



**HAL**  
open science

## Graph problems with obligations

Alexis Cornet, Christian Laforest

► **To cite this version:**

Alexis Cornet, Christian Laforest. Graph problems with obligations. [Research Report] LIMOS (UMR CNRS 6158), universit  Clermont Auvergne, France. 2018. hal-01791148v1

**HAL Id: hal-01791148**

**<https://hal.science/hal-01791148v1>**

Submitted on 14 May 2018 (v1), last revised 29 Jun 2018 (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 destin e au d p t et   la diffusion de documents scientifiques de niveau recherche, publi s ou non,  manant des  tablissements d'enseignement et de recherche fran ais ou  trangers, des laboratoires publics ou priv s.

# Graph problems with obligations

Alexis Cornet, Christian Laforest  
LIMOS (UMR CNRS 6158), Université Clermont-Auvergne, France

May 14, 2018

## Abstract

In this paper we study variants of well-known graph problems as *vertex cover*, *spanning tree*, *matchings*, and *hamiltonian path*. Given a graph  $G = (V, E)$ , we add a partition  $\Pi_V$  (resp.  $\Pi_E$ ) of its vertices (resp. of its edges). Now, any solution  $S$  containing an element (vertex or edge) of a part of this partition must also contain all the others ones. In other words, elements can only be added set by set, instead of one by one as in the classical situation. We propose hardness and approximation results.

## Contents

<b>1</b>	<b>Obligations</b>	<b>1</b>
<b>2</b>	<b>Vertex cover with obligations on vertices</b>	<b>2</b>
<b>3</b>	<b>Connected vertex cover with obligations on vertices</b>	<b>3</b>
<b>4</b>	<b>Spanning tree with obligations on edges</b>	<b>5</b>
<b>5</b>	<b>Matchings with obligations on edges</b>	<b>7</b>
<b>6</b>	<b>Hamiltonian path in complete graphs with obligations on edges</b>	<b>9</b>

## 1 Obligations

In this paper we deal with classical graph problems with additional constraints. Let  $G = (V, E)$  be any undirected graph. We call *system of obligations on vertices of  $G$*  a *partition*  $\Pi_V = V_1, \dots, V_k$  of  $V$  and a *system of obligations on edges of  $G$*  a *partition*  $\Pi_E = E_1, \dots, E_k$  of  $E$ . Each element  $V_i$  (resp.  $E_i$ ) is called a *part* of  $\Pi_V$  (resp.  $\Pi_E$ ). Now, given  $G$  and an associated system  $\Pi_V$  (resp.  $\Pi_E$ ) of obligations on vertices (resp. edges), any solution  $S$  to a problem on  $G$  must have the property that: if  $u \in S$  (resp.  $e \in S$ ) and  $u \in V_i$  (resp.  $e \in E_i$ ) then  $V_i$  (resp.  $E_i$ ) must be entirely included in  $S$ , that is  $V_i \subseteq S$  (resp.  $E_i \subseteq S$ ). In other words, once an “object”  $x$  (vertex or edge), element of a part  $X$ , is in a solution, *all* the others elements of  $X$  *must* also be included in the solution. Obligations can be useful to model situations in which some set of elements (captors, computers, softwares, people, etc.) are interdependent and the presence of one element induces the presence of

all the other ones. From an algorithmic point of view, we will see that in most cases the problems become much harder than the original ones.

This study comes to complete many recent works on a sort of opposite problem, implying what is called *conflict* which is a pair  $\{x, y\}$  of edges or vertices of a graph that *cannot* be both in a solution. Here an instance is then a graph  $G$  and a set of conflicts. Obtaining a solution without conflict is hard in general for many graph problems, as it is shown in these papers [2, 3, 4, 7, 8, 9, 10, 11, 12, 13, 14].

In what follows we give some useful notations (undefined terms can be found in [5] for example). Let  $G = (V, E)$  be any non directed graph, with  $V$  its set of vertices and  $E$  its set of edges. Two vertices  $u$  and  $v$  are *neighbours* if  $G$  contains the edge  $uv$ . The *degree* of a vertex  $u$  is its number of neighbours. We call *graph induced* by the set of edges  $E_i \subseteq E$ , the graph whose set of edges is  $E_i$  and whose vertices are the ones that are at the extremity of at least an edge of  $E_i$ . The graph induced by a set  $S$  of vertices of  $G$ , noted  $G[S]$ , is the graph whose set of vertices is  $S$  and whose edges are the ones of  $G$  connecting two vertices of  $S$ . A *stable* (or *independent*)  $S$  of  $G$  is a subset of its vertices having the property that  $G[S]$  contains no edge.

