Minimum d-blockers and d-transversals in graphs

Marie-Christine Costa, Dominique de Werra, Christophe Picouleau

Minimum d-blockers and d-transversals in graphs

M.-C. Costa * D. de Werra † C. Picouleau ‡

May 13, 2014

Abstract

We consider a set V of elements and an optimization problem on V: the search for a maximum (or minimum) cardinality subset of V verifying a given property $\mathcal{P}$. A d-transversal is a subset of V which intersects any optimum solution in at least d elements while a d-blocker is a subset of V whose removal deteriorates the value of an optimum solution by at least d. We present some general characteristics of these problems, we review some situations which have been studied (matchings, s-t paths and s-t cuts in graphs) and we study d-transversals and d-blockers of stable sets or vertex covers in bipartite and in split graphs.

Keywords: transversal, blocker, cover, bipartite graph, split graph, s-t path, s-t cut, stable set, bilevel programming.

*Ecole Nationale Supérieure des Techniques Avancées-Paristech and CEDRIC laboratory, Paris (France). Email: marie-christine.costa@ensta.fr

‡Conservatoire National des Arts et Métiers, CEDRIC laboratory, Paris (France). Email: christophe.picouleau@cnam.fr

†Ecole Polytechnique Fédérale de Lausanne, Lausanne (Switzerland). Email: dominique.dewerra@epfl.ch

The first version of this paper was completed on January 21, 2010 for the 168^{th} *anniversary of the birth of Alferd Packer to whom it is dedicated.*

1 Introduction

In many instances occurring in practical situations or in a game theoretic context one faces the problem of preventing some move of an opponent with a minimum effort. Such problems have been studied from a theoretical point of view and, in very special cases where a graph representation may be used, it leads to interesting combinatorial problems (see for instance [16], [15]). The purpose of this paper is to discuss some of these problems and to present a few variations with some remarks on their complexity and on open questions which would lead to further developments. In the next section we shall provide a rather general formulation of the concepts of d-blockers and d-transversals with a brief review of earlier results in this direction. In Section 3, we present a general formulation of both problems as bi-level optimization programs. Then the following sections will be devoted to various special cases of applications of these concepts: combinatorial optimization problems in matroids, s-t paths and s-t cuts. In Section 7 we will consider the case of maximum stable sets in split graphs and in bipartite graphs. Finally Section 8 will give concluding remarks and suggestions for other research avenues.

2 Preliminaries

Let us give the general formulation of the various concepts which will be needed in the sequel. We are given a finite ground set $V = \{v_1, ..., v_p\}$ and a property $\mathcal{P}$ defined on V. Let $\mathcal{C}_V$ be the collection of all subsets $C \subseteq V$ satisfying $\mathcal{P}$.
A subset $C \in \mathcal{C}_V$ will be called *maximum* if $|C| = max\,\{|C'| : C' \in \mathcal{C}_V\}$; we denote this cardinality by $\nu(\mathcal{C}_V)$. $C \in \mathcal{C}_V$ will be called *minimum* if $|C| = min\,\{|C'| : C' \in \mathcal{C}_V\}$; we denote this cardinality by $\lambda(\mathcal{C}_V)$.
A subset $C \in \mathcal{C}_V$ will be called *maximal* (resp. *minimal*) if it is inclusionwise maximal (resp. minimal) for property $\mathcal{P}$.
Given a positive integer $d \leq \nu(\mathcal{C}_V)$, a *d-transversal* is a subset $T \subseteq V$ with $|T \cap C| \geq d$ for any maximum $C \in \mathcal{C}_V$ (or any minimum $C \in \mathcal{C}_V$ with $d \leq \lambda(\mathcal{C}_V)$, depending on the context). In the literature, a d-transversal is sometimes called a *d-cover*. If one requires that $|T \cap C| = 1$ for each maximum $C \in \mathcal{C}_V$, then T is sometimes called a *System of Distinct Representatives* (or a *transversal* in [19] or [16]).
A companion concept is the *d-blocker* whose removal deteriorates the optimum size of elements in $\mathcal{C}_V$ by at least d. For subsets C of maximum size, the d-blocker is a subset $B \subseteq V$ such that for any $C \in \mathcal{C}_{V-B}$ we have $|C| \leq \nu(\mathcal{C}_V) - d$. So, we have $\nu(\mathcal{C}_{V-B}) \leq \nu(\mathcal{C}_V) - d$. When we consider subsets C of minimum size, a d-blocker B is such that $\lambda(\mathcal{C}_{V-B}) \geq \lambda(\mathcal{C}_V) + d$. Notice that d-blockers are interesting for problems with ground set V such that the removal of any subset $A \subseteq V$ may deteriorate the optimum size (or value) of the subsets C in $\mathcal{C}_{V-A}$.
One may also consider the search of a subset $P \subseteq V$ such that for all maximum (or minimum) $C \in \mathcal{C}_V$ we have $|C \cap P| \leq d$. We call it a *d-packer*. More on the concept of packer can be found in [20]. In fact, P is a maximum d-packer if and only if

$T = V - P$ is a minimum $(\nu(\mathcal{C}_V) - d)$-transversal ($d = 0, ..., \nu(\mathcal{C}_V) - 1$): recall that if $|C| = \nu(\mathcal{C}_V)$, then $|C \cap P| \leq d \Leftrightarrow |C \cap (V - P)| \geq \nu(\mathcal{C}_V) - d \Leftrightarrow |C \cap T| \geq \nu(\mathcal{C}_V) - d$; moreover, P is maximum if and only if $T = V - P$ is minimum. Note that, obviously, a maximum $\nu(\mathcal{C}_V)$-packer P would be equal to V and $T = V - P$ would be a minimum 0-transversal equal to $\emptyset$. In the following, we will not consider d-packers anymore but only d-transversals with $d = 1, ..., \nu(\mathcal{C}_V)$.
We shall denote by $\beta_d(G)$ (respectively $\tau_d(G)$) the minimum cardinality of a d-blocker (resp. d-transversal) in G.
The following statements can be found in [26] where they are formulated for the special case in which $\mathcal{C}_V$ is the collection of all matchings in a graph.

