



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-01791148v2

HAL Id: hal-01791148

<https://hal.archives-ouvertes.fr/hal-01791148v2>

Submitted on 29 Jun 2018

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

June 29, 2018

Abstract

In this paper we study variants of well-known graph problems: *vertex cover*, *connected vertex cover*, *dominating set*, *total dominating set*, *independent dominating set*, *spanning tree*, *connected minimum weighted spanning graph*, *matching* and *hamiltonian path*. Given a graph $G = (V, E)$, we add a partition Π_V (resp. Π_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 (corresponding to obligations that are singletons). A motivation is to give a general framework and to study the complexity of combinatorial problems coming from systems where elements are interdependent. We propose hardness and approximation results.

Contents

1	Obligations	2
2	Vertex cover with obligations on vertices	3
3	Connected vertex cover with obligations on vertices	4
4	Dominating set with obligations on vertices	6
5	Total dominating set with obligations on vertices	7
6	Independent dominating set with obligations on vertices	8
7	Spanning tree with obligations on edges	10
8	Connected spanning graph of minimum weight with obligations on edges	11
9	Matchings with obligations on edges	13
10	Hamiltonian path in complete graphs with obligations on edges	15
11	Conclusion	16

*christian.laforest@isima.fr, corresponding author

1 Obligations

Systems (production, distribution, network,...) are composed of elements (factories, vehicles, softwares, nodes, links, people...) and must supply outputs (services or goods). These elements are linked (to communicate, to exchange materials,...) and these links form a network modeled as a graph $G = (V, E)$. For the production of outputs or to manage the network, elements must work to complete a task and must be organized. For example, a spanning tree can be useful (to broadcast pieces of information), or a vertex cover (to monitor the links of G) or a dominating set (to monitor the elements). But, in some situations, some sets of these elements must be simultaneously active. This is the case for example when the treatment of a task involves a tool that is distributed on several nodes and to use one of these nodes, all the other ones must also be active. Another case is when nodes are people that are member of teams: if one member of a given team is mobilized for the task then all the other members are also mobilized.

We can model this interdependence between two elements a and b as follows: element a is active (or selected for the task) if and only if element b is active (or selected). We write this dependence $\langle a, b \rangle$, or equivalently $\langle b, a \rangle$. However, by its nature, this relation $\langle \dots \rangle$ is transitive (if $\langle a, b \rangle$ and $\langle b, c \rangle$ then we necessarily get $\langle a, c \rangle$) and reflexive (we have $\langle a, a \rangle$ for any element a). $\langle \dots \rangle$ is then an equivalence relation and it creates a *partition* of the elements, where all the parts are called *obligations* in this article. This means that when an element x is involved, *all* the elements in relation with x in the transitive closure of $\langle \dots \rangle$ are also involved. Note that if an element y is involved in *no* $\langle \dots \rangle$ relation (except with itself), then it is alone in its obligation (singleton $\{y\}$).

We do not address here any specific practical problem but we give a general framework and we treat the underlying combinatorial optimisation problems. Hence, 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* (or *obligation*) of Π_V (resp. Π_E). Now, given G and an associated system Π_V (resp. Π_E) of obligations on vertices (resp. edges), any solution S to a problem on G must *respect (or satisfy) the (constraints on) obligations*, that is must have the following property: 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. As mentioned at the beginning, 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, it is clear that introducing obligations constraints in a classical graph problem $PROB$ leads to a direct generalization of $PROB$ (where obligations are all singletons). But we will see that in most cases the problems with obligations become much harder than the original ones.

In addition to the motivations mentioned above, 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 (x and y are incompatible). 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 [4, 5, 6, 9, 10, 11, 12, 13, 14, 15, 17].

In what follows we give useful notations for the rest of the paper (undefined terms can be found in [7] 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 *neighbors* if G contains the edge uv . The *degree* of a vertex u is its number of neighbors. We call *graph induced* by a 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.

In our paper we reduce some of our problems to well-known NP-complete problems like *set cover*, *X3C* (*exact cover by 3 sets*), *minimum size stable*,... whose strict description can be found for example in the classical textbook [8].

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, Π_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 $V_i \subseteq S$ (i.e. S respects the constraints on obligations).

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, Π_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 Π_V is a singleton, this is the classical NP-complete vertex cover problem [8]. 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 Π_V ($u \in V_i$ 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 Π_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 Π_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 Π_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 *weight vertex cover* S_c in G_c (i.e. a vertex cover of G_c whose total weight is at most two times the minimum one. This approximation can be done in polynomial time, see [1]).