## 2 Vertex cover with obligations on vertices

Let  $G = (V, E)$  be any graph and  $\Pi_V = V_1, \dots, V_k$  a partition of  $V$ , a system of obligations on vertices of  $G$ . A *vertex cover with obligations* ( $\mathcal{VCO}$ )  $S$ , of  $(G, \Pi_V)$  is:

- A vertex cover of  $G$ : each edge  $e = uv \in E$  is covered by  $S$  ( $u \in S$  or  $v \in S$  (both can be in  $S$ )).
- $\forall u \in S$ , if  $u \in V_i$ , then  $V_i \subseteq S$ .

It is easy to see that any instance  $(G = (V, E), \Pi_V)$  always contains at least a  $\mathcal{VCO}$ , namely  $S = V$ .

A  $\mathcal{VCO}$   $S^*$  of the instance  $(G, \Pi_V)$  is said *optimal*, and noted  $\mathcal{VCO}_{OPT}$ , if it is of minimum size. Constructing a  $\mathcal{VCO}_{OPT}$  is hard since even in the very particular case where each part of  $\Pi_V$  is a singleton, this is the classical NP-complete vertex cover problem [6]. In what follows we propose an approximation algorithm for the  $\mathcal{VCO}_{OPT}$  problem. But first we can easily simplify the instance in some cases. Indeed, we can remark that if  $e = uv \in E$  and  $u$  and  $v$  are in the same part  $V_i$  of the partition ( $u \in V_i$  and  $v \in V_i$ ) then any  $\mathcal{VCO}$  (thus any  $\mathcal{VCO}_{OPT}$ ) must contain  $V_i$  since the edge  $e = uv$  must be covered and  $u$  or  $v$  must be in any solution and thus also  $V_i$ . Before running any algorithm, we can include in any solution, all the parts  $V_i$  of  $\Pi_V$  such that  $G$  contains an edge  $e$  with both extremities in  $V_i$ . This can be done in polynomial time. We suppose now that this pre-treatment has been done and that  $G$  does not contain these vertices anymore and  $\Pi_V$  does not contain these parts anymore.

**A 2-approximation algorithm for the  $\mathcal{VCO}_{OPT}$  problem.** At this point we can suppose that an instance is now  $(G = (V, E), \Pi_V = V_1, \dots, V_k)$  where each  $V_i$  is a stable of  $G$ .

1. Construct as follows a new weighted graph  $G_c = (V_c, E_c)$  called *contracted graph*:
  - Each stable  $V_i$  of  $\Pi_V$  is associated to a vertex  $v_i$  of  $G_c$ .

- The *weight* of  $v_i$  is the number of vertices of  $V_i$  ( $|V_i|$ ).
  - Add an edge between  $v_i$  and  $v_j$  in  $G_c$  iff  $G$  contains (at least) an edge having an extremity in  $V_i$  and the other in  $V_j$ .
2. Construct a 2-approximated *weighted vertex cover*  $S_c$  in  $G_c$  (this can be done in polynomial time, see [1]).
  3. Return  $S = \bigcup_{i:v_i \in S_c} V_i$ .

**Theorem 1** *The algorithm described above is a 2-approximation algorithm for the  $\mathcal{VCO}_{OPT}$  problem.*

**Proof.** This algorithm is polynomial. It constructs a vertex cover of  $G$  that satisfies the constraints on obligations.

Note that to respect the conditions on obligations, any  $\mathcal{VCO}$  of  $(G, \Pi_V)$  is an union of some parts of  $\Pi_V$ . We construct now a one-to-one correspondance respecting the weights and the sizes between the  $\mathcal{VCO}$  of  $(G, \Pi_V)$  and the vertex covers of  $G_c$ .

- Let  $S$  be any  $\mathcal{VCO}$  of  $(G, \Pi_V)$ . The set  $S_c = \{v_i : i : V_i \subseteq S\}$  associated to  $S$  is a vertex cover of  $G_c$  of weight  $|S|$ .
- Conversely, let  $S_c = \{v_1, \dots, v_l\}$  be any weighted vertex cover of  $G_c$ . In this case,  $S = \{V_i : i : v_i \in S_c\}$  is a  $\mathcal{VCO}$  of  $(G, \Pi_V)$  whose size is equal to the weight of  $S_c$ .

