



**HAL**  
open science

# On the complexity of partitioning a graph into a few connected subgraphs

Julien Bensmail

► **To cite this version:**

Julien Bensmail. On the complexity of partitioning a graph into a few connected subgraphs. 2012. hal-00762612v1

**HAL Id: hal-00762612**

**<https://hal.science/hal-00762612v1>**

Preprint submitted on 7 Dec 2012 (v1), last revised 18 Apr 2013 (v2)

**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.

# On the complexity of partitioning a graph into a few connected subgraphs

Julien Bensmail

the date of receipt and acceptance should be inserted later

**Abstract** Given a graph  $G$ , a sequence  $\tau = (n_1, \dots, n_p)$  of positive integers summing up to  $|V(G)|$  is said to be *realizable in  $G$*  if there exists a *realization of  $\tau$  in  $G$* , that is a partition  $(V_1, \dots, V_p)$  of  $V(G)$  such that each  $V_i$  induces a connected subgraph of  $G$  on  $n_i$  vertices. We study the computational complexity of some decision problems related to the previous definition. In particular, we show that deciding whether a graph can be partitioned into several connected subgraphs is an NP-complete problem even when a constant number  $c$  of parts with  $c \geq 2$  is requested, or a constant number of vertex-membership constraints must be satisfied. We additionally introduce a  $\Pi_2^p$ -complete graph partition problem asking whether some partial realizations of  $\tau$  in  $G$  can be extended to obtain whole realizations of  $\tau$  in  $G$ .

**Keywords** arbitrarily partitionable graphs · partition into connected subgraphs · complexity · polynomial hierarchy

## 1 Introduction

Let  $G$  be a connected graph. A sequence  $\tau = (n_1, \dots, n_p)$  of positive integers is *admissible for  $G$*  if  $\sum_{i=1}^p n_i = |V(G)|$ . Such a sequence  $\tau$  is additionally said to be *realizable in  $G$*  if there exists a *realization of  $\tau$  in  $G$* , that is a partition  $(V_1, \dots, V_p)$  of  $V(G)$  such that  $V_i$  induces a connected subgraph of  $G$  on  $n_i$  vertices for every  $i \in \{1, \dots, p\}$ .

The problem of finding a realization of a given sequence in a graph gained a lot of attention since the result, proved independently by Gyri and Lovsz, which states that every sequence with size at most  $k \geq 1$ , that is with at most

---

J. Bensmail  
Univ. Bordeaux, LaBRI, UMR 5800, F-33400 Talence, France  
CNRS, LaBRI, UMR 5800, F-33400 Talence, France  
E-mail: julien.bensmail@labri.fr  
Tel.: +33-(0)5-40-00-35-17, Fax: +33-(0)5-40-00-66-69

$k$  elements, admissible for a  $k$ -connected graph  $G$  is also realizable in  $G$  [7,8]. Since then, several graph properties based on the definition above have been investigated.

For instance, we say that  $G$  is *arbitrarily partitionable* (AP for short) if every sequence admissible for  $G$  is also realizable in  $G$ . For the sake of the upcoming next definitions, let us now consider a  $k$ -prescription of  $G$ , that is a sequence of  $k$  pairwise distinct vertices  $(v_1, \dots, v_k)$  of  $G$  with  $k \in \{1, \dots, p\}$ , where  $p$  is the size of  $\tau$ , a sequence that is admissible for  $G$ . We say that  $\tau$  is *realizable in  $G$  under  $(v_1, \dots, v_k)$*  if there exists a realization  $(V_1, \dots, V_p)$  of  $\tau$  in  $G$  such that  $v_i \in V_i$  for every  $i \in \{1, \dots, k\}$ . In other words, a  $k$ -prescription is a set of vertices that were chosen to belong to the first  $k$  parts of a realization of  $\tau$  in  $G$ . Notice that, in our terminology, the  $k$  part sizes associated with these  $k$  prescribed vertices are the first  $k$  ones of the sequence. Finally, the graph  $G$  is said to be *arbitrarily partitionable under  $k$  prescriptions* (AP+ $k$  for short) if every admissible sequence for  $G$  with at least  $k$  elements is realizable in  $G$  under every  $k$ -prescription of  $G$ . All these definitions were introduced to deal with a practical problem of resource sharing among an arbitrary number of users [1,3].

In this paper, we consider the computational complexity of some decision problems derived from the definitions above. Our main result, in Section 2, states that the problem of deciding whether a sequence is realizable in a graph is NP-complete even when restricted to sequences with a constant number  $c$  of elements such that  $c \geq 2$ . This result extends a result of [6] stating that checking whether the sequence  $(\lfloor \frac{|V(G)|}{2} \rfloor, \lfloor \frac{|V(G)|}{2} \rfloor)$  is realizable in a graph  $G$  with even order is NP-complete in general. We then prove in Section 3 that the same decision problem augmented with a prescription has the same complexity no matter what are the size of the prescription or the part sizes associated with the prescribed vertices. The complexity of the problems of deciding whether a graph is AP or AP+ $k$  is then discussed in Section 4. We locate these two problems in the  $\Pi_2^P$  complexity class of the polynomial hierarchy and explain why we cannot modify our previous reductions to prove that these problems are  $\Pi_2^P$ -complete. We then introduce a  $\Pi_2^P$ -complete graph partition problem in the same section.

## 2 Complexity of partitioning a graph into a few connected subgraphs

In this section, we focus on the following decision problem.

REALIZABLE SEQUENCE - REALSEQ

Instance: A graph  $G$  and a sequence  $\tau$  admissible for  $G$ .

Question: Is  $\tau$  realizable in  $G$ ?

It is already known that REALSEQ is computationally hard, even under strong restrictions on  $G$  or  $\tau$ . In particular, this problem remains NP-complete even when  $G$  is a tree with maximum degree 3, or  $\tau = (k, \dots, k)$  has only one integer value  $k \geq 3$  that divides  $|V(G)|$  (see [2] and [4]).

