



HAL
open science

Multiple Bipartite Complete Matching Vertex Blocker Problem: Complexity, polyhedral analysis and Branch-and-Cut

Pierre Laroche, Franc Marchetti, Sébastien Martin, Anass Nagih, Zsuzsanna Roka

► **To cite this version:**

Pierre Laroche, Franc Marchetti, Sébastien Martin, Anass Nagih, Zsuzsanna Roka. Multiple Bipartite Complete Matching Vertex Blocker Problem: Complexity, polyhedral analysis and Branch-and-Cut. Discrete Optimization, 2020, 35, pp.100551. 10.1016/j.disopt.2019.100551 . hal-03489996

HAL Id: hal-03489996

<https://hal.science/hal-03489996>

Submitted on 22 Aug 2022

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

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



Distributed under a Creative Commons Attribution - NonCommercial 4.0 International License



Multiple Bipartite Complete Matching Vertex Blocker Problem: Complexity, Polyhedral Analysis and Branch-and-Cut

Pierre Laroche^a, Franc Marchetti^a, Sébastien Martin^a, Anass Nagih^a, Zsuzsanna Róka^a

^aLaboratoire de Conception, Optimisation et Modélisation des Systèmes, LCOMS EA 7306 Université de Lorraine, Metz 57000, FRANCE

Abstract

Given a bipartite graph $G = (U \cup V, E)$, $|U| \leq |V|$, the surplus of G is defined by the maximum number k such that a matching covering all vertices of U still exists upon removal of any k vertices from V . Given a partition $\mathcal{U} = \{U_1, \dots, U_m\}$ of U , the Multiple Bipartite Complete Matching Vertex Blocker Problem (MBCMVBVP) consists in finding a partition $\mathcal{V} = \{V_1, \dots, V_m\}$ of V such that the smallest surplus among those of the induced subgraphs $G[U_i \cup V_i]$ is maximized. The removed vertices are related to the blocker notion. We show the strong NP-hardness of the MBCMVBVP by using a reduction from the stable set problem. We also propose two integer linear programs for solving this problem. After comparing these two models, we introduce some valid inequalities for the model that outperforms the other one, and we analyze its facial structure. We then derive a Branch-and-Cut algorithm based on these results and conclude by an analysis of the experimental results.

Keywords: matching interdiction/blocker problem, bipartite graph, polytope, integer linear programming, Branch-and-Cut

1. Introduction

Solving interdiction/blocker problems is a mean to determine the maximum possible perturbation such that, beyond that point, no valid solution exists for the studied problem. Practically, these perturbations can be triggered by failures, attacks or absenteeism. Furthermore, such weakness analysis can be used to define tactical and strategic investments to make the model robust.

The *matching interdiction/blocker problem* ([13], [14]) consists in finding the maximum number of edges or nodes that can be removed such that a matching with a given cardinality still exists.

Given a bipartite graph $G = (U \cup V, E)$, $|U| \leq |V|$, the *complete matching problem* on U consists in finding a matching covering all vertices of U ([3], [10]). The *bipartite complete matching vertex blocker problem* (BCMVBVP) ([8]) consists in finding the maximum number k such that a complete matching on U still exists after removal of any k vertices in V . Let $\mathcal{U} = \{U_1, \dots, U_m\}$ be a partition of U . Given a partition $\mathcal{V} = \{V_1, \dots, V_m\}$ of V , we denote by k_i the solution of the BCMVBVP associated with the induced subgraph $G[U_i \cup V_i]$, $i \in \{1, \dots, m\}$. The *Multiple Bipartite Complete Matching Vertex Blocker Problem* (MBCMVBVP) consists in finding a partition $\mathcal{V} = \{V_1, \dots, V_m\}$ of V such that the minimum of $\{k_1, \dots, k_m\}$ is maximized.

To the best of our knowledge, this problem has never been studied before. It can be applied in different fields. For instance, in nurse rostering and staff management, BCMVBVP can be employed to identify the critical number of staff with *specific* skills below which the system remains under-staff. When considering several time shifts, the idea is to have the biggest common flexibility on each shift.