Fact 2.1 *For any $d \geq 1$, a d-blocker is a d-transversal.*

Proof: Let B be a d-blocker and then assume that B is not a d-transversal: there is a maximum C in $\mathcal{C}_V$ with $|C \cap B| < d$, then $|C \cap (V - B)| \geq \nu(\mathcal{C}_V) - d + 1$, so we cannot have $\nu(\mathcal{C}_{V-B}) \leq \nu(\mathcal{C}_V) - d$ and B is not a d-blocker. □

Fact 2.2 *A set $B \subseteq V$ is a 1-blocker if and only if it is a 1-transversal.*

Proof: A 1-blocker is a 1-transversal from Fact 2.1. Let B be a 1-transversal. For every maximum C in $\mathcal{C}_V$ we have $|C \cap B| \geq 1$ so $|C \cap (V - B)| \leq \nu(\mathcal{C}_V) - 1$ and B is a 1-blocker. □

The reader is referred to [3] for all graph theoretical terms not defined here.
It has been shown that finding a minimum d-transversal of all maximum matchings in a bipartite graph is $\mathcal{NP}$-hard for any fixed $d \geq 1$ [26], while it can be found in polynomial time in trees and grid graphs [18]. In [16], Lee reviews and gives some new results for the minimum d-transversal problem in the case where $\mathcal{C}_V$ is the set of maximum cliques of a given graph G; a d-transversal is called there a d-fold maximum clique transversal set. Polynomial algorithms are given to find $\tau_d(G)$ in balanced, split, strongly chordal, triangle free graphs (for fixed $d \geq 2$); it is shown that finding $\tau_d(G)$ is $\mathcal{NP}$-hard in (doubly-)chordal and in planar graphs.

3 Formulation as bi-level programming problems

We will start from the general formulation of a bilevel optimization problem given in [9] for real variables x and y:

$(BLPP)$
$Min_{x \in X, y}\ F(x, y)$
 s.t. $G(x, y) \leq 0$
  $Min_y\ f(x, y)$
  s.t. $g(x, y) \leq 0$
$x \in \Re^n\ y \in \Re^m$

Such models are closely related to game theory: the "leader" chooses a strategy in a set X and the "follower" has an (opposite) strategy set y corresponding to each $x \in X$.
We are going to show that the searches for a minimum d-blocker B and a minimum d-transversal T can be formulated, in a natural way, as special binary bilevel programs. Note that we present one of the possibly equivalent 0-1 formulations for each problem. Consider a maximisation problem defined by a ground set $V = \{v_1, v_2, ..., v_n\}$, a property $\mathcal{P}$ and a collection $\mathcal{C}_V$, $\nu(G)$ being the cardinality of a maximum C in $\mathcal{C}_V$ (see Section 2). We denote by x the vector of variables "controlled" by the leader: $x_i = 1$ if $v_i \in B$ (resp. T), $x_i = 0$ otherwise. We denote by y the vector of variables controlled by the follower: $y_i = 1$ if $v_i \in C$; $y_i = 0$ otherwise.
Let us consider first the d-blocker problem. The associated 0-1 program is the following :

$(BLBlock)$

$Min_x \sum_{i=1,...,n} x_i \quad (1)$

s.t. $\sum_{i=1,...,n} y_i \leq \nu(G) - d \quad (1.1)$

$Max_y \sum_{i=1,...,n} y_i \quad (2)$
s.t. $y_i \leq 1 - x_i$ for all $i = 1, ..., n \quad (2.1)$
$C = \{v_i \ s.t. \ y_i = 1\} \in \mathcal{C}_V \quad (2.2)$
$x \in \{0,1\}^n \ y \in \{0,1\}^n$

The aim of the leader is to select a minimum set B of "elements" to delete (for instance, edges for the matching problem or vertices for the stable set problem), i.e. a minimum set of variables x_i to fix to 1 (objective function (1)), in order to impose a maximum value $\nu(G) - d$ to the best choice of the follower (constraint (1.1)). For each vector x, the follower will choose a maximum set C in $V - B$ verifying $\mathcal{P}$, i.e. the maximum number of variables y_i (objective function (2)) corresponding to undeleted elements (constraints (2.1)) and verifying the constaints associated with property $\mathcal{P}$ (constraints (2.2)). The constraints (2.2) are those corresponding to the considered maximization problem: for instance, for the stable set problem in a graph $G = (X, E)$, $y_i + y_j \leq 1$ for all $(i, j) \in E$.
Consider now the d-transversal problem. The associated $0-1$ program is the following:

$(BLTrans)$

$Min_x \sum_{i=1,...,n} x_i \quad (3)$

s.t. $\sum_{i=1,...,n} x_i y_i \geq d \quad (3.1)$

$Min_y \sum_{i=1,...,n} x_i y_i \quad (4)$
s.t. $\sum_i y_i = \nu(G) \quad (4.1)$
$C = \{v_i \ s.t. \ y_i = 1\} \in \mathcal{C}_V \quad (4.2)$

$x \in \{0,1\}^n$ $y \in \{0,1\}^n$

The aim of the leader is to select a minimum set T of elements to control, i.e. a minimum set of variables x_i to fix to 1 (objective function (3)) in order to be sure to control at least d elements among those selected by the follower (constraint (3.1)). For each vector x, the follower will choose a set of $\nu(G)$ elements verifying $\mathcal{P}$ and containing a minimum number of controlled elements, i.e. $\nu(G)$ variables y_i to fix to 1 (constraint (4.1)) such that $\mathcal{P}$ is verified (constraints (4.2)) and the number of indices i verifying $y_i = x_i = 1$ is minimum (objective function (4)).