Assuming that the size of  $\tau$  is constant, we introduce the following refinement of REALSEQ.

REALIZABLE SEQUENCE WITH SIZE  $k$  -  $k$ -REALSEQ

Instance: A graph  $G$  and a sequence  $\tau$  admissible for  $G$  with size  $k$ .

Question: Is  $\tau$  realizable in  $G$ ?

The properties of the NP-completeness of REALSEQ mentioned above result from some characteristics of the different reductions used. However, these reductions do not involve the existence of a constant threshold  $c \geq 1$  such that:

- $k$ -REALSEQ is in P for every  $k \leq c - 1$ ;
- $k$ -REALSEQ is NP-complete otherwise.

Since partitioning a graph into one single connected subgraph is possible if and only if it is connected, we have  $c \geq 2$ . In what follows, we prove that  $c = 2$ , that is that  $k$ -REALSEQ is NP-complete for every  $k \geq 2$ . Our reduction is from the following variant of 3SAT.

1-IN-3 SAT

Instance: A 3CNF formula  $F$  over variables  $\{x_1, \dots, x_n\}$  and clauses  $\{C_1, \dots, C_m\}$ .

Question: Is there a 1-in-3 truth assignment of the variables of  $F$ , that is a truth assignment such that each clause of  $F$  has exactly one true literal?

Notice that we can suppose that every possible literal appears in  $F$ . Indeed, if  $x_i$  does not appear in any clause of  $F$ , then the 3CNF formula  $F' = F \wedge (x_i \vee \bar{x}_i \vee \bar{x}_{n+1}) \wedge (x_{n+1} \vee \bar{x}_{n+1} \vee x_{n+1})$ , where  $x_{n+1}$  is a new variable, admits a 1-in-3 truth assignment of its variables if and only if  $F$  admits one too. Since there are  $2n$  literals related to the variables of  $F$ , a formula equivalent to  $F$  that contains every possible literal over its variables can be obtained from  $F$  in polynomial time.

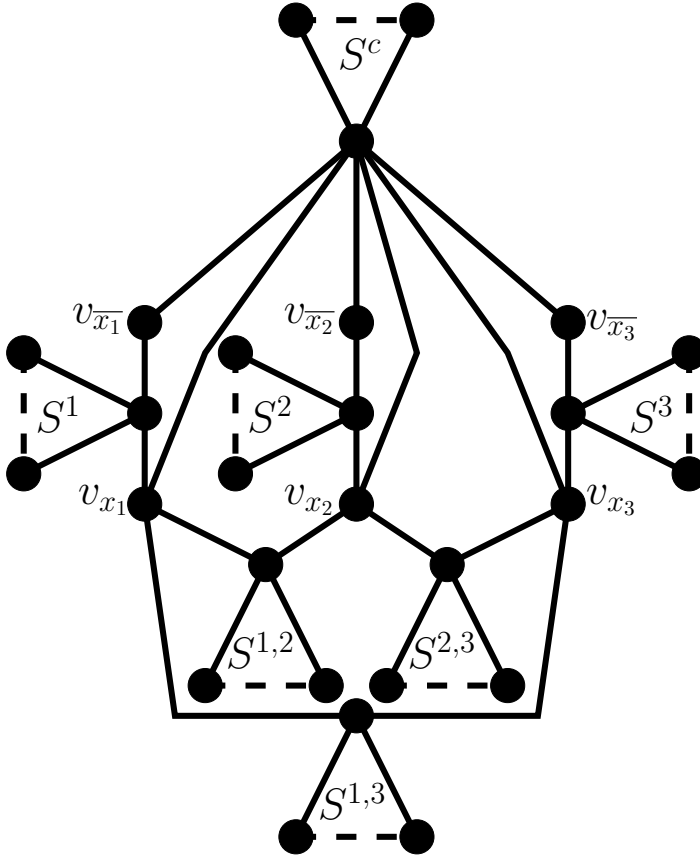
Our proof of the NP-completeness of  $k$ -REALSEQ for every  $k \geq 2$  reads as follows. We first show in Theorem 1 that 2-REALSEQ is NP-complete by reduction from 1-IN-3 SAT. We then explain, in Theorem 2, how to modify our reduction from 1-IN-3 SAT to 2-REALSEQ so that we get a reduction from 1-IN-3 SAT to  $k$ -REALSEQ for any  $k \geq 3$ .

**Theorem 1** 2-REALSEQ is NP-complete.

*Proof* First of all, REALSEQ is clearly in NP. One can indeed design an algorithm that takes the graph  $G$ , the sequence  $\tau$  and a realization  $R$  of  $\tau$  in  $G$  as input and checks whether  $R$  is correct. More precisely, such an algorithm has to check that  $R$  is a partition of  $V(G)$ , that the parts of  $R$  have the correct sizes regarding  $\tau$ , and that the subgraphs of  $G$  induced by  $R$  are connected. This verification can be done in polynomial time no matter what is the size of  $\tau$ .

We now prove that 2-REALSEQ is NP-complete by reduction from 1-IN-3 SAT. For a given formula  $F$  over variables  $\{x_1, \dots, x_n\}$  and clauses  $\{C_1, \dots, C_m\}$ , we construct a graph  $G_F$  and a sequence  $\tau$  such that  $F$  is satisfiable in a 1-in-3 way if and only if  $\tau$  is realizable in  $G_F$ . Our reduction is performed in such a way that  $\tau$  has only two elements.