Email addresses: pierre.laroche@univ-lorraine.fr (Pierre Laroche), franc.marchetti@univ-lorraine.fr (Franc Marchetti), sebastien.martin@univ-lorraine.fr (Sébastien Martin), anass.nagih@univ-lorraine.fr (Anass Nagih), zsuzsanna.roka@univ-lorraine.fr (Zsuzsanna Róka)

Our contributions. The complexity of the problem is analyzed, and two Integer Linear Programs (ILP) are proposed. Some polyhedral properties of the problem are identified including facets and valid inequalities to strengthen their efficiency in a Branch-and-Cut algorithm. All the proposed algorithms have been tested to evaluate the practical effectiveness of the algorithmic ingredients developed. Note that the instances are generated to represent real-life assignment problems such as nurse rostering.

Related work. The notion of interdiction/blocker is relevant in several fields. For example, considering computer networks, it is primordial to guarantee the continuity of services in case of dysfunction of some routers or switches. These problems are usually modeled by shortest paths ([1]), spanning trees ([2]) or network flows ([7]) in the underlying graph.

In the field of human resources, being able to anticipate the potential absenteeism is essential in order to ensure the proper running of a company. This class of problems can be formulated by matching or scheduling models.

The blocker problem allows the understanding of the strength of the graph according to a given property. It has been studied on several well-known graph properties such as the independent set ([18], [19]) and related structures, *i.e.* clique and chromatic number ([15], [17]) or path problem ([16]).

One of the problems related to the matching interdiction/blocker problem has been studied in [12], where the authors consider the minimum d -blocker problem. The goal is to determine a minimum cardinality subset of edges such that their deletion from the graph decreases the matching number by at least d units. This problem corresponds to a particular case of the edge interdiction problem.

In [13] and [15], the authors studied two particular versions of the matching interdiction problem. The first one, called edge interdiction problem, considers a graph in which each edge has a weight and an interdiction cost. The goal is to find a subset of edges such that the value of a maximum-weight matching without these edges is minimized, with respect to a cost constraint. The second problem is defined in a similar way, where costs are applied to vertices instead of edges. These problems are shown as being NP-complete even in bipartite graphs, and some approximation algorithms are proposed to solve them. The same author, in [14], extended his proof to show the NP-hardness of these problems in the particular scope of bipartite graphs.

In [8], the authors showed the polynomiality of this problem when a complete matching is considered. This problem is a particular case of the MBCMBVP when $|\mathcal{U}| = 1$. It has been applied to a robust nurse assignment problem which consists in affecting nurses to jobs according to their skills in such a way that possible absences may not perturb the schedule.

Note that these models can be applied to study problems of assignment in any domain where the management of staff or production with limited resources is needed (air or rail transports, timetabling, ...).

Organization of the paper. In the next section, we give some basic definitions. In Section 3, the MBCMBVP is defined, and its NP-completeness is proved. In Section 4, we propose and study two Integer Linear Programs. The first one is based on the dual of the model used for the BCMVBP. The second one is an alternative with a reduced number of variables but an exponential number of inequalities. We also analyze the associated polytope and present a new family of inequalities. Then, some solving strategies and experimental results are presented in Section 5. In the last section, we propose some directions for further research.

2. Basic notions and definitions

Let us first introduce some notations and recall some basic notions concerning graphs, stable sets, and matchings.

Graphs. Let $G = (W, E)$ be a graph, where W is the set of vertices and E is a set of edges which are unordered pairs of vertices of W . An edge $\{u, v\}$, where u and v are two vertices of W , is denoted uv (or vu). For each $uv \in E$, the vertices u and v are said to be neighbors. In the following, the neighborhood of a vertex $v \in W$ is denoted by $N_G(v) \subseteq W$. We extend this notation to a subset $W' \subseteq W$ by $N_G(W') = \bigcup_{v \in W'} N_G(v)$.

Stable set. A subset S of vertices is a stable set if $N_G(S) \cap S = \emptyset$. Equivalently, each edge in the graph has at most one endpoint in S . The size of a stable set is its cardinality.

Maximal stable set. A stable set S is said to be maximal if it cannot be strictly included in another stable set. The problem of finding a maximal stable set of maximum size is called the maximum stable set problem and is an NP-hard optimization problem. The stable set decision problem concerning the existence of a stable set of a given size K can be solved in polynomial time if K is a fixed integer. However, if K is a part of the instance, the problem becomes NP-complete in the strong sense.