Note that our two bilevel programs are special cases where there is only one constraint (3.1) and (1.1), respectively, which consists in bounding the objective function (4) and (2), respectively.
One can easily verify that, for any value of d, a solution to $(BLBlock)$ is a solution to $(BLTrans)$ and that, for $d = 1$, a solution to $(BLTrans)$ is a solution to $(BLBlock)$. That corresponds to Facts 2.1 and 2.2 of Section 2.

Bilevel programs are generally NP-hard, even if both levels are linear programs (see [9]). Now consider the cases where, for given values of x, the continuous relaxation of the lower program (program (2, 2.1, 2.2) or (4, 4.1, 4.2)) always admits an integral optimal solution, particularly when the associated constraint matrix is totally unimodular. This is the case, for instance, for stable set or matching in bipartite graphs. Then, the lower program can be replaced by the complementary slackness conditions of linear programming and the initial bilevel program becomes a $0-1$ mathematical program which can be polynomially solved in some special cases by using linear programming or with a specific algorithm as will be seen in Section 7. In other cases, the bilevel programming approach would be useful for blockers and transversals problems since many (approximation) methods have been proposed for solving such problems (see [10] or [22]).

4 Matroids

Let now V be the ground set of a matroid M and $\mathcal{C}_V$ the collection of independent sets. The maximum independent sets are the bases of M; their cardinality is $r(M)$ where r is the rank function of M (see [19]). In this context a d-transversal is a subset $T \subseteq V$ such that any basis C of M has at least d elements in T: $|T \cap C| \geq d$. A d-blocker is a subset B such that any basis $C \in V - B$ has $|C| \leq r(M) - d$.

Proposition 4.1 *For any $d \geq 1$, a subset $T \subseteq V$ in a matroid M constructed on V is a d-transversal if and only if it is a d-blocker.*

Proof: Let $T \subseteq V$ and assume that T is not a d-transversal. There exists a basis C of M such that $|T \cap C| < d$. Let $C' = C - T$; since $C' \subseteq C$ and M is a matroid then $C' \in \mathcal{C}_V$ and we have $|C'| \geq |C| - |T| > r(M) - d$. Thus T is not a d-blocker. Conversely, let $T \subseteq V$ such that T is not a d-blocker. There exists $C \in \mathcal{C}_V$, $C \subseteq V - T$, such that $|C| > r(M) - d$. If $|C| \geq r(M)$ then C is a basis of M with $|C \cap T| = 0$ and thus T is not a d-transversal. If $|C| < r(M)$, M being a matroid,

C may be completed by $r(M) - |C|$ elements in V to obtain a basis C' ($C' \in \mathcal{C}_V$ and $|C'| = r(M)$). We have used $r(M) - |C| < r(M) - (r(M) - d) = d$ elements of V (possibly in T). Thus $|C' \cap T| < d$ and T is not a d-transversal. $\square$

However, it is known that finding a minimum 1-transversal in a binary matroid is $\mathcal{NP}$-hard while it is polynomially solvable in regular matroids (see [19]). For illustration purposes, let M be a graphical matroid; then V is the set of edges of a graph G which we assume to be connected. There is a correspondence between bases and spanning trees and finding a minimum 1-transversal is equivalent to determining a minimum cut in G, which can be done in polynomial time (see for instance [12] or chapter 16 in [19]); for fixed $d > 1$, there is a polynomial algorithm for finding a minimum set T of edges which disconnect G into $d + 1$ connected components; if d is not fixed, the problem is $\mathcal{NP}$-hard (see [14]).

5 Shortest s-t paths

A situation of interest is the search of minimum d-blockers and minimum d-transversals of all shortest s-t paths in a directed graph G with a source s and a sink t. V is the set of arcs in G, and a subset V verifies $\mathcal{P}$ if it forms an s-t path; $\mathcal{C}_V$ is the collection of all s-t paths.
A d-transversal is a subset $T \subseteq V$ such that for every shortest s-t path C in G we have $|C \cap T| \geq d$.

Then, after removal of all arcs not contained in any shortest s-t path, T is a 1-transversal if and only if T is an s-t cut. It is known that a minimum cardinality s-t cut can be found in polynomial time (see [1]).
The case $d > 1$ has been studied in [15] where the following observation is stated:

Proposition 5.1 *[15] Let G be a graph such that evrey arc belongs to at least one shortest s-t path. T is an (inclusionwise) minimal d-transversal if and only if T is a union of d disjoint minimal s-t cuts.*

As a consequence, one has the following:

Proposition 5.2 *([15],[23])*
A minimum d-transversal of all s-t paths in a directed graph can be constructed in polynomial time.

For the construction, one may use for instance the algorithm in [23] to find d disjoint s-t cuts $C_1, \ldots, C_d$ such that $|C_1| + \cdots + |C_d|$ is minimum: it is based on a formulation in terms of transshipment problem and it can be solved in $O(mnlog(n^2/m)log(n))$ time where n (resp. m) is the number of vertices (resp. arcs).
In this context, a d-blocker is a subset B of arcs such that every s-t path in $G' = (V, E - B)$ has length at least $l^*(G) + d$ where $l^*(G)$ is the length of a shortest s-t path in G. Finding a minimum d-blocker in this case is known to be NP-hard (see [15]).