A 2-approximation of an optimal weighted vertex cover of  $G_c$  corresponds to a 2-approximated  $\mathcal{VCO}_{OPT}$  of  $(G, \Pi_V)$ . Hence the proposed algorithm is a 2-approximation algorithm for the  $\mathcal{VCO}_{OPT}$  problem.  $\square$

### 3 Connected vertex cover with obligations on vertices

In this section,  $G = (V, E)$  is a *connected* graph. As in section 2, the *obligations* are given by a partition  $\Pi_V = V_1, \dots, V_k$  of  $V$ . A *CVCO*, *connected vertex cover with obligations*,  $S$  of the instance  $(G, \Pi_V)$  is:

- A vertex cover of  $G$  (for any edge  $uv \in E$ ,  $u \in S$  or  $v \in S$  (both can be in  $S$ )),
- A connected set of vertices:  $G[S]$  (the induced graph of  $S$  in  $G$ ) is connected,
- $S$  respects the constraints of obligations of  $\Pi_V$ .

A  $\mathcal{CVCO}_{OPT}$  is a  $\mathcal{VCO}$  of minimum size. Constructing a  $\mathcal{CVCO}_{OPT}$  is a hard problem, even if  $\Pi_V$  is a partition of singletons (in this case this the classical NP-complete connected vertex cover problem [6]).

**Theorem 2** *Any  $\alpha$ -approximation algorithm for the  $\mathcal{CVCO}_{OPT}$  problem can be transformed into a  $2\alpha$ -approximation algorithm for the Optimal Set Cover problem.*

**Proof.** Let  $(A, X)$  be any instance of the Set Cover problem:  $A = \{a_1, \dots, a_n\}$  is a set of  $n$  elements and  $X = X_1, \dots, X_k$  is a family of subsets of  $A$  ( $X_i \subseteq A$ ) covering  $A$ :  $A = \cup_{i=1}^k X_i$ . An optimal set cover is a sub-family of  $X$ , of minimum size, covering  $A$ . We note  $t^*$  the size of such an optimal solution of  $(A, X)$ .

From  $(A, X)$  let us construct an instance of our problem. Each element  $a_i$  is associated to a vertex, also noted  $a_i$ . Each set  $X_i$  of  $X$  is associated to a set noted  $V_i$  of  $n+1$  new vertices, forming a stable. Each of the  $n+1$  vertices of the set  $V_i$  is connected to a vertex  $a_j$  iff the set  $X_i$  contains  $a_j$ . Create now a new vertex  $r$  and connect it to all the vertices of the  $k$  sets  $V_i$ . The degree of  $r$  is then  $k(n+1)$ . We note  $G = (V, E)$  the final graph that is bipartite.

The obligations are the following. Each  $V_i$  is an obligation containing exactly  $n+1$  independent vertices. Add the obligation  $V_0$  containing  $r$  and the  $n$  vertices of  $A$ .  $V_0$  is then also a stable of  $G$  composed of  $n+1$  vertices.  $\Pi_V = V_0, V_1, \dots, V_k$  is a partition of the set  $V$  of vertices of  $G$  and is the system of obligations that we consider here; each  $V_i$  is a stable of  $n+1$  vertices of  $G$ . The instance  $(G, \Pi_V)$  can be constructed in polynomial time from the instance  $(A, X)$ . Consider now the one-to-one mapping between the  $\mathcal{CVCO}$  of  $(G, \Pi_V)$  and the set covers of  $(A, X)$ .

Let  $S_X = X_{i_1}, \dots, X_{i_t}$  be any set cover of size  $t$  of  $(A, X)$ . Consider now the following set  $S$  of vertices of  $G$ :

$$S = V_0 \cup \bigcup_{j=1}^t V_{i_j}$$

$S$  is a vertex cover of  $G$  (all the edges of  $G$  are covered by the vertices of  $V_0$ ),  $G[S]$  is connected (because the vertices of  $V_{i_j}$  are interconnected *via*  $r$  and each  $a_i$  is connected to at least all the vertices of a set  $V_{i_j}$  because  $S_X$  is a covering) and satisfies the obligations of  $\Pi_V$  ( $S$  is composed of an union of obligations of  $\Pi_V$ ). The size of  $S$  is:  $|S| = n+1 + t(n+1) = (n+1)(t+1)$ .