Bipartite graph. Let $\{U, V\}$ be a partition of the set of vertices W (i.e. $U \cup V = W$ and $U \cap V = \emptyset$) such that $N_G(U) \subseteq V$ and $N_G(V) \subseteq U$. $G = (W = U \cup V, E)$ is called a bipartite graph. Let $V' \subseteq V, U' \subseteq U$ and $E[U' \cup V'] = \{uv \mid u \in U', v \in V', uv \in E\}$. We denote by $G[U' \cup V'] = (U' \cup V', E[U' \cup V'])$ the subgraph of G induced by $U' \cup V'$.

Matchings. Given a graph, a matching is a subset of edges such that no two edges share a common vertex. The matching problem consists in finding a matching with a maximal cardinality ([3], [9]).

Complete Matchings. Let $G = (U \cup V, E), |U| \leq |V|$, be a bipartite graph. A complete matching on U is a matching covering all vertices of U , i.e. each vertex of U is incident with an edge of the matching. The associated problem consists in finding a complete matching on U . If $|U| = |V|$, then a complete matching on U is called a perfect matching.

The following theorem is a fundamental result about the existence of a complete matching in bipartite graphs ([6]).

Theorem 1 (Hall’s theorem). Let $G = (U \cup V, E)$ be a bipartite graph. There exists a complete matching on U if and only if $|N_G(U')| \geq |U'|$ for all $U' \subseteq U$.

Surplus. The surplus of a subset $U' \subseteq U$ is defined as $\sigma_G(U') = |N_G(U')| - |U'|$. When $U' = \{u\}$, the surplus of U' is denoted by $\sigma_G(u)$. The surplus of G is defined as $\sigma(G) = \min_{\emptyset \neq U' \subseteq U} \{\sigma_G(U')\}$.

k-Complete Matchings ([8]). Let $G = (U \cup V, E)$ be a bipartite graph. We say that G is *k-Complete Matchable (k-CM) on U* if and only if, for all $V' \subseteq V$ with $|V'| = |V| - k$, there exists a complete matching on U in the induced subgraph $G[U \cup V']$. Note that if a graph is k -CM on U , then it is also $(k - 1)$ -CM on U , for $k > 0$.

Note that the particular case of $k = |V| - |U|$ has been studied in [9], and such a bipartite graph is called *k-critical*.

In [8], the following result has been presented for the case when $0 \leq k \leq \sigma(G)$.

Theorem 2 (Surplus form of Hall’s theorem). Let $G = (U \cup V, E)$ be a bipartite graph. G is *k-CM on U* if and only if, for all non-empty subset $U' \subseteq U, |N_G(U')| \geq |U'| + k$. The maximum value of k such that G is *k-CM on U* is equal to $\sigma(G)$.

Theorem 3. $\sigma(G)$ can be computed in polynomial time.

Proof. As σ is a submodular set function on U ([10]), its minimum value can be computed by a strongly polynomial-time algorithm ([11]). □

The graph presented in Figure 1 is 2-CM (and so 1-CM) on U : after deleting any two vertices (in grey color) from V , there still exists a complete matching on U (bold edges). It is not the case for more than two vertices, hence $\sigma(G) = 2$.

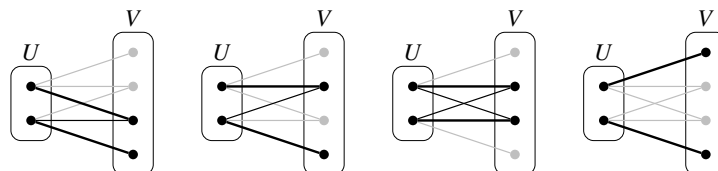


Figure 1: 2-CM on U bipartite graph.

3. Multiple Bipartite Complete Matching Vertex Blocker Problem (MBCMVBVP)

First, we give some definitions, and we illustrate them on some examples. Then, we analyze the complexity of this problem.