6 Minimum $s-t$ cuts

We are given a directed graph G with a source s and a sink t. An s-t cut is defined by a subset A of vertices of G such that $s \in A, t \notin A$; it consists of all arcs (u, v) with $u \in A$ and $v \notin A$. V will be the set of arcs of G and $\mathcal{C}_V$ the collection of all s-t cuts in G.
In this context a d-transversal is a subset T of arcs such that every minimum s-t cut C has $|C \cap T| \geq d$. For finding a minimum d-transversal as far as we know, no polynomial time algorithm is known but, provided the list of all minimum s-t cuts is given, a linear programming model (with totally unimodular matrix) can be used (see chapter 60 in [19]).
Let us now consider the problem of deteriorating the size of a minimum s-t cut. Notice that here we do not have a problem where removal of a subset $A \subset V$ will deteriorate the optimum size of elements in $\mathcal{C}_{V-A}$, i.e., it will not increase the minimum size of an s-t cut or, equivalently, of the maximum value of a flow. In fact, to increase the maximum flow value with the goal of improving the performance of a network, one can either introduce new arcs in the network or, as it occurs in many types of applications, expand the capacity of existing arcs. Here we consider the cardinality of an s-t cut so all our arcs have capacity equal to 1. This is why, mixing both approaches, we will consider an operation of *duplication* of existing arcs which corresponds to an unitary increasing of the capacity of these arcs.
Then, in order to increase by d the minimum value $\lambda(\mathcal{C}_V)$ of $|C|$, $C \in \mathcal{C}_V$, the problem consists in finding a minimum subset of arcs to be duplicated so that the value of a maximum s-t flow is increased by d: we call this set a *d-expander*. The problem can be formalized as will be described now: for any arc $e = (u, v) \in V$ we introduce a twin arc $e^* = (u, v)$; a subset $E = \{e_1, ..., e_p\} \subseteq V$ is duplicated if for every $e_i \in E$ we introduce its twin e_i^*: the set V is then replaced by $V_E^* = V \cup \left\{e_1^*, ..., e_p^*\right\}$.

A *d-expander* is a subset $E = \{e_1, ..., e_p\} \subseteq V$ such that if E is duplicated then for any $C^* \in \mathcal{C}_{V_E^*}$ we have $|C^*| \geq \lambda(\mathcal{C}_V) + d$.

We can now formulate the problem as the construction of a flow (of fixed value) with minimum cost in a capacited network (see [1]). We have to duplicate some arcs in such a way that every s-t cut has now a capacity of at least the minimum capacity of an s-t cut in the original network plus d. Such a set E of arcs to be duplicated is obtained by constructing a minimum cost flow f of value $\nu(\mathcal{C}_V) + d$ in a network obtained as follows: all arcs are duplicated (all have capacity 1); for each arc $e = (u, v)$ the cost is 0 and for each arc $e^* = (u, v)$ the cost is 1. The set E will be the set of arcs e with flow $f(e^*) = 1$. Hence we have:

Proposition 6.1 *A minimum d-expander E of the s-t cuts in a directed graph can be constructed in polynomial time (with the complexity of a min cost flow procedure)*

Note that, if we are allowed to introduce also new arcs in the network in order to increase $\lambda(\mathcal{C}_V)$, then in the above flow model we would just introduce these arcs with unit capacity and unit cost.

7 Stable sets

Here the ground set V is the set of vertices of a graph $G = (V, E)$ and we intend to study d-transversals and d-blockers of maximum stable sets in G. We denote by $N(v)$ the set of neighbors of a vertex $v \in V$. A subset $S \subseteq V$ is *stable* (or *independent*) if no two vertices in S are linked by an edge. The stability number $\alpha(G)$ (corresponding to $\nu(G)$ in the general formulation of Section 2) is the cardinality of a maximum stable set in G. $\mathcal{C}_V$ will be the collection of all stable sets. A d-transversal is then a subset T of vertices such that every maximum stable set S verifies $|S \cap T| \geq d$. A d-blocker is a subset B of vertices such that in the (induced) subgraph $\widehat{G}$ spanned by $V - B$, the stability number has decreased by at least d: $\alpha(\widehat{G}) \leq \alpha(G) - d$. In addition $\mu(G)$ will be the maximum cardinality of a matching in G.
A vertex v is *forced* if every maximum stable set contains v. Following [5] we denote the number of forced vertices in G by $\xi(G)$. A vertex v is *excluded* if no maximum stable set of G contains v. A vertex which is neither forced nor excluded is *free*. For characterizing such vertices in an arbitrary graph, Boros et al. [5] give the following:

Fact 7.1 *[5] A vertex v is free in G if and only if $\alpha(G - N(v)) \leq \alpha(G) - 1$. A vertex v is forced in G if and only if $\alpha(G - v) \leq \alpha(G) - 1$.*

Concerning the complexity of finding a minimum d-transversal of all maximum stable sets the following is known: given any fixed $k \geq 1$, determining if there are more than k forced vertices is $\mathcal{NP}$-complete (see[5]). So determining whether for a given $d \geq 1$ there exists a d-transversal T with $|T| \leq d$ is difficult (such a T exists if and only if there are at least d vertices which are forced).
Notice that from [26], for any fixed $d \geq 1$, finding minimum d-transversals and minimum d-blockers of all maximum stable sets is NP-hard even if G is the line graph of a bipartite graph. It follows from results in [16] that finding minimum d-transversals is polynomial for split graphs, complements of balanced, of strongly chordal, of triangle free graphs (for fixed $d \geq 2$); however finding $\tau_d(G)$ is NP-hard for complements of doubly chordal, of planar graphs, of triangle free graphs (for $d = 1$).

7.1 Split graphs

We shall examine here the case of split graphs $G = (V, E)$; they are defined as graphs in which the vertex set V can be partitionned into a clique K and a stable set S (see [7]). Since the problem of d-transversals has been dealt with in [16], we shall concentrate here on the problem of d-blockers.
We recall that for a split graph $G = (V, E)$ where $n = |V|$ and $\omega(G)$ is the maximum cardinality of a clique we have $n \leq \alpha(G) + \omega(G) \leq n + 1$. We can assume w.l.o.g. that the partition (K, S) is chosen such that $|S| = \alpha(G)$.