The graph  $G_F$  is composed of two main vertex-disjoint subgraphs. The first one is the *clause subgraph*. Each literal  $\ell_i$  of  $F$  is associated with a *literal vertex*  $v_{\ell_i}$  in the clause subgraph. For each pair of literals  $\ell_i$  and  $\bar{\ell}_i$  of  $F$ , we then link the literal vertices  $v_{\ell_i}$  and  $v_{\bar{\ell}_i}$  to the root vertex of a star  $S^i$  with  $n$  vertices of degree 1. Two literal vertices  $v_{\ell_i}$  and  $v_{\ell_j}$  such that  $\ell_j \neq \bar{\ell}_i$  are similarly linked to the root vertex of a star  $S^{i,j}$  with  $n$  vertices of degree 1 if they both appear in a same clause of  $F$ . We finally add a *control star*  $S^c$  with  $n$  vertices of degree 1 to the clause subgraph of  $G_F$  and link its root to every literal vertex so that the clause subgraph is connected. The construction so far is detailed in Figure 1.



**Fig. 1** Resulting subgraph in the clause subgraph of  $G_F$  for a clause  $C_1 = (x_1 \vee x_2 \vee x_3)$  of  $F$

Let  $n_2$  be the number of vertices of the clause subgraph. Then we have

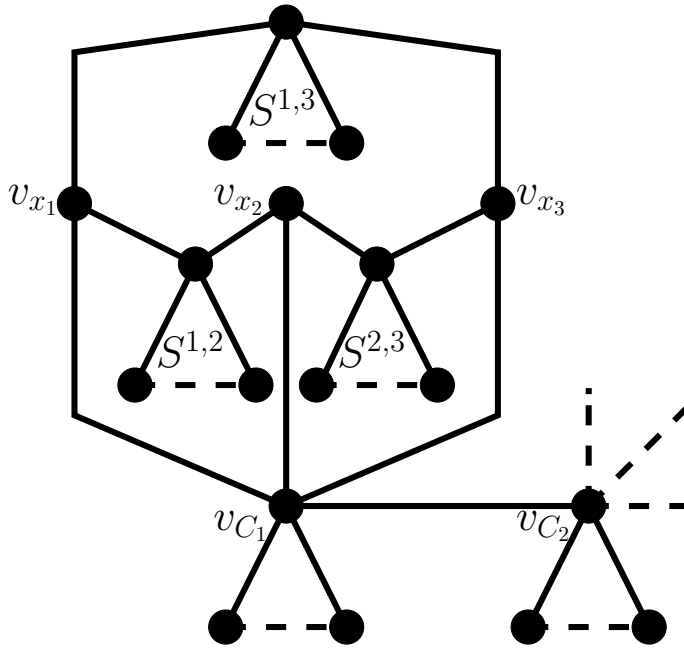
$$n_2 \leq 2n + n(n+1) + 3m(n+1) + n+1$$

since there are exactly  $2n$  literals and  $n$  pairs of literals of the form  $(\ell_i, \bar{\ell}_i)$  in  $F$ , all the clauses of  $F$  can have distinct literals, and the control star  $S^c$  has exactly  $n$  vertices of degree 1.

The second subgraph of  $G_F$  is the *base subgraph*. With each clause  $C_i$  in  $F$  we associate a *clause vertex*  $v_{C_i}$  in the base subgraph that is linked to  $n_2 - n$  vertices of degree 1. For each  $i \in \{1, \dots, m - 1\}$ , we finally add  $v_{C_i} v_{C_{i+1}}$  to  $E(G_F)$  so that the clause vertices induce a path in  $G_F$ . If we denote by  $n_1$  the number of vertices of the base subgraph of  $G_F$ , then we have

$$n_1 = m(n_2 - n + 1).$$

We end up the construction of  $G_F$  by adding some connection between the base and clause subgraphs of  $G_F$  as follows: for each clause  $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$  in  $F$ , we add  $v_{C_i} v_{\ell_{i_1}}$ ,  $v_{C_i} v_{\ell_{i_2}}$  and  $v_{C_i} v_{\ell_{i_3}}$  to  $E(G_F)$ . See Figure 2 for an illustration of this connection.



**Fig. 2** Connection between the base and clause subgraphs of  $G_F$  for a clause  $C_1 = (x_1 \vee x_2 \vee x_3)$  of  $F$

The number of vertices of  $G_F$  is  $n_1 + n_2$ . Thus, the construction of  $G_F$  is performed in polynomial time regarding the size of  $F$ . Consider now the sequence  $\tau = (n_1 + n, n_2 - n)$  admissible for  $G_F$ . Since  $\tau$  has only two elements that are strictly greater than 1, any part  $U$  from a realization  $R$  of  $\tau$  in  $G_F$  that covers the root vertex of any star subgraph in  $G_F$  must also contain all the vertices of degree 1 attached to it. Indeed, if this is not the case, then the

graph  $G_F - U$  contains at least two connected components and, thus, the part of  $R$  different from  $U$  cannot induce a connected subgraph of  $G_F$ .

For this reason, observe that, because of all the induced stars  $S_{n_2-n+1}$  in the base subgraph of  $G_F$ , this latter subgraph must be covered by the part  $V_1$  with size  $n_1 + n$  of a realization  $(V_1, V_2)$  of  $\tau$  in  $G_F$ . Starting from this, we then have to add  $n$  additional vertices from the clause subgraph of  $G_F$  to  $V_1$ . Thanks to an argument similar to the one above, we can only pick up some literal vertices of  $G_F$  since picking up any other of its vertices would disconnect  $G_F$  into too many small components. According to our construction, we cannot also add to  $V_1$  two literal vertices  $v_{\ell_i}$  and  $v_{\ell_j}$  such that  $\ell_i$  and  $\ell_j$  are a variable of  $F$  and its negation, or appear in a same clause of  $F$ , since otherwise this would once again make the subgraph  $G_F - V_1$  disconnected.

