Edge Degeneracy: Algorithmic and Structural Results

Stratis Limnios[*†] Christophe Paul[‡†§] Joanny Perret[¶] Dimitrios M. Thilikos[‡†§‖]

Abstract

We consider a cops and robber game where the cops are blocking edges of a graph, while the robber occupies its vertices. At each round of the game, the cops choose some set of edges to block and right after the robber is obliged to move to another vertex traversing at most s unblocked edges (s can be seen as the speed of the robber). Both parts have complete knowledge of the opponent's moves and the cops win when they occupy all edges incident to the robbers position. We introduce the capture cost on G against a robber of speed s. This defines a hierarchy of invariants, namely $\delta_{\mathrm{e}}^{1}, \delta_{\mathrm{e}}^{2}, \ldots, \delta_{\mathrm{e}}^{\infty}$, where $\delta_{\mathrm{e}}^{\infty}$ is an edge-analogue of the admissibility graph invariant, namely the *edge-admissibility* of a graph. We prove that the problem asking wether $\delta_{\mathrm{e}}^{s}(G) \leq k$, is polynomially solvable when $s \in \{1, 2, \infty\}$ while, otherwise, it is NP-complete. Our main result is a structural theorem for graphs of bounded edge-admissibility. We prove that every graph of edge-admissibility at most k can be constructed using $(\leq k)$-edge-sums, starting from graphs whose all vertices, except possibly from one, have degree at most k. Our structural result is approximately tight in the sense that graphs generated by this construction always have edge-admissibility at most $2k-1$. Our proofs are based on a precise structural characterization of the graphs that do not contain θ_r as an immersion, where θ_r is the graph on two vertices and r parallel edges.

Keywords: Graph Admissibility, Graph degeneracy, Graph Searching, Cops and robber games, Graph decomposition theorems.

1 Introduction

All graphs in this paper are undirected, finite, loopless, and may have parallel edges. We denote by $V(G)$ the set of vertices of a graph G, while we denote by $E(G)$ the multi-set of its edges. We also use the term *s-path of G* for a path of G that has length at most s.

A *(k,s)-hide out* in a graph G is a subset S of its vertices such that, for each vertex $v \in S$, it is not possible to block all s-paths from v to the rest of S by less than k vertices, different than v. The *s-degeneracy* of a graph G, has been introduced in [23] as the minimum k for which G contains a (k,s)-hide out. s-degeneracy defines a hierarchy of graph invariants that, when $s = 1$,

[*]Data Science and Mining (DaSciM) team, Laboratoire d'Informatique (LIX) École Polytechnique, Palaiseau, France.

[†]Supported by project ESIGMA (ANR-17-CE23-0010).

[‡]LIRMM, Univ Montpellier, CNRS, Montpellier, France.

[§]Supported by project DEMOGRAPH (ANR-16-CE40-0028).

[¶]GIPSA-lab, Université de Grenoble, Grenoble, France.

[‖]Supported by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.

gives the classic invariant of graph degeneracy [5, 18, 20] and, when $s = \infty$, gives the parameter of ∞-admissibility that was introduced by Dvořák in [10] and studied in [6, 9, 15, 17, 21, 22, 24].

In this paper we introduce and study the edge analogue of the above hierarchy of graph invariants, namely the *s-edge-degeneracy hierarchy*. The new parameter results from the one of s-degeneracy if we replace (k, s)-hide outs by (k, s)-*edge hide outs* where we ask that, for each vertex v of S, it is not possible to block all s-paths from v to the rest of S by less than k edges. It follows that the value of s-edge-degeneracy may vary considerably than the one of s-degeneracy. For instance, consider the graph θ_k consisting of two vertices and k parallel edges between them. It is easy to see that, for every positive integer s, the s-degeneracy of θ_k is 2, while it s-edge-degeneracy is k (the two vertices form a (k, s)-edge hideout). In other words, s-edge-degeneracy can be seen as an alternative way to extent the notion of degeneracy using edge separators instead of vertex separators.

In Subsection 3.1 we introduce two alternative definitions for s-edge-degeneracy, apart from the one using (k, s)-edge hide outs. The first is in terms of a graph searching game and the second is in terms of graph layouts. Next, we prove a min-max theorem supporting the equivalence of the three definitions. As a consequence of this theorem, we can identify the computational complexity of s-edge-degeneracy: it can be computed in polynomial time when $s \in \{1, 2, \infty\}$, while for all other values of s, desiding whether its value is at most k is an NP-complete problem.

Our next step is to provide a structural theorem for the ∞-edge-degeneracy that, from now on, we call ∞-edge-admissibility. For ∞-degeneracy (also known as ∞-admissibility), Dvořák proved the following structural characterization [9, Theorem 6].

Proposition 1.1. *For every k, there exist constants d_k, c_k and a_k such that every graph G with ∞-admissibility at most k can be constructed by applying $(\leq c_k)$-clique sums starting from graphs where at most d_k vertices have degree at least a_k.*

In the above proposition the $(\leq k)$-*clique sum* operation receives as input two graphs G_1 and G_2 such that each G_i contains a clique K_i with vertex set $\{v_1^i, \ldots, v_\rho^i\}, \rho \leq k$. The outcome of the operation is the graph occurring if we identify v_j^1 and v_j^2 for $j \in \{1, \ldots, \rho\}$ and then remove some of the edges between the identified vertices. While the constants of Proposition 1.1 where not specified in [9], an alternative proof was recently given by Weißauer in [24] where $d_k = k$, $c_k = k$, and $a_k = 2k(k-1)$.

In Section 4 we provide a counterpart of Proposition 1.1 for the ∞-*edge-admissibility* that is the following

Theorem 1.2. *For every k, every graph G with* ∞-edge-admissibility *at most k can be constructed by applying $(\leq k)$-edge sums starting from graphs where at most one vertex has degree at least $k+1$.*

Observe that Theorem 1.2 occurs from Proposition 1.1 if we replace ∞-*admissibility* by ∞-*edge-admissibility*, if, instead of clique sums, we consider edge sums, and if we set $d_k = 1$, $c_k = k$, and $a_k = k + 1$. The $(\leq k)$-edge sum operation (the definition is postponed to Subsection 4.1) was defined in [25] (see also [13]) and can be seen as the edge-counterpart of clique sums.

The proof of our structural theorem is derived by a *precise* structural characterization of the graphs where each pair of vertices is separated by a cut of size at most k. We prove that these graphs are *exactly* those that can be constructed using $(\leq k)$-edge sums from graphs where all but one of their vertices have degree at most k. This directly implies our structural theorem for ∞-*edge-admissibility*, as every pair of two vertices linked by $k+1$ pairwise edge-disjoint paths is a $(k+1,\infty)$-edge hide out.

Our last result is that the converse of the structural characterization in Theorem 1.2 holds in an approximate way: if G can be constructed using $(\leq k)$-edge sums from graphs where all but one of their vertices have degree at most k, then the ∞-edge-admissibility of G is at most $2k-1$. This suggests that our decomposition theorem is indeed the correct choice for the parameter of ∞-edge admissibility.

2 Basic definitions

Sets and integers. Given a non-negative integer s, we denote by $\mathbb{N}_{\geq s}$ the set of all non-negative integers that are not smaller than s. We also denote $\mathbb{N}^{+}_{\geq s} = \mathbb{N}_{\geq s} \cup \{\infty\}$. Given two integers $p \leq q$, we set $[p,q] = \{p, p+1, \ldots, q\}$ and given a $k \in \mathbb{N}_{\geq 0}$ we define $[k] = [1,k]$. Given a set A, we use 2^A for the set of all its subsets, we define $\binom{A}{2} := \{S \mid S \in 2^A \text{ and } |S| = 2\}$, and, given a $k \in \mathbb{N}_{\geq 0}$ we denote by $A^{(\leq k)}$ the set of all subsets of A that have size at most k. A *near-partition* of a set A is a collection of pairwise disjoint sets whose union is A. A *bipartition* of $A, |A| \geq 2$ is a near-partition of A into two non-empty sets.

Graphs. All graphs in this paper are undirected, finite, loopless, and may have parallel edges. We denote by $V(G)$ the set of vertices of a graph G while we use $E(G)$ for the multi-set of its edges. Given a graph G and a vertex v, we define $E_G(v)$ as the multi-set of all edges of G that are incident to v. We define the *neighborhood* of v as $N_G(v) = (\bigcup_{e \in E_G(v)} e) \setminus \{v\}$, the *edge-degree* of v as $\mathbf{deg}_G(v) = |E_G(v)|$. We also define $\Delta(G) = \max\{\mathbf{deg}_G(v) \mid v \in V(G)\}$. Given a $F \subseteq E(G)$, we define $G \setminus F = (V(G), E(G) \setminus F)$.

Given a tree T and two vertices $a, b \in V(T)$ we define aTb as the path of T connecting a and b. Let G be a graph and let $S_1, S_2 \subseteq V(G)$ where $S_1 \cap S_2 = \emptyset$. We define

$$E_G(S_1, S_2) = \{e \in E(G) \mid e \cap V_1 \neq \emptyset \text{ and } e \cap V_2 \neq \emptyset\}.$$

A *cut* of a graph G is any bipartition $(X, \overline{X})$ of its vertices. The *edges* of a cut $(X, \overline{X})$ is the set $E(X, \overline{X})$ while the *size* of $(X, \overline{X})$ is equal to $|E(X, \overline{X})|$. Given two distinct vertices x and y of G, an (x,y)-*cut* of G is a cut $(X, \overline{X})$ of G such that $x \in X$ and $y \in \overline{X}$.