Lemma 7.1 *For a split graph $G = (V, E)$ we have $d \leq \beta_d(G) \leq d + 1$, $(1 \leq d \leq \alpha(G) - 1)$.*

Proof: Clearly $\beta_d(G) \geq d$; if we take for B a set of $d + 1$ vertices in S, then $V - B$ can be covered by $\alpha(G) - d$ cliques (one of them being K and the other ones of the form $v \cup N(v)$ with $v \in S - B$). So $\alpha(G - B) \leq \alpha(G) - d$ and B is a blocker. $\square$

Let us formalize now the associated decision problem **SPLITBLOCK**:
INPUT: A split graph $G = (V, E)$ and a positive integer $d \leq |V|$.
QUESTION: Does G have a d-blocker B with $|B| = d$?

What is the complexity of **SPLITBLOCK**? In G there exists a set $B \subset S$ of d vertices which forms a d-blocker if and only if the vertices of $V - B$ can be covered by $\alpha(G) - d$ cliques. It is also equivalent to saying that the union of the $\alpha(G) - d$ sets $N(v), v \in S - B$ is K. If we consider the family $\mathcal{N} = \{N(v)|v \in S\}$ where $N(v) \subseteq K$ for each v, we have to solve the following set cover problem: Given a ground set K, a positive integer $\alpha(G) - d$ and a collection $\mathcal{N}$ of subsets $N(v) \subseteq K$ does there exists a cover $C' = \{N(v)|v \in S - B\}$ with $|C'| = |S - B| = \alpha(G) - d$, i.e., such that $\bigcup\{N(v)|v \in S - B\} = K$.
We can now state:

Proposition 7.2 **SPLITBLOCK** *is NP-complete even if* $|N(v) \cap S| = 2$ *for each vertex* $v \in K$. *Moreover for any fixed* $d \geq 1$, **SPLITBLOCK** *can be polynomially solved in* $O(n^{d+1})$.

Proof: **SPLITBLOCK** is in NP since in a split graph one can find a maximum stable set in polynomial time.
Consider the Vertex Cover problem in a graph $H = (X, A)$, which is NP-complete (see [13]). We get a split graph G by setting $S = X$ and $K = A$; (x, a) is an edge of G if and only if x is an endpoint of a in H. Then $|N(a) \cap S| = 2$, $\forall a \in K$. Finding a subset $S - B$ in G with $|S - B| = \alpha(G) - d$ such that $\bigcup\{N(v)|v \in S - B\} = K$ is equivalent to finding in H a subset of vertices of cardinality $\alpha(G) - d$ which covers the edges of A. This equivalence implies NP-completeness of **SPLITBLOCK**.
Finally when d is fixed, it suffices to enumerate all subsets $B \subset V$ with $d \leq |B| \leq d + 1$ according to Lemma 7.1. $\square$

It is interesting to observe that the problem of finding a minimum d-blocker of maximum stable sets is NP-hard in the class of split graphs, while the problem of minimum $d-$transversals is polynomially solvable (see[16]).
Since a split graph is perfect, we have:

Corollary 7.3 *In perfect graphs the problem of finding a minimum d-blocker of maximum stable sets is NP-hard.*

Remark 7.1 *If in a split graph* $G = (V, E)$ *for every vertex* $v \in S$ *we have* $|N(v) \cap K| \leq 2$, **SPLITBLOCK** *is polynomially solvable since the corresponding set cover problem is polynomial (see [13]).*

Proposition 7.4 *If* $G = (V, E)$ *is a split graph with* $\alpha(G) + \omega(G) = n + 1$, *then a minimum d-blocker B has cardinality d.*

Proof: In such graphs there is a vertex v_1 in S with $N(v_1) = K$ where (K, S) is the partition of vertices chosen with $|S| = \alpha(G)$. From the above discussion we construct B by chosing d vertices in $S - \{v_1\}$. $\square$

It follows that for threshold graphs (split graphs in which there is an order $v_1, \ldots, v_{\alpha(G)}$ of the vertices in S such that $i < j$ implies $N(i) \supseteq N(j)$ the same result holds since v_1 satisfies the condition of Proposition 7.4.

7.2 Bipartite graphs

Here we shall consider the class of bipartite graphs. Observe that matchings and stable sets are closely related since the König's theorem states that $\alpha(G) + \mu(G) = |V|$ (see [2]). Then we have the two following facts:

Fact 7.2 *Let M be a maximum matching in a bipartite graph G. Any vertex unsaturated by M is forced and there is exactly one endpoint of each edge of M in any maximum stable set of G.*

Proof: If M is a maximum matching, there are $|V| - 2\mu(G) = \alpha(G) - \mu(G)$ vertices which are unsaturated by M. Given any maximum stable set S, it contains at most $\mu(G)$ vertices saturated by M and hence at least $\alpha(G) - \mu(G)$ unsaturated vertices. So all the $\alpha(G) - \mu(G)$ unsaturated vertices are contained in any maximum S and they are forced. Hence S must also contain one endvertex of each one of the $\mu(G)$ edges of M. $\square$

Fact 7.3 *(see [2]) If $G = (V, E)$ is a bipartite graph then the following are equivalent:*

- *G has a perfect matching*
- $\alpha(G) = \mu(G) = |V|/2$
- *denoting by X and Y the partition of V, X and Y are two disjoint minimum vertex covers which are also two disjoint maximum stable sets of size $|V|/2$.*

In addition, we notice that the following proposition stated for trees in [21] can be extended to bipartite graphs :

Proposition 7.5 *Let $G = (V, E)$ be a bipartite graph. G has a perfect matching if and only if G has only free vertices.*

Proof: If G has a perfect matching, from Fact 7.3 there exist two disjoint maximum stable sets and then all vertices are free.
From the proof of Fact 7.2 there are $\xi(G) \geq |V| - 2\mu(G)$ forced vertices in any maximum stable set. If all the vertices are free then $\xi(G) = 0$, i.e. $\mu(G) = |V|/2$ and so G has a perfect matching. $\square$