Definition 1. Let $G = (U \cup V, E)$ be a bipartite graph, $\mathcal{U} = \{U_1, \dots, U_m\}$ be a partition of U and $\mathcal{V} = \{V_1, \dots, V_m\}$ be a partition of V . For all $i \in M = \{1, \dots, m\}$, we denote by H_i the induced subgraph $G[U_i \cup V_i]$. We say that $(G, \mathcal{U}, \mathcal{V})$ is k -Multiple Complete Matchable (k -MCM) on \mathcal{U} if and only if, for all $i \in M$, H_i is k -CM on U_i .

The graph presented on Figure 2 is 1-MCM on \mathcal{U} : $H_1 = G[U_1 \cup V_1]$ is 2-CM on U_1 and $H_2 = G[U_2 \cup V_2]$ is 1-CM on U_2 .

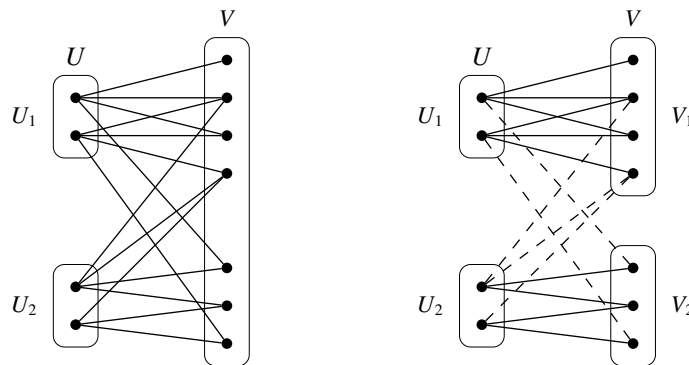


Figure 2: An instance of a MBCMVBVP and its solution.

Definition 2 (MBCMVBVP). Let $G = (U \cup V, E)$ be a bipartite graph and $\mathcal{U} = \{U_1, \dots, U_m\}$ a partition of U . The Multiple Bipartite Complete Matching Vertex Blocker Problem (MBCMVBVP) consists in finding a partition $\mathcal{V} = \{V_1, \dots, V_m\}$ of V such that $\min \{\sigma(H_i) \mid i \in M\}$ is maximum. We denote this maximum number by $\sigma(G, \mathcal{U}, \mathcal{V})$.

In Figure 2, the solution of the MBCMVBVP is given by the partition $\mathcal{V} = \{V_1, V_2\}$ with $\sigma(G, \mathcal{U}, \mathcal{V}) = 1$.

Let us now study the complexity of the MBCMVBVP.

Remark that the case $m = 1$, i.e. $|\mathcal{U}| = 1$, is equivalent to compute $\sigma(G)$. So this subproblem is polynomial from Theorem 3.

We now consider the complexity of the MBCMVBVP when \mathcal{U} is composed of more than one set. The decision problem associated with this problem is denoted by *dec-MBCMVBVP*. An instance of this decision problem is given by a graph $G = (U \cup V, E)$, a partition \mathcal{U} of U and an integer q . It consists in determining if there exists a partition \mathcal{V} of V , $|\mathcal{V}| = |\mathcal{U}|$, such that $(G, \mathcal{U}, \mathcal{V})$ is at least q -MCM on \mathcal{U} . In order to show that *dec-MBCMVBVP* is NP-complete, it is sufficient to show that it is NP-complete when $|\mathcal{U}| = 2$.

Theorem 4. The *dec-MBCMVBVP* is strongly NP-complete when $|\mathcal{U}| = 2$.

Proof. We first show that the *dec-MBCMVBVP* is in NP. Let us consider a graph $G = (\{U_1, U_2\} \cup V, E)$, an integer q and a solution given by the partition $\mathcal{V} = \{V_1, V_2\}$ of V . By Theorem 3, it is possible to test in polynomial time whether the surplus $\sigma(G[U_1 \cup V_1]) \geq q$ and $\sigma(G[U_2 \cup V_2]) \geq q$. Thus, MBCMVBVP is in NP. We now show the NP-completeness in a strong sense of the *dec-MBCMVBVP* by a transformation from the stable set decision problem presented in Section 2.

