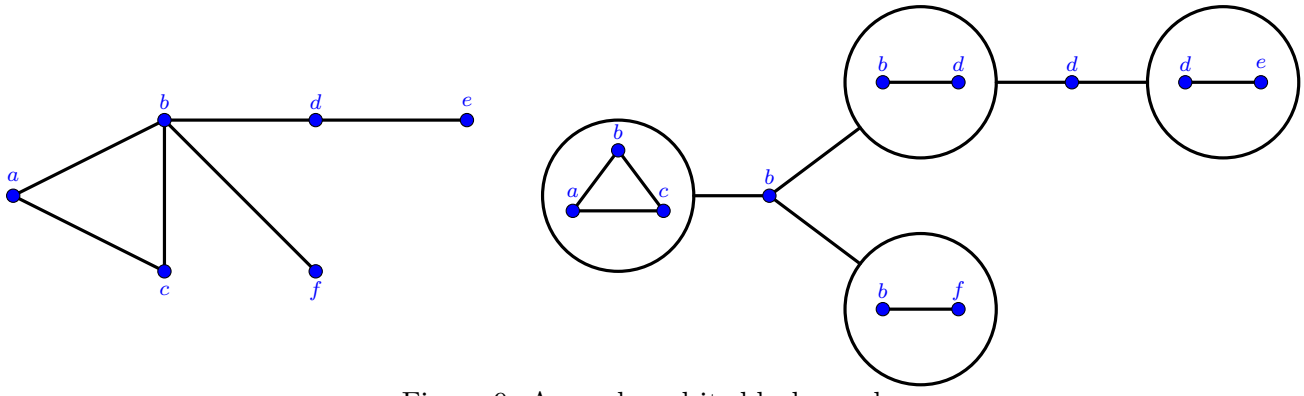


Figure 9: A graph and its block graph.

Proof. If G is 2-connected, then any vertex of G works. Otherwise, let $(F, \mathcal{B})$ be a block decomposition of G. Let B be a leaf block of G. let v be the cut-vertex that connects B to the rest of G. Since v is a cut-vertex in G, $B - v$ is not empty so any vertex in $B - v$ works. □

While the interior property of $\mathcal{D}$ over $\bullet$ ensures that the elimination depth by deleting $\mathcal{D}$ is monotone with respect to $\bullet$, to build graphs with large elimination depths in $\mathcal{C}$, the class of graphs which we are trying to eliminate, we need $\mathcal{C}$ to contain the graphs obtained by $\bullet$, i.e. for $\mathcal{C}$ to respect the closure property under $\bullet$.

Definition 23. Let p be a positive integer. We say that a set of graph $\mathcal{C}$ is *closed under* $\bullet_p$ if and only if for all graphs G and H in $\mathcal{C}$, for all vertices $u \in V(G)$ and $v \in V(H)$, $(G, u) \bullet_p (H, v) \in \mathcal{C}$.

Either every graph of $\mathcal{C}$ can be eliminated in one round, or there exists $G \in \mathcal{C} \setminus \mathcal{D}$ and by gluing copies of G together with $\bullet$, we can create graphs in $\mathcal{C}$ whose structures are similar to that of complete n-ary trees and their elimination depths will be linear in the depth of these trees.

Theorem 24. *Let $\mathcal{C}$ and $\mathcal{D}$ be two sets of connected graphs such that $\mathcal{C} \not\subseteq \mathcal{D}$. If $\mathcal{D}$ contains K_1 and respects the interior property over $\bullet_1$ and if there exists a positive integer p such that $\mathcal{C}$ is closed under $\bullet_p$, then there exists a sequence $(G_k)_{k\geqslant 1}$ of graphs of $\mathcal{C}$ of increasing order and a positive real number c such that for every $k \geqslant 1$, $\mathrm{ed}^{\mathcal{D}}(G_k) \geqslant c \log |V(G_k)|$.*

Proof. We claim that there exists $G \in \mathcal{C} \setminus \mathcal{D}$ of order at least 3. Indeed, $K_1 \in \mathcal{D}$ so there exists $G' \in \mathcal{C} \setminus \mathcal{D}$ of order at least 2. Since $\mathcal{C}$ is closed under $\bullet_p$, there exists $u \in V(G')$ such that $(G', u) \bullet_p (G', u) \in \mathcal{C}$ and $|(G', u) \bullet_p (G', u)| \geqslant 3$. Let G be $(G', u) \bullet_p (G', u)$. Since $\mathcal{D}$ respects the interior property over $\bullet_1$ (and therefore $\bullet_p$), $G \notin \mathcal{D}$. By Lemma 22, we choose $r \in V(G)$ that is not a cut-vertex. We define the sequence $(G_k)_{k\geqslant 1}$ as follows: $G_1 := G$ and we rename r to r_1, and G_{k+1} is constructed by the following procedure.

$G_{k+1} \leftarrow G$
for $v \in V(G) - r$ **do**
  $G_{k+1} \leftarrow (G_{k+1}, v) \bullet_p (G_k, r_k)$
end for

and we rename r to r_{k+1}. Since r is not a cut-vertex of G, r_k is not a cut-vertex of G_k. We denote by R_k the vertices of the copy of G in G_k that contains r_k. For every $k \geqslant 1$, $G_k \in \mathcal{C}$ because $\mathcal{C}$ is closed under $\bullet_p$. For example, $(G_k)_{k\geqslant 1}$ is exactly $(T_k)_{k\geqslant 1}$ when G is a triangle and $p = 1$ (Figure 6). The following is the key argument to obtain the desired lower bound on $\mathrm{ed}^{\mathcal{D}}(G_k)$.

Claim 25. *For every $k \geqslant 3$, $\mathrm{ed}^{\mathcal{D}}(G_k) \geqslant 1 + \mathrm{ed}^{\mathcal{D}}(G_{k-2})$.*

Proof of claim. Let W be an optimal $\mathcal{D}$-deletion for G_k. There exists a vertex v in $R_k \setminus W$, indeed, every connected supergraph of $G_k[R_k]$ in G_k can be obtained from a series of dumbbell operators $\bullet_1$ between G and some connected induced graphs of $G_k - R_k$. Therefore, $G_k[W] \in \mathcal{D}$ cannot contain R_k which induces a copy of G since $G \notin \mathcal{D}$ and $\mathcal{D}$ respects the interior property over $\bullet_1$. There are two cases to consider for v.

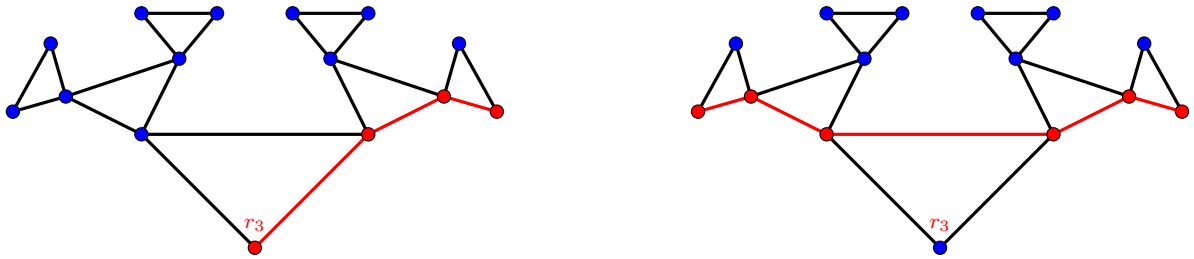
- Suppose that $v \neq r_k$ (see Figure 10a). The connected component C containing v of $G_k - W$ is isomorphic to $(G_{k-1}, r_{k-1}) \bullet_1 (H, h)$ for some graph H and vertex $h \in V(H)$. Therefore, by Corollary 21,

$$\mathrm{ed}^{\mathcal{D}}(G_k) = 1 + \mathrm{ed}^{\mathcal{D}}(G_k - W) \geqslant 1 + \mathrm{ed}^{\mathcal{D}}(C) \geqslant 1 + \mathrm{ed}^{\mathcal{D}}(G_{k-1}) \geqslant 1 + \mathrm{ed}^{\mathcal{D}}(G_{k-2}).$$