Consider now any  $\mathcal{CVCO}$   $S$  of  $(G, \Pi_V)$ . As  $S$  satisfies the constraints on obligations, it is composed of an union of obligations. As  $G[S]$  is connected and  $G$  is bipartite, it must contain some of the obligations  $V_i, i \geq 1$ . But as  $S$  must contain  $r$  or a vertex  $a_i$  to ensure the connectivity it must contain the obligation  $V_0$ . Note  $V_0, V_{i_1}, \dots, V_{i_t}$  the obligations composing  $S$ :  $S = V_0 \cup V_{i_1} \cup \dots \cup V_{i_t}$ . Let  $S_X = X_{i_1}, \dots, X_{i_t}$  be the sub-family associated to this  $\mathcal{CVCO}$   $S$ . As  $V_0 \subseteq S$ , each vertex  $a_i$  is connected to the other vertices of  $S$  via the vertices of at least a  $V_{i_j}$ . Thus  $S_X$  is a set cover of  $(A, X)$ . We get:  $|S_X| = t$  and  $|S| = (t+1)(n+1)$ .

This one-to-one mapping associates to each set cover of size  $t$  a  $\mathcal{CVCO}$  of size  $(t+1)(n+1)$  and reciprocally. The transformations in one direction or the other can be done in polynomial time.

Suppose that a  $\mathcal{CVCO}_{OPT}$  can be approximated with a ratio  $\alpha$  in polynomial time. Then, for any instance  $(A, X)$  one can: construct the associated instance  $(G, \Pi_V)$ , then use this approximation algorithm to construct a  $\alpha$ -approximated  $\mathcal{CVCO}$   $S$ :  $(t+1)(n+1) = |S| \leq \alpha|S^*|$ . Then with the one-to-one transformation, one can construct the associated set cover  $S_X$ , of size  $t$ . This chain of constructions is polynomial. Let  $S_X^*$  be an optimal set cover, of size  $t^*$ . By the one-to-one transformation, this corresponds to a  $\mathcal{CVCO}$  of size  $(t^*+1)(n+1)$ . This  $\mathcal{CVCO}$  is optimal (otherwise it would be possible to construct a smaller one with the one-to-one transformation). Hence,

$$|S| = (t+1)(n+1) \leq \alpha(t^*+1)(n+1)$$

then,  $t + 1 \leq \alpha(t^* + 1)$  and  $t \leq \alpha t^* + (\alpha - 1) \leq \alpha(t^* + 1) \leq 2\alpha t^*$  (because  $1 \leq t^*$ ). The algorithm described above is then a  $2\alpha$ -approximation algorithm for the set cover problem.  $\square$

**Corollary 1** *The  $\mathcal{CVCO}_{\mathcal{OPT}}$  problem cannot be approximated by a ratio better than  $c \log(n)/2$  unless  $P = NP$ .*

*Proof.* Theorem 2 shows that the  $\mathcal{CVCO}_{\mathcal{OPT}}$  problem cannot be approximated by a ratio better than  $c \log(n)/2$  since the optimal set cover problem cannot be approximated within  $c \log(n)$  for some  $c > 0$ , unless  $P = NP$ , see [1].  $\square$

## 4 Spanning tree with obligations on edges

In this section, an instance is  $(G = (V, E), \Pi_E = E_1, \dots, E_k)$  where  $G$  is any connected graph and  $\Pi_E$ , the obligations, is a partition of  $E$ .

The objective is, given an instance,  $(G = (V, E), \Pi_E = E_1, \dots, E_k)$ , to decide if there is a *tree spanning  $G$  with obligations* ( $\mathcal{TSO}$ )  $T = (V, E_T)$  which is a tree spanning  $G$  and such that for any  $e \in E_T$ , if  $e \in E_i$  then all the edges of  $E_i$  must also be in  $T$ .

**Theorem 3** *Deciding if  $(G = (V, E), \Pi_E = E_1, \dots, E_k)$  contains a  $\mathcal{TSO}$  is NP-complete, even if:*

- $G$  is bipartite, of maximum degree 4 and
- each  $E_i$  induces a star (that is a tree with a vertex directly connected to all the others) with exactly 3 edges ( $|E_i| = 3$ ).

*Proof.* The problem is in NP.

Let  $(X = \{x_1, \dots, x_{3q}\}, Z_1, \dots, Z_k)$  be any instance of the X3C problem (exact cover by 3 sets) where  $X$  is a set of  $3q$  elements and each  $Z_i$  is a subset of 3 elements of  $X$  ( $Z_i \subseteq X$  and  $|Z_i| = 3$ ) with the property:  $X = \bigcup_{i=1}^k Z_i$  (the sets  $Z_i$  cover  $X$ ). The X3C problem consists in deciding if this instance contains an *exact cover* of  $X$ , i.e., if there exist  $Z_{i_1}, \dots, Z_{i_q}$  two-by-two mutually disjoint sets such that  $X = \bigcup_{j=1}^q Z_{i_j}$ . This is a well-known NP-complete problem, even if each element  $x_i$  is in at most 3 sets, see [6]. It is this restricted formulation that we consider here.