3. Return $S = \bigcup_{i: v_i \in S_c} V_i$ (for each vertex v_i of S_c , put the corresponding obligation V_i in S).

Theorem 1 *The algorithm described above is a 2-approximation algorithm for the $\mathcal{VCO}_{\mathcal{OPT}}$ 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, Π_V) is a union of some parts of Π_V . We construct now a one-to-one correspondance respecting the weights and the sizes between the \mathcal{VCO} of (G, Π_V) and the weighted vertex covers of G_c .

- Let S be any \mathcal{VCO} of (G, Π_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, Π_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 $\mathcal{VCO}_{\mathcal{OPT}}$ of (G, Π_V) . Hence the proposed algorithm is a 2-approximation algorithm for the $\mathcal{VCO}_{\mathcal{OPT}}$ 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 \mathcal{CVCO} , *connected vertex cover with obligations*, S of the instance (G, Π_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 Π_V .

It is easy to see that any instance $(G = (V, E), \Pi_V)$ always contains at least a \mathcal{VCO} , namely $S = V$ since G is connected. A $\mathcal{CVCO}_{\mathcal{OPT}}$ is a \mathcal{CVCO} of minimum size. Constructing a $\mathcal{CVCO}_{\mathcal{OPT}}$ is a hard problem, even if Π_V is a partition of singletons (in this case this is the classical NP-complete connected vertex cover problem [8]).

Theorem 2 *Any α -approximation algorithm for the $\mathcal{CVCO}_{\mathcal{OPT}}$ problem can be transformed into a 2α -approximation algorithm for the minimum size 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 : $A = \bigcup_{i=1}^k X_i$. 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 the element 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, Π_V) can be constructed in polynomial time from the instance (A, X) . Consider now the following one-to-one mapping between the \mathcal{CVCO} of (G, Π_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 Π_V (S is composed of a union of obligations of Π_V). The size of S is: $|S| = n + 1 + t(n + 1) = (n + 1)(t + 1)$.

Consider now any \mathcal{CVCO} S of (G, Π_V) . As S satisfies the constraints on obligations, it is composed of a 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 $\mathcal{CVCO}_{\mathcal{OPT}}$ can be approximated with a ratio α in polynomial time. Then, for any instance (A, X) one can: construct the associated instance (G, Π_V) , then use this approximation algorithm to construct a α -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 2α -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}_{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 Dominating set with obligations on vertices

In this section, an instance is $(G = (V, E), \Pi_V = V_1, \dots, V_k)$ where G is a graph and Π_V is a partition of V . A *dominating set with obligations* S (\mathcal{DO}) of (G, Π_V) satisfies:

- S dominates G (for any $u \in V - S$, u has at least a neighbor in S),
- S respects the constraints of obligations of Π_V .

Remark 1 *There is always a \mathcal{DO} : V , the set of vertices of G .*

The minimization problem is NP-complete and cannot be approximated with a better ratio than $c \cdot \log |V|$ for any $c > 0$ (unless $P = NP$): indeed when the obligations are all singletons, we get the classical dominating set problem having this bound on approximation ratio, see [16].

Let us show now that it is possible to construct a $\mathcal{O}(\log(|V|))$ -approximation for our problem of dominating set with obligations. For that purpose we reduce it to the *weighted set cover* for which there is such an approximation ratio $\mathcal{O}(\log(|V|))$, see [3].

Theorem 3 *Given (G, Π_V) , it is possible to approximate an optimal \mathcal{DO} with ratio $\mathcal{O}(\log(|V|))$.*

Proof. From instance $(G = (V, E), \Pi_V)$, we construct (U, S, w) an instance of the weighted set cover. Let $U = V$. For any obligation $V_i \in \Pi_V$, we construct a set S_i composed of the union of the closed neighborhoods of vertices of V_i (the closed neighborhood of x is the set of neighbors of x plus x itself). We can remark that a set S_i contains exactly the vertices dominated by V_i . The weight of this set is the size of the obligation (which is, in general, different from the size of S_i) i.e. $w(S_i) = |V_i|$. The family S of sets of the instance (U, S, w) is composed of all these S_i . Figure 1 shows an example of construction of S_1 from V_1 . Here, the set constructed has weight 3 (the size of V_1) and dominates V_1 and its neighbors.