- Suppose that $v = r_k$ (see Figure 10b). As a consequence, $R_k \setminus \{r_k\} \cap W$ is not empty. Consider a copy of G_{k-1} in G_k and the vertices W' of W intersecting this copy. Since $R_k \setminus \{r_k\} \cap W$ is not empty, we have that $r_{k-1} \in W'$. Similarly to the above arguments, $R_{k-1} \setminus W'$ is not empty and contains $v' \neq r_{k-1}$ such that the connected component C containing v' of $G_{k-1} - W'$ (and therefore of $G_k - W$) is isomorphic to $(G_{k-2}, r_{k-2}) \bullet_1 (H, h)$ for some graph H and vertex $h \in V(H)$. Therefore, by Corollary 21,

$$\mathrm{ed}^{\mathcal{D}}(G_k) = 1 + \mathrm{ed}^{\mathcal{D}}(G_k - W) \geqslant 1 + \mathrm{ed}^{\mathcal{D}}(C) \geqslant 1 + \mathrm{ed}^{\mathcal{D}}(G_{k-2}).$$

♢



(a) the case where $r_k \in W$. (b) the case where $r_k \notin W$.

Figure 10: The two cases of the previous proof, W is in red in both cases.

Since $G_1 = G \notin \mathcal{D}$, $\mathrm{ed}^{\mathcal{D}}(G_1) \geqslant 2$ and $\mathrm{ed}^{\mathcal{D}}(G_2) \geqslant 2$. Therefore, for every $k \geqslant 1$, $\mathrm{ed}^{\mathcal{D}}(G_k) \geqslant 1 + \lceil \frac{k}{2} \rceil$ as a direct corollary of the previous claim. Moreover, we claim that for every $k \geqslant 1$, $|V(G_k)| = \frac{(|V(G)|-1)^{k+1}-1}{|V(G)|-2}$. Indeed, we will proceed by induction on k. If $k = 1$, then $|V(G_1)| = |V(G)| = \frac{(|V(G)|-1)^2-1}{|V(G)|-2}$. Let $k \geqslant 1$ be such that $|V(G_k)| = \frac{(|V(G)|-1)^{k+1}-1}{|V(G)|-2}$. Since G_{k+1} is constructed by $|V(G)| - 1$ copies of G_k and one additional vertex (r_{k+1}), $|V(G_{k+1})| = (|V(G)| - 1)|V(G_k)| + 1 = (|V(G)| - 1)\frac{(|V(G)|-1)^{k+1}-1}{|V(G)|-2} + 1$ by induction hypothesis. Therefore, $|V(G_{k+1})| = \frac{(|V(G)|-1)^{k+2}-1}{|V(G)|-2}$. Finally, we get $\mathrm{ed}^{\mathcal{D}}(G) \geqslant 1 + \lceil \frac{k}{2} \rceil = 1 + \left\lceil \frac{1}{2} \left(\frac{\log((|V(G)|-2)|V(G_k)|+1)}{\log(|V(G)|-1)} - 1 \right) \right\rceil$. In other words, there exists a positive real number c such that for every $k \geqslant 1$, $\mathrm{ed}^{\mathcal{D}}(G_k) \geqslant c \log |V(G_k)|$, which concludes the proof. □

Another way to interpret this proof on the lower bound on $\mathrm{ed}^{\mathcal{D}}(G_k)$ is through the construction of a $\mathcal{D}$-shelter of G_k of thickness $\lceil \frac{k}{2} \rceil$ defined recursively as the union of $V(G_k)$ and $\mathcal{D}$-shelters of copies of G_{k-2}. As for G_1 and G_2, $\mathcal{D}$-shelters of thickness 1 exist because they are not in $\mathcal{D}$.

Theorem 24 is stated only for $\mathcal{C}$ with connected graphs. However, in general, the extremal values for elimination depths for any class of graphs are the same as when restricted to only connected graphs in that class by definition of elimination depth.

This theorem provides constructive examples for logarithmic lower bounds on $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ for a wide range of classes of graphs $\mathcal{D}$ and $\mathcal{C}$. There are examples of applications of this theorem even when $\mathcal{D}$ and $\mathcal{C}$ are really close to each other. The following table is a list of classes of graphs that respect the hypothesis of Theorem 24, i.e. $\mathcal{D}$ (contains connected graphs and K_1 and) has the interior property over $\bullet_1$ and $\mathcal{C}$ (contains connected graphs and) has the closure property under $\bullet_p$. The proofs that those classes of graphs indeed respect the hypothesis are in Appendix A. For example, elimination depth is at least logarithmic when we eliminate chordal graphs by deleting graphs of degeneracy at most k or when we eliminate graphs with mad $\leqslant r + \varepsilon$ by deleting graphs with mad $\leqslant r$.

$\mathcal{D}$
K_1
girth $\geqslant k+1$
treewidth $\leqslant k$
degeneracy $\leqslant k$
chromatic number $\leqslant k$
trees
cycles
cliques
C_k-minor free
mad $\leqslant r$, $r > 2$
eulerian graphs

$\mathcal{C}$	p
girth $\geqslant k$	1
treewidth $\leqslant k+1$	1
degeneracy $\leqslant k+1$	3
chromatic number $\leqslant k+1$	1
planar graphs	1
chordal graphs	1
K_t-minor free	1
C_{k+1}-minor free	1
mad $\leqslant r+\varepsilon$, $r > 2$	$2+2/(r+\varepsilon-2)$

Table 1: Examples of classes of graphs respecting the hypothesis of Theorem 24.

4 Upper bound

In the previous section, we saw sufficient conditions on $\mathcal{D}$ and $\mathcal{C}$ for $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ to be at least logarithmic. For instance, when we delete cycles from planar graphs. A result by Miller [23] implies that this bound is tight, i.e. for every planar graph, there exists a cycles-elimination forest of depth at most logarithmic in the number of vertices. We discuss this result further in Section 4.4. More generally, we will explore methods to prove that $\mathrm{ed}^{\mathcal{D}}(\mathcal{C})$ is at most logarithmic. The main idea behind these methods is a divide-and-conquer paradigm where we try to find graphs of $\mathcal{D}$ that "cut the graph (of $\mathcal{C}$) in half". Doing so repeatedly will yield our desired logarithmic bound. This leads to the notion of balanced separators.

4.1 Balanced separators

Definition 26. Let G be a graph or order n and let $\alpha \in (0,1)$. We say that $S \subseteq V(G)$ is an α-separator if and only if every connected component of $G - S$ contains at most αn vertices.

For example, Lipton and Tarjan proved that any planar graph contains a $\frac{2}{3}$-separator on $\mathcal{O}(\sqrt{n})$ vertices [19]. Small separators have applications in the design of algorithms in the field of parameterized complexity [1]. Lemma 3 is a simple example of the existence of a vertex that is a $\frac{1}{2}$-separator in trees. In the following, we will be interested in balanced separators with structural constraints, as they have to be in $\mathcal{D}$, rather than size constraints. We say that a set of graphs is *hereditary* if and only if it is closed under taking induced subgraphs.

Proposition 27. *Let $\mathcal{D}$ be a set of connected graphs containing K_1. Let $\mathcal{C}$ be a hereditary set of graphs. If there exists a positive integer c and $\alpha \in (0,1)$ such that for every graph $G \in \mathcal{C}$, G has an α-separator that has a $\mathcal{D}$-elimination path on at most c vertices, then* $\mathrm{ed}^{\mathcal{D}}(\mathcal{C}) \leqslant 1 + c\log_{\alpha^{-1}}(n)$.