From this instance, let us construct a graph  $G$ . For each element  $x_i$  of  $X$  create a new vertex, also noted  $x_i$ . For each set  $Z_i$  create a new vertex, also noted  $Z_i$ . Add an edge between each vertex  $Z_i$  and the 3 vertices that are in the set  $Z_i$ . Now, create a tree  $T_r$  to connect the  $k$  vertices  $Z_i$  that will become leaves of  $T_r$ . The  $Z_i$  are connected two-by-two by a new vertex. Then these  $\lceil k/2 \rceil$  new vertices are connected two-by-two by a new vertex, and so on until there is only one final new vertex that we call  $r$  (as “root” of  $T_r$ ). Each vertex  $u$ , except the leaves  $Z_i$ , have one or two children. For each such  $u$  we add a new vertex  $l_u$  (or two is necessarily) that is only connected to  $u$  ( $l_u$  is a leaf). These 3 vertices are called the 3 children of  $u$ . We get now the final tree. All of these vertices and edges form the final graph  $G = (V, E)$  that is bipartite and, thanks to the restriction on X3C instances, the maximum degree of  $G$  is 4. An illustration of the construction is

given in Figure 1: the bottom vertices are elements of  $X$ , squared vertices are the  $Z_i$ , black ones are the additional children and colored vertices are the internal ones of tree  $T_r$ . The dashed ellipses represent the obligations that are described now.

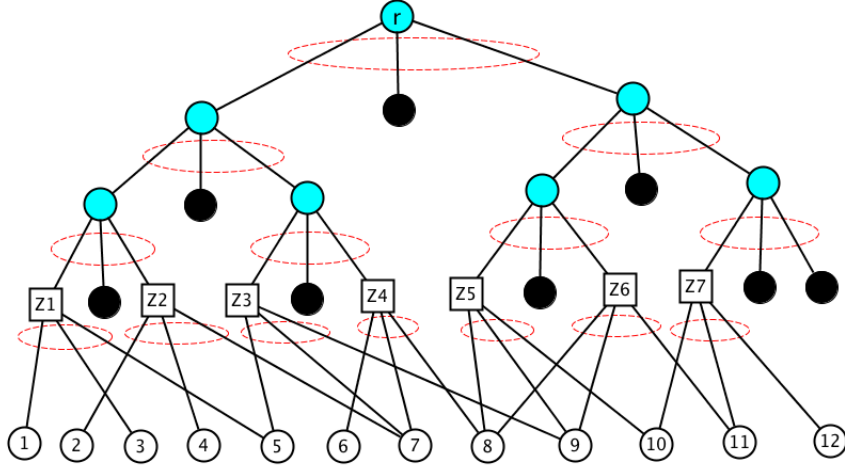


Figure 1: Construction of  $G$  from a  $X3C$  instance

For each vertex  $Z_i$  we group in a same obligation noted  $E_i$  the 3 edges connecting  $Z_i$  to the 3 vertices representing the 3 elements that are in set  $Z_i$ . We group in a same obligation the 3 edges connecting any internal vertex  $u$  to its 3 children. All these obligations are called *tree obligations*. Each edge of  $G$  is now in exactly one obligation (tree one or in a  $E_i$ ) and the set of all these obligations is  $\Pi_E$ , composed of stars of exactly 3 edges.

The construction described above is polynomial.

Suppose that the  $X3C$  instance has a solution  $Z_{i_1}, \dots, Z_{i_q}$ . In this case, we can select the following obligations: all the tree obligations and all the edges in the obligations  $E_{i_1}, \dots, E_{i_q}$ . This gives a tree spanning  $G$  (each vertex  $x_i$  is a leaf because it is a neighbour of exactly one vertex  $Z_{i_j}$  and each vertex  $Z_l$  is connected to the other vertices *via* the tree  $T_r$ ). This tree respects the obligations of  $\Pi_E$  and is then a  $\mathcal{TSO}$  of  $(G, \Pi_E)$ .