Let us now state the following:

Proposition 7.6 *If $G = (V, E)$ is a bipartite graph the vertex set can be partitioned into three subsets F, H and $V - F - H$ such that:*

- *F is the set of all forced vertices (belonging to any maximum stable set)*
- *H is the set of all excluded vertices (contained in no maximum stable set)*
- *The subgraph induced by $V - F - H$ has a perfect matching*

Proof: If G has a perfect matching, the proposition is obvious ($F = H = \emptyset$). Assume now that G has no perfect matching. Let F be the set of all forced vertices of G ($F \neq \emptyset$ from Fact 7.2) and let H be the set of all excluded vertices. Consider now the subgraph $\widehat{G}$ induced by $V - F - H$. A vertex adjacent to a forced vertex is obviously excluded; so $\widehat{G}$ has no vertex adjacent to a vertex in F. A maximum stable set S^* in G is the union of F with a stable set $\widehat{S}$ in $\widehat{G}$: since $\widehat{G}$ has no vertex adjacent to a vertex in F, $\widehat{S}$ could be any maximum stable set in $\widehat{G}$. Then, a forced (resp. excluded) vertex in $\widehat{G}$ would be a forced (resp. excluded) vertex in G: so such a vertex cannot belong to $\widehat{G}$. Thus $\widehat{G}$ has only free vertices, so from Proposition 7.5, $\widehat{G}$ has a perfect matching. $\square$

We now explain how to determine the excluded and the forced vertices in a bipartite graph.

Proposition 7.7 *The sets of excluded and of forced vertices in a bipartite graph can be determined in* $O(|V|^{\frac{5}{2}})$ *time.*

Proof:
We use the following procedure:

Input: $G = (V, E)$: a bipartite graph;
Output: F and H: the sets of forced and of excluded vertices;
1. Construct a maximum matching M in G;
2. Label with f all the unsaturated vertices;
3. *while* it_is_possible *do*
    3.1 label with h any vertex y such that $\{x, y\} \in E$ and x is labelled with f;
    3.2 label with f any vertex x such that $\{x, y\} \in M$ and y is labelled with h;
   end do
4. $F=$ {vertices of G with label f};
   $H=$ {vertices of G with label h}.

Justification of the procedure:
If M is perfect, then $F = H = \emptyset$. So assume now that there is at least one unsaturated vertex. From Fact 7.2, any unsaturated vertex is forced (step 2). Any vertex adjacent to a forced vertex is excluded (step 3.1). In a bipartite graph, there is exactly one endpoint of each edge of a maximum matching M in any maximum stable set (Fact 7.2): if an endpoint of an edge in M is excluded then the other is forced (step 3.2).
The vertices which are unlabelled at the end of loop 3 are all saturated (otherwise they would be labelled with f), they are all matched (in M) to an unlabelled vertex (otherwise they would be labelled at step 3.1 or 3.2) and they are not adjacent to a vertex in F (otherwise they would be labelled with h at step 3.1). Thus, the subgraph $\widehat{G}$ induced by the unlabelled vertices has a perfect matching. Thus the procedure produces a partition of V into sets F, H and $V - F - H$ as required by Proposition 7.6.

Complexity of the procedure:

We need to compute a maximum matching: in a bipartite graph, it can be done in time $O(|V|^{\frac{5}{2}})$ (see [2]). Then at step 3, each edge is considered at most once. So the whole procedure runs in time $O(|V|^{\frac{5}{2}})$. □

The previous proof shows that in bipartite graphs a vertex which is neither forced nor adjacent to a forced vertex is free; thus any excluded vertex is adjacent to a forced vertex. On the contrary, in an arbitrary graph, we may have excluded vertices which are not adjacent to any forced vertex: take two triangles having a common vertex v; v is excluded and all remaining vertices are free.
We can now state:

Proposition 7.8 *In a bipartite graph G with $\xi(G)$ forced vertices, the minimum cardinality of a d-transversal of all maximum stable sets is*

$$\tau_d(G) = \begin{cases} d & \text{if } d \le \xi(G) \\ 2d - \xi(G) & \text{if } \xi(G) < d \le \alpha(G) \end{cases}$$

Moreover a minimum d-transversal can be constructed in time $O(|V|^{\frac{5}{2}})$.

Proof: We use the following procedure.

Input: $G = (V, E)$: a bipartite graph; d: $1 \le d \le \alpha(G)$
Output: $T \subseteq V$: a minimum d-transversal

1. Determine the sets F of forced vertices and H of excluded vertices
2. If $1 \le d \le \xi(G)$: choose d vertices in F. This gives the required T
3. If $d > \xi(G)$, let $\widehat{G}$ be the subgraph of G induced by $V - F - H$.
4. Construct a perfect matching $\widehat{M}$ in $\widehat{G}$. Then set $T = F \cup \widehat{T}$ where $\widehat{T}$ is the set of endpoints of $d - \xi(G)$ edges of $\widehat{M}$.

Justification of the procedure:
Clearly, for $d \le \xi(G)$, the set formed by d (forced) vertices in F is a minimum d-transversal. Assume now that $d > \xi(G)$. From Proposition 7.6, $\widehat{G}$ admits a perfect matching $\widehat{M}$. Any maximum stable set S in G contains $\xi(G)$ forced vertices and exactly one endpoint of each edge in $\widehat{M}$, i.e. $\mu(\widehat{G}) = \alpha(G) - \xi(G)$ vertices in $\widehat{G}$. So, the set T constructed by the procedure is a d-transversal. Furthermore, we have two maximum stable sets in G: $S_1 = F \cup \widehat{X}$ and $S_2 = F \cup \widehat{Y}$ where $\widehat{X}$ and $\widehat{Y}$ are the right and left sets of vertices of $\widehat{G}$. So a d-transversal must contain at least $\xi(G) + 2(d - \xi(G)) = 2d - \xi(G)$ vertices. Since $|T| = 2d - \xi(G)$ then T is minimum.