Proof. Let G be a graph of order n in $\mathcal{C}$. Let $T(n)$ be the number of rounds need to eliminate G by deleting graphs in $\mathcal{D}$. We proceed by induction on n. If $n = 1$, then $\mathrm{ed}^{\mathcal{D}}(G) = 1$. Otherwise, if $n > 1$, then we can eliminate an α-separator of G in at most c rounds. Therefore, $T(n) \leqslant cT(\alpha n)$ and $T(1) = 1$. So, $T(n) \leqslant 1 + c\log_{\alpha^{-1}}(n)$. Notice that similarly to Proposition 9, we cannot replace "$\mathcal{D}$-elimination path" by "$\mathcal{D}$-elimination forest". □

4.2 Central blocks

Our methods are based on the notion of central blocks. A block is central if its "interior" is a $\frac{1}{2}$-separator and thus the block itself is a $\frac{1}{2}$-separator. More formally,

Definition 28. Let G be a connected graph and let $(F, \mathcal{B})$ be a block decomposition of G. Let $B \in \mathcal{B}$. Let T be a tree of $F - B$. We call $C \coloneqq G[\{u \in U : U \in V(T)\}]$ a *branch of the block decomposition of G rooted at B*. A block B is *central* if and only if each branch of the block decomposition rooted at B contains at most $\frac{1}{2}n$ vertices.

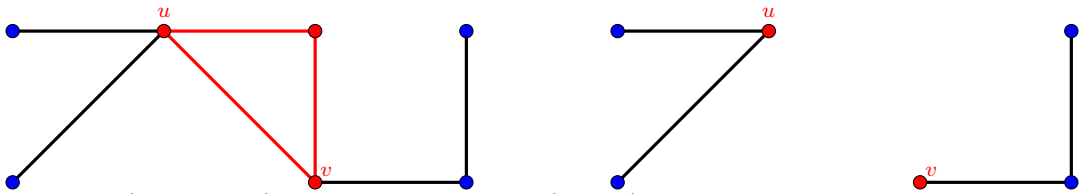


Figure 11: A graph (on the left) with a central block (in red) whose cut-vertices are u and v and the same graph (on the right) after the removal of the interior of its central block.

It is possible that graphs do not contain central blocks. The following is a characterization of graphs that have central blocks.

Proposition 29. *Every connected graph G contains a central block or a cut-vertex that is a $\frac{1}{2}$-separator.*

Proof. Suppose that G has no cut-vertex that is a $\frac{1}{2}$-separator. Let T be a *spanning tree* of G (T is a subgraph of G that is a tree and $V(T) = V(G)$), rooted at v such that $\{v\}$ is a $\frac{1}{2}$-separator of T which exists due to Lemma 3. We distinguish two cases.

- Suppose that v is a cut-vertex of G. By hypothesis, $\{v\}$ is not a $\frac{1}{2}$-separator of G. Therefore, there is a graph H that is a connected component of $G - v$ such that $|V(H)| > \frac{1}{2}n$. Let B be the block of G containing v that intersects H. We claim that B is central.
  If B is a leaf block of G, then v is the only cut-vertex of G in B. Thus, $|V(B)| = |V(H) \cup \{v\}| > \frac{1}{2}n$. Hence, $|V(G - (B \setminus \{v\}))| \leqslant \frac{1}{2}n$. In other words, B is central.
  Otherwise, let $G_1, \ldots, G_k$ be the branches of the block decomposition of G rooted at B. Suppose without loss of generality that $v \in V(G_1)$. Observe that $|V(G_1)| < \frac{1}{2}n$ because $V(G_1)$ is contained in $G - V(H)$. For $i > 1$, $V(G_i)$ is contained in a connected component of $T - v$. Therefore, $|V(G_i)| \leqslant \frac{1}{2}n$ because $\{v\}$ is a $\frac{1}{2}$-separator of T. Thus, B is central.
- Otherwise, let B be a block of G such that $v \in B$. Let $G_1, \ldots, G_k$ be the branches of the block decomposition of G rooted at B. For $i \geqslant 1$, $V(G_i)$ is contained in a connected component of $T - v$. Therefore, $|V(G_i)| \leqslant \frac{1}{2}n$ because $\{v\}$ is a $\frac{1}{2}$-separator of T. Thus, B is central. □

Propositions 27 and 29 imply that we can eliminate G efficiently either by deleting a cut-vertex or a central block in a constant number of rounds.

4.3 Eliminating the central block

While Proposition 27 requires an α-separator to have a $\mathcal{D}$-elimination path, when this separator is a central block, we only need it to have a $\mathcal{D}$-elimination forest. Indeed, the purpose is to ensure that we can eliminate the separator in a constant number of rounds. In the case of a block B of a graph G, during the elimination process, two connected components (induced subgraphs of $G[B]$) cannot be connected in G through vertices in $G - B$. This leads to the following proposition.

Proposition 30. *Let $\mathcal{D}$ be a set of connected graphs containing K_1. Let $\mathcal{C}$ be a hereditary set of graphs. If there exists a positive integer c such that for every graph $G \in \mathcal{C}$, if G has a central block B such that $\mathrm{ed}^{\mathcal{D}}(G[B]) \leqslant c$, then $\mathrm{ed}^{\mathcal{D}}(\mathcal{C}) \leqslant 1 + c\log_2(n)$.*

As a consequence of Proposition 30, if 2-connected graphs in $\mathcal{C}$ have constant elimination depths, then every graph in $\mathcal{C}$ would have logarithmic elimination depths. An example of this idea is for the elimination depths of graphs with a forbidden cycle as a minor by deleting graphs with a smaller forbidden cycle as a minor. A graph H is a *minor* of a graph G, denoted by $H \subseteq_m G$ if H can be obtained from G by vertex deletions, edge deletions, and *edge contractions* (contracting an edge uv means that we identify u and v). We say that a graph G is *H-minor-free* if and only if H is not a minor of G. The seminal Graph Minors series (starting with [32]) of Robertson and Seymour is the foundation of modern structural graph theory. See also [20] for a survey about Graph Minor theory. A famous characterization of planar graphs is Kuratowski's theorem [17] stating that a graph is planar if and only if it is K_5-minor-free and $K_{3,3}$-minor free (the graph formed by two independent sets of size 3 with all 9 edges in between). Let $\mathcal{H}$ be a (finite) set of graphs. We define $\mathcal{F}orb_m(\mathcal{H}) := \{G\colon \text{ for all } H \in \mathcal{H}, H \not\subseteq_m G\}$. So the set of planar graphs can be defined as $\mathcal{F}orb_m(\{K_5, K_{3,3}\})$. From now on, we also use another characterization of 2-connected graphs which is a consequence of Menger's Theorem [22].

Observation 31. *A graph G of order at least 3 is 2-connected if and only if for all S and T disjoint vertex sets of G, there are two internally vertex-disjoint paths where each has one endpoint in S and one in T.*

To illustrate our idea, in the following, we will be interested in eliminating C_{k+1}-minor-free graphs by deleting C_k-minor-free graphs. We will write $\mathcal{F}orb_m(C_k)$ instead of $\mathcal{F}orb_m(\{C_k\})$ for conciseness.

Lemma 32. *Let k be an integer with $k \geqslant 3$. Let G be a 2-connected graph in $\mathcal{F}orb_m(C_{k+1})$. Any pair of cycles of length k (as subgraphs) in G intersect on at least two vertices.*

Proof. Let C and C' be cycles of length k (as subgraphs) in G. Suppose that $|V(C) \cap V(C')| \leqslant 1$.