We construct now a one-to-one mapping between the dominating sets with obligations of (G, Π_V) and the set covers of (U, S, w) .

Let D be any dominating set with obligations of (G, Π_V) . As D respects the obligations, D is a union of obligations V_{i_1}, \dots, V_{i_t} . Construct $C = \bigcup_{j=1}^t S_{i_j}$. As D is a dominating set of G , each vertex u of V is dominated by a vertex in a set V_{i_j} and, hence, each element u of $U = V$ is covered by S_{i_j} , i.e. by C . We also have $|D| = \sum_{j=1}^t |V_{i_j}| = \sum_{j=1}^t w(S_{i_j}) = w(C)$.

Reciprocally, let $C = S_{i_1}, \dots, S_{i_l}$ be a set cover of (U, S, w) . Construct $D = \bigcup_{i=1}^l V_{i_i}$. As C is a set cover, each element v is covered by at least a S_{i_j} , and then each corresponding vertex v is dominated by itself if it is in V_{i_j} , or by one of its neighbors in V_{i_j} , D is then a dominating set of G . Moreover, by construction, D respects the obligations. As previously, $w(C) = \sum_{j=1}^l w(S_{i_j}) = \sum_{j=1}^l |V_{i_j}| = |D|$.

The final result follows from this polynomial transformation and one-to-one mapping, preserving size/weight and the result of [3]. \square

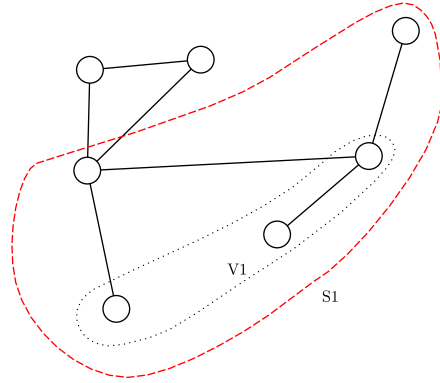


Figure 1: Construction of S_1 from V_1 .

5 Total dominating set with obligations on vertices

In this section, an instance is $(G = (V, E), \Pi_V = V_1, \dots, V_k)$ where G is a graph and Π_V is a partition of V . A *total dominating set with obligations* S (\mathcal{TDO}) of (G, Π_V) satisfies:

- S totally dominates G (for any $u \in V$, u has at least a neighbor in S),
- S respects the constraints of obligations of Π_V .

Remark 2 $(G = (V, E), \Pi_V)$ contains a \mathcal{TDO} (the set V) iff G has no isolated vertices.

An *optimal \mathcal{TDO}* is a \mathcal{TDO} of minimum size. The minimization problem is NP-complete and cannot be approximated with a better ratio than $c \cdot \log |V|$ for any $c > 0$: indeed when the obligations are all singletons, we get the classical total dominating set problem having this bound on approximation ratio, see [2].

Let us show now that it is possible to construct a $\mathcal{O}(\log(|V|))$ -approximation for our problem of total dominating set with obligations. For that purpose we reduce it to the *weighted Set Cover* for which there is such an approximation ratio $\mathcal{O}(\log(|V|))$, see [3]. The reduction is very similar to the one of Section 4, however, to be complete we include it.

Theorem 4 Given (G, Π_V) , it is possible to approximate an optimal \mathcal{TDO} with ratio $\mathcal{O}(\log(|V|))$.

Proof. Let $(G = (V, E), \Pi_V)$ be any instance of our problem. We suppose here that G has no isolated vertices, i.e., $(G = (V, E), \Pi_V)$ contains a \mathcal{TDO} . We construct (U, S, w) an instance of the weighted set cover. Let $U = V$. For any obligation $V_i \in \Pi_V$, we construct a set S_i composed of the union of the *open* neighborhoods of vertices of V_i (the open neighborhood of a vertex x is the set of neighbors of x , without x itself). The weight of this set is the size of the obligation (which is, in general, different from the size of S_i) i.e. $w(S_i) = |V_i|$. The family S of sets of the instance (U, S, w) is composed of all these S_i . Figure 2 shows an example of construction of S_1 from V_1 . Here, the set constructed has weight 3 (the size of V_1) and dominates the open neighborhood of V_1 . We can remark that all the vertices of V_1 are not necessarily in S_1 .