Complexity of the procedure:
Step 1 requires time $(O(|V|^{\frac{5}{2}})$ (see Proposition 7.7) and it produces the graph $\widehat{G}$ and the matching $\widehat{M}$ used at step 4. Thus, the whole procedure runs in time $(O(|V|^{\frac{5}{2}})$.
□

Remark 7.2 *In a bipartite graph with a perfect matching, d-transverals may not be d-blockers : for the graph in Figure 1 the black vertices form a 2-transversal but it is not a 2-blocker since the white vertices form a stable set S with $|S| = 2 > \alpha(G) - 2 = 1$.*

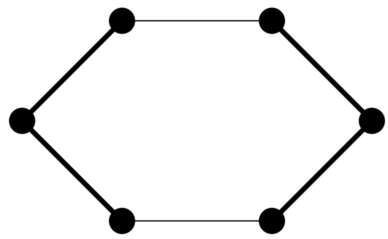

Figure 1: A 2-transversal which is not a 2-blocker

Proposition 7.9 *In a bipartite graph G, the minimum cardinality of a d-blocker is* $\beta_d(G) = \begin{cases} d & \text{if } d \le |V| - 2\mu(G) \\ 2d - |V| + 2\mu(G) & \text{if } |V| - 2\mu(G) < d \le \alpha(G) \end{cases}$.
Moreover a minimum d-blocker for the maximum stable set problem can be constructed in time $O(|V|^{\frac{5}{2}})$.

Proof: We use the following procedure.

Input: $G = (V, E)$ bipartite; d: $1 \le d \le \alpha(G)$
Output: $B \subseteq V$ a minimum d-blocker

1. Construct a maximum matching M; $|M| = \mu(G)$
2. Let F^* be the set of vertices not saturated by M (they are forced vertices); $|F^*| = |V| - 2\mu(G)$
3. If $1 \le d \le |F^*|$: choose d vertices in F^*. This gives the required B
4. If $d > |F^*|$, set $B = F^* \cup \widehat{B}$ where $\widehat{B}$ is the set of endpoints of $d - |V| + 2\mu(G)$ edges of M

Justification of the procedure:
If $d \le |V| - 2\mu(G)$, the result is obvious: we have $|V| - 2\mu(G)$ vertices not saturated by a maximum matching which are forced vertices (see 7.2 and its proof): removing d of them will reduce the size of a maximum stable set by d.
Assume now that $|V| - 2\mu(G) < d \le \alpha(G)$, i.e. $d = |V| - 2\mu(G) + p$ (with $0 < p \le \mu(G)$). Let $\overline{G}$ be the subgraph induced by $V - B$. $\overline{G}$ admits a perfect matching with $\mu(\overline{G}) = \mu(G) - p$. Thus $\alpha(\overline{G}) = \mu(\overline{G}) = \mu(G) - p = \mu(G) - (d - |V| + 2\mu(G)) = |V| - \mu(G) - d = \alpha(G) - d$; so B is a d-blocker. Now, let us show that B is minimum. We have $|B| = 2d - |V| + 2\mu(G)$; if we removed a set B' with $|B'| < |B|$, we would have more than $2(|V| - \mu(G) - d) = 2(\alpha(G) - d)$ vertices in the remaining graph G' (which is bipartite) and then $\alpha(G') > \alpha(G) - d$, so B' would not be a d-blocker.
The complexity of the procedure is the complexity of finding a maximum matching in a bipartite graph (at step 1), i.e. $O(|V|^{\frac{5}{2}})$ (see [2]). □

7.3 Vertex covers in bipartite graphs

Here we intend to study d-transversals of minimum vertex covers in a bipartite graph $G = (V, E)$. A subset $C \subseteq V$ is a *covering set* or a *cover* if every edge $\{x, y\} \in E$

has at least one endpoint in C, i. e. $|\{x,y\} \cap C| \geq 1, \forall \{x,y\} \in E$. The covering number $\beta(G)$ (corresponding to $\lambda(G)$ in the general formulation of Section 2) is the cardinality of a minimum covering set in G. A d-transversal is then a subset T of vertices such that every minimum covering set C verifies $|C \cap T| \geq d$.
A vertex v is *forced* if every minimum cover contains v. We denote the number of forced vertices in G by $\zeta(G)$. A vertex v is *excluded* if no minimum cover of G contains v. A vertex which is neither forced nor excluded is *free*.
We easily derive the following statement from Proposition 7.8:

Corollary 7.10 *In a bipartite graph G with $\zeta(G)$ forced vertices, the minimum cardinality of a d-transversal of all minimum covers is*
$$\tau_d(G) = \begin{cases} d & \text{if } d \leq \zeta(G) \\ 2d - \zeta(G) & \text{if } \zeta(G) < d \leq \beta(G) \end{cases}$$
Moreover a minimum d-transversal can be constructed in time $O(|V|^{\frac{5}{2}})$.

Proof: If $S \subset V$ is a stable set then its complement $C = V - S$ is a covering set. So the complement of a maximum stable set is a minimum cover. From Proposition 7.6, F and H are the sets of excluded vertices and forced vertices, respectively, for all covers. The sequel of the proof is an immediate adaptation of the proof of Proposition 7.8 in which $\xi(G)$ is replaced by $\zeta(G)$. $\square$