- Suppose that $|V(C) \cap V(C')| = 0$. Since G is 2-connected and is of order at least 3, there are two internally vertex-disjoint paths P and Q with length at least 1 between $V(C)$ and $V(C')$. Therefore, $G[V(C) \cup V(C') \cup V(P) \cup V(Q)]$ contains a cycle of length at least $k+1$ as a subgraph. It suffices to contract some edges this cycle (if necessary) to obtain C_{k+1}. This contradicts the fact that G is C_{k+1}-minor-free.
- Suppose that $|V(C) \cap V(C')| = 1$. Since G is 2-connected, their intersection is not a cut-vertex. Therefore, there exists a path P internally vertex-disjoint from this intersection between $V(C)$ and $V(C')$. Similarly to the previous point, $G[V(C) \cup V(C') \cup V(P)]$ is not C_{k+1}-minor-free. □

Lemma 33. *Let k be an integer with $k \geqslant 3$. Let $G \in \mathcal{F}orb_m(C_{k+1})$. Let $\mathcal{D}$ be the set of connected graphs in $\mathcal{F}orb_m(C_k)$. If G is 2-connected, then $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 2$.*

Proof. If $G \in \mathcal{F}orb_m(C_k)$, then $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 1$. Otherwise, G contains cycles of length k as minors. Since G is C_{k+1}-minor-free, G contains cycles of length k as subgraphs. Let C be such a cycle and let $v \in V(C)$. First, we can delete the connected subgraph induced by $V(C-v)$ in G. This graph is order $k-1$ so it is C_k-minor-free. By Lemma 32, $G - V(C-v)$ contains no cycles of length k. Therefore, $G - V(C-v)$ is C_k-minor-free and can be eliminated in one round. □

Theorem 34. *Let k be an integer with $k \geqslant 3$. Let $G \in \mathcal{F}orb_m(C_{k+1})$. Let $\mathcal{D}$ be the set of connected graphs in $\mathcal{F}orb_m(C_k)$. If G of order n, then $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 1 + 2\log_2(n)$.*

Proof. If G is 2-connected, then $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 2$ by Lemma 33. Otherwise, by Proposition 29, G contains a cut-vertex that is a $\frac{1}{2}$-separator, which can be deleted in one round, or a central block in two rounds (again by Lemma 33). By Proposition 30, we get our desired result. □

We can generalize this result to eliminating graphs in $\mathcal{F}orb_m(C_{k+c+1})$ by deleting connected graphs in $\mathcal{F}orb_m(C_k)$ for any nonnegative integer c.

Theorem 35. *Let k and c be integers such that $k \geqslant 3$ and $c \geqslant 0$. Let $G \in \mathcal{F}orb_m(C_{k+c+1})$. Let $\mathcal{D}$ be the set of connected graphs in $\mathcal{F}orb_m(C_k)$. If G is of order n, then $\mathrm{ed}^{\mathcal{F}orb_m(C_k)}(G) \leqslant 1 + \lambda\log_2(n)$, with $\lambda = \frac{(c+1)(4k+c-2)}{2(k-1)}$.*

Proof. To prove this result, we will proceed by first considering G to be 2-connected, and then we will generalize to any graphs in $\mathcal{D}$. Suppose that G is 2-connected and C_{k+c+1}-minor free. We will prove that we can eliminate G in at most $\frac{(c+1)(4k+c-2)}{2(k-1)}$ rounds by induction on c. If $c = 0$, then we can eliminate G in at most 2 rounds by Lemma 33 and $\frac{(4k-2)}{2(k-1)} \geqslant 2$ since $k \geqslant 3$. Let $c \geqslant 1$ and let G be C_{k+c+1}-minor free. If G contains no cycle of length $k+c$ then G is $C_{k+(c-1)+1}$-minor free and by induction hypothesis, we can eliminate G in at most $\frac{((c-1)+1)(4k+(c-1)-2)}{2(k-1)} \leqslant \frac{(c+1)(4k+c-2)}{2(k-1)}$ rounds. Otherwise, G contains at least one cycle of length $k+c$. By Lemma 32, deleting $k+c-1$ vertices of a cycle of length $k+c$ kills all the cycles of length $k+c$ in G. A strategy to eliminate G is to first eliminate all cycles of length $k+c$ in G in at most $\left\lceil \frac{k+c}{k-1} \right\rceil \leqslant \frac{k+c}{k-1} + 1 = \frac{2k+c-1}{k-1}$ rounds, and then to eliminate the rest of G in at most $\frac{((c-1)+1)(4k+(c-1)-2)}{2(k-1)}$ rounds by induction hypothesis. In total, we need at most $\frac{((c-1)+1)(4k+(c-1)-2)}{2(k-1)} + \frac{2k+c-1}{k-1} = \frac{(c+1)(4k+c-2)}{2(k-1)}$ rounds. □

4.4 Separating the central block

A second idea to obtain logarithmic upper bounds on elimination depths is that a separator of a central block might extend to a balanced separator of the whole graph. This strategy can be used to delete cycles in planar graphs. A *weighted graph* is a graph with an assignment of positive weights on the vertices. The *total weight* of a graph is the sum of the weights on the vertices. For $\alpha \in (0,1)$, a *weighted α-separator* of a graph G is a set $S \subseteq V(G)$ such that the connected components of $G - S$ have total weight at most α times the total weight of G.

Theorem 36 (Miller, 1986 [23])**.** *Let G be a 2-connected planar weighted graph. There exists an induced cycle in G that is a weighted $\frac{2}{3}$-separator of G.*

Let $\mathcal{C}$ be hereditary class of graphs. A weighted balanced separator with the right structural constraints for every 2-connected weighted graph of $\mathcal{C}$ always extends to a balanced separator with the same constraints for all graphs of $\mathcal{C}$. Indeed, let $G \in \mathcal{C}$. By Proposition 29, if G contains no cut-vertex that is a $\frac{1}{2}$-separator, then G has a central block B. Consider the branches of the block decomposition of G rooted at B. We can assign the following weights to the vertices in B. If $v \in B$ is a cut-vertex of G, then its weight is the order of its associated branch. Otherwise, $v \in B$ has weight 1. A weighted balanced separator of $G[B]$ with such a weight assignment is a balanced separator of G. This leads to the following theorem.

Theorem 37 (Miller, 1986 [23])**.** *Let G be a connected planar graph. Either G contains a cut-vertex that is a $\frac{1}{2}$-separator, or an induced cycle that is a $\frac{2}{3}$-separator.*

As a consequence, we get the following results on elimination depths of planar graphs.

Corollary 38. *Let G be a planar graph of order n and let $\mathcal{D}$ be the set of cycles and K_1, $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 1 + \log_{3/2}(n)$.*

Deleting an induced cycle is the same as deleting two induced paths, so Corollary 38 also proves that $\mathrm{ed}^{\mathrm{paths}}(G) \leqslant 1 + 2\log_{3/2}(n)$. There are actually planar graphs which cannot be separated by deleting a (single) tree, see Appendix B for more details.

4.5 Bounding elimination depths for graphs of bounded treewidth

Treewidth is a well-known parameter introduced by Robertson and Seymour in [33]. Graphs with bounded treewidth are known for their balanced separators. Therefore, we are also interested in studying elimination depths for such graphs. First, we introduce the notion of tree decomposition.

Definition 39. A *tree decomposition of a graph G* is a pair $(T, (B_t)_{t \in V(T)})$ where T is a tree, $(B_t)_{t \in V(T)}$ are subsets of $V(G)$ associated to the vertices of T and called *bags* such that

- For every $uv \in E(G)$, there exists $t \in V(T)$ such that u and v are in B_t.
- For every $v \in V(G)$, T_v is connected, where T_v is the induced subgraph of T such that $t \in V(T_v)$ if and only if $v \in B_t$.

The *width of a tree decomposition* is one less than the size of the largest bag in the tree decomposition. The *treewidth* of a graph G, denoted by $\mathrm{tw}(G)$, is the minimal integer k such that G has a tree decomposition of width k. For instance, the graphs of treewidth 1 are forests and the graphs of treewidth 2 are series-parallel graphs [3]. See [4] for more about treewidth.