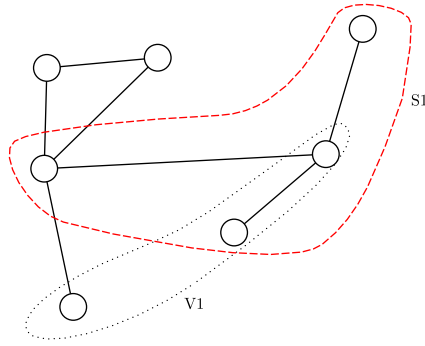


Figure 2: Construction of S_1 from V_1 .

We construct now a one-to-one mapping between the dominating sets with obligations of (G, Π_V) and the set covers of (U, S, w) .

Let D be any total dominating set with obligations of (G, Π_V) . As D respects the obligations, D is a union of obligations V_{i_1}, \dots, V_{i_t} . Construct $C = \bigcup_{j=1}^t S_{i_j}$. As D is a dominating set of G , each vertex u of V is dominated by a vertex in a set V_{i_j} and, hence, each element u of $U = V$ is covered by S_{i_j} , i.e. by C . We also have $|D| = \sum_{j=1}^t |V_{i_j}| = \sum_{j=1}^t w(S_{i_j}) = w(C)$.

Reciprocally, if $C = S_{i_1}, \dots, S_{i_l}$ is a set cover of (U, S, w) , let $D = \bigcup_{i=1}^l V_{i_i}$. As C is a set cover, each element v is covered by at least a S_{i_j} , and then each corresponding vertex v is dominated by a neighbor in V_{i_j} , D is then a total dominating set of G . Moreover, by construction, D respects the obligations. As previously, $w(C) = \sum_{j=1}^l w(S_{i_j}) = \sum_{j=1}^l |V_{i_j}| = |D|$.

The final result follows from this polynomial transformation and polynomial one-to-one mapping, preserving size/weight and the result of [3]. \square

6 Independent dominating set with obligations on vertices

In this section, an instance is $(G = (V, E), \Pi_V = V_1, \dots, V_k)$ where G is a graph and Π_V is a partition of V . An *independent dominating set with obligations* S ($\mathcal{ID}\mathcal{O}$) of (G, Π_V) satisfies:

- S dominates G (for any $u \in V - S$, u has at least a neighbor in S),
- S is a stable of G (no edges between vertices of S),
- S respects the constraints of obligations of Π_V .

In this particular variant of domination, a solution is not always guaranteed.

Theorem 5 *Determining if (G, Π_V) contains an $\mathcal{ID}\mathcal{O}$ is NP-complete.*

Proof. The problem is clearly in NP. Let (X, Z) be a X3C instance (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

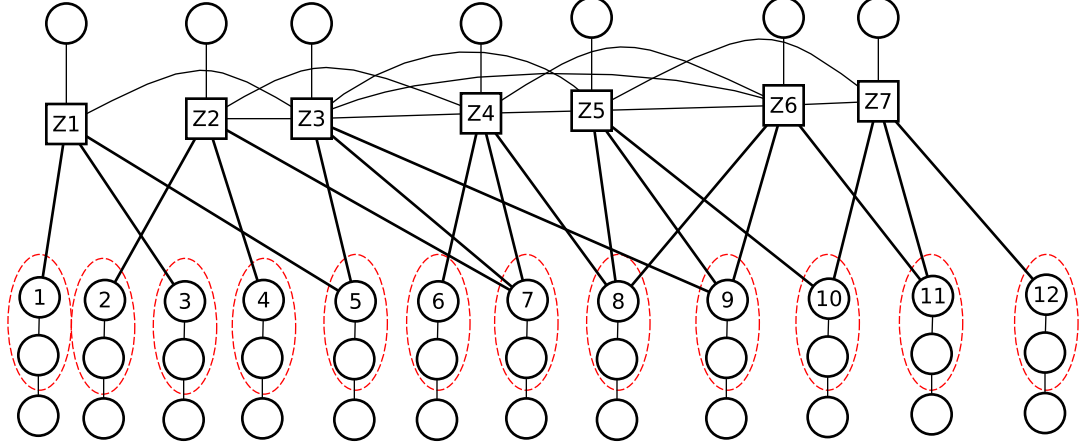


Figure 3: Construction of (G, Π_V) from (X, Z) .

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 (each element of X is in exactly one subset of the solution). This problem is NP-complete, see [8].

