



HAL
open science

Connected Partitions of Graphs with Vertex Excess

Olivier Baudon, Julien Bensmail, Lyn Vayssieres

► **To cite this version:**

Olivier Baudon, Julien Bensmail, Lyn Vayssieres. Connected Partitions of Graphs with Vertex Excess. 2026. <hal-05543481>

HAL Id: hal-05543481

<https://hal.science/hal-05543481v1>

Preprint submitted on 9 Mar 2026

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons CC BY-NC 4.0 - Attribution - Non-commercial use - International License

Connected Partitions of Graphs with Vertex Excess

Olivier Baudon^a, Julien Bensmail^b, Lyn Vayssieres^a

^aUniv. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR 5800, F-33400, Talence, France

^bUniversité Côte d'Azur, CNRS, Inria, I3S, France

Abstract

In this work, we generalise the notions of connected partitions and arbitrarily partitionable graphs by allowing parts to induce connected graphs through external vertices. Namely, for a graph G of order n and any partition $\pi = (n_1, \dots, n_p)$ of n , we are interested in partitions $\mathcal{P} = (V_1, \dots, V_p)$ of $V(G)$ where $|V_i| = n_i$ for all $i \in \{1, \dots, p\}$, such that there exist (possibly empty) sets $\mathcal{C} = (C_1, \dots, C_p)$ for which $G[V_i \cup C_i]$ is connected for every $i \in \{1, \dots, p\}$. That is, vertices in the C_i 's act as “helpers” to make the V_i 's induce connected graphs. However, from a practical point of view, it is preferable that a given vertex of G does not appear in too many parts of \mathcal{C} , which parameter we call its excess. We introduce and investigate the new notion of x -XAP graphs, which are graphs that can be partitioned this way whatever π is, and so that every vertex, individually, has excess at most x .

We initiate the study of these notions, focusing in particular on possible generalisations of earlier results without the extra excess requirement. We establish several properties of x -XAP graphs, in particular for classes of graphs that are common to consider in this field. We also provide algorithmic results, establishing that finding a certain \mathcal{P} or \mathcal{C} as above is NP-complete, whatever the excess. We finish off with directions for further work.

Keywords: connected partition; arbitrarily partitionable graph; excess.

1. Introduction

Let G be a graph of order n (*i.e.*, having n vertices). An n -partition $\pi = (n_1, \dots, n_p)$ into $p \geq 1$ parts n_1, \dots, n_p is a set of p (possibly repeating) positive integers n_1, \dots, n_p fulfilling $n_1 + \dots + n_p = n$. A π -realisation (V_1, \dots, V_p) in G is a partition of $V(G)$ into p connected parts V_1, \dots, V_p (that is, $G[V_i]$ is connected for every $i \in \{1, \dots, p\}$) and $|V_i| = n_i$ for every $i \in \{1, \dots, p\}$. Finally, we say that G is *arbitrarily partitionable* (or AP, for short) if all n -partitions are realisable in G ; or, in other words, if G can be partitioned (vertex-wise) into arbitrarily many connected graphs, whatever their orders are. Note that a necessary, implicit condition for a graph to be AP is that it is connected.

The notion of AP graph was introduced independently by Barth, Baudon, and Puech in [1], and by Horňák and Woźniak in [12]. In the former reference, the authors motivated the introduction of this notion by the fact that it models the following practical problem. Suppose we own a network of n computing resources, and that, as managers, we are asked to share/split our resources between p customers, each of which requires a prescribed number of resources. Of course, resources should not be split arbitrarily; rather, all resources attributed to some customer should form a network, in that these resources should be able to communicate within the subnetwork they form. It is preferable also, to avoid concurrency conflicts, that no resource is attributed to two different customers. This problem can quite naturally be modeled under a graph formalism, where our network would be represented as a graph (vertices being resources, edges indicating direct communication between pairs

of resources) and the resource sharing problem above would correspond to partitioning this graph into connected subgraphs, in the sense we introduced earlier. And, because we *a priori* can imagine we can anticipate neither the numbers p of customers and their requested numbers n_1, \dots, n_p of resources, ideally we would rather prefer the graph modeling our network to be AP, so that there is not bad scenario where we cannot satisfy customers.

Several aspects of AP graphs have been investigated since their introduction, see *e.g.* [2, 7, 8, 9, 13, 16] and the pointers there for an overview of the latest progress on the topic. Note that, later, previous results will be surveyed as they relate to our own investigations.

Although the practical network sharing problem above is rather natural, it is also quite general, and one may thus argue the notion of AP graph is not the most appropriate to model it properly. For instance, one may argue that sharing our network at once, once all resources are requested, is not too natural, and it would thus be preferable that our resources can be attributed on the fly, as customers request them. Likewise, one could argue that it could be preferable that all subnetworks we attribute can recursively be shared at will, in case customers themselves want to be able to reshare their network later on. Or one could also argue that, in some contexts, our resources might not be all identical, and thus customers might request their subnetwork to contain some particular of these resources. Such concerns led authors to introduce more restricted notions of AP graphs (such as OLAP graphs [11], RAP graphs [5], and AP+ k graphs [4], respectively), and something interesting is investigating how these modified notions differ from the original one.

In this work, we introduce another modified notion of AP graphs, motivated by the following thoughts. It is obvious that not all graphs are AP; for an example, consider realising the n -partition $(2, \dots, 2)$ in any star with even order $n \geq 4$. This implies, regarding the practical problem we introduced earlier, that, in some contexts, depending on our network's structure and the customers' requests, we cannot satisfy customers as desired. In such situations, one may instead try to maximise our customers' satisfaction as much as possible, while trying to minimise disturbance/dissatisfaction. Again, these objectives are quite vague, and one could thus imagine several ways to measure satisfaction/dissatisfaction.

Here, we focus on the following ideas and concepts, which we believe are among the most natural ones to consider regarding the original practical problem. When attributing resources to a customer Q_1 , the ideal scenario is when these resources form a subnetwork N_1 in which any two resources can communicate (not necessarily directly), which, from the graph theoretical point of view, translates to connectivity. In cases, however, where these resources do not form such a subnetwork (thus, equivalently, a subgraph which is not connected), any two resources of N_1 willing to communicate might have to involve, as intermediaries, resources belonging to networks N_2, \dots, N_p attributed to other customers Q_2, \dots, Q_p . From Q_1 's point of view, allowing such outside communications is transparent and thus fine. However, having resources of N_2, \dots, N_p supporting communications between resources of N_1 might result in resources of N_2, \dots, N_p to handle extra workload, which might cause inconveniences (slowdown, etc.) to Q_2, \dots, Q_p 's subnetworks. Thus, though allowing such outside communications might be fine, they should be limited.

We treat these concerns through the following terminology (see Figure 1). Let G be a graph of order n , let $\pi = (n_1, \dots, n_p)$ be an n -partition, and let $\mathcal{P} = (V_1, \dots, V_p)$ be a partition of $V(G)$ into p parts V_1, \dots, V_p of cardinality n_1, \dots, n_p , respectively, where the V_i 's are not necessarily connected parts. A set $\mathcal{C} = (C_1, \dots, C_p)$ of p (possibly empty and/or intersecting) subsets C_1, \dots, C_p of $V(G)$ is called a \mathcal{P} -connector if $G[V_i \cup C_i]$ is connected for all $i \in \{1, \dots, p\}$. If \mathcal{C} is a \mathcal{P} -connector, then we call $(\mathcal{P}, \mathcal{C})$ a π -realisation in G . That is, \mathcal{P} may not be a π -realisation (if some V_i 's are not connected), but each