Proposition 40. *Every graph with a tree decomposition of width at most t has a $\frac{1}{2}$-separator that is a bag (of size at most $t+1$).*

Proof. Let G be a graph of order n and $(T, (B_t)_{t \in V(T)})$ be a tree decomposition of G of width t. Let $uv \in E(T)$. Let T_u (resp. T_v) be the subtree of $T - v$ (resp. $T - u$) containing u (resp. v). We orient the edge u to v if and only if the graph induced by the vertices of the bags in T_u is of higher order than the one by T_v. We do so for every edge of T. Since T is acyclic, there always exists a sink s, a vertex where all edges are oriented towards it, in T. We claim that the bag of s is a $\frac{1}{2}$-separator of G. Indeed, a connected component of $G - B_s$ is a subgraph of the graph induced by the vertices of the bags of a connected component of $T - s$. Since s is a sink, by construction, every connected component of $T - s$ corresponds to an induced subgraph of G of order at most $\frac{n}{2}$. This concludes the proof. □

By Proposition 40, for every graph G or order n, $\mathrm{td}(G) \leqslant 1 + (\mathrm{tw}(G)+1)\log_2(n)$. One can try a similar approach to bound elimination depths, but the issue is that we have no structural constraints on the graphs

induced by the bags of a tree decomposition. Since we are interested in deleting connected graphs, the worst-case scenario is for a bag to induce an independent set. Therefore, we cannot find better bounds with just the treewidth. However, consider chordal graphs. A graph is *chordal* if and only if its minimal separators are cliques. Such a graph can have very large treewidth compared to its number of vertices but if it has a tree decomposition where every bag is a clique. Therefore, the treewidth itself does not provide any good bound on the elimination depth of a chordal graph by deleting cliques, but the structure of the graphs induced by the bags of the tree decomposition gives us a $1 + \log_2(n)$ upper bound. This motivates the idea of considering tree decompositions with structural constraints on the graphs induced by the bags. For instance, the *connected treewidth* of a graph G (introduced in [8]), denoted by $\mathrm{ctw}(G)$, is the smallest integer k such that there exists a *connected tree decomposition* of G of width k, a tree decomposition with bags that induce connected graphs in G. A strategy to eliminate a graph G can then be used to eliminate a bag of a connected tree decomposition that is a $\frac{1}{2}$-separator. Unlike the case of a block, we have no hypothesis on the interactions between this bag and the rest of the graph. Therefore, to compute the cost of eliminating a bag, we cannot consider its elimination depth but rather the following parameter.

The *deletion depth* of a graph G by deleting graphs in $\mathcal{D}$, denoted by $\mathrm{dd}^{\mathcal{D}}(G)$, is the minimal integer k such that there exists a $\mathcal{D}$-elimination path of G on k vertices.

Proposition 41. *Let $\mathcal{D}$ be a set of connected graphs that contains K_1 and let $\mathcal{C}$ be a hereditary set of graphs. For every graph $G \in \mathcal{C}$, $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 1 + c\log_2(n)$ where $c = \max\{\mathrm{dd}^{\mathcal{D}}(G) \colon H \in \mathcal{C}$ and $|V(H)| = \mathrm{ctw}(G) + 1\}$.*

In general, ctw can be arbitrarily large compared to tw. For example, $\mathrm{tw}(C_n) = 2$ whereas $\mathrm{ctw}(C_n) = \Theta(n)$. However, we can hope for c in the previous proposition to be smaller than $\mathrm{tw}(G)$, which would lead to a better bound as in the case of deleting cliques in chordal graphs. More generally, one can also investigate other variants of tree decomposition with required structures depending on the graphs that we can delete.

5 About monotonicity of elimination depth

Previously, we observed that elimination depth is not monotone under taking induced subgraph. In this subsection, we will characterize the classes of graphs $\mathcal{D}$ for which $\mathrm{ed}^{\mathcal{D}}$ is monotone. We say that $\mathcal{D}$ is *connected-hereditary* if and only if it is closed under taking connected induced subgraphs.

Theorem 42. *Let $\mathcal{D}$ be a set of connected graphs containing K_1. The elimination depths when deleting graphs in $\mathcal{D}$ is monotone under taking induced subgraph if and only if $\mathcal{D}$ is either the set of all connected graphs, the set of all complete graphs, or the set of all complete graphs of order at most n for some integer $n \geqslant 1$.*

Proof. If $\mathcal{D}$ is the set of all connected graphs, then $\mathrm{ed}^{\mathcal{D}}$ is monotone under taking induced subgraph because $\mathrm{ed}^{\mathcal{D}}(G) = 1$ for every nonnull graph G, and $\mathrm{ed}^{\mathcal{D}} = 0$ otherwise. Therefore, we can suppose that $\mathcal{D}$ is not the set of all connected graphs. First, let $\mathcal{D}$ be either the set of all complete graphs, or the set of all complete graphs of order at most n for some integer $n \geqslant 1$. We will prove that $\mathrm{ed}^{\mathcal{D}}$ is monotone under taking induced subgraph. It is sufficient to prove that for all graph G and for all vertex $v \in V(G)$, $\mathrm{ed}^{\mathcal{D}}(G) \geqslant \mathrm{ed}^{\mathcal{D}}(G - v)$. Notice that for every graph $D \in \mathcal{D}$, and for every vertex $v \in V(D)$, $D - v$ is either the null graph or is a graph of $\mathcal{D}$. Let $(F, (B_f)_{f \in V(F)})$ be a $\mathcal{D}$-elimination forest for G of depth d. We can construct a $\mathcal{D}$-elimination forest for $G - v$ from $(F, (B_f)_{f \in V(F)})$ by deleting the occurrence of v in the bags of $(F, (B_f)_{f \in V(F)})$, then deleting the empty bag B containing v (if it is empty), and make the parent of B the parent of the children of B (if they exist). This $\mathcal{D}$-elimination forest of $G - v$ has depth not greater than d therefore $\mathrm{ed}^{\mathcal{D}}(G) \geqslant \mathrm{ed}^{\mathcal{D}}(G - v)$.

Secondly, we will prove that with any other set of connected graph $\mathcal{D}$ containing K_1, $\mathrm{ed}^{\mathcal{D}}$ is not monotone under taking induced subgraph. Let $\mathcal{D}$ be a set of connected graphs that contains K_1 that is not the set of all connected graphs, the set of all complete graphs, or the set of all complete graphs of order at most n for some integer $n \geqslant 1$. If $\mathcal{D}$ is not connected-hereditary, then there exists $G \in \mathcal{D}$ and $v \in V(G)$ such that $G - v$ is connected and $G - v \notin \mathcal{D}$. Therefore, $\mathrm{ed}^{\mathcal{D}}(G) = 1$ and $\mathrm{ed}^{\mathcal{D}}(G - v) > 1$ so $\mathrm{ed}^{\mathcal{D}}$ is not monotone under taking induced subgraph. Suppose from now on that $\mathcal{D}$ is connected-hereditary.

Moreover, since $\mathcal{D}$ contains a graph D that is not complete and is connected-hereditary, $\mathcal{D}$ must contain a P_j with $j \geqslant 3$ which is formed by two non-adjacent vertices in D together with a shortest path between them. We distinguish two cases.

Suppose that paths in $\mathcal{D}$ are on at most k vertices. Consider the following construction of the graph G. Start from a cycle C_{4k+4} and add a path P on k vertices between two vertices that are diametrically opposite (see Figure 12 for an example). The resulting graph G verifies $\mathrm{ed}^{\mathcal{D}}(G) \leqslant 3$ by first deleting P, then by deleting the middle vertex of each remaining path, and finally by deleting the remaining connected components which are isomorphic to P_k. Now, consider C_{4k+4}. After any first deletion (at most a path on k vertices), the remaining graph is a path on at least $3k+4$ vertices. After another deletion, the remaining graph contains at least $2k+4$ vertices, more precisely, there are at most two connected components, one of which must be a path on at least $k+2$ vertices. This path cannot be eliminated in 1 round. Therefore, $\mathrm{ed}^{\mathcal{D}}(C_{4k+4}) > 3$ and $C_{4k+4} \subseteq_i G$, so $\mathrm{ed}^{\mathcal{D}}$ is not monotone.