Let us consider an instance of the stable set problem $G' = (V', E')$ and a positive integer $K \leq \frac{|V'|}{2}$ (see Figure 3a). As the stable set decision problem is NP-complete for any $K \leq |V'|$ (see [4]), it is sufficient to show that the stable set decision problem with $K \leq \frac{|V'|}{2}$ can be polynomially reduced to *dec-MBCMVBVP*. We assume, without loss of generality, that the graph G' is connected. Otherwise, we consider each connected component separately. We obtain an instance of the *dec-MBCMVBVP* by constructing a graph $G = (U \cup V, E)$ and a partition $\mathcal{U} = \{U_1, U_2\}$ (see Figure 3b), as follows:

- $U = U_1 \cup U_2$, where $U_1 = \{u_{e'}^1 | e' \in E'\}$ and $U_2 = \{u^2\}$.
- $V = W_1 \cup W_2 \cup W_3$, where $W_1 = \{v_{e'}^1 | e' \in E'\}$, $W_2 = \{v_{v'}^2 | v' \in V'\}$ and $W_3 = \{v_j^3 | j \in \{1, \dots, |V'| - 2K\}\}$. Note that W_3 is a set of fictitious vertices.
- $E = E_1 \cup E_2 \cup E_3 \cup E_4$, where $E_1 = \{v_{e'}^1, u_{e'}^1 | e' \in E'\}$, $E_2 = \{v^2, u^2 | v^2 \in W_2\}$, $E_3 = \{v_{v'}^2, u_{v'}^1 | v_{v'}^2 \in W_2, u_{v'}^1 \in U_1, e' \text{ not incident to } v' \text{ in } G'\}$ and $E_4 = \{v^3, u^1 | v^3 \in W_3, u^1 \in U_1\}$.

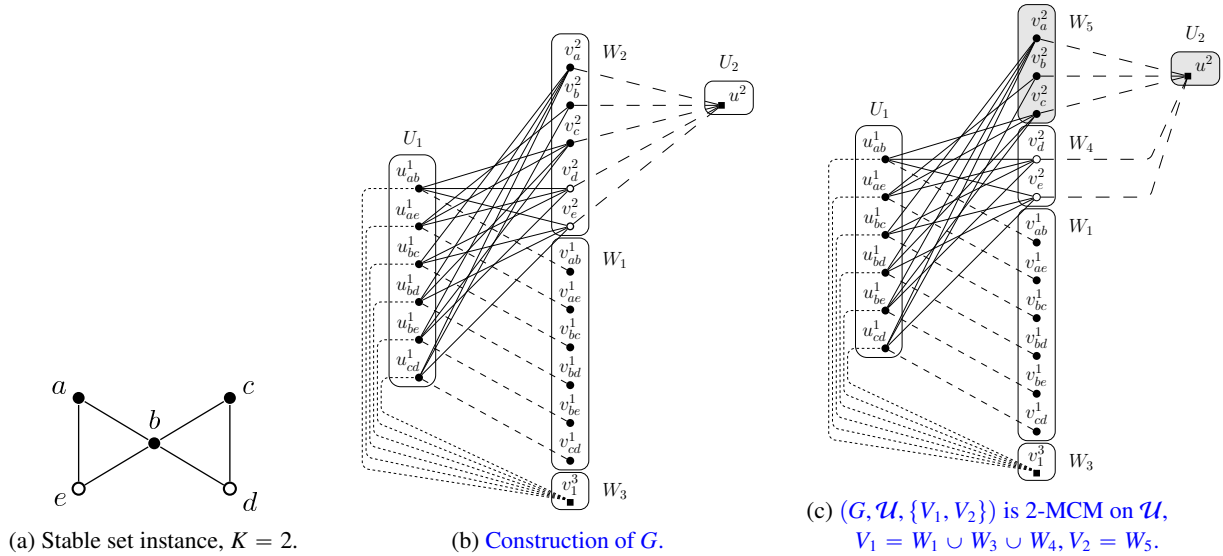


Figure 3: Stable set problem instance and the associated bipartite graph.

Recall that G' is supposed to be connected. Then, by the construction of graph G , the following observations hold.

Observation 1. For each non-empty subset W'_2 of W_2 , there exists at least one vertex of U_1 having less than $|W'_2|$ neighbors in W'_2 .

Indeed, if each vertex of U_1 had exactly $|W'_2|$ neighbors in W'_2 , the vertices of G' associated with the vertices of W'_2 would be isolated vertices, and G' would not be connected.