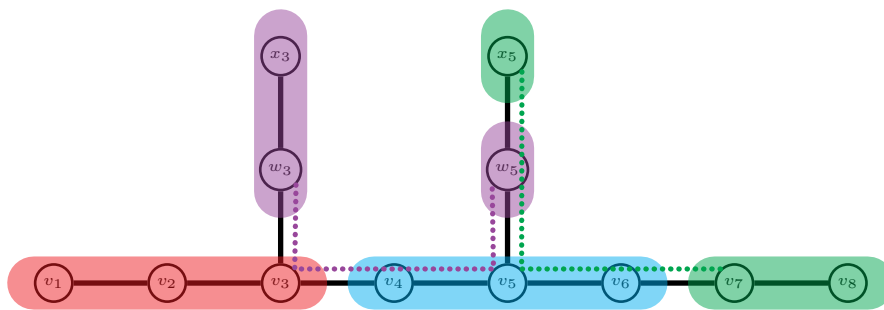


Figure 1: A $(3,3,3,3)$ -realisation $(\mathcal{P}, \mathcal{C})$ in some graph G . The four parts of \mathcal{P} are represented by regions with different colours surrounding vertices. Note that the red and blue parts are connected, while the purple and green parts are not. In the connector \mathcal{C} (whose parts are represented by dotted lines with colours matching the colours of the parts they connect), the parts associated to the red and blue parts are thus empty, while, regarding the purple and green parts, they can be reduced to $\{v_3, v_4, v_5\}$ and $\{v_6, v_5, w_5\}$, respectively. As a result, we have $x(v_3) = x(v_4) = x(v_6) = x(w_5) = 1$ and $x(v_5) = 2$; so $(\mathcal{P}, \mathcal{C})$ has excess 2. Actually, G is not AP (it admits no $(3,3,3,3)$ -realisation with excess 0); it is thus, at best, 1-XAP. Let us add that G admits $(3,3,3,3)$ -realisations with excess 1, since $G - \{x_3, w_3, v_4\}$ admits an obvious $(3,3,3,3)$ -realisation with excess 0, while $\{x_3, w_3, v_4\}$ is not connected but becomes connected when added v_3 .

part V_i that is not connected becomes connected when added the vertices in C_i . Regarding the concerns above, roughly put, any part C_i of the connector thus indicates intermediary resources supporting communication between resources of V_i . Obviously, if $C_i = \emptyset$, then V_i is connected. Actually, if \mathcal{P} is already a π -realisation, then $\mathcal{C} = (\emptyset, \dots, \emptyset)$ is a \mathcal{P} -connector.

Now, as mentioned earlier, we would rather be interested in connectors that do not involve some given vertices too much. Namely, every vertex u of G is, in some sense, more or less “stressed” depending on the numbers of C_i ’s containing u . That is, the more C_i ’s contain u , the more u helps in connecting parts of the realisation that are not connected. W.r.t. \mathcal{C} , we thus denote by $x(u) \geq 0$ the *excess* of u , being the number of C_i ’s of \mathcal{C} in which x appears. Likewise, we denote by $x(\mathcal{P}, \mathcal{C})$ the *excess* of $(\mathcal{P}, \mathcal{C})$, being the minimum $x \geq 0$ such that $x(u) \leq x$ for all vertices u of G . And since we would rather try to minimise this parameter, we say G is x -XAP if $x \geq 0$ is the minimum integer for which every n -partition admits a realisation with excess at most x . Note that APness is exactly 0-XAPness.

This work is organised as follows. We begin in Section 2 by raising first remarks and observations on all notions we have just introduced. In particular, we there end up raising Conjecture 2.7 on the maximum value x such that graphs are x -XAP, which conjecture we then investigate further in Section 3 for some common classes of graphs of the field. We complement our results with algorithmic ones in Section 3. In particular, we prove that, in a graph, both realising a given partition with given excess, and finding a connector with given excess, are NP-complete problems. We finish off in Section 5 with questions and problems we think might be appealing for further work on the topic.

2. Preliminary remarks, observations, and questions

To begin with, we start off by stating an obvious fact, being that if we do not mind about minimising the resulting excess, then, for every connected graph G of order n , every n -partition admits a realisation in G .

Observation 2.1. *If G is a connected graph of order n and π is an n -partition, then there are π -realisations $(\mathcal{P}, \mathcal{C})$ (thus with some excess) in G .*

Proof. Just consider any partition $\mathcal{P} = (V_1, \dots, V_p)$ of $V(G)$ where the (not necessarily connected) parts have the required cardinalities w.r.t. π , and consider $\mathcal{C} = (C_1, \dots, C_p)$

where $C_i = V(G)$ for all $i \in \{1, \dots, p\}$. Then, obviously, since G is connected, so is every $G[V_i \cup C_i]$. It follows that \mathcal{C} is a \mathcal{P} -connector, and thus that $(\mathcal{P}, \mathcal{C})$ is a π -realisation. \square

Although, in the definitions we introduced earlier, what matters in connectors is the maximum resulting excess over all vertices, obviously we would rather be interested, regarding our practical motivations, in connectors containing the bare minimum number of vertices. We thus say, given a graph G and some π -realisation $(\mathcal{P}, \mathcal{C})$ in G where $\mathcal{P} = (V_1, \dots, V_p)$ and $\mathcal{C} = (C_1, \dots, C_p)$, that \mathcal{C} is *minimal* if there are no C'_1, \dots, C'_p with $C'_i \subseteq C_i$ for every $i \in \{1, \dots, p\}$ such that $(\mathcal{P}, \mathcal{C}')$ is a π -realisation (for $\mathcal{C}' = (C'_1, \dots, C'_p)$) and $\sum_{i=1}^p |C'_i| < \sum_{i=1}^p |C_i|$. In other words, \mathcal{C} is minimal if removing any vertex from its parts makes it lose its property of being a connector for \mathcal{P} .

By considering minimal connectors, we can come up with a number of results.

Observation 2.2. *Let G be a graph, and $(\mathcal{P}, \mathcal{C})$ be a realisation in G where $\mathcal{P} = (V_1, \dots, V_p)$ and $\mathcal{C} = (C_1, \dots, C_p)$. If \mathcal{C} is minimal, then $V_i \cap C_i = \emptyset$ for every $i \in \{1, \dots, p\}$.*

Proof. This follows from the definitions, since if $u \in V_i \cap C_i$ for some $i \in \{1, \dots, p\}$, then $G[V_i \cup C_i] = G[V_i \cup (C_i \setminus \{u\})]$ and thus $V_i \cup (C_i \setminus \{u\})$ is a connected part, implying that removing u from C_i in \mathcal{C} results in a \mathcal{P} -connector that is smaller than \mathcal{C} , a contradiction. \square

A consequence is then:

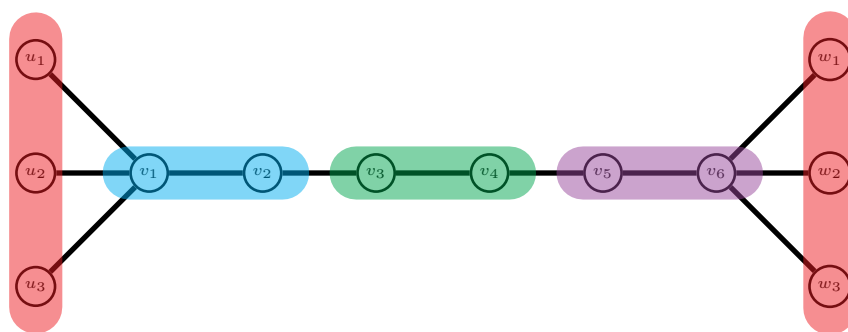
Theorem 2.3. *If G is any graph of order n , then the maximum excess of a vertex of G by any realisation is $\lfloor \frac{n-1}{2} \rfloor$. Thus, G is x -XAP for some $x \leq \lfloor \frac{n-1}{2} \rfloor$.*

Proof. Let $(\mathcal{P}, \mathcal{C})$ be a π -realisation in G , where $\mathcal{P} = (V_1, \dots, V_p)$ and $\mathcal{C} = (C_1, \dots, C_p)$. We can assume \mathcal{C} is minimal. Consider any vertex u of G , and assume u belongs to V_1 w.l.o.g. By Observation 2.2, we have $u \notin C_1$. Then $x(u)$ reaches its maximum value when $u \in C_i$ for all $i \in \{2, \dots, p\}$, which is when p is maximised, none of V_2, \dots, V_p is connected, and, thus, $|V_1| = 1$ and $|V_i| = 2$ for all $i \in \{2, \dots, p\}$. In this case, we have $p - 1 = \lfloor \frac{n-1}{2} \rfloor$. \square