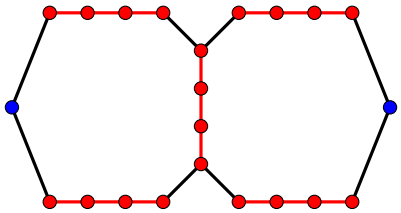


Figure 12: The construction with $k = 4$ with P_4 highlighted in red. The elimination ordering of depth 3 is first the central P_4, then the two blue vertices, then the four remaining P_4.

Otherwise, suppose that $\mathcal{D}$ contains arbitrarily long paths. We call a connected graph G $\mathcal{D}$-*apex* if and only if $G \notin \mathcal{D}$ and there exists $v \in V(G)$ such that $G - v \in \mathcal{D}$. Since $\mathcal{D}$ is not the class of all graphs, there exists a $\mathcal{D}$-apex. Let G be a $\mathcal{D}$-apex and let $v \in V(G)$ be such that $G - v \in \mathcal{D}$. Consider G_3 as described in the proof of Theorem 24, with G as the building block, v as the special vertex, and $\bullet_2$ as the dumbbell operator. Let $v_0, \ldots, v_c$ be the vertices of G_3 that correspond to v. Construct the graph G_3' from a copy of G_3 by adding c new vertices $u_1, \ldots, u_c$ and then add the edges $v_{i-1}u_i$ and u_iv_i for all integer i such that $1 \leqslant i \leqslant c$ (see Figure 13 for an example). Observe that $G_3 \subseteq_i G_3'$ by removing the vertices $u_1, \ldots, u_c$. Let $\mathcal{C}$ be the class of all connected graphs and observe that $\mathcal{D}$ respects the interior property over $\bullet_1$ since it is connected-hereditary. As a result, by Theorem 24, $\mathrm{ed}^{\mathcal{D}}(G_3) \geqslant 3$. Since $\mathcal{D}$ contains arbitrarily long paths, we can delete the path $v_0u_1v_1 \ldots u_cv_c$ in 1 round. The remaining connected components are all isomorphic to $G - v$ so they can be deleted in 1 round. Therefore, $\mathrm{ed}^{\mathcal{D}}(G_3') \leqslant 2$, so $\mathrm{ed}^{\mathcal{D}}$ is not monotone under taking induced subgraph. □

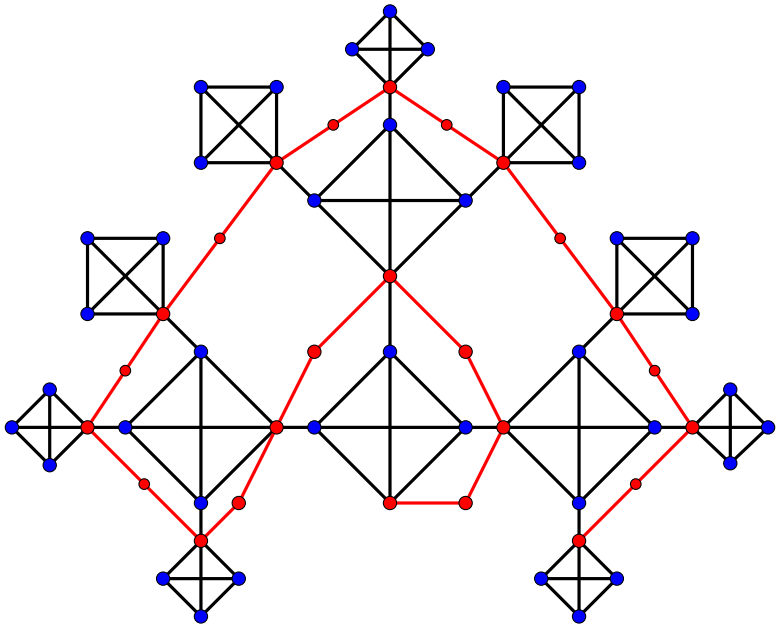


Figure 13: The construction with $G = K_4$. The order of elimination is the red path and then the remaining triangles.

A Example of classes of graphs that respect the interior property or the closure property

In this section, we will see examples of classes of graphs that respect the hypothesis of Theorem 24. So we will prove that some classes of graphs respect the interior property over $\bullet_1$, or that some other classes of graphs respect the closure property under $\bullet_p$ for some $p \geqslant 1$.

Connected-hereditary classes of Graphs A class of connected graphs $\mathcal{D}$ is closed under taking connected induced subgraph if and only if for all connected graph G and H, if $G \in \mathcal{D}$ and $H \subseteq_i G$ then $H \in \mathcal{D}$. We say that $\mathcal{D}$ is a connected-hereditary class of graphs. Every connected-hereditary classes of graphs respects the interior property over $\bullet_1$. It is the case of graphs of girth $\geqslant k+1$ (considering the girth of trees as $+\infty$), graphs of treewidth $\leqslant k$, graphs of degeneracy $\leqslant k$, graphs of chromatic number $\leqslant k$, trees, graphs that are H-minor free, graphs of maximum average degree $\leqslant r$. Every graphs classes that contain no graph with a cut-vertex also respect the interior property over $\bullet_1$. It is the case for cliques and for cycles. All definitions of previous classes are in [7]. For these reasons, we can consider the interior property over $\bullet_1$ as a generalization of connected-hereditary.

Eulerian graphs A graph is *eulerian* if and only if all its vertices have even degree. This means that for every eulerian graph E, if $E \cong (G,u) \bullet_1 (H,v)$ then both G and H must be eulerian. Indeed, by the handshaking lemma, the sum of the degrees of vertices in G and in H must be even. So u must have even degree in G and v must have even degree in H because by hypothesis, every vertex of $V(G) \setminus \{u\}$ has even degree in G and every vertex of $V(H) \setminus \{v\}$ has even degree in H and. Therefore, both G and H are eulerian.

Degeneracy A graph G of order n is said to have *degeneracy* k if there exists an ordering of its vertices $v_1, \dots v_n$ such that for all $1 \leqslant i \leqslant n$, $|\{v_j : j < i \text{ and } v_j v_i \in E(G)\}| \leqslant k$. Let G and H be two graphs of degeneracy less than k. Let $u_1, \dots, u_{|V(G)|}$ and $v_1, \dots, v_{|V(H)|}$ be an ordering of the vertices of G and H of degeneracy less that k. Let r be the new vertex of $(G,u_1) \bullet_3 (H,v_1)$. If $k \geqslant 2$, then $(G,u_1) \bullet_3 (H,v_1)$ is of degeneracy less than k with the following ordering : r, $u_1, \dots, u_{|V(G)|}$, $v_1, \dots, v_{|V(H)|}$. Notice that there are examples for which $\bullet_1$ and $\bullet_2$ are not sufficient.

Maximum average degree The average degree of a graph G of order n, denoted by $\mathrm{ad}(G) = \sum_{v \in V(G)} \deg(v)/n = 2|E(G)|/|V(G)|$. The maximum average degree of a graph G, denoted $\mathrm{mad}(G) = \max\{\mathrm{ad}(H) : H \subseteq_i G\}$.