Conversely, suppose that the instance  $(G, \Pi_E)$  has a  $\mathcal{TSO}$  noted  $T$ . As  $T$  respects the obligations, it necessarily contains *all* the tree obligations, this is mandatory to include the leaves of the form  $l_u$ . It also contains other obligations. But each vertex  $x_i$  is a leaf of  $T$ . Otherwise, if it is neighbour of 2 vertices,  $Z_a$  and  $Z_b$  then we would have a cycle with some edges of the tree obligations, that is not allowed because  $T$  is a tree. As  $T$  covers all the  $3q$  vertices/leaves  $x_i$  it must contain exactly  $q$  vertices of type  $Z_i$ , noted  $Z_{i_1}, \dots, Z_{i_q}$ , and their 3 edges incident from the associated obligations  $E_{i_1}, \dots, E_{i_q}$ . The sets  $Z_{i_1}, \dots, Z_{i_q}$  cover  $X$  and are pairwise mutually disjoint and is then a solution for the  $X3C$  instance.  $\square$

## 5 Matchings with obligations on edges

In this section, an instance in  $(G, \Pi_E)$  where  $G = (V, E)$  is any undirected graph and  $\Pi_E = E_1, \dots, E_k$  is a partition of  $E$ , the set of edges of  $G$ .

A *matching with obligations* ( $\mathcal{MO}$ )  $M$  of the instance  $(G, \Pi_E)$  is a matching of  $G$  (set of pairwise non incident edges of  $G$ ) such that for any edge  $e$  of  $M$ , if  $e \in E_i$  then  $E_i \subseteq M$ .

It is polynomial to determine if  $(G, \Pi_E)$  contains a  $\mathcal{MO}$ . Indeed, there is a non empty  $\mathcal{MO}$  iff at least a part  $E_i$  induces a matching. From this we can simplify an instance  $(G, \Pi_E)$ : if a part  $E_i$  of  $\Pi_E$  induces a graph in which a vertex has more than one neighbour then  $E_i$  can be deleted from  $\Pi_E$  and the edges of  $E_i$  can be deleted from  $G$ . This pretreatment can be done in polynomial time. From now we suppose that  $(G = (V, E), \Pi_E = E_1, \dots, E_k)$  is an instance where each  $E_i$  induces a matching of  $G$  and thus contains a  $\mathcal{MO}$  (possibly empty). A  $\mathcal{MO}$  of *maximum size* is noted  $\mathcal{MO}_{OPT}$ .

**Theorem 4** *Let  $(G = (V, E), \Pi_E = E_1, \dots, E_k)$  be an instance where each  $E_i$  induced a matching of  $G$ . Any  $\alpha$ -approximation algorithm for the  $\mathcal{MO}_{OPT}$  problem can be transformed into a  $\alpha$ -approximation algorithm for the maximum size stable problem.*

**Proof.** Let  $H = (V_H, E_H)$  be any graph, instance of the maximum size stable problem.

Note  $V_H = \{h_1, \dots, h_n\}$  the  $n$  vertices of  $H$ . We construct an instance of our problem from  $H$ . Let us create two new vertices  $a_i$  and  $b_i$  and associate a new edge  $a_i b_i$  to each vertex  $h_i$  of  $H$ . For each edge  $h_j h_i$  with  $j < i$  let us associate a new edge  $a_j b_{ji}$  where  $b_{ji}$  is a new vertex, exclusively created for the edge  $h_j h_i$  of  $H$ . The union of all these edges and vertices gives a graph noted  $Q$  (not yet the final graph  $G$ ).

For each vertex  $h_i$  of  $H$  let us associate a set  $D_i$  of edges composed of edge  $a_i b_i$  and all the edges  $a_j b_{ji}$  with  $j < i$  and  $h_j h_i \in E_H$ :

$$D_i = \{a_i b_i\} \cup \{a_j b_{ji} : j < i, h_j h_i \in E_H\}$$

$D_1, \dots, D_n$  form a partition of the edges of graph  $Q$  and each  $D_i$  induces a matching of  $Q$ . Figure 2 gives an example of this construction if  $H$  is a cycle with 4 vertices.

The sets  $D_i$  can have different sizes. Let  $D_a$  be the one of maximum size. The next steps consist in adding new independent edges, between new vertices, to each  $D_i$  such that the  $n$  sets all have the same size  $|D_a|$ .

Note  $G = (V, E)$  the graph obtained from  $Q$  by the addition of these new vertices and edges. We note  $\Pi_E = E_1, \dots, E_n$  the sets  $D_1, \dots, D_n$  previously obtained. We now have the following properties.