Let us construct an instance of our problem from (X, Z) . For each element x of X , a P_3 (a path with 3 vertices) is created and one extremity is called the *vertex representing the element*. For each subset z of Z , a path P_2 is created and one extremity is called the *vertex representing the subset*. Additional edges are added between:

- Each vertex representing a subset and each vertex representing an element inside this subset.
- Each pair of vertices representing subsets whose associated subsets have non-empty intersection.

For each element x , an obligation containing the vertex representing x and its neighbor in its P_3 is created. They are called *obligations of elements*. All the other obligations are singletons. An example of result of this (polynomial) construction is given in Figure 3.

Let D be an independent dominating set respecting the obligations of (G, Π_V) . D contains no obligation of elements because these obligations are between two vertices linked by an edge. Hence, each vertex representing an element can only be dominated by vertices representing subsets. Let S be the family of subsets corresponding to the vertices representing subsets of D . Then as each vertex representing an element is dominated by D , each element is covered by S . Moreover, as only subsets with non-empty intersection are neighbors, D is an independent set, and the subsets of S are pairwise disjoint: S is then an exact cover of (X, Z) .

Now, let S be an exact cover of (X, Z) . Let us construct D . For each Z_i , the corresponding vertex is added to D iff Z_i is in S . Otherwise, the neighbor of Z_i in the P_2 is added to D . Also add to D all the vertices that are the opposite extremities of the vertices representing an element in each P_3 . It is easy to see that D is an independent set. Moreover, D respects the obligations (since each vertex of D is in a singleton obligation). Finally, each element is covered by S : each vertex representing an element is then dominated by a vertex representing a subset. The paths P_2 are dominated either by the vertex representing the subset or by the other extremity. For each

path P_3 the vertex, opposite extremity of the vertex representing the element, is in D and covers itself and its unique neighbor. D is then an independent dominating set, respecting the obligations of (G, Π_V) . \square

7 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 Π_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 (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 6 *Deciding if $(G = (V, E), \Pi_E = E_1, \dots, E_k)$ contains a 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} pairwise 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 [8]. 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 new vertices. Then these $\lfloor k/2 \rfloor$ new vertices are connected two-by-two by new vertices, 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 if necessary) 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 4: 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.

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*.

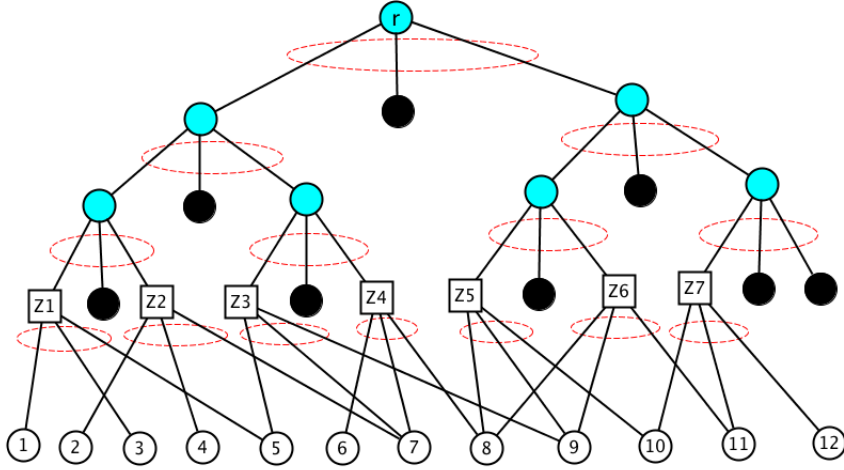


Figure 4: Construction of G from a $X3C$ instance.

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 Π_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 neighbor of exactly one vertex Z_{i_j} and each vertex Z_i is connected to the other vertices *via* the tree T_r). This tree respects the obligations of Π_E and is then a \mathcal{TSO} of (G, Π_E) .

Conversely, suppose that the instance (G, Π_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 neighbor 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 disjoint and is then a solution for the $X3C$ instance. \square

8 Connected spanning graph of minimum weight with obligations on edges

In this section $G = (V, E)$ is a weighted connected graph: each edge $e \in E$ has a weight $w(e) > 0$. The obligations form a partition $\Pi_E = E_1, \dots, E_k$ of E . The objective is to extract from G a subset S of edges, inducing a connected graph spanning all the vertices of V , having a minimum weight and respecting the obligations. Such an object is called a $CSGO_{OPT}$ (*Minimum Weight Connected*

Spanning Graph with obligations). We call *CSGO* a *Connected spanning Graph with obligations* (a *CSGO_{OPT}* is a minimum weight *CSGO*).