Lemma 43. *Let $r > 2$. Let G and H be two connected graphs of average degree less than r. Then for all $u \in V(G)$, and for all $v \in V(H)$, the average degree of $(G,u) \bullet_{2+\lceil 2/(r-2)\rceil} (H,v)$ is less that r.*

Proof. Let $n = |V(G)|$, $n' = |V(H)|$, $d = \sum_{v \in V(G)} \deg(v)$, $d' = \sum_{v \in V(H)} \deg(v)$. So $d/n \leqslant r$ and d'/n'. In $(G,u) \bullet_{2+\lceil 2/(r-2)\rceil} (H,v)$, both u and v have their degree that have been increased by 1, and there are $\lceil 2/(r-2)\rceil$ new vertices of degree 2. Therefore, the density of the new graph is

$$\begin{aligned}
\frac{d+d'+2+2\lceil 2/(r-2)\rceil}{n+n'+\lceil 2/(r-2)\rceil} &\leqslant \frac{nr+n'r+2+2\lceil 2/(r-2)\rceil}{n+n'+\lceil 2/(r-2)\rceil} \\
&= r\frac{n+n'+(2+2\lceil 2/(r-2)\rceil)/r}{n+n'+\lceil 2/(r-2)\rceil} \\
&\leqslant r \text{ because } (2+2\lceil 2/(r-2)\rceil)/r \leqslant \lceil 2/(r-2)\rceil.
\end{aligned}$$

Indeed, $r > 2$ so $2/r < 1$. Therefore,

$$\begin{aligned}\frac{2+2\lceil 2/(r-2)\rceil}{r} - \lceil 2/(r-2)\rceil &= \frac{2}{r} + \lceil 2/(r-2)\rceil(\frac{2}{r}-1)\\ &< \frac{2}{r} + (2/(r-2)+1)(\frac{2}{r}-1)\\ &= \frac{2}{r} + \frac{r}{r-2}\frac{-r+2}{r}\\ &= \frac{2}{r} - 1 < 0\end{aligned}$$

□

Proposition 44. *Let $r > 2$. The class of graphs of maximum average degree at most r is closed under $\bullet_{2+\lceil 2/(r-2)\rceil}$.*

Proof. For the sake of contradiction, suppose that there are two graphs G and H of maximum average degree less than r and such that there are two vertices $u \in V(G)$ and $v \in V(H)$ such that $\text{mad}((G,u) \bullet_{2+2/(r-2)} (H,v)) > r$. Let D be a densest subgraph of $(G,u) \bullet_{2+2/(r-2)} (H,v)$. So the average degree of D is strictly bigger than r. We can suppose that D is connected because otherwise, we can only consider a connected component of D. $\mathcal{D}$ cannot be a subgraph of G nor of H because their maximum average degree is less than r. Therefore, there exists $G' \subseteq G$, $H' \subseteq H$ such that $D \cong (G',u) \bullet_{2+2/(r-2)} (H',v)$. The average degree of G' is less than r because G' is a subgraph of G. The same goes for H'. Therefore, by Lemma 43, the average degree of $(G',u) \bullet_{2+2/(r-2)} (H',v)$ is less than r. But D being a densest subgraph, its average degree is equal to its maximum average degree. So $\text{mad}(D) \leqslant r$ which is absurd. □

B A planar graph that cannot be separated by a tree

As mentioned in section 4.4, there exist planar graphs with no separator that induces a tree. An example of such planar graphs are the Apollonian networks defined as follows:

- $A_1 \cong K_3$
- A_{k+1} is defined from A_k by adding a vertex in each of the faces of A_k except for the infinite face, and then adding edges between each of the newly created vertices and the vertices that form the border of their face.

The construction of A_k comes with a natural embedding. We can consider that $V(A_k) \subseteq V(A_{k+1})$ and we can then name the vertices of A_1 as a_1, a_2, a_3 and the new vertex in A_2 as c. For all $k \geqslant 2$ and for all $i, j \in [3]$ with $i \neq j$, the cycle $\{a_i, a_j, c\}$ as an interior $I_{i,j}$. One can observe that $\{a_i, a_j, c\} \cup I_{i,j}$ induces a subgraph in A_k that is isomorphic to A_{k-1}. We denote these subgraphs by $A_k^{i,j}$ and we abuse notation by saying that $A_k^{i,j} = A_k^{j,i}$ (see Figure 14 for an example).

Proposition 45. *For all $k \geqslant 1$, there is no $W \subseteq V(A_k)$ such that $A_k[W]$ is a tree and $A_k - H$ is disconnected.*

Proof. We will proceed by induction on k. It is clear for A_1. Let $k \geqslant 1$. For the sake of contradiction, let $W \subseteq V(A_{k+1})$ be such $A_{k+1}[W]$ is a tree and $A_{k+1} - W$ is disconnected. Let $u \in V(A_{k+1}^{i,j})$ and $v \in V(A_{k+1}^{i,j'})$ for some $i, j, j' \in [3]$ such that $u, v \notin W$. It suffices to prove that u and v are still connected to reach a contradiction. Four cases :

- If $|\{a_1, a_2, a_3\} \cap W| = 0$: If u and a_i are disconnected, then $V(A_{k+1}^{i,j}) \cap W$ disconnects $A_{k+1}^{i,j}$ which is absurd by induction hypothesis. The same holds for v and a_i. Therefore, there is a path from u to a_i and from a_i to v so u and v are still connected.
- If $|\{a_1, a_2, a_3\} \cap W| = 1$: If $a_i \notin W$, then u and v are connected as in the previous case. Otherwise, both a_j and $a_{j'}$ are not in W. If u and a_j are disconnected, then $V(A_{k+1}^{i,j}) \cap W$ disconnects $A_{k+1}^{i,j}$ which is absurd by induction hypothesis. The same holds for v and $a_{j'}$, moreover, a_j and $a_{j'}$ are neighbors. Therefore, there is a path from u to a_j, from a_j to $a_{j'}$, and from $a_{j'}$ to v. So u and v are still connected.
- If $|\{a_1, a_2, a_3\} \cap W| = 2$: c cannot be in W otherwise, a cycle is formed. If u and c are disconnected, then $V(A_{k+1}^{i,j}) \cap W$ disconnects $A_{k+1}^{i,j}$ which is absurd by induction hypothesis. The same holds for v and c. Therefore, there is a path from u to c and from c to v. So u and v are still connected.

- If $|\{a_1, a_2, a_3\} \cap W| = 3$: Then $\{a_1, a_2, a_3\}$ form a cycle, which is absurd.