- All the sets  $E_i$  have the same size noted  $t$ ,
- $E_1, \dots, E_n$  is a partition of the set  $E$  of edges of  $G$ ,
- each  $E_i$  induces a matching in  $G$ ,
- $G$  is a bipartite graph.
- $E_i \cup E_j$  is a matching of  $G$  iff  $h_i h_j \notin E$ .



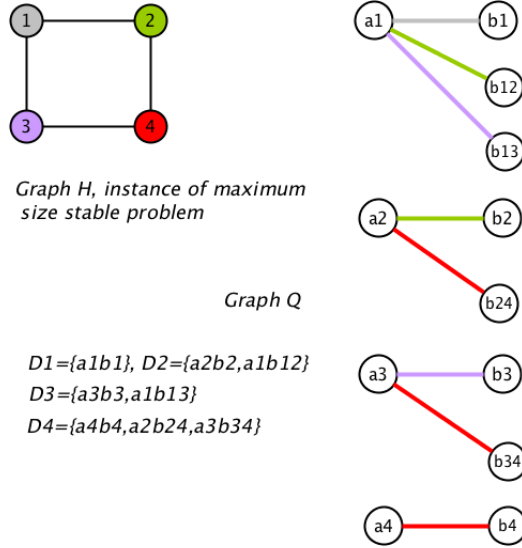


Figure 2: Construction of  $Q$  from  $H$

This instance  $(G, \Pi_E)$  can be constructed in polynomial time from the instance  $H$  of the maximum size problem.

Let  $S = \{h_{i_1}, \dots, h_{i_q}\}$  be any stable, of size  $q$ , in  $H$ . Let us consider the associated obligations to  $S$ :  $E_{i_1}, \dots, E_{i_q}$ . As  $S$  is a stable,  $M_S = \bigcup_{j=1}^q E_{i_j}$  is a matching of  $G$  of size  $qt$ .

Conversely, let  $M$  be any matching of  $G$ , composed of the obligations  $E_{i_1}, \dots, E_{i_q}$ . As  $M$  is a matching of  $G$  of size  $qt$ ,  $S = \{h_{i_1}, \dots, h_{i_q}\}$  is a stable of size  $q$  in  $H$ .

There is a one-to-one mapping between the  $\mathcal{MO}$  of  $(G, \Pi_E)$  and the stables of  $H$ . The sizes are all the same, up to a factor  $t$ . Hence, if an approximation algorithm of ratio  $\alpha$  exists for the  $\mathcal{MO}_{\mathcal{OPT}}$  problem then it would be possible to approximate the maximum size stable problem with a ratio  $\alpha$  via the previously described transformations: transform  $H$  into the instance  $(G, \Pi_E)$ , then apply the approximation algorithm on this instance and then transform its result into a stable of  $H$ . The conservation of the sizes (up to a factor  $t$ ) by these transformations insures the approximation ratio.  $\square$

**Corollary 2** *The  $\mathcal{MO}_{\mathcal{OPT}}$  problem cannot be approximated with a ratio better than  $|V|^{1/2-\epsilon}$  unless  $P = NP$ .*

**Proof.** Theorem 4 shows that it is not possible to approximate the  $\mathcal{MO}_{\mathcal{OPT}}$  problem with a better ratio than the one of the minimum size stable. But this last problem cannot be approximated within  $|V|^{1/2-\epsilon}$  for any  $\epsilon > 0$ , unless  $P = NP$ , see [1].  $\square$

## 6 Hamiltonian path in complete graphs with obligations on edges

Here an instance is  $(G = (V, E), \Pi_E = E_1, \dots, E_k)$  where  $G$  is any connected graph and  $\Pi_E$  a partition of  $E$ .

A *hamiltonian path with obligation* ( $\mathcal{HPO}$ ) of  $(G, \Pi_E)$  is a hamiltonian path of  $G$  satisfying all the constraints of obligations of  $\Pi_E$  (if an edge  $e$  is in the path then all the edges belonging to the same obligation must also be in the path).

**Theorem 5** *Deciding if  $(G, \Pi_E)$  contains a  $\mathcal{HPO}$  is NP-complete, even if  $G$  is a complete graph.*

**Proof.** The problem is in NP. Let  $H = (V, E)$  be any connected graph, instance of the hamiltonian path problem, which is an NP-complete problem, see [6]. Let  $n = |V|$ . We suppose here that  $n \geq 4$  (if  $n$  is smaller then the problem can easily be solved in constant time). The graph for our problem is  $K_n$ , the complete graph on the  $n$  vertices  $V$  of  $H$ . The obligations are the following. For each edge  $uv$  of  $H$ , the edge  $uv$  of  $K_n$  is the only element (singleton) of this part. All the edges  $uv$  outside  $H$  ( $uv \notin E$ ) are grouped in a single obligation  $E_0$ . This instance  $(K_n, \Pi_E)$  can be constructed in polynomial time. We divide our study in two cases.