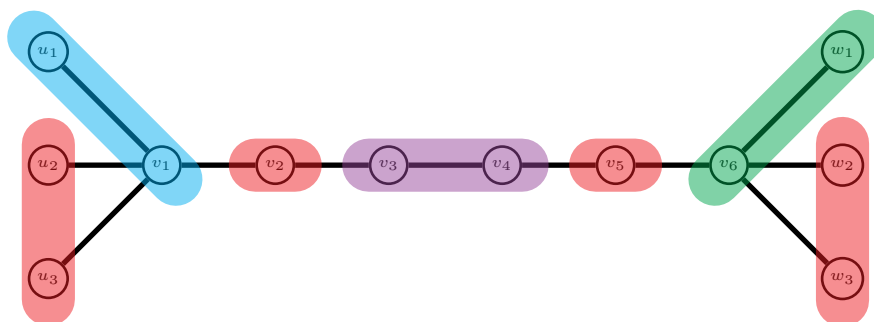
To comprehend how optimal Theorem 2.3 is, we can consider stars, which have frequently been mentioned as simple graphs that are somewhat the worst ones w.r.t. the AP property. Note indeed that every star of order n at least 4 admits no (quasi-) perfect matching, thus no realisation of either $(2, \dots, 2)$ (if n is even) or $(1, 2, \dots, 2)$ (otherwise). We formalise this fact through the XAP property (showing that, in general, the possible margin for improving upon Theorem 2.3 is very thin):

Theorem 2.4. *Every star of even order $n \geq 4$ is $(\frac{n}{2} - 1)$ -XAP, while every star of odd order $n \geq 3$ is $(\frac{n-1}{2} - 1)$ -XAP.*

Proof. Let G be a star of order $n \geq 4$, with center r (of degree $n - 1 \geq 3$), and π be any n -partition. Note that, in any π -realisation $(\mathcal{P}, \mathcal{C})$ in G , every part of \mathcal{P} that does not contain r and has cardinality at least 2 contributes 1 to $x(\mathcal{P}, \mathcal{C})$. Note, in particular, that, assuming \mathcal{C} is minimal, every non-empty part of \mathcal{C} reduces to $\{r\}$, and thus $x(\mathcal{P}, \mathcal{C}) = x(r)$. This implies that, so that $x(\mathcal{P}, \mathcal{C})$ is minimum, the part of \mathcal{P} containing r must be the largest one. From these arguments, we deduce that the worst-case scenario for π (i.e., generating the most excess for r in any π -realisation) is when $\pi = (2, \dots, 2)$ (when n is even) and $\pi = (1, 2, \dots, 2)$ (otherwise), as these configurations maximise the number of elements of π with value at least 2, while they minimise the value of the largest element with value at least 2. In the first case, we get $x(r) = \frac{n}{2} - 1$. In the second case, we get $x(r) = \frac{n-1}{2} - 1$. \square



(a) 1st realisation



(b) 2nd realisation

Figure 2: Two $(6, 2, 2, 2)$ -realisations $(\mathcal{P}, \mathcal{C})$ in a graph G . In each case, the four parts of \mathcal{P} are represented by regions with different colours surrounding vertices. Note that, in both cases, only the red part (of cardinality 6) is not connected. Connectors \mathcal{C} are not represented on the figure, but it can be checked that, assuming minimality of \mathcal{C} , in both cases the realisation has excess 1 due to the red part. While the realisation in (a) looks more natural (as it minimises the number of connected components in the red part), the one in (b) is thus equivalent excess-wise. Note that G admits no $(6, 2, 2, 2)$ -realisation with excess 0.

Note that this simple case shows that x -XAP graphs exist for all $x \geq 0$. Also, although the case of stars is rather simple, still we can deduce general properties of connectors from it. For instance, one could wonder whether, by any realisation $(\mathcal{P}, \mathcal{C})$ (with minimal \mathcal{C}) of a partition in some graph, any non-connected part V_i of \mathcal{P} induces a bounded number of connected components. From stars, we deduce this is not true, regardless of \mathcal{C} . Thus, parts of minimal connectors may have to “connect” arbitrarily many connected components.

Observation 2.5. *There is no constant $k \geq 1$ such that, for every graph G of order n and every n -partition π , there is necessarily a π -realisation $(\mathcal{P}, \mathcal{C})$ in G such that, for every part V_i of \mathcal{P} , the number of connected components of $G[V_i]$ is at most k .*

Proof. To see this is true, consider *e.g.* any star G with even order $n \geq 4$, and the n -partition $\pi = (n/2, n/2)$. By arguments used in the proof of Theorem 2.4, in any π -realisation $(\mathcal{P}, \mathcal{C})$, for the part V_i (of cardinality $n/2$) of \mathcal{P} not containing the center vertex of G we have that $G[V_i]$ consists in exactly $n/2$ isolated vertices. The statement then follows from the fact that these arguments hold however large n is. \square

On the other hand, the way we defined minimal connectors is rather wide, in that parts of a connector do not have to fulfil particular properties. Note in particular that if $(\mathcal{P}, \mathcal{C})$ is a realisation of some partition in a graph G where \mathcal{C} is minimal, then, for any part C of \mathcal{C} , nothing imposes that $G[C]$ is connected. While one may conceptualise that the parts in a minimal connector behave well, such as inducing connected subgraphs, etc., nothing guarantees this has to be the case (see Figure 2). In the same line of thoughts, one

may wonder whether there are situations where parts of a minimal connector necessarily induce some structures (such as cycles, cliques, etc.). Due to the freedom allowed by our definitions, nothing prevents such phenomena to occur.

To further illustrate these concerns a bit, in what follows we construct situations where minimal connectors have their parts inducing graphs of arbitrarily large diameter. This contrasts with the situation investigated in Observation 2.5, where all parts in minimal connectors induced graphs with very small diameter. The proof we give illustrates why establishing such a seemingly easy result is not that straightforward.

Observation 2.6. *There exist graphs G of order n such that, for some n -partitions π , all π -realisations $(\mathcal{P}, \mathcal{C})$ in G with the least excess have excess 1 and have parts C in \mathcal{C} such that $G[C]$ has connected components with arbitrarily large diameter.*

Proof. Consider e.g. the following construction. Start from an edge uv , add five leaves u_1, \dots, u_5 adjacent to u , five leaves v_1, \dots, v_5 adjacent to v , and subdivide uv an arbitrary number of times but so that the order $n \geq 12$ of the whole resulting graph G is even. We denote by $l \geq 0$ the (even) number of inner vertices on the unique path joining u and v .

Let us consider the n -partition $\pi = (10, 2, \dots, 2)$. If any π -realisation with excess 0 in G existed, then, in that realisation, no part of cardinality 2 could include one of the ten leaves. Indeed, if, say, $\{u_1, u\}$ was a part, then it would be impossible to cover u_2, \dots, u_5 via connected parts of cardinality 2 or 10. Thus, the ten leaves of G would necessarily have to be covered by the only connected part of cardinality 10 of the realisation, which would make it impossible for that part to be connected, a contradiction.

On the other hand, π -realisations $(\mathcal{P}, \mathcal{C})$ with excess 1 exist; an obvious one consists in having all ten leaves belonging to the unique part of \mathcal{P} of cardinality 10, having the associated part C^* of \mathcal{C} containing all vertices of the unique path from u to v , and having all other parts (of cardinality 2) of \mathcal{P} being connected parts over the consecutive vertices of the unique path joining u and v (thus with the associated parts of \mathcal{C} being empty).