□

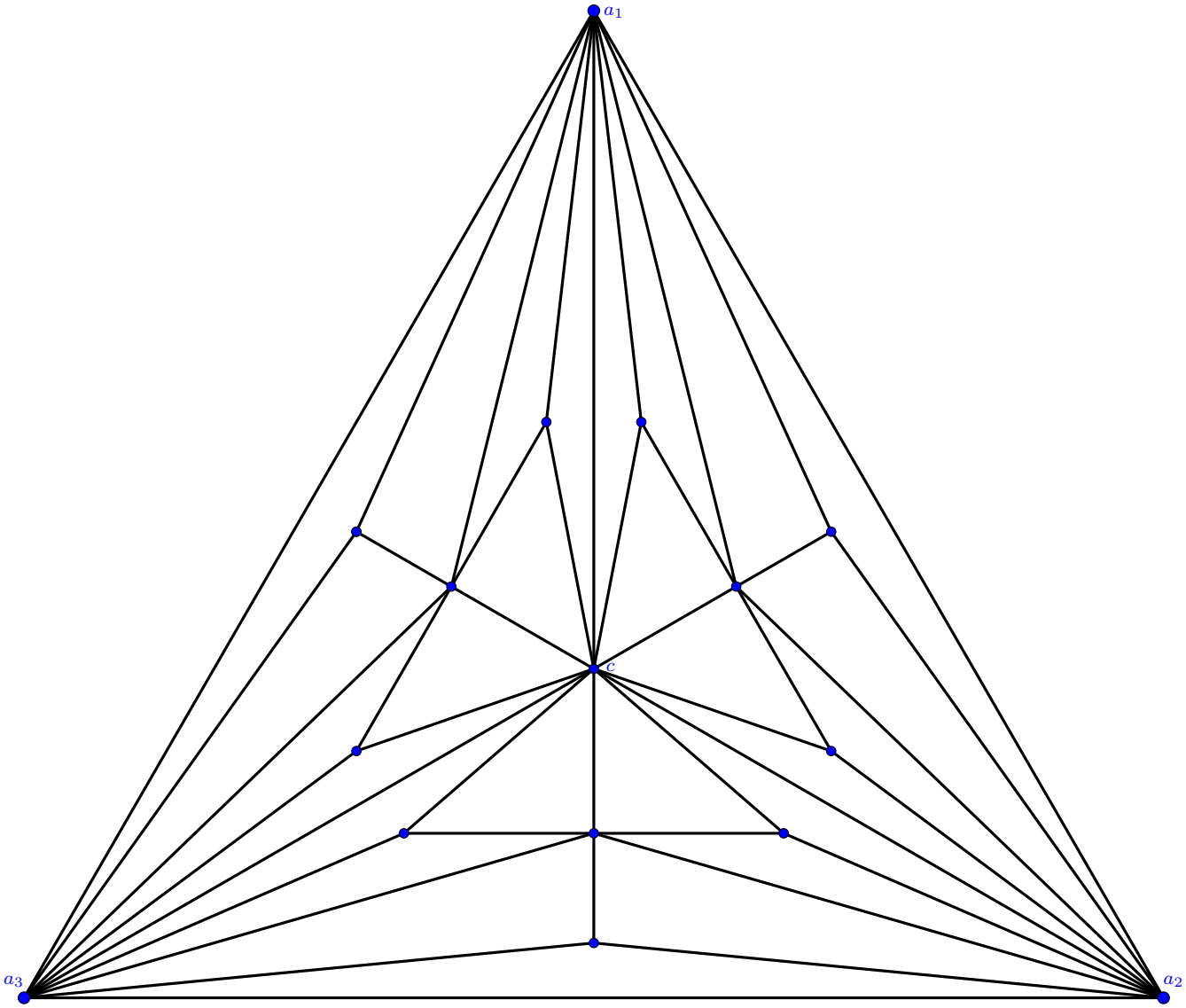


Figure 14: The third Apollonian network A_3.

References

[1] Jochen Alber, Henning Fernau, and Rolf Niedermeier. Graph separators: a parameterized view. *Journal of Computer and System Sciences*, 67(4):808–832, 2003. Parameterized Computation and Complexity 2003.

[2] Michael O Albertson, Robert E Jamison, Stephen T Hedetniemi, and Stephen C Locke. The subchromatic number of a graph. In *Annals of Discrete Mathematics*, volume 39, pages 33–49. Elsevier, 1989.

[3] Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1):1–45, 1998.

[4] Hans L Bodlaender. Discovering treewidth. In *International Conference on Current Trends in Theory and Practice of Computer Science*, pages 1–16. Springer, 2005.

[5] Jannis Bulian and Anuj Dawar. Graph isomorphism parameterized by elimination distance to bounded degree. *Algorithmica*, 75(2):363–382, 2016.

[6] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

[7] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[8] Reinhard Diestel and Malte Müller. Connected tree-width. *Combinatorica*, 38:381–398, 2018.

[9] Zdeněk Dvořák, Archontia C. Giannopoulou, and Dimitrios M. Thilikos. Forbidden graphs for tree-depth. *European Journal of Combinatorics*, 33(5):969–979, 2012. EuroComb '09.

[10] Guillaume Fertin, Emmanuel Godard, and André Raspaud. Minimum feedback vertex set and acyclic coloring. *Information Processing Letters*, 84(3):131–139, 2002.

[11] Fedor V Fomin, Archontia C Giannopoulou, and Michał Pilipczuk. Computing tree-depth faster than 2^n. *Algorithmica*, 73(1):202–216, 2015.

[12] Archontia C. Giannopoulou, Paul Hunter, and Dimitrios M. Thilikos. Lifo-search: A min–max theorem and a searching game for cycle-rank and tree-depth. *Discrete Applied Mathematics*, 160(15):2089–2097, 2012.

[13] Falko Hegerfeld and Stefan Kratsch. Solving connectivity problems parameterized by treedepth in single-exponential time and polynomial space. *arXiv preprint arXiv:2001.05364*, 2020.

[14] Yoichi Iwata, Tomoaki Ogasawara, and Naoto Ohsaka. On the power of tree-depth for fully polynomial fpt algorithms. *arXiv preprint arXiv:1710.04376*, 2017.

[15] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 2010.

[16] Kolja Knauer, Hoang La, and Petru Valicov. Feedback vertex sets in (directed) graphs of bounded degeneracy or treewidth, 2022.

[17] Casimir Kuratowski. Sur le problème des courbes gauches en topologie. *Fundamenta Mathematicae*, 15(1):271–283, 1930.

[18] Alexander Lindermayr, Sebastian Siebertz, and Alexandre Vigny. Elimination distance to bounded degree on planar graphs. *Fundamenta Informaticae*, 191, 2024.

[19] Richard J Lipton and Robert Endre Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36(2):177–189, 1979.

[20] László Lovász. Graph minor theory. *Bulletin of the American Mathematical Society*, 43(1):75–86, 2006.

[21] Flaminia L. Luccio. Almost exact minimum feedback vertex set in meshes and butterflies. *Information Processing Letters*, 66(2):59–64, 1998.

[22] Karl Menger. Zur allgemeinen kurventheorie. *Fundamenta Mathematicae*, 1927.

[23] Gary L. Miller. Finding small simple cycle separators for 2-connected planar graphs. *Journal of Computer and System Sciences*, 32(3):265–279, 1986.

[24] Wojciech Nadara, Michał Pilipczuk, and Marcin Smulewicz. Computing treedepth in polynomial space and linear fpt time, 2022.

[25] Wojciech Nadara and Marcin Smulewicz. Decreasing the maximum average degree by deleting an independent set or a d-degenerate subgraph, 2020.

[26] Jaroslav Nešetřil and Patrice Ossona De Mendez. *Sparsity: graphs, structures, and algorithms*, volume 28. Springer Science & Business Media, 2012.

[27] Jaroslav Nešetřil and Patrice Ossona de Mendez. Tree-depth, subgraph coloring and homomorphism bounds. *European Journal of Combinatorics*, 27(6):1022–1041, 2006.

[28] Jaroslav Nešetřil and Patrice Ossona de Mendez. On low tree-depth decompositions. *Graphs and Combinatorics*, 31(6):1941–1963, April 2015.

[29] Victor Neumann-Lara. The dichromatic number of a digraph. *Journal of Combinatorial Theory, Series B*, 33(3):265–270, 1982.

[30] Alex Pothen. The complexity of optimal elimination trees. *Technical Report*, 1988.

[31] Felix Reidl, Peter Rossmanith, Fernando Sanchez Villaamil, and Somnath Sikdar. A faster parameterized algorithm for treedepth, 2014.

[32] Neil Robertson and Paul D. Seymour. Graph minors. i. excluding a forest. *Journal of Combinatorial Theory, Series B*, 35(1):39–61, 1983.

[33] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of tree-width. *Journal of algorithms*, 7(3):309–322, 1986.

[34] Neil Robertson and Paul D. Seymour. Graph minors. v. excluding a planar graph. *Journal of Combinatorial Theory, Series B*, 41(1):92–114, 1986.

[35] Abraham Siberschatz and Peter B. Galvin. *Operating System Concepts, 4th Ed.* Addison-Wesley Longman Publishing Co., Inc., USA, 4th edition, 1993.

[36] Jeffrey Travers and Stanley Milgram. An experimental study of the small world problem. In Samuel Leinhardt, editor, *Social Networks*, pages 179–197. Academic Press, 1977.