We then claim that we can deduce a 1-in-3 truth assignment of the variables of  $F$  from a realization  $R = (V_1, V_2)$  of  $\tau$  in  $G_F$  and vice-versa. If  $R$  is a correct realization of  $\tau$  in  $G_F$ , then there are exactly  $n$  literal vertices  $v_{\ell_{i_1}}, \dots, v_{\ell_{i_n}}$  from the clause subgraph of  $G_F$  that belong to  $V_1$ . Since  $G_F[V_2]$  is connected, setting the literals  $\ell_{i_1}, \dots, \ell_{i_n}$  true makes  $F$  evaluated true in a 1-in-3 way. In particular, no pair of these literals is a variable of  $F$  and its negation, or appears in a same clause of  $F$ . Conversely, if  $F$  is satisfiable in a 1-in-3 way, then let  $\phi : \{\ell_1, \dots, \ell_{2n}\} \rightarrow \{0, 1\}$  be a satisfying truth assignment of its literals. Then observe that  $(V_1, V_2)$ , where

- $V_1$  contains all the vertices from the base subgraph of  $G_F$  and every literal vertex  $v_{\ell_i}$  from the clause subgraph of  $G_F$  such that  $\phi(\ell_i) = 1$ ,
- $V_2 = V(G_F) - V_1$ ,

is a correct realization of  $\tau$  in  $G_F$  according to the arguments above.  $\square$

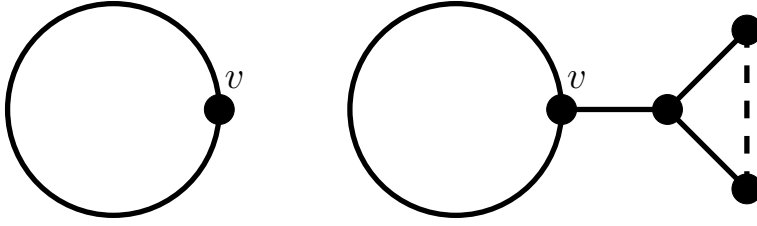
We finally explain how to generalize the reduction of Theorem 1 so that we get a reduction from 1-IN-3 SAT to  $k$ -REALSEQ for any  $k \geq 3$ .

**Theorem 2**  *$k$ -REALSEQ is NP-complete for every  $k \geq 2$ .*

*Proof*  $k$ -REALSEQ is in NP for every  $k \geq 2$  as mentioned in the proof of Theorem 1. Besides, recall that 2-REALSEQ is NP-complete by Theorem 1. The proof that  $k$ -REALSEQ is NP-complete for every  $k \geq 3$  is based on our reduction from 1-IN-3 SAT to 2-REALSEQ. More precisely, we want to augment the reduced instance, that is the graph  $G_F$  and the sequence  $\tau$ , in such a way that the arguments given in the proof of Theorem 1 are still correct and not altered by the modifications.

For the sake of this proof, let us introduce the following definition. Given a graph  $H$ , a vertex  $v \in V(H)$  and an arbitrary integer  $a \geq 3$ , the  $(a, v)$ -star-augmentation of  $H$  is the graph obtained as follows:

1. consider the disjoint union of  $H$  and a star  $S^a$  with  $a - 1$  vertices of degree 1 whose root is denoted by  $r$ ,
2. add an edge between  $v$  and  $r$ .



**Fig. 3** A graph  $H$  and an arbitrary  $(a, v)$ -star-augmentation of  $H$

An example of an  $(a, v)$ -star-augmentation of a graph is depicted in Figure 3.

Let us now show that 3-REALSEQ is NP-complete by reduction from 1-IN-3 SAT. From a 3CNF formula  $F$ , we construct a graph  $G_F$  and a sequence  $\tau = (n_1, n_2, n_3)$  admissible for  $G_F$  such that  $F$  is satisfiable in a 1-in-3 way if and only if  $\tau$  is realizable in  $G_F$ .

By applying the reduction from 1-IN-3 SAT to 2-REALSEQ, we get a graph  $G'_F$  and a sequence  $\tau' = (n'_1, n'_2)$  admissible for  $G'_F$  that is realizable in  $G'_F$  if and only if  $F$  admits a 1-in-3 assignment of its variables. Now consider, as  $G$ , an  $(a, v)$ -star-augmentation of  $G'_F$  where  $a = n_1 + n_2 + 1$  and  $v \in V(G'_F)$  is arbitrary, and  $\tau = (a, n'_1, n'_2)$ . In a realization  $(U, V_1, V_2)$  of  $\tau$  in  $G_F$ , notice that, because  $n'_1, n'_2 \geq 2$ , the star subgraph  $S^a$  of  $G_F$  must be covered entirely by the part  $U$  with size  $a$  since otherwise the remaining subgraph  $G_F - U$  would contain too many small components and it would be impossible for  $V_1$  and  $V_2$  to induce two connected subgraphs of  $G_F$ . Therefore,  $\tau$  is realizable in  $G_F$  if and only if  $\tau'$  is realizable in  $G'_F$ , and by transitivity we get that  $F$  is satisfiable in a 1-in-3 way if and only if  $\tau$  is realizable in  $G_F$ .

Of course, this reduction modification can be applied as many times as necessary to get a reduction from 1-IN-3 SAT to  $k$ -REALSEQ for any  $k \geq 4$ .  $\square$

### 3 Complexity of partitioning a graph into connected subgraphs following a prescription

In this section, we investigate the computational complexity of the following decision problem.

PRESCRIPTIBLE SEQUENCE - PRESCSEQ

Instance: A graph  $G$ , a sequence  $\tau = (n_1, \dots, n_p)$  admissible for  $G$ , and a  $k$ -prescription  $P$  of  $G$  with  $k \in \{1, \dots, p\}$ .

Question: Is  $\tau$  realizable in  $G$  under  $P$ ?

This problem has the same complexity as REALSEQ, this statement being not affected by the size of  $P$  or by the elements of  $\tau$  associated with  $P$ . This claim is proved by reduction from REALSEQ which was shown to be NP-complete in Section 2.