We define the function $\rho : 2^{V(G)} \to \mathbb{N}$ such that $\rho(X) = |E_G(X, V(G) \setminus X)|$. It is easy to see that ρ is a submodular function, ie.,

$$\forall X, Y \in 2^{V(G)} \quad \rho(X \cap Y) + \rho(X \cup Y) \leq \rho(X) + \rho(Y). \tag{1}$$

Given a graph G and two distinct $x, y \in V(G)$, we call an (x,y)-s-*path* every s-path in G starting from x and finishing on y. We also use the term (x,y)-*path* as a shortcut for (x,y)-∞-path. We define the function $\mathbf{cut}_{G,s} : \binom{V(G)}{2} \to \mathbb{N}_{\geq 0}$ so that $\mathbf{cut}_{G,s}(x,y)$ is equal to the minimum size of a $F \subseteq E(G)$ such that $G \setminus F$ does not contain any (x,y)-s-path. The complexity of computing $\mathbf{cut}_{G,s}(x,y)$ is provided by the next proposition (see [3, 16, 19]).

Proposition 2.1. *If $s \in \{1, 2, \infty\}$, then the problem that, given a graph G, a $k \in \mathbb{N}$, and two distinct vertices a and b of G, asks whether $\mathbf{cut}_{G,s}(a,b) \leq k$ is polynomially solvable, while it is* NP*-complete if $s \in \mathbb{N}_{\geq 3}$.*

3 Graph searching and s-edge-degeneracy

3.1 A search game.

The study of graph searching parameters is an active field of graph theory. Several important graph parameters have their search-game analogues that provide useful insights on their combinatorial and algorithmic properties. (For related surveys, see [1, 2, 4, 11, 12].)

We introduce a graph searching game, where the opponents are a group of cops and a robber. In this game, the cops are blocking *edges* of the graph, while the robber resides on the *vertices*. The first move of the game is done by the robber, who chooses a vertex to occupy. Then, the game is played in rounds. In each round, first the cops block a set of edges and next the robber moves to another vertex via a path consisting of at most s unblocked edges. The robber *cannot stay put* and he/she is captured if, after the move of the cops, all the edges incident to his/her current location are blocked. Both cops and robbers have full knowledge of their opponent's current position and they take it into consideration before they make their next move. We next give the formal definition of the game.

The game is parameterized by the speed $s \in \mathbb{N}^{+}_{\geq 1}$ of the robber. A *search strategy on G* for the cops is a function $f : V(G) \to 2^{E(G)}$ that, given the current position $x \in V(G)$ of the robber in the end of a round, outputs the set $f(v)$ of the edges that should be blocked in the beginning of the next round. The *cost* of a cop strategy f is defined as $\mathbf{cost}(f) = \max\{|f(v)| \mid v \in V(G)\}$, i.e., the maximum number of edges that may be blocked by the robbers according to f.

An *escape strategy on G* for the robber is a pair $R = (v_{\text{start}}, g)$ where v_{start} is the vertex of robber's first move and $g : 2^{E(G)} \times V(G) \to V(G)$ is a function that, given the set F of blocked edges in the beginning of a round and the current position x of the robber, outputs the vertex $u = g(F, v)$ where the robber should move. Here the natural restriction for g is that there is an s-path from v to u in $G \setminus F$. Clearly, if F is the set of edges that are incident to v, then $g(F, v)$ should be equal to v and this expresses the situation where the robber is captured.

Let f and $R = (v_{\text{start}}, g)$ be strategies for the cop and the robber respectively. The *game scenario* generated by the pair (f, R) is the infinite sequence $v_0, F_1, v_1, F_2, v_2, \ldots,$ where $v_0 = v_{\text{start}}$ and for every $i \in \mathbb{N}_{\geq 1}$, $F_i = f(v_{i-1})$ and $v_i = g(F_i, v_{i-1})$. If $v_i = v_{i-1}$ for some $i \in \mathbb{N}_{\geq 1}$, then (f, R) is a *cop-winning* pair, otherwise it is a *robber-winning* pair.

The *capture cost against a robber of speed s* in a graph G, denoted by $\mathbf{cc}_s(G)$ is the minimum k for which there is a cop strategy f, of cost at most k, such that for every robber strategy R, (f, R) is a cop-winning pair.

3.2 A min-max theorem for s-edge-degeneracy

s-edge-degeneracy. Let G be a graph, $x \in V(G)$, $S \subseteq V(G) \setminus \{x\}$, and $s \in \mathbb{N}^{+}_{\geq 1}$. We say that a set $A \subseteq E(G)$ is an (s, x, S)-*edge-separator* if every s-path of G from x to some vertex in S, contains some edge from A. We define $\mathbf{supp}_{G,s}(x, S)$ to be the minimum size of an (s, x, S)-edge-separator in G.

Let G be a graph and let $L = \langle v_1, \ldots, v_r \rangle$ be a layout (i.e. linear ordering) of its vertices. Given an $i \in [r]$, we denote $L_{\leq i} = \langle v_1, \ldots, v_i \rangle$. Given an $s \in \mathbb{N}^{+}_{\geq 1}$, we define the *$s$-edge-support* of a vertex v_i in L as $\mathbf{supp}_{G,s}(v_i, L_{\leq i-1})$. The *$s$-edge-degeneracy* of L, is the maximum s-edge-support of a vertex in L. The *s-edge-degeneracy* of G, denoted by $\delta^{s}_{\mathrm{e}}(G)$ is the minimum s-edge-degeneracy over all layouts of G.

(k, s)**-edge-hide-outs.** Let $s \in \mathbb{N}^{+}_{\geq 1}$ and $k \in \mathbb{N}$. A (k, s)-*edge-hide-out* in a graph G is any set $R \subseteq V(G)$ such that, for every $x \in R$, $\mathbf{supp}_{G,s}(x, R\setminus\{x\}) \geq k$. A (k, s)-*edge-hide-out* S is *maximal* there is no other (k, s)-*edge-hide-out* S' with $S \subsetneq S'$. It is easy to verify that every graph contains a unique maximal (k, s)-edge-hide-out.

(k, s)-edge-hide-outs can be seen as obstructions to small s-edge-degeneracy. In particular we prove the following min-max theorem, characterizing the search game that we defined in Subsection 3.1.

Theorem 3.1. *Let G be a graph and let $s \in \mathbb{N}^{+}_{\geq 1}$ and $k \in \mathbb{N}$. The following three statements are equivalent.*

(1) $\mathbf{cc}_s(G) \leq k$, i.e., there is a cop strategy f on G of cost less than k, such that for every robber strategy R on G, (f, R) is cop-winning.

(2) G has no $(k + 1, s)$-edge-hide-out.

(3) $\delta^s_{\mathrm{e}}(G) \leq k$.

Proof. *(1)* $\Rightarrow$ *(2)*. We prove that the negation of *(2)* implies the negation of *(1)*. Suppose that S is a $(k + 1, s)$-edge-hide-out of G. We use S in order to build an escape strategy $R = (v_{\text{start}}, g)$ on G as follows: Let v_{start} be any vertex in S. Let now $v \in S$ and $F \in 2^{E(G)}$. If $|F| > k$, then $g(v, F) = v$. We next define $g(v, F)$ for every $F \in E(G)^{\leq k}$. As S is a $(k + 1, s)$-edge-hide-out of G, we know that $\mathbf{supp}_{G,s}(v, S \setminus \{v\}) \geq k + 1$, therefore there is an s-path from v to some vertex $u \in S \setminus \{v\}$ that avoids all edges in F. We define $g(v, F) = u$. Notice now that if f is a cop strategy on G of cost at most k, and $v_0, F_1, v_1, F_2, v_2, \ldots,$ is the game scenario generated by the pair (f, R), then $v_{i-1} \neq v_i$ for every $i \in \mathbb{N}_{\geq 1}$. This means that R is a robber-winning strategy against any cop strategy of cost at most k, therefore $\mathbf{cc}_s(G) \geq k + 1$.

(2) $\Rightarrow$ *(3)*. Let $n = |V(G)|$. As G has no $(k+1, s)$-edge-hide-out, it follows that for every $R \subseteq V(G)$ there is a vertex $v \in R$, such that $\mathbf{supp}_{G,s}(v, R\setminus\{v\}) \leq k$. We pick such a vertex for every $R \subseteq V(G)$ and we denote it by $v(R)$. We now set $V_n = V(G)$, $v_n = v(V_n)$, and for $i \in \langle n - 1, \ldots, 1\rangle$ we set $V_i = V_{i+1} \setminus \{v_{i+1}\}$, $v_i = v(V_i)$. We now set $L = \langle v_1, \ldots, v_n\rangle$ and observe that for every $i \in [n]$, $\mathbf{supp}_{G,s}(v_i, L_{\leq i-1}) = \mathbf{supp}_{G,s}(v_i, V_{i-1}) \leq k$. Therefore, the s-edge-degeneracy of L is at most k, hence $\delta^s_{\mathrm{e}}(G) \leq k$.