We can note that, because of the obligations, a *CSGO* is not necessarily a tree. Indeed, if each obligation induced a cycle for example, no spanning tree is possible. We can remark that since G is connected, G itself is a *CSGO* of (G, Π_E) (the problem always has a solution) and that if Π_E only contains singletons, this is the traditional minimum weight spanning tree problem that can be polynomially solved with the classical Prim algorithm for example.

Theorem 7 *Let $(G = (V, E), \Pi_E = E_1, \dots, E_k)$ be an instance with G a weighted connected graph. Determining if there is a *CSGO* of weight at most $|V| - 1$ is NP-complete, even if:*

- G is bipartite, of maximum degree 4,
- all the weights are 1 and
- each obligation induces a star with 3 edges.

Proof. This problem is clearly in NP. Any spanning graph contains at least $n - 1$ edges, with $n = |V|$. Hence, in the case where each edge has weight 1, there is no *CSGO* with weight strictly less than $n - 1$. Deciding if there exists a *CSGO* of weight at most $n - 1$ is then strictly equivalent to decide if there exists a *TSO* in this instance, which is NP-complete, even if G verifies the hypotheses, thanks to Theorem 6. \square

Theorem 7 shows that deciding whether an instance contains a *CSGO* is NP-complete, even if all the weights are equal. The next result shows that there is no constant approximation algorithm for the weighted case.

Theorem 8 *Any α -approximation algorithm for the *CSGO* problem in bipartite graphs where obligations induce stars can be transformed into a α -approximation algorithm for the minimum size set cover problem.*

Proof. Let $(X = \{x_1, \dots, x_n\}, F = \{F_1, \dots, F_k\})$ be any instance of the set cover problem. Let us construct an instance of *CSGO*. Put in V the n vertices corresponding to x_1, \dots, x_n , k vertices corresponding to sets F_1, \dots, F_k and an additional new vertex r . Link r to each vertex F_i and link each vertex F_i to all the x_j such that $x_j \in F_i$. Clearly the graph obtained is bipartite. Put in a same obligation, noted O_0 , all the incident edges of r and assign a weight ϵ/k on each edge of O_0 (where ϵ can be as small as desired). For each F_i , put all incident edges to F_i , except the one between F_i and r , in an obligation noted O_i and assign to each such edge of O_i a weight $1/|F_i|$. Hence, the total weight of each obligation is 1, except O_0 with weight ϵ . Each obligation induces a star. An exemple of this construction is given in Figure 5.

We show now that each solution of the set cover problem can polynomially be transformed in a solution of equivalent weight for the *CSGO* problem, and reciprocally.

Let S be a solution of the set cover, of size t . We construct C the set of edges as follows: put O_0 in C , and for each $0 < i \leq k$, put O_i in C iff F_i is in S . The vertices r and F_i are connected in C (via O_0). Each element x_j is covered by a set F_i of the set cover: the corresponding vertex x_j is connected to the vertex F_i via the obligation O_i , hence C is a *CSGO*. C contains O_0 and t other obligations, its weight is then $t + \epsilon$.

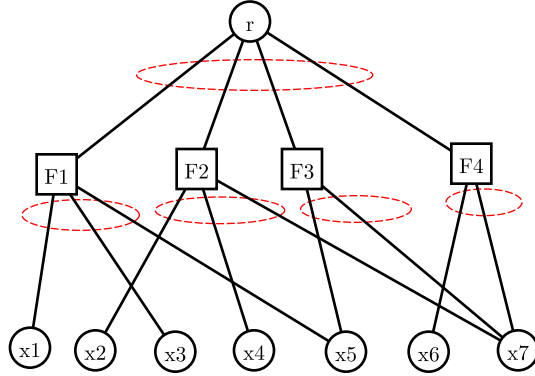


Figure 5: Construction of the $CSGO$ instance. Dashed ellipses represent the obligations.

Let now C be a $CSGO$. C contains O_0 (to connect r) and t other obligations. Its weight is then $t + \epsilon$. Let us construct a solution S of the set cover. For each $0 < i \leq k$, put F_i in S iff $O_i \in C$. Let x_j be an element of X . The corresponding vertex is connected by an edge which is an element of an obligation O_i . Hence, the corresponding set F_i belongs to S and the element is covered, S is then a set cover. Moreover, the size of S is t .