In that realisation $(\mathcal{P}, \mathcal{C})$, note that $G[C^*]$ indeed contains its unique connected component covering all vertices of G on the unique path joining u and v ; thus, that connected component has diameter $l + 2$, which can be arbitrarily large as l can be chosen arbitrarily large in the first place. The tricky part, now, is that $(\mathcal{P}, \mathcal{C})$ is not unique, as one can easily perform slight modifications to it while preserving excess at most 1. For instance, one could (similarly as what is depicted in Figure 2(b)) have \mathcal{P} containing $\{u_1, u\}$ as a connected part of cardinality 2 instead, having the first two inner vertices on the path from u to v belonging to the part of cardinality 10 instead, and have \mathcal{C} modified accordingly. However, it can be noted that not all u_i 's (and similarly v_i 's) can be covered by parts of \mathcal{P} of cardinality 2 while preserving excess 1: the best we can perform is having, say, $\{u, u_1\}$ as a connected part, $\{u_2, u_3\}$ as a non-connected part (thus with associated part $\{u\}$ in the connector), which prevents $\{u_4, u_5\}$ from being covered by a part being either connected or not increasing the excess of u . Thus, at least one of the u_i 's must be covered by the part of cardinality 10, and similarly for at least one for the v_i 's. Because that part has cardinality 10, we deduce that the number of inner vertices of the main path this part can cover is at most 8. Then, because l can be arbitrarily large, one connected component of the associated part of the connector must have large diameter (function of l). \square

Back to implications of Theorem 2.4, a natural question is about the maximum value x such that graphs can be x -XAP. Due to Theorems 2.3 and 2.4, this maximum value should be expressed as a graph function. The case of stars is a bit misleading however, as, in our proof of Theorem 2.4, the important point is that $n - 1$ corresponds to the maximum degree Δ . Our guess is that Theorem 2.3 should be generalised to a result alike:

Conjecture 2.7. *There is a linear function f of the maximum degree such that if G is a graph of maximum degree Δ , then G is x -XAP for some $x < f(\Delta)$.*

It is important to make clear that Conjecture 2.7 is only about an upper bound on the value of x such that a given graph is x -XAP. In particular, note that 0-XAP graphs can have arbitrarily large maximum degree (consider *e.g.* any complete graph), so, in general, there is no systematic connection between maximum degree and excess.

In the next section, we investigate Conjecture 2.7 for classes of graphs that have been commonly investigated in the context of AP graphs. In particular, through the case of trees, we will get to establish that f indeed exists.

3. Classes of graphs

In this section, we investigate the XAPness of classes of graphs that are common to consider when it comes to studying AP graphs. All results we establish here should be taken in the context of Conjecture 2.7, which we essentially confirm for all classes of graphs we get to consider. Another crucial point for focusing on particular classes of graphs is the following obvious generalisation of a key result on AP graphs (see *e.g.* [1]):

Observation 3.1. *For any $x \geq 0$, if G is a graph spanned by an x -XAP graph, then G is x' -XAP for some $x' \leq x$. Conversely, adding edges to an x -XAP graph results in an x' -XAP graph for some $x' \leq x$.*

Thus, understanding the XAPness of particular graph classes may be a way to better comprehend Conjecture 2.7.

3.1. Combs and subdivided suns

Trees form surely the most studied class of graphs in the context of AP graphs, as, due to Observation 3.1, admitting an AP spanning tree is a sufficient condition for a graph to be AP (although this approach is limited, as there exist AP graphs admitting no AP spanning trees [16]). This resulted in works of the literature being dedicated to understanding the class of AP trees, such as [1, 2, 12, 16] for some of the most influential ones.

One of the most investigated classes of trees in this context is that of so-called *multipodes*, which are precisely subdivided stars. Indeed, in the very early works on the topic [1, 2, 12], the main question was about the possible maximum degree of AP trees, which was ultimately proved to be at most 4 in [2], through studying exactly properties of AP multipodes. More precisely, understanding the class of *tripodes*, *i.e.*, subdivided claws, was a key point in these aforementioned previous works.

We here focus on a generalisation of tripodes called *combs*, introduced and investigated in [16], being essentially subdivided subcubic caterpillars. More precisely, for any $l \geq 3$ and any set $S = (l_2, \dots, l_{l-1})$ of $l - 2$ positive integers l_2, \dots, l_{l-1} (possibly 0), the comb $\text{Comb}(l, S)$ denotes the graph obtained from a path $u_1 \dots u_l$ on l vertices by making u_i adjacent to one end-vertex of a new path of order l_i , for every $i \in \{2, \dots, l - 1\}$.

While combs are not always 0-XAP (as one can easily design combs of even order admitting no perfect matching; thus no $(2, \dots, 2)$ -realisation with excess 0), we prove they can actually be no worse than 1-XAP.

Theorem 3.2. *Every comb is either 0-XAP or 1-XAP.*

Proof. Let $G = \text{Comb}(l, S)$ be a comb (with main path $u_1 \dots u_l$) for some $l \geq 3$ and $S = (l_2, \dots, l_{l-1})$, and let n denote the order of G . Consider $\pi = (n_1, \dots, n_p)$, any n -partition. We explain how to build a π -realisation $(\mathcal{P}, \mathcal{C})$ with excess at most 1, whatever π is.

Let \mathcal{S} be the following ordering over the vertices of G :

- The first vertex of \mathcal{S} is u_1 .
- Assuming the last vertex added to \mathcal{S} is some u_i for $i < l$, we denote by $a_1 \dots a_{l_{i+1}}$ the (possibly empty, if $l_{i+1} = 0$) path of order l_{i+1} attached at u_{i+1} , where $d(a_1) = 1$ and $d(a_{l_{i+1}}) = 2$ (that is, a_1 and $a_{l_{i+1}}$ have degree 1 and 2, respectively). We then add $a_1, \dots, a_{l_{i+1}}, u_{i+1}$ to \mathcal{S} , following this order. In the particular case where $l_{i+1} = 0$, note that the only vertex added to \mathcal{S} here is u_{i+1} .
- We repeat the previous step for all i 's, until u_{l-1} gets added to \mathcal{S} .
- Eventually, we add u_l to \mathcal{S} .

Note that all vertices get added to \mathcal{S} . Essentially, \mathcal{S} follows $u_1 \dots u_l$, except that before adding any u_i , it first adds all vertices of the path attached at u_i , if any.

Now, to obtain $(\mathcal{P}, \mathcal{C})$, we proceed as follows. For the part V_1 of cardinality n_1 of \mathcal{P} , we pick the n_1 first vertices of \mathcal{S} . If $G[V_1]$ is connected, then the associated part C_1 in \mathcal{C} is set to \emptyset . Now, if $G[V_1]$ is not connected, then it must be because, since we followed \mathcal{S} , the last vertex of the path $u_1 \dots u_l$ added to V_1 is some u_i while the last vertices added to V_1 (at most l_{i+1}) belong to the path attached at u_{i+1} , but we did not go all the way up to adding u_{i+1} to V_1 (as otherwise V_1 would be connected). In this case, assuming $a_1 \dots a_{l_{i+1}}$ denote the consecutive vertices of the path attached at u_{i+1} (where a_1 is the leaf and $a_{l_{i+1}}$ is thus adjacent to u_{i+1}), and $j \in \{1, \dots, l_{i+1}\}$ is the largest value such that a_j is part of V_1 , we set $C_1 = \{a_{j+1}, \dots, a_{l_{i+1}}, u_{i+1}\}$. Clearly, $G[V_1 \cup C_1]$ is connected, and the only vertices of G for which the excess was increased (by 1) are exactly those in C_1 .

We repeat the exact same process to obtain the parts V_2, \dots, V_p of \mathcal{P} and C_2, \dots, C_p of \mathcal{C} , by resuming the picking process along \mathcal{S} where it stopped, and defining parts as previously for V_1 and C_1 . Once the whole process stops, $(\mathcal{P}, \mathcal{C})$ is a realisation of π in G . In terms of excess, the parts of \mathcal{C} , by arguments above, essentially induce pairwise disjoint paths. Thus, every vertex has excess at most 1. \square

By Observation 3.1, recall that Theorem 3.2 implies that every graph spanned by a comb is 0-XAP or 1-XAP. Also, regarding Conjecture 2.7, since, paths apart, combs have maximum degree $\Delta = 3$, we derive that every comb is x -XAP for some $x < \Delta - 1$.