(3) $\Rightarrow$ *(1)*. Suppose now that $L = \langle v_1, \ldots, v_n\rangle$ is a layout of $V(G)$ such that, for every $i \in [n]$, $\mathbf{supp}_{G,s}(v_i, L_{\leq i-1}) \leq k$. We use L to build a cop strategy $f : V(G) \to 2^{E(G)}$ as follows. Let $i \in [n]$ and let F_i be an $(s, v_i, L_{\leq i-1})$-edge-separator of G. We define f by setting $f(v_i) = F_i$. This means that if at some point the robber occupies vertex v_i, then there is no s-path in $G \setminus F_i$ from v_i to $L_{\leq i-1}$. As a consequence of this, no matter what the robber strategy $R = (v_{\text{start}}, g)$ is, it should hold that $g(v_i, F_i) \in L_{\geq i}$. Therefore if $x_0, F_1, x_1, F_2, x_2, \ldots,$ is the game scenario generated by the pair (f, R), then $x_i = x_{i-1}$ for some $i < n$. $\square$

3.3 The complexity of s-edge-degeneracy, for distinct values of s

We now combine Proposition 2.1 with the min-max theorem of the previous subsection in order to identify the computational complexity of δ^s_{e} for different values of s. Our main result is the following.

Theorem 3.2. *If $s \in \{1, 2, \infty\}$, then the problem that, given a graph G and a $k \in \mathbb{N}$, asks whether $\delta_{\mathrm{e}}^{s}(G) \leq k$, is polynomially solvable, while it is* NP*-complete if $s \in \mathbb{N}_{\geq 3}$.*

Proof. Notice first that checking whether $\delta_{\mathrm{e}}^{s}(G) \leq k$ can be done by the algorithm **check s-edge degeneracy** in Figure 1. Indeed, if the maximal $(k+1, s)$-edge-hideout S is non-empty then the above algorithm will report that $\delta_{\mathrm{e}}^{s}(G) > k$ after visiting, in line 3, every vertex not in S, as, by the maximality of S, for every $S' \supsetneq S$ there is a vertex $x \in S' \setminus S$ where $\mathbf{supp}_{G,s}(x, S' \setminus \{x\}) \leq k$. On the other hand, if S is empty, then the procedure will produce a layout $L = \langle v_1, \ldots, v_n \rangle$ with s-edge-degeneracy at most k.

Algorithm check s-edge degeneracy

Input: a graph G and an integer $k \in \mathbb{N}_{\geq 0}$.
Output: a report on whether $\delta_{\mathrm{e}}^{s}(G) \leq k$.

1. $n \leftarrow |V(G)|$, $S \leftarrow V(G)$.
2. for $i = n, \ldots, 1$,
3. if there is an $x \in S$ with $\mathbf{supp}_{G,s}(x, S \setminus \{x\}) \leq k$ then $v_i \leftarrow x$,
   else report that "$\delta_{\mathrm{e}}^{s}(G) > k$" and **stop**
   // S is the maximal $(k+1, s)$-edge-hideout of G,
   witnessing that $\delta_{\mathrm{e}}^{s}(G) > k$, because of Theorem 3.1.//
4. $S \leftarrow S - v_i$.
5. Output "$\delta_{\mathrm{e}}^{s}(G) \leq k$, witnessed by layout $L = \langle v_1, \ldots, v_n \rangle$."

Figure 1: An algorithm checking whether $\delta_{\mathrm{e}}^{s}(G) \leq k$.

Clearly, **check s-edge degeneracy** runs in polynomial time if checking whether $\mathbf{supp}_{G,s}(x, S \setminus \{x\}) \leq k$ can be done in polynomial time, which is equivalent to checking whether $\mathbf{cut}_{G',s}(x, x') \leq k$ where G' is the graph obtained by G after we identify all vertices of $S \setminus \{x\}$ to a single vertex x'. As this is possible for $s \in \{1, 2, \infty\}$, due to Proposition 2.1, the polynomial part of the theorem follow.

It now remains to prove that checking whether $\delta_{\mathrm{e}}^{s}(G) \leq k$ is an NP-hard problem when $s \in \mathbb{N}_{\geq 3}$. For this we will reduce the problem of checking whether $\mathbf{cut}_{G,s}(a, b) \leq k$ to the problem of checking whether $\delta_{\mathrm{e}}^{s}(G) \leq k$ and the result will follow from the hardness part of Proposition 2.1.

Let $\mathbf{T}_s = (G, a, b, k)$ be a quadruple where G is a graph on n vertices, $k \in \mathbb{N}_{\geq 0}$, and a, b two distinct vertices of G. We construct the graph $G_{\mathbf{T}_s}$ as follows: Take $k+n+1$ copies $G_1, \ldots, G_{k+n+1}$ of G and identify all a's of these copies to a single vertex that we call again a, while we set $B := \{b_1, \ldots, b_{k+n+1}\}$ where b_i is the copy of b in G_i. Next, we add n new vertices $C = \{c_1, \ldots, c_n\}$ and, for every $(i, j) \in [n] \times [k+n+1]$, we add the edge $e_{i,j} = c_i b_j$. The construction of $G_{\mathbf{T}_s}$ is completed by subdividing each edge $e_{i,j}$ $s-1$ times.

For every $(i, j) \in [n] \times [k+n+1]$, we denote by $P_{i,j}$ the (c_i, b_j)-s-path that replaces $e_{i,j}$ after this subdivision. Also we set

$$\mathcal{P}_j = \{P_{i,j} \mid i \in [n]\}, \text{ for } j \in [k+n+1],$$

$$\mathcal{Q}_i = \{P_{i,j} \mid j \in [k+n+1]\}, \text{ for } i \in [n],$$

and $P = \bigcup_{j \in [k+n+1]} \mathcal{P}_j$.

For the correctness of the reduction, it remains to prove the following.

$$\delta_{\mathrm{e}}^{s}(G_{\mathbf{T}_s}) \leq k + n \iff \mathbf{cut}_{G,s}(a, b) \leq k \tag{2}$$

We first claim that, for every $j \in [k+n+1]$,

$$\mathbf{cut}_{G,s}(a,b) = \mathbf{cut}_{G_{\mathbf{T}_s},s}(a,b_j). \tag{3}$$

To see (3) observe that none of the (b_j,a)-s-paths of $G_{\mathbf{T}_s}$ contains any vertex outside G_j, therefore $\mathbf{cut}_{G_{\mathbf{T}_s},s}(a,b_j) = \mathbf{cut}_{G_j,s}(a,b_j) = \mathbf{cut}_{G,s}(a,b)$.

We first prove the $(\Rightarrow)$ direction of the (2). For this we assume that $\mathbf{cut}_{G,s}(a,b) \geq k+1$ and we show that $G_{\mathbf{T}_s}$ contains a $(k+n+1,s)$-edge-hide-out, which, by Theorem 3.1, yields $\delta_{\mathrm{e}}^{s}(G_{\mathbf{T}_s}) \geq k+n+1$. We claim that $S := C \cup B \cup \{a\}$ is a $(k+n+1,s)$-edge-hide-out of $G_{\mathbf{T}_s}$. As $\mathbf{cut}_{G,s}(a,b) \geq k+1 \geq 1$, we know that for each $j \in [k+n+1]$ there is a (b_j,a)-s-path, say R_j, in $G_{\mathbf{T}_s}$ whose internal vertices are not vertices of any path in P. Moreover, every two paths in $\mathcal{R} := \{R_j \mid j \in [k+n+1]\}$ have only one vertex, that is a in common. The fact that $|\mathcal{R}| = k+n+1$ implies that $\mathbf{cut}_{G_{\mathbf{T}_s},s}(a,B) \geq k+n+1$. Therefore, as $\mathbf{cut}_{G_{\mathbf{T}_s},s}(a,S \setminus \{a\}) \geq \mathbf{cut}_{G_{\mathbf{T}_s},s}(a,B)$, we have that

$$\mathbf{cut}_{G_{\mathbf{T}_s},s}(a,S \setminus \{a\}) \geq k+n+1. \tag{4}$$

Consider now the vertex b_j, for some $j \in [k+n+1]$, and notice that that $\mathbf{cut}_{G_{\mathbf{T}_s},s}(b_j, W \cup \{a\}) \geq \mathbf{cut}_{G_{\mathbf{T}_s},s}(a,b_j) + |\mathcal{P}_j|$. Combining this with (3) and the fact that $|\mathcal{P}_j| = n$, we obtain that $\mathbf{cut}_{G_{\mathbf{T}_s},s}(b_j, W \cup \{a\}) \geq \mathbf{cut}_{G,s}(a,b) + n \geq k+n+1$. As $\mathbf{cut}_{G_{\mathbf{T}_s},s}(b_j, S \setminus \{b_j\}) \geq \mathbf{cut}_{G_{\mathbf{T}_s},s}(b_j, W \cup \{a\})$, we have that

$$\forall j \in [k+n+1] \quad \mathbf{cut}_{G_{\mathbf{T}_s},s}(b_j, S \setminus \{b_j\}) \geq k+n+1. \tag{5}$$

Consider now the vertex c_i, for some $i \in [n]$. Notice that $\mathbf{cut}_{G_{\mathbf{T}_s},s}(c_i,B) \geq |\mathcal{Q}_i| = k+n+1$. As $\mathbf{cut}_{G_{\mathbf{T}_s},s}(c_i, S \setminus \{c_i\}) \geq \mathbf{cut}_{G_{\mathbf{T}_s},s}(c_i,B)$ we obtain that

$$\forall i \in [n] \quad \mathbf{cut}_{G_{\mathbf{T}_s},s}(c_i, S \setminus \{c_i\}) \geq k+n+1. \tag{6}$$

It now follows from (4), (5), and (6), that S is an $(k+n+1,s)$-edge-hide-out of $G_{\mathbf{T}_s}$, as required.