As ϵ can be arbitrary small, using these transformations, one can use a α -approximation algorithm for our $CSGO$ to create a α -approximation algorithm for the set cover problem. \square

Corollary 2 *The minimum weight $CSGO$ cannot be approximated with a constant approximation ratio (unless $P = NP$), even if G is a bipartite graph and if each obligation induces a star.*

Proof. Theorem 8 shows that it is not possible to approximate the $CSGO$ problem with a better ratio than the one of the minimum size set cover, even in bipartite graphs where each obligation induces a star. But this last problem cannot be approximated within $c \log(c)$ for some c , unless $P = NP$, see [1]. \square

At this point, an open question remains: determine if there is a constant approximation ratio for the unweighted case (i.e. for unitary weights).

9 Matchings with obligations on edges

In this section, an instance in (G, Π_E) where $G = (V, E)$ is any 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, Π_E) is a matching of G (set of pairwise non incident edges of G) respecting the obligations (for any edge e of M , if $e \in E_i$ then $E_i \subseteq M$).

It is polynomial to determine if (G, Π_E) contains a \mathcal{MO} . Indeed, there is a non empty \mathcal{MO} iff at least an obligation E_i induces a matching. From this we can simplify an instance (G, Π_E) : if a part E_i of Π_E induces a graph in which a vertex has more than one neighbor then E_i can be deleted from Π_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 $\mathcal{MO}_{\mathcal{OPT}}$.

Theorem 9 *Let $(G = (V, E), \Pi_E = E_1, \dots, E_k)$ be an instance where each E_i induced a matching of G . Any α -approximation algorithm for the $\mathcal{MO}_{\mathcal{OPT}}$ problem can be transformed into a α -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 .

For each edge $h_i h_j$ of H , we create a new P_3 (path with 3 vertices) associated to this edge. The union of these $|E_H|$ pairwise disjoint paths form a graph noted Q (not yet the final graph G). Now, for each i , $1 \leq i \leq n$, we create D_i a subset of edges of Q as follows. For each edge $h_i h_j$ of H , put an edge of the associated P_3 path in D_i and the other one in D_j . These n sets D_1, \dots, D_n form a partition of the edges of Q and each D_i is a matching. Figure 6 gives an example of this construction.

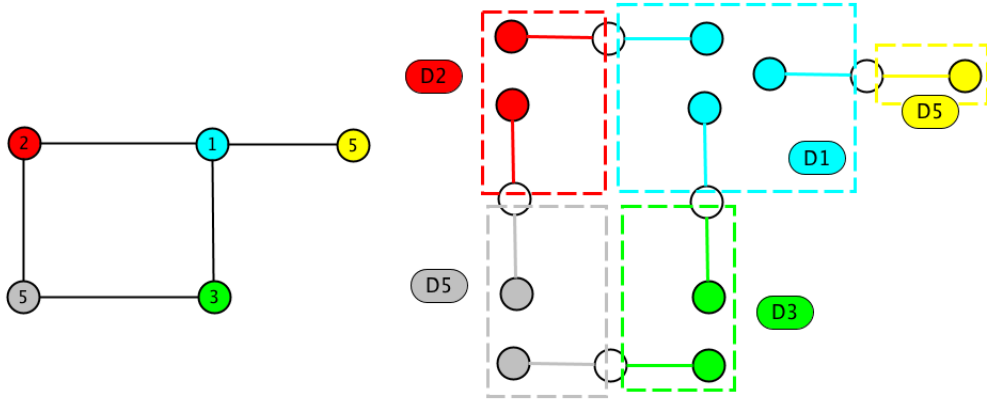


Figure 6: Construction of Q (right) from H (left). Sets D_i are in dashed boxes.

The sets D_i can have different sizes. Let D_a be the one of maximum size (this corresponds to the maximum degree of H). 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 $\Pi_E = E_1, \dots, E_n$ obtained by the previous operation of homogenization of size. 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 ,
- $E_i \cup E_j$ is a matching of G iff $h_i h_j \notin E$.