In the context of AP graphs, another investigated class of graphs is that of *sun*s, which are subcubic graphs that can be obtained from a cycle by attaching exactly one leaf at some of the vertices. These graphs appeared in [5, 14], as studying these stood as a way to go beyond the case of trees by considering an easy class of unicyclic graphs.

It is easy to see that every sun is spanned by a subcubic caterpillar. Likewise, every subdivided sun is spanned by a comb. Therefore, we directly derive the following result (which, again, is best possible excess-wise) from Observation 3.1 and Theorem 3.2:

Corollary 3.3. *Every subdivided sun is either 0-XAP or 1-XAP.*

For subdivided suns, regarding Conjecture 2.7 we thus have, excluding cycles, that every subdivided sun with maximum degree $\Delta = 3$ is x -XAP for some $x < \Delta - 1$.

3.2. Trees

As mentioned earlier, trees form definitely the most investigated class of graphs when it comes to APness. It turns out that the class of AP trees is actually quite restricted, as AP trees have maximum degree at most 4 (see [2]). By somewhat generalising the main ideas in the proof of Theorem 3.2, we establish the following result:

Theorem 3.4. *Every tree with maximum degree Δ is x -XAP for some $x < \Delta$.*

Proof. Let T be a tree with order n and maximum degree Δ . By rooting T at any vertex r , we get a (virtual) orientation from the root r to the leaves, where each non-root vertex has a unique parent and ancestors (through iterating the parent-child relationship), and each non-leaf vertex has children and descendants (again through the parent-child relationship). For every non-leaf vertex u with $d \geq 1$ children, we suppose that there is a fixed, arbitrary ordering over these children v_1, \dots, v_d ; for any two distinct $i, j \in \{1, \dots, d\}$, we write $v_i < v_j$ if v_i appears before v_j in that ordering. For every $i \in \{1, \dots, d\}$, we also denote by T_{v_1}, \dots, T_{v_d} the subtrees of T rooted at v_1, \dots, v_d , respectively. We infer the ordering over the v_i 's to the T_{v_i} 's; that is, we write $T_{v_i} < T_{v_j}$ if $v_i < v_j$, for any two distinct $i, j \in \{1, \dots, d\}$.

Let $\pi = (n_1, \dots, n_p)$ be an n -partition. Our goal is to build a π -realisation $(\mathcal{P}, \mathcal{C})$ in T with excess at most $\Delta - 1$. For that, as in the proof of Theorem 3.2, we first need an ordering \mathcal{S} over the vertices of T . This ordering \mathcal{S} we consider is any order in which the vertices of T are traversed during the course of a Depth-First Search (DFS) Algorithm performed from r . That is, recall, \mathcal{S} first contains r , and then vertices are added to \mathcal{S} recursively by applying the same process over the vertices of T_{v_1}, \dots, T_{v_d} following that order, assuming T_{v_1}, \dots, T_{v_d} are the $d = d(r)$ subtrees at r and $T_{v_1} < \dots < T_{v_d}$.

To build $(\mathcal{P}, \mathcal{C})$, as earlier we create the parts V_1, \dots, V_p (of cardinality n_1, \dots, n_p , respectively) of \mathcal{P} one after the other, by just adding vertices to parts as following \mathcal{S} from its first vertex to its last vertex. Assume we have reached the point of adding n_i vertices to V_i , for some $i \in \{1, \dots, p\}$. If V_i is a connected part, then we can simply set $C_i = \emptyset$. Otherwise, as following \mathcal{S} , at some point, two consecutive vertices u and v get added to V_i , say u first and v second, but uv is not an edge. Then, by definition, u has an ancestor w such that u and v belong to two successive subtrees T_{v_j} and $T_{v_{j+1}}$, respectively, of w . Note that v is a child of w . In this case, we add to C_i the inner vertices (omitting u and v) of the unique path from u to v in T (where we may omit the vertices belonging to V_i). Essentially, this path goes “upwards” from u to w (through an arbitrarily long path) and then to v (through the edge wv). Note that this process may repeat several times, in case V_i missed more vertices to reach cardinality n_i ; so this process of adding vertices to C_i may repeat an arbitrary number of times. But, eventually, $T[V_i \cup C_i]$ is connected.

We claim that $(\mathcal{P}, \mathcal{C})$ has excess at most $\Delta - 1$. To see this is true, focus on any vertex u , say different from r for now. Note that, in the process above, the only moment where u may have been added to some part of \mathcal{C} , is due to some descendant v added to some V_i while the next vertex w added to V_i belongs to another part of T , and u lies on the unique path joining these two pieces of V_i . This implies u has excess 0 if u is a leaf. Now, if u is not a leaf, then it has children v_1, \dots, v_d , where $d = d(u) - 1$. Assuming T_{v_1}, \dots, T_{v_d} denote the subtrees rooted at v_1, \dots, v_d , respectively, and $v_1 < \dots < v_d$ (and thus $T_{v_1} < \dots < T_{v_d}$), recall that \mathcal{S} went through vertices of T_{v_1}, \dots, T_{v_d} in that order, exhausting all vertices of any T_{v_i} before proceeding with those of $T_{v_{i+1}}$. This implies that, for every $i \in \{1, \dots, d\}$, there is at most one vertex of T_{v_i} that caused u to be added to some part of \mathcal{C} . More precisely, either $i < d$ and u was added to some part of \mathcal{C} to connect a piece of a part of \mathcal{P} in T_{v_i} and another piece of the same part in $T_{v_{i+1}}$, or $i = d$ in which case u was added to a part of \mathcal{C} to connect a piece of a part \mathcal{P} in T_{v_d} and another piece of the same part in some subtree at some ancestor of u . After that latter case, the next vertices treated following \mathcal{S} were in other places of T , implying u has not been added to parts of \mathcal{C} after that. All in all, u may thus have been added to at most d parts of \mathcal{C} , the worst-case scenario being when it was added to connect two pieces of a same part in T_{v_i} and $T_{v_{i+1}}$ for all $i \in \{1, \dots, d-1\}$, and similarly for two pieces of a same part in T_{v_d} and another part of T accessible through the parent of u . The excess of u is thus at most $d = d(u) - 1 \leq \Delta - 1$.

Similar arguments apply when considering $u = r$, except that $d = d(r)$ here. Again, r may appear in at most $d(r) - 1$ parts of \mathcal{C} and thus has excess at most $d(r) - 1 \leq \Delta - 1$. Thus $(\mathcal{P}, \mathcal{C})$ has excess at most $\Delta - 1$, which concludes the proof. \square

Combining Observation 3.1 and Theorem 3.4, we thus deduce the following result regarding Conjecture 2.7:

Corollary 3.5. *Every graph of maximum degree Δ is x -XAP for some $x < \Delta$.*

Corollary 3.5 is about right for combs and subdivided suns, recall Theorem 3.2 and Corollary 3.3, although this might be misleading given how small the maximum degree is here. There is some gap for stars however, recall Theorem 2.4, while stars should be, in our opinion, the worst graphs (with given maximum degree) in terms of XAPness.

Note that Corollary 3.5 improves right away for every graph of maximum degree Δ admitting spanning trees with maximum degree smaller than Δ . Recall however that the problem of finding a spanning tree with minimum maximum degree of a graph is NP-complete, as there is an obvious connection with the HAMILTONIAN PATH problem.

As a complement, let us also add that we will prove, in Theorem 4.2, that determining whether a partition is realisable in a given tree is NP-complete, whatever the excess.

3.3. Balloons

We now focus on so-called *balloons*, which have been quite investigated in the context of AP graphs, in particular to understand the structure of AP graphs with connectivity 2 (see [3]). For any $k \geq 3$ and set $S = (l_1, \dots, l_k)$ of strictly positive integers¹, the k -balloon $\text{Bal}(k, S)$ is the graph obtained from k disjoint paths of order l_1, \dots, l_k , respectively, where each i th path has end-vertices u_i and v_i , by adding a new vertex y joined to all u_i 's and a new vertex z joined to all v_i 's. Essentially, $\text{Bal}(k, S)$ is the graph where two vertices y and z are joined by k disjoint paths, their numbers of vertices being as indicated by S .