We now prove the $(\Leftarrow)$ direction of (2). The assumption that $\mathbf{cut}_{G,s}(a,b) \leq k$ implies that $\mathbf{cut}_{G_{\mathbf{T}_s},s}(a,b_j) \leq k$, because of (3). Therefore there is a set F_j of edges in G_i that blocks every (b_j,a)-s-path of $G_{\mathbf{T}_s}$.

Let $L = \langle v_1, \ldots, v_\ell \rangle$ be any layout of the vertices of $G_{\mathbf{T}_s}$ where

$$L_{\leq k+2n+2} = \langle a, c_1, \ldots, c_n, b_1, \ldots, b_{k+n+1} \rangle \tag{7}$$

In order to prove that $\delta_{\mathrm{e}}^{s}(G_{\mathbf{T}_s}) \leq k+n$ it suffices to show that, for each $h \in [\ell]$,

$$\mathbf{supp}_{G,s}(v_h, L_{\leq h-1}) \quad \leq \quad k+n. \tag{8}$$

Notice that the vertices of $L_{\leq k+2n+2}$ are the vertices of $S = W \cup B \cup \{a\}$. As each other vertex $v \in V(G) \setminus S$, has degree at most $n-1$ in $G_{\mathbf{T}_s}$, we directly have that (8) holds when $h \in [k+2n+3, \ell]$. Let now $v_h = b_j$ for some $j \in [k+n+1]$. Let F_j^* be the edges incident to b_j that are edges of the paths in $\mathcal{P}_j$. Observe that $F_j \cup F_j^*$ blocks in $G_{\mathbf{T}_s}$ all the s-paths from $L_{\leq h-1}$ to b_j. As all the edges in $F_j \cup F_j^*$ have some endpoint in $L_{\geq h}$ and $|F_j| + |F_j^*| \leq k+n$, we conclude that (8) holds when $h \in [n+2, k+2n+2]$. Let now $v_h = c_i$, $i \in [n]$. Notice that the distance in $G_{\mathbf{T}_s}$ between c_i and any vertex in $\{a\} \cup (W \setminus \{c_i\})$ is bigger than s, therefore $\mathbf{supp}_{G,s}(v_h, L_{\leq h-1}) \leq |F_j| + |F_j^*| \leq k+n$ and (8) holds when $h \in [2, n+1]$. Finally (8) holds trivially when $h = 1$. This completes the proof of (2), and the theorem follows. $\square$

4 A structural theorem for edge-admisibility

This section is dedicated to the statement and proof of our structural characterization for $\delta_{\mathrm{e}}^{\infty}$.

4.1 Basic definitions

Edge-admissibility The ∞-*admissibility* of a graph G is the minimum k for which there exists a layout $L = \langle v_1, \ldots, v_n \rangle$ of $V(G)$ such that for every $i \in [n]$ there are at most k vertex-disjoint, except for v_i, paths from v_i to $L_{\leq i-1}$ in G. If in this definition we replace "vertex-disjoint" by "edge-disjoint" (and we obviously drop the exception of v_i) we have an edge analogue of the admissibility invariant that, because of Menger's theorem is the same invariant as $\delta_{\mathrm{e}}^{\infty}$. This encourages us to alternatively refer to $\delta_{\mathrm{e}}^{\infty}(G)$ as the ∞-*edge-admissibility* of the graph G.

The purpose of this section is to give a structural characterization for graphs of bounded edge-admissibility. For this we need first a series of definitions.

Immersions. Given a graph G and two incident edges e and f of G (i.e., edges with a common endpoint) the result of *lifting e and f in G* is the graph obtained from G after removing e and f and then adding the edge formed by the symmetric difference of e and f. We say that a graph H is an *immersion* of a graph G, denoted by $H \leq G$, if a graph isomorphic to H can be obtained from some subgraph of G after a series of liftings of incident edges. Given a graph H, we define the class of *H-immersion free graphs* as the class of all graphs that do not contain H as an immersion.

Edge sums. Let G_1 and G_2 be graphs, let v_1, v_2 be vertices of $V(G_1)$ and $V(G_2)$ respectively such that $k = \mathbf{deg}_G(v_1) = \mathbf{deg}_G(v_2)$, and consider a bijection $\sigma : E_{G_1}(v_1) \to E_{G_2}(v_2)$, where $E_{G_1}(v_1) = \{e_1^i \mid i \in [k]\}$. We define the *$k$-edge sum of G_1 and G_2 on v_1 and v_2, with respect to σ*, as the graph G obtained if we take the disjoint union of G_1 and G_2, identify v_1 with v_2, and then, for each $i \in \{1, \ldots, k\}$, lift e_1^i and $\sigma(e_1^i)$ to a new edge e^i and remove the vertex v_1. We say that G is a *$(\leq k)$-edge sum of G_1 and G_2* if either G is the disjoint union of G_1 and G_2 or there is some $k' \in [k]$, two vertices v_1 and v_2, and a bijection σ as above such that G is the k'-edge sum of G_1 and G_2 on v_1 and v_2, with respect to σ.

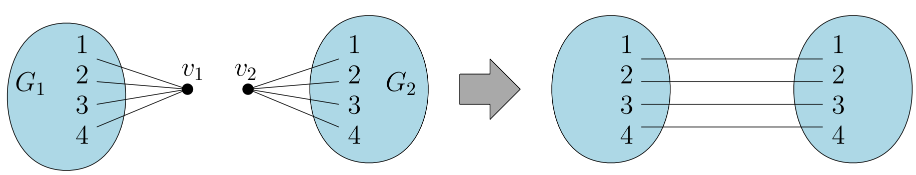

Figure 2: The graphs G_1 and G_2 and the graph created after the edge-sum of G_1 and G_2.

Let $\mathcal{G}$ be some graph class. We recursively define the *$(\leq k)$-sum closure of $\mathcal{G}$*, denoted by $\mathcal{G}^{(\leq k)}$, as the set of graphs containing every graph $G \in \mathcal{G}$ that is the $(\leq k)$-edge sum of two graphs G_1 and G_2 in $\mathcal{G}$ where $|V(G_1)|, |V(G_2)| < |V(G)|$.

A graph G is *almost k-bounded edge-degree* if all its vertices, except possibly from one, have edge-degree at most k. We denote this class of graphs by $\mathcal{A}_k$.

The rest of this section is devoted to the proof of the the following result.

Theorem 4.1. *For every graph G and $k \in \mathbb{N}_{\geq 0}$, if G has edge-admissibility at most k, then G can be constructed by almost k-bounded edge-degree graphs after a series of $(\leq k)$-edge sums, i.e., $G \in \mathcal{A}_k^{(\leq k)}$. Conversely, for every $k \in \mathbb{N}_{\geq 1}$, every graph in $\mathcal{A}_k^{(\leq k)}$ has edge-admissibility at most $2k-1$.*

4.2 A structural characterizations of θ_k-immersion free graphs

Recall that given a $k \in \mathbb{N}_{\geq 1}$, θ_k is the graph with two vertices and k parallel edges between them. In this subsection we prove that θ_k-immersion free graphs are exactly the graphs in $\mathcal{A}_k^{(\leq k)}$ (Theorem 4.7).

We need some more definitions in order to translate edge-sums to their decomposition equivalent that will be more easy to handle.

Tree-partitions. A *tree-partition* of a graph G is a pair $\mathcal{D} = (T, \mathcal{B})$ where T is a tree and $\mathcal{B} = \{B_t \mid t \in V(T)\}$ is a near-partition of $V(G)$. We refer to the sets in $\mathcal{B}$ as the *bags* of $\mathcal{D}$. Given a tree-partition $\mathcal{D} = (T, \mathcal{B})$ of G and an edge $e \in E(T)$, we define $\mathbf{cross}_{\mathcal{D}}(e) = E_G(V_1, V_2)$, where $V_i = \bigcup_{t \in V(T_i)} B_t$, for $i \in [2]$ and T_1 and T_2 are the two connected components of $T \setminus e$.