Case 1:  $E_0$  induces a graph of maximum degree greater than or equal to 3. In this case, the edges of  $E_0$  cannot be in a  $\mathcal{HPO}$  of  $K_n$ . Hence,  $H$  contains a hamiltonian path iff  $(K_n, \Pi_E)$  contains a  $\mathcal{HPO}$ .

Case 2:  $E_0$  induces a graph of maximum degree at most 2. In this case, each vertex  $u$  has degree at least  $n - 2$  in  $H$ . But, by hypothesis  $n \geq 4$ , this implies that the degree in  $H$  of each vertex is at least  $n/2$ . This is the well-known (see [5] for example) Dirac sufficient condition for  $H$  to have a hamiltonian cycle, i.e. also a hamiltonian path. Hence,  $H$  has a hamiltonian path and  $(K_n, \Pi_E)$  has a  $\mathcal{HPO}$ .

In all cases,  $H$  has a hamiltonian path iff  $(K_n, \Pi_E)$  has a  $\mathcal{HPO}$ . □

## References

- [1] AUSIELLO, G., CRESCENZI, P., GAMBOSI, G., KANN, V., MARCHETTI-SPACCAMELA, A., AND PROTASI, M. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.
- [2] CORNET, A., AND LAFOREST, C. Total Domination, Connected Vertex Cover and Steiner Tree with Conflicts. *Discrete Mathematics & Theoretical Computer Science Vol. 19 no. 3* (Dec. 2017).
- [3] CORNET, A., AND LAFOREST, C. Domination problems with no conflicts. *Discrete Applied Mathematics* (Accepted, to appear 2018).
- [4] DELBOT, F., LAFOREST, C., AND PHAN, R. Hardness Results and Approximation Algorithms for Discrete Optimization Problems with Conditional and Unconditional Forbidden Vertices. Research report hal-01257820, Jan. 2016.

- [5] DIESTEL, R. *Graph Theory, 4th Edition*, vol. 173 of *Graduate texts in mathematics*. Springer, 2012.
- [6] GAREY, M. R., AND JOHNSON, D. S. *Computers and intractability*. W.H. freeman New York, 1979.
- [7] KANTÉ, M. M., LAFOREST, C., AND MOMÈGE, B. An exact algorithm to check the existence of (elementary) paths and a generalisation of the cut problem in graphs with forbidden transitions. In *SOFSEM 2013* (2013), vol. 7741 of *Lecture Notes in Computer Science*, Springer, pp. 257–267.
- [8] KANTÉ, M. M., LAFOREST, C., AND MOMÈGE, B. Trees in graphs with conflict edges or forbidden transitions. In *TAMC 2013, Hong Kong* (2013), vol. 7876 of *Lecture Notes in Computer Science*, Springer, pp. 343–354.
- [9] KANTÉ, M. M., MOATAZ, F. Z., MOMÈGE, B., AND NISSE, N. Finding paths in grids with forbidden transitions. In *WG 2015* (2015), vol. 9224 of *Lecture Notes in Computer Science*, Springer, pp. 154–168.
- [10] KOLMAN, P., AND PANGRÁC, O. On the complexity of paths avoiding forbidden pairs. *Discrete Applied Mathematics* 157, 13 (2009), 2871–2876.
- [11] KOVÁČ, J. Complexity of the path avoiding forbidden pairs problem revisited. *Discrete Applied Mathematics* 161, 10 (2013), 1506–1512.
- [12] LAFOREST, C., AND MOMÈGE, B. Some hamiltonian properties of one-conflict graphs. In *IWOCA 2014* (2014), vol. 8986 of *Lecture Notes in Computer Science*, Springer, pp. 262–273.
- [13] LAFOREST, C., AND MOMÈGE, B. Nash-williams-type and chvátal-type conditions in one-conflict graphs. In *SOFSEM 2015* (2015), vol. 8939 of *Lecture Notes in Computer Science*, Springer, pp. 327–338.
- [14] YINNONE, H. On paths avoiding forbidden pairs of vertices in a graph. *Discrete applied mathematics* 74, 1 (1997), 85–92.