Theorem 3.6. *Every k -balloon is x -XAP for some $x < \lfloor \frac{k}{2} \rfloor$.*

Proof. This follows, in particular, from Corollary 3.5, as a k -balloon G has maximum degree k and admits a spanning tree with maximum degree $\lfloor k/2 \rfloor$: assuming y and z are the two vertices of degree k of G , just remove $\lfloor k/2 \rfloor$ edges incident to y and $\lfloor k/2 \rfloor$ edges incident to z , chosen so that no two edges removed are incident to a same path joining y and z . Essentially, this means any partition can be realised in G by picking parts along the k paths joining y and z , going from y to z and *vice versa* repeatedly, and generating 1 more excess for either y or z alternatively whenever transitioning from a path to the next one, except for exactly two transitions (as y and z belong to parts too). \square

This proves, in particular, that balloons agree with Conjecture 2.7. To be more precise, let us insist on the fact that the maximum degree of a k -balloon is k .

3.4. Complete bipartite graphs

We now focus on complete bipartite graphs, where, recall, for any two integers $n, m \geq 1$, we denote by $K_{n,m}$ the complete bipartite graph with $V(K_{n,m}) = X \cup Y$, where $|X| = n$, $|Y| = m$, X and Y are independent sets, and any two vertices in X and Y are joined by an edge. We mostly consider these graphs because they stand as a class of graphs denser

¹Technically, in previous works, investigated balloons could have one of the l_i 's being equal to 0; as this case is not worth considering here, we voluntarily omit it.

than the ones we have been considering up to this point, and also because they have been quite considered in the context of AP graphs as they generalise stars (which are complete bipartite graphs of the form $K_{1,m}$).

Theorem 3.7. *Every complete bipartite graph $K_{n,m}$ with $n \leq m$ is x -XAP for some $x < \lceil \frac{m-n-1}{n} \rceil + 2$.*

Proof. We can assume $m - n \geq 2$ as otherwise $K_{n,m}$ admits a Hamiltonian path and thus is 0-XAP. Set $X = \{u_1, \dots, u_n\}$ and $Y = \{v_1, \dots, v_m\}$. We obtain the result from Corollary 3.5; indeed, consider the following spanning tree T of $K_{n,m}$. Start from the maximum path $v_1 u_1 v_2 u_2 \dots v_n u_n v_{n+1}$ of $K_{n,m}$ as T . At this point, all vertices have degree at most 2 in T , and only vertices in $\{v_{n+2}, \dots, v_m\}$ do not belong to T yet. The number of such vertices is $m - n - 1$. We then repeatedly modify T as follows: until all vertices in $\{v_{n+2}, \dots, v_m\}$ get connected to T , we keep on considering each u of u_1, \dots, u_n one by one in order, a $v \in Y$ that is not part of T yet, and add the edge uv to T . This increases the degrees of the u_i 's in T by at most $\lceil \frac{m-n-1}{n} \rceil$, so their degrees in T in total is at most $\lceil \frac{m-n-1}{n} \rceil + 2$. Meanwhile, vertices in Y have degree at most 2 in T . \square

Note that, when $n \leq m$, the maximum degree Δ of $K_{n,m}$ is m . Thus, the bound of Theorem 3.7, when injecting Δ , rewrites to $\lceil \frac{\Delta-n-1}{n} \rceil + 2$. The worst case is then when $n = 1$ (stars), where the bound reduces to Δ . This is off by a factor 2, recall Theorem 2.4.

3.5. Split graphs

We finish off by considering *split graphs*, which are graphs where vertices can be partitioned into $K \cup S$, where K is a clique and S is a stable set. We consider these graphs because they stand as another class of somewhat denser graphs, and because they have also been the subject of interesting works and results in the context of AP graphs [10].

Theorem 3.8. *Every split graph with stable set of size s is x -XAP for some $x < \lceil \frac{s}{2} \rceil + 1$.*

Proof. Let G be a split graph with clique K and stable set S , where $s = |S|$. To realise any partition in G , one strategy consists in picking parts in K first, which creates connected parts only. Once all vertices of K have been added to parts, it remains to cover parts of S only. At this point, we can see the situation as the case of stars, by contracting all vertices of K to a single vertex. Then the rest of the realisation can be obtained following arguments in the proof of Theorem 2.4, where, in the worst-case scenario, we generate 1 more excess for some vertex of K per pair of vertices of S . Then the maximum excess we generate is at most $\lceil s/2 \rceil$ (cases where the remaining parts to pick have cardinality 2). \square

4. Algorithmic results

In this section, we provide several results regarding the complexity of problems related to the XAPness of graphs. When it comes to the original notions behind AP graphs, perhaps the two most natural problems to consider read as follows:

REALISATION

Instance: A graph G of order n , and an n -partition $\pi = (n_1, \dots, n_p)$.

Question: Does G admit a π -realisation (V_1, \dots, V_p) ?

AP

Instance: A graph G .

Question: Is G AP?

For clarity, let us insist on the fact that, in REALISATION, we wonder about realisations with excess 0 (*i.e.*, with connected parts only), while, in AP, we wonder about 0-XAPness. To remain brief, let us mention that the former problem is well known to be NP-complete even under various constraints on either G and/or π , while the status of the latter problem is much more open (it is not even clear whether this problem lies in NP or co-NP). We refer the reader to *e.g.* [9] for more information regarding latest works on the topic.

Below, we focus on two types of problems, which we believe are the most natural ones to consider in our context. Namely, we focus first on the complexity of finding connected partitions with given excess, before focusing on finding connectors with given excess.

4.1. Realising partitions with given excess

In our case, direct analogues to the REALISATION problem read as follows:

REALISATION WITH EXCESS

Instance: A graph G of order n , an n -partition π , and an integer $x \geq 0$.

Question: Does G admit a π -realisation with excess x ?

REALISATION WITH EXCESS x

Instance: A graph G of order n , and an n -partition π .

Question: Does G admit a π -realisation with excess x ?

It should be clear that both problems lie in the complexity class NP. Note that for the first problem, REALISATION WITH EXCESS, its NP-hardness follows directly from the fact that REALISATION is NP-hard (as any instance of REALISATION is nothing but an instance of REALISATION WITH EXCESS with additional input $x = 0$). Likewise, regarding the second problem, REALISATION WITH EXCESS x , the fact that REALISATION is NP-hard implies REALISATION WITH EXCESS 0 is an NP-hard problem as well.

In what follows, we generalise this latter fact by proving that REALISATION WITH EXCESS x is also NP-hard for all $x \geq 1$. Our proof relies on the following key fact:

Observation 4.1. *Let G be a graph of order n , and π be an n -partition where all elements have value at least 2. Assume further that the largest element of π has value m . Then, if G has a vertex u adjacent to k leaves for some $k \geq m$, then, by any realisation $(\mathcal{P}, \mathcal{C})$ of π in G , we have $x(u) \geq \left\lceil \frac{k-(m-1)}{m} \right\rceil$, and thus $(\mathcal{P}, \mathcal{C})$ has excess at least $\left\lceil \frac{k-(m-1)}{m} \right\rceil$.*

Proof. Because no element of π has value 1, any part containing a leaf adjacent to u must either contain u (connected part) or contribute 1 to the excess of u (otherwise). So the best-case scenario (minimising $x(u)$) is when a part of cardinality m of $(\mathcal{P}, \mathcal{C})$ contains u and $m - 1$ leaves adjacent to u , while the remaining $k - (m - 1)$ leaves are covered by (non-connected) parts. Because m is the maximum element value in π , at best we can cover, through $(\mathcal{P}, \mathcal{C})$, these $k - (m - 1)$ remaining leaves with $\left\lceil \frac{k-(m-1)}{m} \right\rceil$ parts of cardinality m , thereby contributing this much to $x(u)$. The claim thus holds. \square

Observation 4.1 can also be of some use to bound, for a given graph G , the value $x \geq 0$ such that G is x -XAP. This is particularly appropriate for graphs having vertices adjacent to many leaves, such as some trees (our proof of Theorem 2.4 being some illustration).