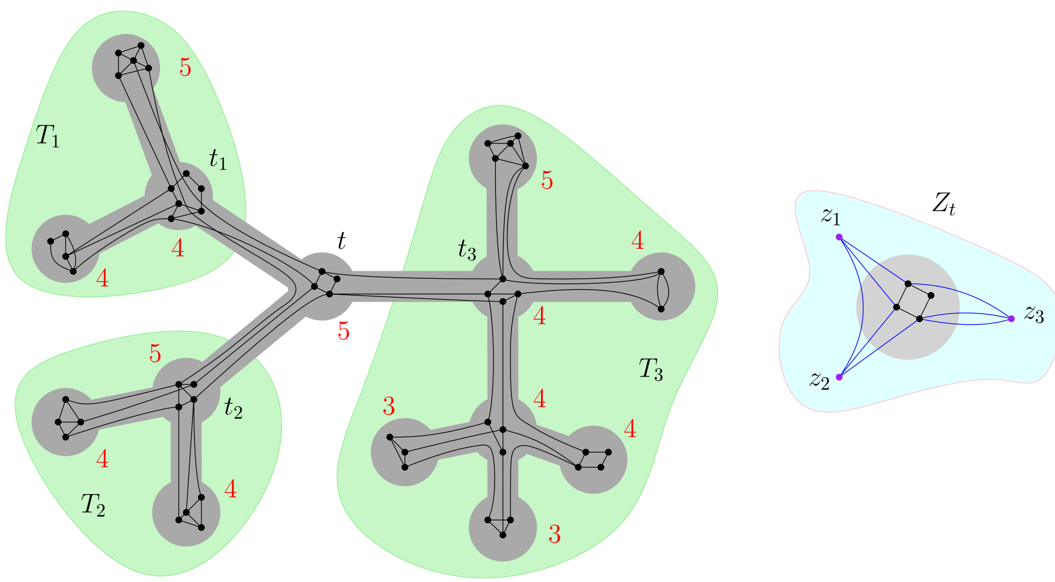

Figure 3: A graph G, a tree-partition of G with adhesion 3, and the torso Z_t of the vertex t.

For each $t \in V(T)$, we define the t-torso of $\mathcal{D}$ as follows: Let $T_1, \ldots, T_{q_t}$ be the connected components of $T \setminus t$ and let $t_1, \ldots, t_{q_t}$ be the neighbors of t in T such that $t_i \in V(T_i)$. We set $\bar{B}_i = \bigcup_{h \in V(T_i)} B_h$, for $i \in [q_t]$. Next, we define the graph Z_i as the graph obtained from G if, for every $i \in [q_t]$, we identify all the vertices of $\bar{B}_i$ to a single vertex z_i (maintaining the multiple edges created after such an identification). We call Z_t *the t-torso of $\mathcal{D}$* or, simply *a torso* of $\mathcal{D}$. We call the new vertices $z_1, \ldots, z_{q_t}$ *satellites* of the torso Z_t. For each $i \in [q_t]$, we say that z_i *represents* the vertex t_i in T and *subsumes* the connected component T_i of $T \setminus t$. For an example of a tree-partition, see Figure 3.

Let $\mathcal{D} = (\mathcal{B}, T)$ be a tree-partition of a graph G. The *adhesion* of $\mathcal{D} = (T, \mathcal{B})$ is $\max\{|\mathbf{cross}_{\mathcal{D}}(e)| \mid e \in E(T)\}$ (the adhesion of the tree-partititon of Figure 3 is 3). The *strength* of $\mathcal{D} = (T, \mathcal{B})$ is $\min\{\Delta(Z_t) \mid t \in V(T)\}$ (in the tree-partititon of Figure 3 the red numbers are the values of $\Delta(Z_t)$ for each node of the tree T).

Observe that if $\mathcal{D}$ has strength at least $k+1$, then every torso of $\mathcal{D}$ contains a vertex of degree at least $k+1$.

Notice that each graph G, where $\Delta(G) \leq k$, has a tree-partition $(T, \mathcal{B})$ where both adhesion and strength are at most k: let T be a star with center r and $|V(G)|$ leaves $\ell_1, \ldots, \ell_{|V(G)|}$, consider a numbering $v_1, \ldots, v_{|V(G)|}$ of $V(G)$, and then set $B_r = \emptyset$, while $B_{\ell_i} = \{v_i\}, i \in [|V(G)|]$.

The next observation follows directly from the definitions and provides a "translation" of edge-sums in terms of tree-partitions.

Observation 4.2. Let $\mathcal{G}$ be a graph class and let $k \in \mathbb{N}$. The class $\mathcal{G}^{(\leq k)}$ contains exactly the graphs that have a tree-partition of adhesion at most k whose torsos are graphs in $\mathcal{G}$.

Lemma 4.3. *Let $k \in \mathbb{N}_{\geq 0}$ and let G be a graph and $\mathcal{D} = (T, \mathcal{B})$ be a tree-partition of G of adhesion at most k. If $\theta_{k+1} \leq G$, then there is a $t \in V(T)$ such that $\theta_{k+1} \leq Z_t$.*

Proof. Observe that if $\theta_{k+1} \leq G$,then there are two vertices x and y in G that are connected by $k+1$ pairwise edge-disjoint (x, y)-paths, $P_1, \ldots, P_{k+1}$ in G. As $\mathcal{D}$ has adhesion at most k, there is some $t \in V(T)$ such that $x, y \in B_t$. Let $T_1, \ldots, T_{q_t}$ be the connected components of $T \setminus t$ and let $z_1, \ldots, z_{q_t}$ be the satelites of the t-torso Z_t of $\mathcal{D}$. Let $i \in [k]$ and notice that, among the edges of the (x, y)-path P_i, those missing from Z_t are those that do not have endpoints in B_t. Notice also that for every $j \in [q_t]$ the edges of P_i with both endpoints in $\bigcup_{t' \in V(T_j)} B_{t'}$ appear as consecutive edges in P_i. We now contract each such set of edges to the vertex z_j for each $j \in [q_t]$ and observe that the resulting path P'_i is a path of Z_t. Observe that $P'_1, \ldots, P'_{k+1}$ are pairwise edge-disjoint (x, y)-paths of Z_t and we conclude that $\theta_{k+1} \leq Z_t$ as required. □

Let $\mathcal{D} = (T, \mathcal{B})$ be a tree-partition of a graph G and $k \in \mathbb{N}_{\geq 0}$. We say that a torso Z_t of $\mathcal{D}$ is

- *k-splitable*: if it contains a cut $(X, \overline{X})$ of size smaller than or equal to k where both X and $\overline{X}$ contain some vertex of degree at least $k+1$.
- *k-overloaded*: if at least two of its vertices have degree at least $k+1$.

Given a tree-partition $\mathcal{D} = (T, \mathcal{B})$, we define

$$\mathbf{w}(\mathcal{D}) = \sum_{t \in V(T)} (\mathbf{s}_{\mathcal{D}}(t) - 1)$$

where $\mathbf{s}_{\mathcal{D}}(t)$ is the number of vertices in B_t that have degree at least $k+1$.

Observation 4.4. Let G be a graph, $k \in \mathbb{N}_{\geq 0}$, and $\mathcal{D}$ be a tree-partition of G that has strength at least $k+1$. Then $\mathbf{w}(\mathcal{D}) > 0$ iff some of its torsos are k-overloaded.

Given a $k \in \mathbb{N}_{\geq 0}$, we say that a tree-partition $\mathcal{D} = (\mathcal{B}, T)$ is *k-tight* if, its adhesion is at most k and its strength is at least $k+1$.

Lemma 4.5. *For every graph G and $k \in \mathbb{N}_{\geq 0}$, if $\mathcal{D}$ is a k-tight tree-partition of G with a k-splittable torso, then there is a k-tight tree-partition $\mathcal{D}'$ of G where $\mathbf{w}(\mathcal{D}') < \mathbf{w}(\mathcal{D})$.*

Proof. Let Z_t be a splittable torso of $\mathcal{D}$ and let $L_t = \{z_1, \ldots, z_{q_t}\}$ be the satellite vertices of Z_t. We denote by $t_1, \ldots, t_{q_t}$ be the vertices of T represented by $z_1, \ldots, z_{q_t}$ respectively. Also we denote by $T_1, \ldots, T_{q_t}$ the connected components of $T \setminus t$ that are subsumed by $z_1, \ldots, z_{q_t}$, respectively. Let also Q_t be the vertices of Z_t that have degree at least $k+1$. As the adhesion of $\mathcal{D}$ is at most k, it follows that each vertex in L_t has degree at most k. Therefore, $Q_t \subseteq B_t$.