8 Concluding remarks

For all problems discussed here the determination of the optimal elements of $\mathcal{C}_V$ was polynomial. It appears that the complexity of finding minimum d-transversals and d-blockers and the complexity of finding an optimal element of $\mathcal{C}_V$ are not directly related. We have seen that in the case of line graphs of bipartite graphs the first problem is NP-hard while the second one is polynomial. Conversely, there are problems for which finding minimum d-transversals or d-blockers is polynomial while finding an optimal element of $\mathcal{C}_V$ is NP-hard; such a situation occurs for instance when we want to find a d-transversal, $d \geq 1$, in case where $\mathcal{C}_V$ is the family of all longest elementary s-t paths, s has degree one and s is an endpoint of an induced path of length at least d. It could be interesting to examine other similar situations, i.e., where finding an optimal element of $\mathcal{C}_V$ is NP-hard.
In this paper, after having formalized the concepts of d-transversals and d-blockers we have solved the case where $\mathcal{C}_V$ is the collection of all maximum stable sets in bipartite graphs. Notice that we have simultaneously solved the d-packer problem where one searches a maximum set of vertices which has no more than d vertices in any maximum stable set (recall that it corresponds to an $(\alpha(G) - d)$-transversal ($d = 1, \ldots, \alpha(G)$)). We have also solved the case of d-blockers in split graphs. It would be interesting to examine the case of other classes of graphs for which the complexity status is still open. One could consider other families $\mathcal{C}_V$ which may lead to interesting results and to further research directions, like the weighted cases of the above problems.

Acknowledgements: This research was carried out when M.-C. Costa and Ch. Picouleau were visiting EPFL and when D. de Werra was visiting CNAM in 2009 and 2010. The support of both institutions is gratefully acknowledged. We also would like to thank A. Sebö for fruitful discussions during the elaboration of the paper. Furthermore, D. de Werra expresses his gratitude to Dr B. Pelet who provided the necessary logistics.

References

[1] R.K. Ahuja, T. L. Magnanti, J.B. Orlin, *Networks flows: Theory, Algorithm, and Applications*, Prentice Hall (1993).

[2] A.J. Asratian, T.M.J. Denley , R. Häggkvist *Bipartite graphs and their applications*, Cambridge University Press, (Cambridge, 1998).

[3] C. Berge, *Graphes*, Gauthier-Villars, (Paris, 1983).

[4] F. Bonomo, G. Durán (2004), *Computational complexity of classical problems for hereditary classes of graphs*, Pesquisa Operacional 24, 413-434 or Electronic Notes in Discrete Mathematics 18, 41-46.

[5] E. Boros, M.C. Golumbic, V.E. Levit (2002), *On the number of vertices belonging to all maximum stable sets of a graph*, Discrete Applied Mathematics 124 (1), 17-25.

[6] E. Boros, K. Elbassioni, V. Gurvich (2006), *Transversal Hypergraphs to Perfect Matchings in Bipartite Graphs: Characterization and Generation Algorithms*, Journal of Graph Theory 53 (3), 209-232.

[7] A. Branstädt, J. P. Spinrad, V. B. Le, *Graph classes : A survey*, SIAM Monographs on Discrete Math. and Appl. (1999).

[8] P. Burzyn, F. Bonomo, G. Durán (2006), *$\mathcal{NP}$-completeness results for edge modification problems*, Discrete Applied Mathematics 154(13), 1824-1844.

[9] B. Colson, P. Marcotte, G. Savard (2007), *An overview of bilevel optimization*, Annals of Oper. Research, 153, pp. 235-256.

[10] S. Dempe, *Foundations of bilevel programming*, Kluwer Academic Publishers (2002).

[11] P. Erdos, T. Gallai, Z. Tuzc (1992), *Covering the cliques of a graph with vertices*, Discrete Mathematics 108, 279-289.

[12] A. Frank (1994), *On the edge connectivity algorithm of Nagamochi and Ibaraki*, Unpublished report.

[13] M.R. Garey and D.S. Johnson, *Computers and intractability, a guide to the theory of NP-completeness*, Freeman (New York, 1979).

[14] O. Goldschmidt and D.S. Hochbaum (1994), *A polynomial algorithm for the k-cut problem for fixed k*, Mathematics of Operations Research 19, 24-37.

[15] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf, J. Zhav (2008), *On short paths interdiction problems: total and node-wise limited interdiction*, Theory of Computing Systems 43, 204-233.

[16] C.-M. Lee (2009), *Variations of maximum-clique transversal sets on graphs*, Annals of Oper. Research, to appear.

[17] J.M. Lewis, M. Yannakakis (1980), *The node-deletion problem for hereditary properties is $\mathcal{NP}$-complete*, Journal of Computer and System Sciences 20, 219-230.

[18] B. Ries, C. Bentz, C. Picouleau, D. de Werra, M.-C. Costa, R. Zenklusen (2010), *Blockers and Transversals in some subclasses of bipartite graphs: when caterpillars are dancing on a grid*, Discrete Mathematics, 310(1), 132-146.

[19] A. Schrijver, *Combinatorial Optimization: Polyhedra and Efficiency*, Springer, (New York, 2003).

[20] W. Spurr, K. Spurr, *Aferd Packer's High Protein Gourmet Cookbook*, Centennial Publications, (Grand Junction, 1995).

[21] P.J. Slater (1978), *A constructive characterization of trees with at least k disjoint maximum matchings*, J. Combinatorial Theory Ser. B 25, 326-338.

[22] L.N. Vicente, G. Savard, J.J. Judice (1996), *The discrete linear bilevel programming problem*, J. of Optimization Theory and Applications, 89, 597-614.

[23] D. Wagner (1990), *Disjoint st-cuts in a network*, Networks 20, 361-371.

[24] M. Yannakakis (1981), *Edge-deletion problems*, SIAM Journal of Computing 10 (2), 297-309.

[25] M. Yannakakis (1981), *Node-deletion problems on bipartite graphs*, SIAM Journal of Computing 10 (2), 310-327.

[26] R. Zenklusen, B. Ries, C. Picouleau, D. de Werra, M.-C. Costa, C. Bentz (2009), *Blockers and Transversals*, Discrete Mathematics, 309(13), 4306-4314.

[27] J. Zito (1991), *The structure and maximum number of maximum independent sets in trees*, J. Graph Theory, 15, 207-221.