We now prove our first complexity result:

Theorem 4.2. *For every $x \geq 0$, the problem REALISATION WITH EXCESS x is NP-hard. This remains true when restricted to subdivided stars.*

Proof. If $x = 0$, then the problem is equivalent to REALISATION, which was indeed proved to be NP-hard when restricted to subdivided stars in [9]. Thus, we can assume $x \geq 1$.

The proof is by reduction from the REALISATION problem restricted to linear forests (*i.e.*, forests of paths) and partitions where all elements have value at least 2 (this restriction is indeed NP-hard, see [9]). From a forest of paths G of order n and an n -partition $\pi = (n_1, \dots, n_p)$ where all elements have value at least 2, we construct, in polynomial time, a subdivided star G' of order n' and an n' -partition $\pi' = (n'_1, \dots, n'_p)$ such that G admits a π -realisation (with excess 0) if and only if G' admits a π' -realisation with excess x .

Let us denote by m the biggest element value appearing in π , and set $M = m + 1$. For each i th of the t paths in the linear forest (forming G), let u_i denote any one of the two end-vertices. Start from G' being G , add a vertex v , make v adjacent to u_1, \dots, u_t , and, setting $q = m + xM$, add q leaves w_1, \dots, w_q adjacent to v . Regarding π' , just start from π , and add $x + 1$ elements with value M . Note that the number of vertices we have added to obtain G' from G is $q + 1 = m + xM + 1 = M(x + 1)$, while the sum of the element values added to π to obtain π' is $M(x + 1)$; thus, π' is an n' -partition, where $n' = |V(G')|$. Clearly, the construction of G' and π' is achieved in polynomial time, and G' is a subdivided star.

To see that the claimed equivalence holds, consider any hypothetical π' -realisation $(\mathcal{P}', \mathcal{C}')$ in G' . Assume \mathcal{C}' is minimal. Because the largest element value in π' is M , and v is adjacent to q leaves, by Observation 4.1 we deduce that

$$x(v) \geq \left\lfloor \frac{q - (M - 1)}{M} \right\rfloor = \left\lfloor \frac{M - 1 + xM - (M - 1)}{M} \right\rfloor = \left\lfloor \frac{xM}{M} \right\rfloor = x.$$

In particular, by arguments in the proof of Observation 4.1, the only way for v to have excess x by $(\mathcal{P}', \mathcal{C}')$ is to have one connected part of \mathcal{P}' containing v and $M - 1$ of its adjacent leaves, x non-connected parts each containing M distinct leaves, and having the rest of the parts being connected and covering the vertices originating from the linear forest G . In particular, the cardinalities of these connected parts are exactly the values in π ; in other words, they form a π -realisation (with excess 0) of G . Thus, one can deduce a π -realisation with excess 0 in G from a π' -realisation with excess x in G' , and *vice versa*. \square

Theorem 4.2 has notable consequences. In particular, it implies that the problem of finding a realisation is not FPT when parameterised by the excess parameter.

4.2. Finding connectors with given excess

We now turn to a different problem, being more specific to the new notions we have introduced. Namely, we focus on:

CONNECTOR WITH EXCESS x

Instance: A graph G of order n , an n -partition $\pi = (n_1, \dots, n_p)$, and a partition $\mathcal{P} = (V_1, \dots, V_p)$ of $V(G)$ where $|V_i| = n_i$ for all $i \in \{1, \dots, p\}$.

Question: Does G admit a \mathcal{P} -connector \mathcal{C} such that $(\mathcal{P}, \mathcal{C})$ has excess x ?

Clearly, CONNECTOR WITH EXCESS x lies in NP for every x . Note also that CONNECTOR WITH EXCESS 0 can be solved in polynomial time, since an instance is here positive if and only if \mathcal{P} is a π -realisation (thus with excess 0). On the other hand, we prove that CONNECTOR WITH EXCESS x is NP-hard for $x \geq 1$. We prove this formally for $x = 1$ below, before explaining afterwards how this can be generalised.

Theorem 4.3. *The problem REALISATION WITH EXCESS 1 is NP-hard.*

Proof. The proof is by reduction from the DISJOINT PATHS problem, which is well known to be NP-hard [15]. Recall that, in an instance of the DISJOINT PATHS problem, a graph H is given together with a positive integer $q \geq 1$ and $2q$ pairwise-distinct vertices s_1, \dots, s_q and t_1, \dots, t_q of G , and the question is whether there exist q vertex-disjoint paths of H joining each pair $\{s_1, t_1\}, \dots, \{s_q, t_q\}$. From H , q , and $s_1, \dots, s_q, t_1, \dots, t_q$, we construct, in polynomial time, an equivalent instance of REALISATION WITH EXCESS 1.

We start by building the graph of the instance, called G . Start initially from G being H . Then, for every s_i , add a new vertex s'_i joined to s_i only, and similarly, for every t_i , add a new vertex t'_i joined to t_i only. Let n denote the order of G . As π , we consider the n -partition containing q occurrences of the value 2, and $n - 2q$ occurrences of the value 1, while, as \mathcal{P} , we consider the partition of $V(G)$ where $\{s'_i, t'_i\}$ is a (non-connected) part of cardinality 2 for every $i \in \{1, \dots, q\}$, and $\{u\}$ is a part of cardinality 1 for every vertex u of H . Note that the cardinalities of the parts in \mathcal{P} indeed meet the values indicated by π . This instance of REALISATION WITH EXCESS 1 is built in polynomial time.

To see that the equivalence holds, consider the following arguments. Note that the only parts of \mathcal{P} that are not connected are all parts of the form $\{s'_i, t'_i\}$. Then, any \mathcal{P} -connector \mathcal{C} must have, for every $i \in \{1, \dots, q\}$, a part C_i such that $G[\{s'_i, t'_i\} \cup C_i]$ is connected. Due to the structure of G , this C_i must contain a path from s_i to t_i through vertices of H only. Also, two parts C_i and C_j of \mathcal{C} cannot have a vertex u in common, as otherwise we would have $x(u) > 1$. This implies that the C_i 's contain vertex-disjoint paths through H joining the s_i 's and t_i 's. So if $(\mathcal{P}, \mathcal{C})$ is a π -realisation with excess at most 1, then we can deduce a solution to the original instance of the DISJOINT PATHS problem. Since the other direction clearly holds as well, the equivalence holds. \square

Generalising Theorem 4.3 can be done through the following observation.

Observation 4.4. *Let G be a graph of order n , $\pi = (n_1, \dots, n_p)$ be an n -partition, and $\mathcal{P} = (V_1, \dots, V_p)$ be a partition of $V(G)$ where $|V_i| = n_i$ for all $i \in \{1, \dots, p\}$. Let G' be the graph of order $n' = 3n$ obtained from G by adding two leaves u' and u'' adjacent to every vertex u , $\pi' = (n_1, \dots, n_{p+n})$ be the n' -partition obtained from π by adding n occurrences of the value 2, and $\mathcal{P}' = (V_1, \dots, V_{p+n})$ be the partition of $V(G')$ obtained from \mathcal{P} by adding the part $\{u', u''\}$ for every vertex u of G . If G' admits a \mathcal{P}' -connector \mathcal{C}' such that $(\mathcal{P}', \mathcal{C}')$ has excess at most x , then the restriction of \mathcal{C}' to G is a \mathcal{P} -connector with excess at most $x - 1$.*

Proof. Since $\{u', u''\}$ is a part of \mathcal{P}' for every vertex u of G , we have that $\{u\}$ must be contained in a part of \mathcal{C}' , thereby contributing 1 to the excess of u . Thus, in G' , all original vertices u of G have their excess altered by exactly 1 due to the pairs of vertices $\{u', u''\}$ we have added to build G' . Then the restriction of \mathcal{C}' to G must have excess at most $x - 1$. \square

Theorem 4.5. *For every $x \geq 1$, the problem CONNECTOR WITH EXCESS x is NP-hard.*

Proof. It suffices to perform the reduction in the proof of Theorem 4.3, with additionally performing $x - 1$ times the modifications described in Observation 4.4. This artificially increases by $x - 1$ the excess of every vertex, implying that, omitting these modifications, we are still left with the task of finding disjoint paths/a connector with excess at most 1. Clearly, these modifications can still be performed in polynomial time. \square

It is worth mentioning that the DISJOINT PATHS problem becomes polynomial-time solvable when q (the number of pairs of vertices to be joined) is a fixed constant not part of the input [17]. From this and arguments developed earlier, we deduce that any algorithm solving the DISJOINT PATHS problem in polynomial time (such as when q is fixed and not

part of the input) could be invoked to solve, in polynomial time, some particular instances of CONNECTOR WITH EXCESS x .