**Theorem 3** PRESCSEQ is NP-complete.

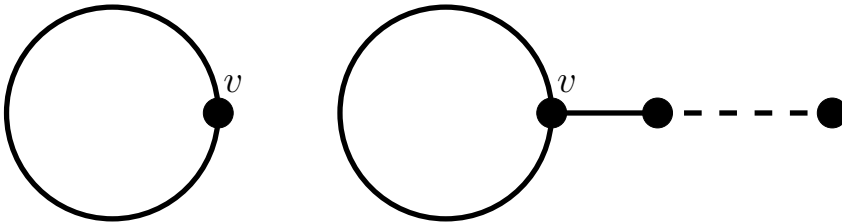
*Proof* It should be clear that one can modify the checking algorithm for REALSEQ we gave in the proof of Theorem 1 so that it also takes the prescription  $P$  as an input and makes sure that the vertices of  $P$  belong to the corresponding parts of  $R$ . This modification does not alter the time complexity of the algorithm. Therefore, PRESCSEQ is in NP.

We now show that PRESCSEQ is NP-complete by reduction from REALSEQ. Given a graph  $G$  and a sequence  $\tau = (n_1, \dots, n_p)$  admissible for  $G$ , we construct a new graph  $G'$ , a new sequence  $\tau'$  admissible for  $G'$  and a prescription  $P$  of  $G'$  such that  $\tau$  is realizable in  $G$  if and only if  $\tau'$  is realizable in  $G'$  under  $P$ .

Let  $a \geq 1$  be an arbitrary positive integer and  $v$  be an arbitrary vertex of some graph  $H$ . The  $(a, v)$ -path-augmentation of  $H$  is the graph obtained from  $H$  as follows:

1. consider the disjoint union of  $H$  and  $P_a$ , a path of order  $a$  whose vertices are consecutively denoted by  $u_1, \dots, u_a$ ;
2. add an edge between  $u_1$  and  $v$ .

This construction is depicted in Figure 4.



**Fig. 4** A graph  $H$  and an arbitrary  $(a, v)$ -path-augmentation of  $H$

Let us denote by  $G'$  an  $(a, v)$ -path-augmentation of  $G$  for some arbitrary integer  $a \geq 1$  and  $v \in V(G)$ , and let  $\tau' = (a, n_1, \dots, n_p)$  and  $P = (u_a)$ . Since the first part of a realization of  $\tau'$  in  $G'$  must induce a connected subgraph of  $G'$  on  $a$  vertices including  $u_a$ , it should be clear that we must set  $U = \{u_1, \dots, u_a\}$ . Once this part has been picked up, we still have to find a realization  $(V_1, \dots, V_p)$  of the remaining sequence  $(n_1, \dots, n_p) = \tau$  in the remaining graph  $G' - U = G$ . If the latter realization of  $\tau$  in  $G$  exists, then  $(U, V_1, \dots, V_p)$  is a realization of  $\tau'$  in  $G'$  under  $P$ .

Conversely, if there exists a realization  $(V_1, \dots, V_p)$  of  $\tau$  in  $G$ , then it should be clear that  $(U, V_1, \dots, V_p)$ , where  $U = \{u_1, \dots, u_a\}$ , is a correct realization of  $\tau'$  in  $G'$  under  $P$ . Thus, this reduction is correct since it can clearly be performed in polynomial time.  $\square$

Notice that our reduction is correct for every  $(a, v)$ -path-augmentation of  $G$  and that we can perform several arbitrary path-augmentations of  $G$  as long as we add the corresponding part sizes to  $\tau$  and vertices to  $P$ . Additionally,

one can perform the reduction above from one of the  $k$ -REALSEQ problems instead of REALSEQ so that the size of  $\tau$  in a reduced instance of PRESCSEQ is constant. Recall also that in our reduction from 1-IN-3 SAT to 2-REALSEQ (proof of Theorem 1), the base subgraph of  $G_F$  has to be covered by the first part of a realization of  $\tau$  in  $G_F$  while the vertices of the control star of the clause subgraph must belong to the second part of such a realization. One can thus request a 1- or 2-prescription while realizing  $\tau$  in  $G_F$ .

Because of all the previous remarks, we get that PRESCSEQ is NP-complete for every constant size  $c \geq 2$  of  $\tau$  and every constant size  $c' \in \{1, \dots, c\}$  of  $P$ .

#### 4 Some $\Pi_2^P$ problems

The  $\Pi_2^P$  complexity class belongs to the second level of the polynomial hierarchy [5]. In this section, we investigate the relationship between some graph partition problems derived from our definitions and  $\Pi_2^P$ .

We start with the following problem.

AP GRAPH

Instance: A graph  $G$ .

Question: Is  $G$  an AP graph?

This problem is not known to belong to either NP or co-NP. However, it is clearly in  $\Pi_2^P$  since one can design a polynomial-time algorithm that takes  $G$  and a sequence  $\tau$  admissible for  $G$  not realizable in  $G$  as input and checks that  $\tau$  is indeed not realizable in  $G$  using an oracle for REALSEQ.

The problem defined below

AP+ $k$  GRAPH

Instance: A graph  $G$ .

Question: Is  $G$  an AP+ $k$  graph?

is also in  $\Pi_2^P$  for every  $k \geq 1$ . Indeed, recall that PRESCSEQ restricted to  $k$ -prescriptions is an NP problem. One can thus design a similar algorithm as the one we just mentioned for AP GRAPH, except that this algorithm would use an oracle for PRESCSEQ.

We do not know whether AP GRAPH and AP+ $k$  GRAPH are  $\Pi_2^P$ -complete problems. Indeed, to design a polynomial-time reduction from a  $\Pi_2^P$ -complete problem  $A$  to one of these two problems, it would be necessary to "translate" the restrictions associated with an instance of  $A$  to some graph substructures just like we did in the proof of Theorem 1 by introducing a lot of star subgraphs in the reduced graphs. But introducing these graph substructures would make the whole graph being not AP. That is why, for example, our reduction from 1-IN-3 SAT to REALSEQ does not seem to be generalizable into some reduction from a  $\Pi_2^P$ -complete version of 1-IN-3 SAT to AP GRAPH.