We now construct a tree-partition $\mathcal{D}'$ of G. As Z_t is k-splittable, there is a cut $(X, \overline{X})$ of Z_t, of size at most k and two vertices x, y where $\mathbf{deg}_{Z_t}(x), \mathbf{deg}_{Z_t}(y) \geq k+1$, and $x \in X$ and $y \in \overline{X}$. We set $Q_t^{(x)} = Q_t \cap X$ and $Q_t^{(y)} = Q_t \cap \overline{X}$ and keep in mind that $x \in Q_t^{(x)}$ and $y \in Q_t^{(y)}$. Note that there is a set $I \subseteq [q_t]$ such that $X \cap Z_t = \{z_i \mid i \in I\}$ and $\overline{X} \cap Z_t = \{z_i \mid i \in [q_t] \setminus I\}$. We construct the tree T' as follows: we start from $T \setminus t$, then add two new adjacent vertices t_x and t_y, make t_x adjacent with all vertices in $\{t_i \mid i \in I\}$ and make t_y adjacent with all vertices in $\{t_i \mid i \in [q_t] \setminus I\}$. We also define $\mathcal{B}' = \{B'_h \mid h \in V(T')\}$ such that if $h \in V(T) \setminus \{t\}$, then $B'_h = B_h$. Finally, set $B'_{t_x} = B_t \cap X$ and $B'_{t_y} = B_t \cap \overline{X}$. Observe that

- if $e = t_x t_y$, then $|\mathbf{cross}_{\mathcal{D}'}(e)| = \mathbf{cut}_{Z_t}(X, \overline{X}) \leq k$,
- if $e = t_y t_i, i \in [q_t] \setminus I$, then $|\mathbf{cross}_{\mathcal{D}'}(e)| = |\mathbf{cross}_{\mathcal{D}}(tt_i)| \leq k$,
- if $e = t_x t_i, i \in I$, then $|\mathbf{cross}_{\mathcal{D}'}(e)| = |\mathbf{cross}_{\mathcal{D}}(tt_i)| \leq k$, and
- if $e \in E(T') \setminus E(T)|$, then $|\mathbf{cross}_{\mathcal{D}'}(e)| = |\mathbf{cross}_{\mathcal{D}}(e)| \leq k$.

From the above, we deduce that the adhesion of $\mathcal{D}'$ is at most k.

Let now $v \in V(T')$. As $\mathcal{D}$ has strength at least $k+1$, then for each $h \in V(T) \setminus \{t\}$ there is a vertex in B'_h that has degree at least $k+1$. This, together with the fact that $x \in B'_{t_x}$ and $y \in B'_{t_y}$ implies that $\mathcal{D}'$ has strength at least $k+1$. Therefore $\mathcal{D}'$ is k-tight.

We finally observe the following:

- $\mathbf{s}_{\mathcal{D}'}(t_x) = |Q_t^{(x)}|$,
- $\mathbf{s}_{\mathcal{D}'}(t_y) = |Q_t^{(y)}|$, and
- if $t \in V(T') \setminus \{t_x, t_y\}$, then $\mathbf{s}_{\mathcal{D}'}(t) = \mathbf{s}_{\mathcal{D}}(t)$

From the above, $(\mathbf{s}_{\mathcal{D}'}(t_x) - 1) + (\mathbf{s}_{\mathcal{D}'}(t_x) - 1) = |Q_t| - 2 = (\mathbf{s}_{\mathcal{D}}(t) - 1) - 1$, therefore $\mathbf{w}(\mathcal{D}') < \mathbf{w}(\mathcal{D})$ as required. ☐

Given a tree T and two members a, a' of $E(T) \cup V(T)$ we define aTa' as the unique path in T starting from a and finishing on a'. Also, given a vertex $t \in V(T)$ we define its *status* of t as

$$\mathbf{status}(T, t) = \sum_{t' \in V(T)} |E(tTt')|,$$

i.e., the sum of all the lengths of all the paths from t to the rest of the vertices of T.

Let $(X, \overline{X})$ and $(Y, \overline{Y})$ be two cuts of a graph G. We say that the cuts $(X, \overline{X})$ and $(Y, \overline{Y})$ are *parallel* if $X \subseteq Y$, or $\overline{X} \subseteq \overline{Y}$, or $X \subseteq \overline{Y}$, or $\overline{Y} \subseteq X$.

Lemma 4.6. *Let $k \in \mathbb{N}_{\geq 0}$. If G is a θ_{k+1}-immersion free graph with at least one vertex of degree at least $k+1$, Then G has a k-tight tree-partition where each torso has exactly one vertex of degree greater than k.*

Proof. Notice that G has at least one k-tight tree-partition that consists of a single bag containing all the vertices of G. Among all k-tight tree-partitions of G, consider the set $\mathfrak{D}$ containing every k-tight tree-partition of G, where $\mathbf{w}(\mathcal{D})$ takes the minimum possible value, say ℓ. From Observation 4.4 it is enough to prove that $\ell = 0$, i.e., the tree-partitions in $\mathfrak{D}$ contain no k-overloaded torsos. Assume, towards a contradiction, that $\ell > 0$. Consider two vertices x and y, of G each of degree at least $k+1$, that belong to the same bag of some tree-partition of $\mathfrak{D}$. Among all tree-partitions in $\mathcal{D}$ containing x, y in the same bag, say B_t, we choose $\mathcal{D} = (T, \mathcal{B})$ to be one where $\mathbf{status}(T, t)$ is minimized.

As $\theta_{k+1} \not\leq G$, the graph G contains some (x, y)-cut $(X, \overline{X})$ of size at most k. Let $\mathcal{S}_{x,y}$ be the set of all such cuts.

We say that an edge $e \in E(T)$ is *crossed* by $(X, \overline{X})$ if the cut of G corresponding to $\mathbf{cross}_{\mathcal{D}}(e)$ and the cut $(X, \overline{X})$ are not parallel. As both x and y have degree at least $k+1$, there should be two edges e_x and e_y in $\mathbf{cross}_{\mathcal{D}}(e)$ such that $e_x \subseteq X$ and $e_y \subseteq \overline{X}$.

Let $(X, \overline{X}) \in \mathcal{S}_{x,y}$. Let $e = t't''$ be an edge of $E(T)$ that is crossed by $(X, \overline{X})$. We make the convention that, whenever we consider such an edge, we assume that $|E(tTt'')| < |E(tTt')|$, i.e., t'' is closer to t than t', in T. We say that such an edge e is $(X, \overline{X})$-*extremal for* (T, t) if there is no other edge $e' \neq e$ of T that is crossed by $(X, \overline{X})$ and such that $e \in E(e'Tt)$. We denote by $\mathbf{extr}(X, \overline{X})$ the set of edges of T that are $(X, \overline{X})$-extremal for (T, t). Notice that $\mathbf{extr}(X, \overline{X})$ should be non-empty, as otherwise $(X, \overline{X})$ should induce a cut of Z_t, therefore Z_t would be k-splittable and this, due to Lemma 4.5, would contradict the minimality of $\mathbf{w}(\mathcal{D})$. We next define the *cost* of the cut $(X, \overline{X})$ as

$$\mathbf{cost}_{T,t}(X, \overline{X}) = \sum_{t't'' \in \mathbf{extr}(X, \overline{X})} |E(tTt')|.$$

We now pick the (x, y)-cut $(X, \overline{X}) \in \mathcal{S}_{x,y}$ as one of minimum possible cost, in other words, $\mathbf{cost}(X, \overline{X}) = \min\{\mathbf{cost}(X', \overline{X}') \mid (X', \overline{X}') \in \mathcal{S}_{x,y}\}$.

Let $e = t't''$ be an $(X, \overline{X})$-extremal edge of T. Let $(A, \overline{A})$ be the cut of G whose edges are $\mathbf{cross}_{\mathcal{D}}(e)$ and w.l.o.g., we assume that $x, y \in A$. Recall that

$$\rho(X) = \rho(\overline{X}) \leq k \quad \text{and} \quad \rho(A) = \rho(\overline{A}) \leq k. \tag{9}$$

We next claim that

$$\rho(A \cap X) = |E(A \cap X, \overline{A} \cup \overline{X})| > k. \tag{10}$$

To see (10), notice that if this is not the case, then $(A \cap X, \overline{A} \cup \overline{X}) \in \mathcal{S}_{x,y}$, because $x \in A \cap X$ and $y \in \overline{A} \cup \overline{X}$. Notice that if $t = t'$, then $\mathbf{extr}(A \cap X, \overline{A} \cup \overline{X}) = \mathbf{extr}(X, \overline{X}) \setminus \{t''t'\}$ while if t^* is the unique neighbor of t' in the path joining t' and t, then $\mathbf{extr}(A \cap X, \overline{A} \cup \overline{X}) = \mathbf{extr}(X, \overline{X}) \setminus \{t''t'\} \cup \{t't^*\}$. In both cases, $\mathbf{cost}(A \cap X, \overline{A} \cup \overline{X}) = \mathbf{cost}(X, \overline{X}) - 1$, a contradiction to the minimality of the choice of $(X, \overline{X})$. Working symmetrically on $\overline{A}$, instead of A, it follows that

$$\rho(A \cap \overline{X}) = |E(A \cap \overline{X}, \overline{A} \cup X)| > k. \tag{11}$$

By the submodularity of ρ, we have that

$$\rho(A \cap X) + \rho(A \cup X) \leq \rho(X) + \rho(A). \tag{12}$$

$$\rho(A \cap \overline{X}) + \rho(A \cup \overline{X}) \leq \rho(\overline{X}) + \rho(A). \tag{13}$$

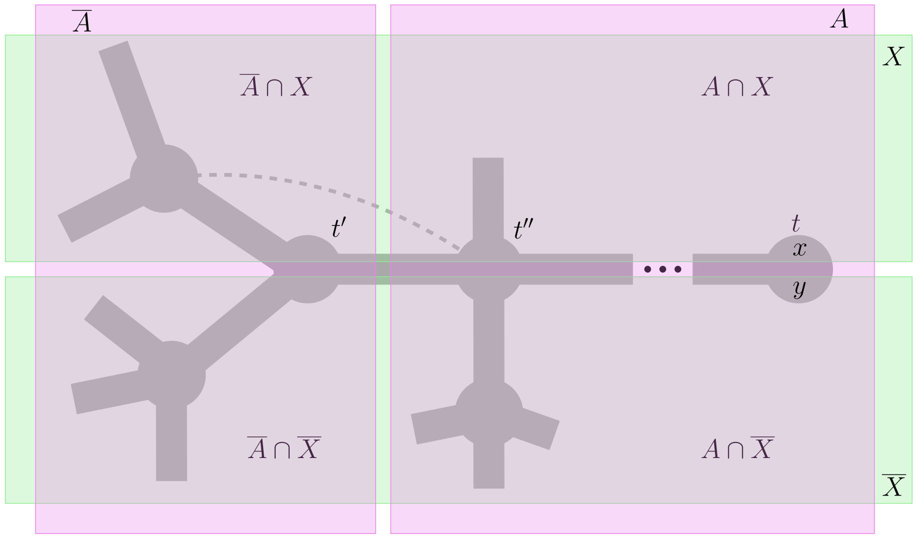

Figure 4: A visualization of the proof of Lemma 4.6.