5. Conclusion and perspectives

Motivated by practical concerns, in this work we generalised the notions of connected partitions and AP graphs by introducing the notion of vertex excess, where, essentially, we allow parts of realisations to be not connected, as long as we provide outside vertices to support connectivity. As explained in the introductory section, this justifies from a practical point of view. As seen through Observation 2.1, allowing excess expectedly makes graphs partitionable into arbitrarily many connected graphs of arbitrary orders. However, still from a practical point of view, it is preferable to minimise excess as much as possible. This led us to introduce the notion of x -XAP graphs, which we have studied throughout this work, both from the structural and algorithmic points of view. In particular, an interesting point was wondering how previous results on AP graphs generalise to XAP graphs.

The main question we came up with is that of determining the value x such that a given graph is x -XAP. This led us to raise Conjecture 2.7, which we have studied throughout Section 3. From our experience with the field, and given the results we were able to establish, if we had to make a guess, then we would estimate that the best function f we can establish is about $\Delta/2$. This would match the results we proved, and also would be along the case of stars (Theorem 2.4) which we believe is the most constrained one.

Following our investigations, we believe a number of directions could be worth considering in further research. In particular:

- In Section 2, we wondered about properties of \mathcal{P} -connectors, and, in particular, on hypothetical contexts and graphs where, for some realisation with given excess, all connectors have particular properties. For instance, around Observation 2.5, we discussed situations where connectors might have to induce graphs with large diameter, which was not that obvious to spot later on given the restricted classes of graphs we focused on in Section 2.5. In the line of Observation 2.5, we would be interested in knowing whether there are situations where connectors must induce graphs with counterintuitive properties, such as being quite disconnected, or containing certain structures such as cycles, cliques, etc. We believe this would be a first step towards understanding better the new notions we have introduced.
- As mentioned above, the main question we raised is Conjecture 2.7. We would be interested in any progress towards it. In particular, our best result regarding it is Corollary 3.5, while, as mentioned above, we expect it should be possible to establish a result that is better by about a factor 2.
- In this line, a first step could be to consider Conjecture 2.7 for more classes of graphs. An obvious one we could consider is that of multipodes mentioned in Section 3, which was the key behind understanding structural properties of AP trees [1, 2, 12, 16]. In turn, understanding the XAPness of multipodes could certainly allow to generalise Theorem 3.2 and Corollary 3.3 to generalisations of combs and subdivided suns with arbitrarily large maximum degree.
- The case of trees is of particular interest, given both that we established our best result, Corollary 3.5, from it, and that trees have been, in the context of AP graphs, the most investigated class of graphs to date.

In particular, a major result regarding AP trees established in [2] states that AP trees have maximum degree at most 4 and that this is tight. In our terminology, this translates to stating that 0-XAP trees have maximum degree at most 4. Naturally, we would be interested in knowing the maximum value that the maximum degree of an x -XAP tree can attain, for any fixed value of x . In particular, we wonder whether establishing such a result could be done by just generalising the arguments from [2], which mostly consist in studying the APness of multipodes.

As a first remark regarding this matter, we note that, as x increases, x -XAP multipodes can have larger and larger maximum degree. In particular, it is not hard to check that from an AP multipode with maximum degree 4 (such exist, see [2]), one obtains an x -XAP multipode with maximum degree $2x + 4$ when adding $2x$ leaves adjacent to the center vertex.

- Regarding complexity results, just as for APness, it is largely unclear what is the status of the problem of determining whether a given graph is x -XAP for a fixed value of x . One could estimate that this problem is at least NP-hard, but, despite several works on the topic (see [2, 6, 9, 10]), this is not known to hold even for $x = 0$. Actually, it is not even clear whether the problem lies in NP or co-NP at all, though previous results seem to tend towards the fact that the problem lies in NP. Although we do not expect this problem becomes any easier when adding the excess requirement, still be believe it is worth recalling this interesting question.

Finally, the way we defined the notions of connector and excess is only one possible way to go, and one could introduce and investigate more or less distant variations. For instance, instead of minimising the maximum number of times a vertex appears in the parts of a connector, one may wonder about minimising the total number of (distinct or not) vertices appearing in a connector. In particular, we wonder whether such a slight variation over our definitions changes the problem a lot. Likewise, as mentioned in the introductory section, several variants of AP graphs have been introduced in literature, and one could legitimately wonder about adding the excess requirement there as well.

References

- [1] D. Barth, O. Baudon, J. Puech. Decomposable trees: a polynomial algorithm for tripodes. *Discrete Applied Mathematics*, 119(3):205–216, 2002.
- [2] D. Barth, H. Fournier. A degree bound on decomposable trees. *Discrete Mathematics*, 306(5):469–477, 2006.
- [3] O. Baudon, J. Bensmail, F. Foucaud, M. Piłśniak. Structural properties of recursively partitionable graphs with connectivity 2. *Discussiones Mathematicae Graph Theory*, 37:89–115, 2017.
- [4] O. Baudon, J. Bensmail, J. Przybyło, M. Woźniak. Partitioning powers of traceable or Hamiltonian graphs. *Theoretical Computer Science*, 520:133–137, 2014.
- [5] O. Baudon, F. Gilbert, M. Woźniak. Recursively arbitrarily vertex-decomposable graphs. *Opuscula Mathematica*, 32(4):689–706, 2012.
- [6] J. Bensmail. On the complexity of partitioning a graph into a few connected subgraphs. *Journal of Combinatorial Optimization*, 30(1):174–187, 2015.

- [7] J. Bensmail. A σ_3 condition for arbitrarily partitionable graphs. *Discussiones Mathematicae Graph Theory*, 44(2):755–776, 2024.
- [8] J. Bensmail. Toughness Properties of Arbitrarily Partitionable Graphs. *Discrete Applied Mathematics*, 378:316–328, 2026.
- [9] J. Bensmail, B. Li. More Aspects of Arbitrarily Partitionable Graphs. *Discussiones Mathematicae Graph Theory*, 42(4):1237–1261, 2022.
- [10] H. Broersma, D. Kratsch, G.J. Woeginger. Fully decomposable split graphs. *European Journal of Combinatorics*, 34(3):567–575, 2013.
- [11] M. Horňák, Z. Tuza, M. Woźniak. On-line arbitrarily vertex decomposable trees. *Discrete Applied Mathematics*, 155:1420–1429, 2007.
- [12] M. Horňák, M. Woźniak. On arbitrarily vertex decomposable trees. *Discrete Mathematics*, 308(7):1268–1281, 2008.
- [13] R. Kalinowski, M. Pilśniak, I. Schiermeyer, M. Woźniak. Dense arbitrarily partitionable graphs. *Discussiones Mathematicae Graph Theory*, 36:5–22, 2016.
- [14] R. Kalinowski, M. Pilśniak, M. Woźniak, I. A. Ziolo. On-line arbitrarily vertex decomposable suns. *Discrete Mathematics*, 309(22):6328–6336, 2009.
- [15] R. M. Karp. On the computational complexity of combinatorial problems. *Networks*, 5:45–68, 1975.
- [16] R. Ravaux. *Graphes arbitrairement partitionnables : propriétés structurelles et algorithmiques*. Ph.D. thesis (in French), Université Versailles Saint-Quentin, 2009.
- [17] N. Robertson, P. D. Seymour. Graph minors. XIII. The disjoint paths problem. *Journal of Combinatorial Theory, Series B*, 63(1):65–110, 1995.