There are however  $\Pi_2^P$ -complete problems related to our graph partition problems. For the sake of the illustration problem below, let us introduce more definitions. Let  $G$  be a graph and  $\tau = (n_1, \dots, n_p)$  be a sequence that is admissible for  $G$ . Given a  $\ell \in \{1, \dots, p\}$ , a  $n_\ell$ -partition-level for  $\tau$  and  $G$  is a set  $L_\ell$  of subsets of  $V(G)$  that induce connected subgraphs of  $G$  with order  $n_\ell$ .

A  $(n_1, \dots, n_\ell)$ -partition-hierarchy  $L$  for  $\tau$  and  $G$  is a collection  $L = (L_1, \dots, L_\ell)$  of  $n_i$ -partition-levels for  $\tau$  and  $G$  for  $i$  up to  $\ell$  such that no subsets in  $L_i$  and  $L_j$  intersect for  $i \neq j$ . We finally say that  $\tau$  is *realizable in  $G$  under  $L$*  if for every collection of subsets  $(V_1, \dots, V_\ell)$  from the partition levels of  $L$  such that  $V_1 \in L_1, \dots, V_\ell \in L_\ell$  there exists a realization  $(V_1, \dots, V_\ell, \dots, V_p)$  of  $\tau$  in  $G$ . In other words, we are given partial realizations of  $\tau$  in  $G$ , that is some ways for picking up the parts associated with the  $\ell$  first elements of  $\tau$ , whose parts are dispatched into  $\ell$  partition levels, and we ask whether each of these partial realizations is extendable to a whole realization of  $\tau$  in  $G$ . A partition hierarchy for  $\tau$  and  $G$  can be seen as a compact way to describe a large number of partial realizations of  $\tau$  in  $G$ .

As an illustration of these definitions, consider the two graphs  $K_{1,4}$  and  $K_5$  of Figure 5. Let  $\tau = (1, 1, 3)$  be an admissible sequence for  $K_{1,4}$  and  $K_5$ , let  $L_1 = (\{a\}, \{c\})$  and  $L_2 = (\{b\}, \{e\})$  be two 1-partition-levels for  $\tau$  and both  $K_{1,4}$  and  $K_5$ , and  $L = (L_1, L_2)$  be a  $(1, 1)$ -partition-hierarchy for  $\tau$  and both  $K_{1,4}$  and  $K_5$ . Clearly,  $\tau$  is not realizable in  $K_{1,4}$  under  $L$  since  $(\{c\}, \{b\}, V(K_{1,4}) - \{c, b\})$  is not a correct realization of  $\tau$  in  $K_{1,4}$ . However,  $\tau$  is realizable in  $K_5$  under  $L$  since  $(\{a\}, \{b\}, V(K_5) - \{a, b\})$ ,  $(\{a\}, \{e\}, V(K_5) - \{a, e\})$ ,  $(\{c\}, \{b\}, V(K_5) - \{c, b\})$  and  $(\{c\}, \{e\}, V(K_5) - \{c, e\})$  are correct realizations of  $\tau$  in  $K_5$ .

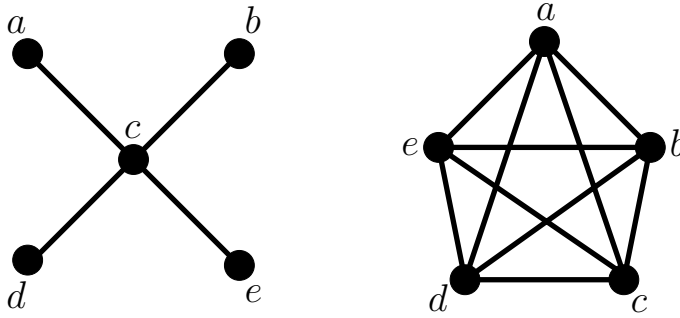


Fig. 5 The graphs  $K_{1,4}$  and  $K_5$

We now investigate the computational complexity of the problem associated with the definition above.

DYNAMIC REALIZABLE SEQUENCE - DYNREALSEQ

Instance: A graph  $G$ , a sequence  $\tau = (n_1, \dots, n_{p'}, \dots, n_p)$  admissible for  $G$  with  $p \geq p'$  elements, and a  $(n_1, \dots, n_{p'})$ -partition-hierarchy  $L$  for  $\tau$  and  $G$ .

Question: Is  $\tau$  realizable in  $G$  under  $L$ ?

As mentioned above, DYNREALSEQ is a  $\Pi_2^p$ -complete problem. Our proof of this claim is based on our reduction from 1-IN-3 SAT to REALSEQ. In order to reuse it, we need a  $\Pi_2^p$ -complete version of 1-IN-3 SAT. We thus show that the following problem

$\forall\exists 1\text{-IN-3 SAT}$ 

Instance: A 3CNF formula  $F$  over variables  $X \cup Y$ , where  $X = \{x_1, \dots, x_{n'}\}$ ,  $Y = \{x_{n'+1}, \dots, x_n\}$  and  $n' \leq n$ , and clauses  $\{C_1, \dots, C_m\}$ .

Question: For every truth assignment of the variables of  $X$ , does there exist a truth assignment of the variables of  $Y$  such that  $F$  is satisfied in a 1-in-3 way? is  $\Pi_2^p$ -complete. This is done by reduction from the following classical  $\Pi_2^p$ -complete problem.

 $\forall\exists 3\text{SAT}$ 

Instance: A 3CNF formula  $F$  over variables  $X \cup Y$ , where  $X = \{x_1, \dots, x_{n'}\}$ ,  $Y = \{x_{n'+1}, \dots, x_n\}$  and  $n' \leq n$ , and clauses  $\{C_1, \dots, C_m\}$ .