Combining now (9), (10), and (12) and (9), (11), and (13) we have that $\rho(A \cup X) \leq k$ and $\rho(A \cup \overline{X}) \leq k$ which can be rewritten

$$\rho(\overline{A} \cap \overline{X}) \leq k \quad \text{and} \quad \rho(\overline{A} \cap X) \leq k. \tag{14}$$

Note that the vertices of $B_{t'}$ that have degree at least $k+1$ should all be in exactly one of $\overline{A} \cap \overline{X}$ and $\overline{A} \cap X$. Indeed, if this is not correct, then $Z_{t'}$ should be k-splittable and this, due to Lemma 4.5, would contradict the minimality of $\mathbf{w}(\mathcal{D})$. W.l.o.g. we assume that $Q = B_{t'} \cap \overline{A} \cap X$ contains only vertices of degree at most k.

Let $z_1, \ldots, z_{q_{t'}}$ be the satellites of $Z_{t'}$ and let t_i be the vertex of T represented by $z_i, i \in [q_{t'}]$, assuming, w.l.o.g., that z_1 represents t'' in T (that is $t_1 = t''$). Let also T_i be the connected component of $T \setminus t'$ subsumed by z_i, for $i \in [q_{t'}]$. As $t't'' \in \mathbf{extr}(X, \overline{X})$, there is some non-empty $I \subseteq [2, q_{t'}]$ such that

$$\bigcup_{i \in I} \bigcup_{s \in V(T_i)} B_s = (\overline{A} \cap X) \setminus B_{t'} \text{ and } \bigcup_{i \in [2, q_{t'}] \setminus I} \bigcup_{s \in V(T_i)} B_s = (\overline{A} \cap \overline{X}) \setminus B_{t'}. \tag{15}$$

We now add the set Q to $B_{t''}$ and remove it from $B_{t'}$, and also remove from T all edges in $\{t_i t' \mid i \in I\}$ and add the edges $\{t_i t'' \mid i \in I\}$ to get T' (in Figure 4, the new edge is depicted by the dashed edge). Observe that $\mathcal{D}' = (T', \mathcal{B})$ is a tree-partition of G with adhesion at most k and where all its nodes contain some vertex of degree at least $k+1$. Therefore $\mathcal{D}'$ is k-tight. Notice that, by the construction of T', $\mathbf{status}(T', t) < \mathbf{status}(T, t)$ a contradiction to the minimality of $\mathbf{status}(T, t)$ in the choice of $\mathcal{D} = (T, \mathcal{B})$. $\square$

Theorem 4.7. *For every graph G and $k \in \mathbb{N}$, G is θ_{k+1}-immersion free if and only if $G \in \mathcal{A}_k^{(\leq k)}$.*

Proof. We prove first "only if" direction. If G has no vertices of degree at least $k+1$, then $G \in \mathcal{A}_k$ and the result follows trivially. If G has at least one vertex of degree at least $k+1$, then, because of Lemma 4.6, G has a k-tight tree-partition of adhesion at most k and whose torsos belong to $\mathcal{A}_k$. Then, from Observation 4.2, $G \in \mathcal{A}_k^{(\leq k)}$.

We next prove the "if" direction. Suppose that $G \in \mathcal{A}_k^{(\leq k)}$, therefore, from Observation 4.2, G has a tree-partition $\mathcal{D}$ of adhesion at most k whose torsos are all in $\mathcal{A}_k$. As none of the torsos of $\mathcal{D}$ contains θ_{k+1} as an immersion, because of Lemma 4.3, the same holds for G and we are done. $\square$

As mentioned by one of the reviewers, Theorem 4.7 can alternatively be proved by a suitable application of the theorem of Gomory and Hu [14] (see also [8] and [7]).

4.3 An upper bound to edge-admissibility

In this subsection we prove that θ_{k+1}-immersion free graphs have edge-admissibility at most $2k-1$. In the end of this section, this will serve for proving Theorem 4.1.

Carving decompositions. Given a tree T we denote by $L(T)$ the set of all the vertices of T that have degree at most 1 and we call them the *leaves* of T. A *rooted tree* is a pair $\mathbf{T} = (T, r)$ where T is a tree and $r \in V(T)$. A *binary rooted tree* is a rooted tree $\mathbf{T} = (T, r)$ where all its non-leaf vertices have exactly two children. If $v \in V(T)$, we define $\mathbf{descl}_{\mathbf{T}}(v)$ as the set containing every leave ℓ of T such that $v \in V(rT\ell)$.

Let G be a graph and $S \subseteq V(G)$. A *rooted carving decomposition* of G is a pair $(\mathbf{T}, \sigma)$ consisting of a rooted binary tree $\mathbf{T} = (T, r)$ and a function $\sigma : V(G) \to L(T)$. We stress that σ is not a bijection, i.e., we permit many vertices of G to be mapped to the same leaf of T. The *weight* of a vertex t in $V(T) \setminus L(T)$ is defined as

$$\mathbf{w}(t) = |E_G(S_1, S_2)|$$

where $S_i = \sigma^{-1}(\mathbf{descl}_{\mathbf{T}}(t_i)), i \in [2]$ and t_1, t_2 are the children of t in T. For every edge $e = tt'$ of $E(T)$, where t' is a child of t, we define $\mathbf{cut}(e)$ as the set $E_G(V_1, V_2)$ where $V_1 = \sigma^{-1}(\mathbf{descl}_{\mathbf{T}}(t'))$ and $V_2 = V(G) \setminus V_1$. We also define the *weight* of $e = tt'$ as $\mathbf{w}(e) = |\mathbf{cut}(e)|$.

Lemma 4.8. *Let G be a graph and $k \in \mathbb{N}_{\geq 1}$. If $\theta_{k+1} \nleq G$, then $\delta_{\mathrm{e}}^{\infty}(G) \leq 2k-1$.*

Proof. We show that if G is θ_{k+1}-immersion free, then G cannot contain a $(2k, \infty)$-edge-hideout and therefore, from Theorem 3.1, $\delta_{\mathrm{e}}^{\infty}(G) \leq 2k-1$. Suppose to the contrary that $S, |S| \geq 2$, is a $(2k, \infty)$-edge-hideout of G. We build a rooted carving decomposition of G by applying the following precedure:

Step 1. Consider $(\mathbf{T}, \sigma)$ where $\mathbf{T} = (T, v)$, T consists of only one vertex, that is the root r, and $\sigma(v) = r$ for all $v \in V(G)$.
Step 2. Let ℓ be a vertex of T where $|\sigma^{-1}(\ell) \cap S| \geq 2$. If no such vertex exists, then **stop**.
Step 3. Pick, arbitrarily, two distinct vertices x_1 and x_2 in $\sigma^{-1}(\ell) \cap S$. Notice that G contains a (x_1, x_2)-cut (X^1, X^2) of at most k edges where $x_i \in X^i, i \in [2]$, otherwise, from Menger's theorem there are $k+1$ pairwise edge disjoint paths from x_1 to x_2 in G, which implies the existence of θ_{k+1} as an immersion in G, a contradiction. We now add in T two new vertices ℓ_1 and ℓ_2 make them the children of ℓ and update σ so that the vertices in $X^i \cap \sigma^{-1}(\ell)$ are now mapped in $\ell_i, i \in [2]$, i.e. we remove from σ $(t, \sigma^{-1}(\ell))$ and we add $(t_1, X^1 \cap \sigma^{-1}(\ell))$ and $(t_2, X^2 \cap \sigma^{-1}(\ell))$.
Step 4. Go to **Step 2**.

Let $(\mathbf{T}, \sigma)$ be the rooted carving decomposition produced by the above procedure. By the construction of $(\mathbf{T}, \sigma)$, each vertex of T has weight at most k and for each leaf $\ell \in L(G)$, $|\sigma^{-1}(\ell) \cap S| = 1$. We construct a path P of T by applying the following procedure.