This instance (G, Π_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, Π_E) and the stables of H . The sizes are all the same, up to a factor t . Hence, if an approximation algorithm of ratio α exists for the \mathcal{MO}_{OPT} problem then it would be possible to approximate the maximum size stable problem with a ratio α via the previously described transformations: transform H into the instance (G, Π_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 3 *The \mathcal{MO}_{OPT} problem cannot be approximated with a ratio better than $|V|^{1/2-\epsilon}$ unless $P = NP$.*

Proof. Theorem 9 shows that it is not possible to approximate the \mathcal{MO}_{OPT} problem with a better ratio than the one of the maximum size stable. But this last problem cannot be approximated within $|V|^{1/2-\epsilon}$ for any $\epsilon > 0$, unless $P = NP$, see [1]. \square

10 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 Π_E a partition of E .

A *hamiltonian path with obligation (HPO)* of (G, Π_E) is a hamiltonian path of G (a path of $|V| - 1$ edges, spanning V) satisfying all the constraints of obligations of Π_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 10 *Deciding if (G, Π_E) contains a 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 [8]. 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, Π_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 HPO of K_n . Hence, H contains a hamiltonian path iff (K_n, Π_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 [7] 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, Π_E) has a \mathcal{HPO} .

In all cases, H has a hamiltonian path iff (K_n, Π_E) has a \mathcal{HPO} . \square

11 Conclusion

In this paper we shown that adding obligations drastically increases the approximation ratio of classical graph problems. This is the case for the connected vertex cover with obligations that has no constant approximation ratio algorithm (while there is a 2-approximation algorithm for the original problem), the minimum connected weighted spanning graph and the maximum size matching. For the last two problems, the classical versions are polynomial but the version with obligations are as hard as set cover or maximum stable problems. For some other problems, the situation is even worst: it becomes NP-complete to know whether there is a solution, regardless of its size (while it is trivial or polynomial in the original problem). This is the case for the following problems: independent dominating set, spanning tree and hamiltonian path in complete graphs. For the dominating and total dominating set problem, the approximation ratios are almost the same with or without obligations constraints (but these ratios are not constant). Only the vertex cover problem keeps the same constant approximation ratio 2.

One might imagine that a perspective could be to refine our results by studying more specific/restricted instances. Unfortunately in some cases, the problem is “equivalent” to another hard problem (minimum set cover, maximum size stable problem) that already received a lot of attention and improving them is known as a hard challenge in itself since a long time. In other cases, the instances for which our problem is hard are basic in a sense: bipartite graphs of maximum degree 4 and very small obligation sizes for the spanning tree problem, complete graph for the hamiltonian path problem.

Other combinatorial problems can be studied with our framework. But our results show that dealing with obligations can lead to very complex problems that could be unsolvable. Organizing practical systems with obligations should be done with a lot of attention.

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] CHLEBÍK, M., AND CHLEBÍKOVÁ, J. Approximation hardness of dominating set problems. In *Algorithms – ESA 2004* (Berlin, Heidelberg, 2004), S. Albers and T. Radzik, Eds., Springer Berlin Heidelberg, pp. 192–203.

- [3] CHVATAL, V. A greedy heuristic for the set-covering problem. *Mathematics of operations research* 4, 3 (1979), 233–235.
- [4] 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).
- [5] CORNET, A., AND LAFOREST, C. Domination problems with no conflicts. *Discrete Applied Mathematics* 244 (2018), 327–338.
- [6] 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.
- [7] DIESTEL, R. *Graph Theory, 4th Edition*, vol. 173 of *Graduate texts in mathematics*. Springer, 2012.
- [8] GAREY, M. R., AND JOHNSON, D. S. *Computers and intractability*. W.H. freeman New York, 1979.
- [9] 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.
- [10] 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.
- [11] 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.
- [12] KOLMAN, P., AND PANGRÁC, O. On the complexity of paths avoiding forbidden pairs. *Discrete Applied Mathematics* 157, 13 (2009), 2871–2876.
- [13] KOVÁČ, J. Complexity of the path avoiding forbidden pairs problem revisited. *Discrete Applied Mathematics* 161, 10 (2013), 1506–1512.
- [14] 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.
- [15] 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.
- [16] RAZ, R., AND SAFRA, S. A sub-constant error-probability low-degree test, and a sub-constant error-probability pcp characterization of np. In *Proceedings of the twenty-ninth annual ACM symposium on Theory of computing* (1997), ACM, pp. 475–484.

- [17] YINNONE, H. On paths avoiding forbidden pairs of vertices in a graph. *Discrete applied mathematics* 74, 1 (1997), 85–92.