Question: For every truth assignment of the variables of  $X$ , does there exist a truth assignment of the variables of  $Y$  such that  $F$  is satisfied?

**Lemma 4**  $\forall\exists 1\text{-IN-3 SAT}$  is  $\Pi_2^p$ -complete.

*Proof*  $\forall\exists 1\text{-IN-3 SAT}$  is clearly in  $\Pi_2^p$ . One can indeed design an algorithm that takes  $F$  and a truth assignment  $\phi_1$  to the variables of  $X$  for which there is no truth assignment  $\phi_2$  to the variables in  $Y$  making  $F$  evaluated in a 1-in-3 way as input. It just has to check that  $\phi_2$  does not exist thanks to an oracle dealing with 1-IN-3 SAT. Such a checking algorithm runs in polynomial time regarding the size of  $F$ .

We now show that  $\forall\exists 1\text{-IN-3 SAT}$  is  $\Pi_2^p$ -complete by reduction from  $\forall\exists 3\text{SAT}$ . From a 3CNF formula  $F$  over variables  $X \cup Y$ , we construct a new 3CNF formula  $F'$  over variables  $X' \cup Y'$  such that for every truth assignment  $\phi_1$  to the variables in  $X$  there exists a truth assignment  $\phi_2$  to the variables in  $Y$  making  $F$  evaluated true if and only if for every truth assignment  $\phi'_1$  to the variables in  $X'$  there exists a truth assignment  $\phi'_2$  to the variables in  $Y'$  such that  $F'$  is evaluated true in a 1-in-3 way.

The reduction is straightforward. First, replace each clause  $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$  in  $F$  by four clauses  $(\overline{\ell_{i_1}} \vee a_i \vee b_i)$ ,  $(\overline{\ell_{i_2}} \vee c_i \vee d_i)$ ,  $(\overline{\ell_{i_3}} \vee e_i \vee f_i)$  and  $(a_i \vee c_i \vee e_i)$  in  $F'$  where  $a_i, b_i, c_i, d_i, e_i$  and  $f_i$  are six new variables. Finally, let  $X' = X$  and  $Y' = Y \cup \bigcup_{i=1}^m \{a_i, b_i, c_i, d_i, e_i, f_i\}$ .

First suppose that for every truth assignment  $\phi'_1$  to the variables of  $X'$  there exists a truth assignment  $\phi'_2$  to the variables in  $Y'$  such that  $F'$  is satisfied in a 1-in-3 way. Because every clause of  $F'$  has exactly one true literal under  $\phi'_1$  and  $\phi'_2$ , it means that only one element in  $\{a_i, c_i, e_i\}$  is evaluated true by  $\phi'_2$  for every  $i \in \{1, \dots, m\}$ . Let us suppose that for such an  $i$  we have  $\phi'_2(a_i) = 1$  and  $\phi'_2(c_i) = \phi'_2(e_i) = 0$  without loss of generality. Thus, we have  $\ell_{i_1}$  evaluated true by either  $\phi'_1$  or  $\phi'_2$ . It follows that the following truth assignment  $\phi_1$  and  $\phi_2$  of the variables in  $X$  and  $Y$ , respectively,

- $\phi_1 = \phi'_1$ ,
- $\phi_2(x) = \phi'_2(x)$  for every  $x \in Y$ ,

is such that  $F$  is satisfied. Conversely, suppose that for every truth assignment  $\phi_1$  to the variables in  $X$  there is a truth assignment  $\phi_2$  to the variables in  $Y$  such that  $F$  has all its clauses satisfied under  $\phi_1$  and  $\phi_2$ . We explain how to

get a truth assignment  $\phi'_2$  to the variables in  $Y'$  so that  $F'$  is evaluated true in a 1-in-3 way under  $\phi'_2$  and the truth assignment  $\phi'_1 = \phi_1$  to the variables in  $X'$ . First, let  $\phi'_2(x) = \phi_2(x)$  for every  $x \in Y$ . We then have to provide a truth assignment of  $a_i, b_i, c_i, d_i, e_i$  and  $f_i$  via  $\phi'_2$  for every  $i \in \{1, \dots, m\}$ . This assignment depends on the number of true literals in  $C_i = (\ell_{i_1} \vee \ell_{i_2} \vee \ell_{i_3})$  via  $\phi_1$  and  $\phi_2$ . Let  $\phi_3 : X \cup Y \rightarrow \{0, 1\}$  be the truth assignment of the variables in  $X \cup Y$  deduced from  $\phi_1$  and  $\phi_2$  as follows:

- if  $i \in \{1, \dots, n'\}$ , then  $\phi_3(x_i) = \phi_1(x_i)$ ;
- if  $i \in \{n' + 1, \dots, n\}$ , then  $\phi_3(x_i) = \phi_2(x_i)$ .

Consider now that the images of the  $a_i$ 's,  $b_i$ 's,  $c_i$ 's,  $d_i$ 's,  $e_i$ 's and  $f_i$ 's by  $\phi'_2$  are the ones depicted in Table 1. It should then be clear that  $F'$  is evaluated true in a 1-in-3 way under  $\phi'_1$  and  $\phi'_2$ .

$(\phi_3(\ell_{i_1}), \phi_3(\ell_{i_2}), \phi_3(\ell_{i_3}))$	$\phi'_2(a_i)$	$\phi'_2(b_i)$	$\phi'_2(c_i)$	$\phi'_2(d_i)$	$\phi'_2(e_i)$	$\phi'_2(f_i)$
(1, 0, 0)	1	0	0	0	0	0
(0, 1, 0)	0	0	1	0	0	0
(0, 0, 1)	0	0	0	0	1	0
(1, 1, 0)	1	0	0	1	0	0
(1, 0, 1)	1	0	0	0	0	1
(0, 1, 1)	0	0	1	0	0	1
(1, 1, 1)	1	0	0	1	0	1

**Table 1** Truth assignment of  $\phi'_2$  for the variables in  $Y - Y'$

□

We finally prove that DYNREALSEQ is  $\Pi_2^P$ -complete.