Step 1. Let P be the path of T consisting of r and one (arbitrarily chosen), say t', of the children of r (i.e., P is just an edge). Notice that $\mathbf{w}(\{r, t'\}) = \mathbf{w}(r) \leq k \leq 2k - 1$ (recall that $k \geq 1$).
Step 2. Let e be the the last edge of P (starting from r) and let t be its endpoint that is also an endpoint of P (different than r). If t is a leaf of T, then **stop**.
Step 3. Let t_1 and t_2 be the children of t and let $e_i = tt_i, i \in [2]$. We partition the edges of $\mathbf{cut}(e)$ into two sets, namely F_1 and F_2 so that F_i contains edges with an endpoint in $\mathbf{descl}_{\mathbf{T}}(t_i), i \in [2]$. Notice that $\mathbf{cut}(e_i) = F_i \cup E_G(\sigma^{-1}(\mathbf{descl}_{\mathbf{T}}(t_1)), \sigma^{-1}(\mathbf{descl}_{\mathbf{T}}(t_2)))$, therefore, for $i \in [2]$,

$$\mathbf{w}(e_i) = |\mathbf{cut}(e_i)| \quad = \quad |F_i| + |\mathbf{w}(t)|. \tag{16}$$

As $\mathbf{w}(e) \leq 2k - 1$, one, say F_1, of F_1, F_2 should have at most $k - 1$ edges. By applying (16) for $i = 1$, we obtain that $|\mathbf{w}(e_1)| \leq k - 1 + \mathbf{w}(t) \leq 2k - 1$. We now extend P by adding in it the vertex t_1 and the edge e_1 and we update $e := e_1$.
Step 4. Go to **Step 2**.

We just constructed a path P in T between r and a leaf of ℓ of T such that for every edge $e \in E(P)$, $\mathbf{w}(e) \leq 2k - 1$. Notice that $\sigma^{-1}(\ell)$ contains exactly one vertex, say x, of S. Moreover, if f is the edge of T that is incident to ℓ, then $\rho(\sigma^{-1}(\ell)) = \mathbf{w}(f) \leq 2k - 1$, as f is an edge of P (the last one). This implies that there is a set of $2k - 1$ edges blocking every path from x to $S \setminus \{x\}$. Therefore, $\mathbf{supp}_G(\infty, x, S \setminus \{x\}) \geq 2k - 1$, contradicting to the fact that S is a $(2k, \infty)$-edge-hideout of G. □

Observation 4.9. If H and G are graphs then $H \leq G \Rightarrow \delta_{\mathrm{e}}^{\infty}(H) \leq \delta_{\mathrm{e}}^{\infty}(G)$.

Proof. Suppose that $H \leq G$ and that $k \leq \delta_{\mathrm{e}}^{\infty}(H)$. From Theorem 3.1 H contains a (k, ∞)-edge-hide-out $S \subseteq V(H)$. Because of Menger's theorem, for every vertex $v \in S$ there are at least $k + 1$ pairwise edge-disjoint paths from v to vertices of $S \setminus \{v\}$. Notice that these paths also exist in G as the "inverse" of the lift operation does not alter the paths from a vertex of S to the rest of the vertices of S. These paths, again using Menger's theorem, imply that S is also a $(k + 1, \infty)$-edge-hide-out of G, therefore, again from Theorem 3.1, $k \leq \delta_{\mathrm{e}}^{\infty}(G)$. □

We are now ready to give the proof of Theorem 4.1.

Proof of Theorem 4.1. For the first part of the theorem, observe that $\delta_{\mathrm{e}}^{\infty}(\theta_{k+1}) = k + 1$, therefore, from Observation 4.9, $\theta_{k+1} \not\leq G$. Using now the "only if" direction of Theorem 4.7 we obtain that $G \in \mathcal{A}_k^{(\leq k)}$, as required.

For the second part of the theorem, let $G \in \mathcal{A}_k^{(\leq k)}$, which by the "if" direction of Theorem 4.7 implies that $\theta_{k+1} \not\leq G$. Using now Lemma 4.8, we conclude that $\delta_{\mathrm{e}}^{\infty}(G) \leq 2k - 1$. □

Acknowledgements

We wish to thank the anonymous reviewers for their comments and remarks that considerably improved the presentation of this paper.

References

[1] Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous.* International Series in Operations Research & Management Science, 55. Kluwer Academic Publishers, Boston, MA, 2003.

[2] Brian Alspach. Searching and sweeping graphs: a brief survey. *Matematiche (Catania)*, 59(1-2):5–37 (2006), 2004.

[3] Georg Baier, Thomas Erlebach, Alexander Hall, Ekkehard Köhler, Heiko Schilling, and Martin Skutella. Length-bounded cuts and flows. In *Automata, languages and programming. Part I*, volume 4051 of *LNCS*, pages 679–690. Springer, Berlin, 2006.

[4] Daniel Bienstock. Graph searching, path-width, tree-width and related problems (a survey). In *Reliability of computer and communication networks (New Brunswick, NJ, 1989)*, volume 5 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 33–49. Amer. Math. Soc., Providence, RI, 1991.

[5] Hans L. Bodlaender, Thomas Wolle, and Arie M. C. A. Koster. Contraction and treewidth lower bounds. *J. Graph Algorithms Appl.*, 10(1):5–49 (electronic), 2006.

[6] Guantao Chen and Richard H. Schelp. Graphs with linearly bounded ramsey numbers. *Journal of Combinatorial Theory, Series B*, 57(1):138 – 149, 1993.

[7] Matt DeVos, Jessica McDonald, Bojan Mohar, and Diego Scheide. A note on forbidding clique immersions. *Electr. J. Comb.*, 20(3):P55, 2013.

[8] Reinhard Diestel, Fabian Hundertmark, and Sahar Lemanczyk. Profiles of separations: in graphs, matroids, and beyond. *Combinatorica*, 39(1):37–75, 2019.

[9] Zdeněk Dvořák. A stronger structure theorem for excluded topological minors. *arXiv:1209.0129*, 2012.

[10] Zdeněk Dvořák. Constant-factor approximation of the domination number in sparse graphs. *European Journal of Combinatorics*, 34(5):833 – 840, 2013.

[11] Fedor V. Fomin and Nikolai N. Petrov. Pursuit-evasion and search problems on graphs. In *Proceedings of the Twenty-seventh Southeastern International Conference on Combinatorics, Graph Theory and Computing*, volume 122 of *Congr. Numer.*, pages 47–58, 1996.

[12] Fedor V. Fomin and Dimitrios M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoret. Comput. Sci.*, 399(3):236–245, 2008.

[13] Archontia C. Giannopoulou, Marcin Jakub Kaminski, and Dimitrios M. Thilikos. Forbidding kuratowski graphs as immersions. *Journal of Graph Theory*, 78(1):43–60, 2015.

[14] Ralph E. Gomory and T. C. Hu. Multi-terminal network flows. *Journal of the Society for Industrial and Applied Mathematics*, 9(4):551–570, 1961.

[15] Martin Grohe, Stephan Kreutzer, Roman Rabinovich, Sebastian Siebertz, and Konstantinos Stavropoulos. Colouring and covering nowhere dense graphs. In *Graph-Theoretic Concepts in Computer Science - 41st International Workshop, WG 2015, Garching, Germany, June 17-19, 2015, Revised Papers*, pages 325–338, 2015.

[16] Alon Itai, Yehoshua Perl, and Yossi Shiloach. The complexity of finding maximum disjoint paths with length constraints. *Networks*, 12(3):277–286, 1982.

[17] Hal A. Kierstead and William T. Trotter. Planar graph coloring with an uncooperative partner. In *Planar Graphs, Proceedings of a DIMACS Workshop, New Brunswick, New Jersey, USA, November 18-21, 1991*, pages 85–94, 1991.

[18] Lefteris M. Kirousis and Dimitrios M. Thilikos. The linkage of a graph. *SIAM J. Comput.*, 25(3):626–647, 1996.

[19] Ali Ridha Mahjoub and S. Thomas McCormick. Max flow and min cut with bounded-length paths: complexity, algorithms, and approximation. *Math. Program.*, 124(1-2):271–284, 2010.

[20] David W. Matula. A min–max theorem for graphs with application to graph coloring. *SIAM Reviews*, 10:481–482, 1968.

[21] Jaroslav Nesetril and Patrice Ossona de Mendez. *Sparsity - Graphs, Structures, and Algorithms*, volume 28 of *Algorithms and combinatorics*. Springer, 2012.

[22] Jaroslav Nešetřil and Patrice Ossona de Mendez. Fraternal augmentations, arrangeability and linear ramsey numbers. *European Journal of Combinatorics*, 30(7):1696 – 1703, 2009. EuroComb'07: Combinatorics, Graph Theory and Applications.

[23] David Richerby and Dimitrios M. Thilikos. Searching for a visible, lazy fugitive. *SIAM J. Discrete Math.*, 25(2):497–513, 2011.

[24] Daniel Weißauer. On the block number of graphs. *SIAM J. Discrete Math.*, 33(1):346–357, 2019.

[25] Paul Wollan. The structure of graphs not admitting a fixed immersion. *J. Comb. Theory, Ser. B*, 110:47–66, 2015.