**Theorem 5** DYNREALSEQ is  $\Pi_2^P$ -complete.

*Proof* DYNREALSEQ is clearly a  $\Pi_2^P$  problem. One can provide a combination of parts  $(V_1, \dots, V_{p'})$  from the  $(n_1, \dots, n_{p'})$ -partition-hierarchy for  $\tau$  and  $G$  of the problem instance to a polynomial-time algorithm checking that these parts cannot be extended to a realization of  $\tau$  in  $G$ . It just has to make sure that the sequence  $(n_{p'+1}, \dots, n_p)$  is not realizable in  $G - \bigcup_{i=1}^{p'} V_i$  using an oracle for REALSEQ.

We now show that DYNREALSEQ is complete in  $\Pi_2^P$  by reduction from  $\forall\exists 1$ -IN-3 SAT that was shown to be  $\Pi_2^P$ -complete in Lemma 4. Our reduction is inspired by the reduction from 1-IN-3 SAT to REALSEQ we gave in the proof of Theorem 1. Remember that in the latter reduction, setting a variable of  $F$  to true is simulated in an instance of REALSEQ by adding a literal vertex of  $G_F$  to the part with size  $n_1 + n$  of a realization of  $\tau$  in  $G_F$ . We here somehow want to keep that relationship between setting a variable of  $F$  to true and putting a literal vertex of the  $G_F$  into a part of the realization. Given a truth assignment  $\phi_1$  to the variables in  $X$ , it means that we have to check whether every partial realization of  $\tau$  in  $G_F$  whose part with size  $n_1 + n$  contains

the literal vertices associated with the true literals via  $\phi_1$  is extendable to a realization of  $\tau$  in  $G_F$ . All these possible partial realizations are considered using a partition-hierarchy for  $\tau$  and  $G_F$ .

First of all, let  $G_F$  be the graph obtained from  $F$  using the reduction we gave in the proof of Theorem 1. Then, let  $\tau = (1, \dots, 1, n_1 + n - n', n_2 - n)$  be a sequence admissible for  $G_F$  with size  $n' + 2$ , let  $L_i = \{\{v_{x_i}\}, \{v_{\bar{x}_i}\}\}$  be a 1-partition-level for  $\tau$  and  $G_F$  for every  $x_i \in X$ , and  $L = \bigcup_{i=1}^{n'} L_i$  be a  $(1, \dots, 1)$ -partition-hierarchy for  $\tau$  and  $G_F$ . With a truth assignment  $\phi_1$  of the variables in  $X$  setting  $n'$  literals of  $F$  to true is then associated a combination of vertex-disjoint subsets  $(V_1, \dots, V_{n'})$  from the 1-partition-levels in  $L$ , where  $V_i = \{x_i\}$  if  $\phi_1(x_i) = 1$  or  $V_i = \{\bar{x}_i\}$  otherwise.

Let us now suppose that for every truth assignment  $\phi_1$  to the variables in  $X$  there exists a truth assignment  $\phi_2$  to the variables of  $Y$  such that  $F$  is evaluated true in a 1-in-3 way. Then the realization  $(V_1, \dots, V_{n'+2})$  of  $\tau$  in  $G_F$ , where

- for every  $i \in \{1, \dots, n'\}$ , we have  $V_i = \{v_{x_i}\}$  if  $\phi_1(x_i) = 1$  or  $V_i = \{v_{\bar{x}_i}\}$  otherwise,
- $V_{n'+1}$  contains all the vertices from the base subgraph of  $G_F$  and every literal vertex  $v_{\ell_i}$  of the clause subgraph of  $G_F$  such that  $\phi_2(\ell_i) = 1$ ,
- $V_{n'+2} = V(G_F) - \bigcup_{i=1}^{n'+1} V_i$ ,

is correct according to the arguments we gave in the proof of Theorem 1. Conversely, suppose that every combination  $(V_1, \dots, V_{n'})$  of subsets from the 1-partition-levels of  $L$  is extendable to a realization  $(V_1, \dots, V_{n'+2})$  of  $\tau$  in  $G_F$ . As explained before, the partition  $(V_1, \dots, V_{n'})$  is associated with a truth assignment  $\phi_1$  to the variables in  $X$  and from the literal vertices contained in  $V_{n'+1}$  we can deduce a truth assignment  $\phi_2$  to the variables in  $Y$  (see the proof of Theorem 1). Clearly,  $F$  is evaluated true in a 1-in-3 way under  $\phi_1$  and  $\phi_2$ .  $\square$

## References

1. D. Barth, O. Baudon, and J. Puech. Decomposable trees: a polynomial algorithm for tripodes. *Discret. Appl. Math.*, 119(3):205–216, July 2002.
2. D. Barth and H. Fournier. A degree bound on decomposable trees. *Discret. Math.*, 306(5):469–477, 2006.
3. O. Baudon, J. Bensmail, J. Przybyło, and M. Woźniak. Partitioning powers of traceable or Hamiltonian graphs. *Preprint*, 2012. Available at <http://hal.archives-ouvertes.fr/hal-00687278>.
4. M.E. Dyer and A.M. Frieze. On the complexity of partitioning graphs into connected subgraphs. *Discret. Appl. Math.*, 10:139–153, 1985.
5. M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979.
6. N. Guttmann-Beck and R. Hassin. Approximation algorithms for min-max tree partition. *J. Algorithms.*, 24(2):266–286, 1997.
7. E. Györi. On division of graphs to connected subgraphs. In *Combinatorics*, pages 485–494, Colloq. Math. Soc. János Bolyai 18, 1978.
8. L. Lovász. A homology theory for spanning trees of a graph. *Acta Math. Acad. Sci. Hung.*, 30(3-4):241–251, 1977.