Observation 2. Let S be a non-empty proper subset of V' and let W'_2 be a non empty proper subset of W_2 such that $S = \{x \in V' | v_x^2 \in W'_2\}$ and $W'_2 = \{v_x^2 \in W_2 | x \in S\}$. Then S is a stable set of G' if and only if each vertex of U_1 has at least $|W'_2| - 1$ neighbors in W'_2 .

Indeed, if S is a stable set of G' , then each edge $e' \in E'$ is incident to at most one vertex of S and hence each vertex of U_1 is adjacent to at least $|W'_2| - 1$ vertices of W'_2 . Conversely, if W'_2 is such that each vertex of U_1 is adjacent to at least $|W'_2| - 1$ vertices of W'_2 , then each edge $e' \in E'$ is incident to at most one vertex of S and hence S is a stable set of G' . Remark that the number of vertices of U_1 having exactly $|W'_2| - 1$ neighbors in W'_2 is positive (by Observation 1) and equal to the number of edges of E' which are incident to exactly one vertex of S .

To complete the proof of the theorem, it is sufficient to show the following claim.

Claim 1. G' contains a stable set of size at least K , $K \leq \lfloor \frac{|V'|}{2} \rfloor$, if and only if there exists a partition $\mathcal{V} = \{V_1, V_2\}$ of V such that $(G, \mathcal{U}, \mathcal{V})$ is at least $(|V'| - K - 1)$ -MCM on \mathcal{U} .

Proof. (\Rightarrow) Let us suppose that G' contains a stable set S of size K ($S = \{d, e\}$ in Figure 3a). We split W_2 into W_4 and W_5 , W_4 being the set of vertices associated to the vertices of S ($W_4 = \{v_d^2, v_e^2\}$ in Figure 3c) and $W_5 = W_2 \setminus W_4$. Note that $|W_4| = K$ and $|W_5| = |V'| - K$. Let $H_1 = G[U_1 \cup V_1]$ where $V_1 = W_1 \cup W_3 \cup W_4$, and $H_2 = G[U_2 \cup V_2]$ where $V_2 = W_5$.

By construction of G , each non-empty subset U'_1 of U_1 has $|U'_1|$ neighbors in W_1 and $|V'| - 2K$ neighbors in W_3 . Thus, $\sigma(G[U_1 \cup (W_1 \cup W_3)]) = |V'| - 2K$. By Observation 2, each non-empty subset U'_1 of U_1 has at least $K - 1$ neighbors in $G[U_1 \cup W_4]$ and hence $\sigma_{H_1}(U'_1) \geq (|V'| - 2K) + (K - 1) = |V'| - K - 1$. Therefore, $\sigma(H_1) \geq |V'| - K - 1$.

Recall that $U_2 = \{u^2\}$. Since $N_{H_2}(u_2) = W_5$, the surplus of H_2 is $\sigma(H_2) = |W_5| - 1 = |V'| - K - 1$.

By Theorem 2, H_1 is $(|V'| - K - 1)$ -CM on U_1 and H_2 is $(|V'| - K - 1)$ -CM on U_2 . Thus, $(G, \mathcal{U}, \mathcal{V})$ is $(|V'| - K - 1)$ -MCM on \mathcal{U} .

(\Leftarrow) Suppose that there exists a partition $\mathcal{V} = \{V_1, V_2\}$ such that $H_i = G[U_i \cup V_i]$ is $(|V'| - K - 1)$ -CM on U_i (i.e. $\sigma(H_i) \geq |V'| - K - 1$), $i \in \{1, 2\}$.

Let W_5 be any subset of W_2 of cardinality at least $|V'| - K$. By choosing $V_2 = W_5$, we have $\sigma(H_2) \geq |V'| - K - 1$.

Let W_4 be any subset of $W_2 \setminus W_5$. Notice that $|W_4| \leq K$. A necessary condition for $\sigma(H_1)$ to be greater than or equal to $|V'| - K - 1$ is that $|N_{H_1}(U_1)| \geq |U_1| + (|V'| - K - 1) = |E'| + (|V'| - K - 1)$. Hence, the minimum size of V_1 is $|E'| + |V'| - K - 1$ and, by construction of G , its maximum size is $|E'| + |V'| - K$. According to this necessary condition, V_1 can be composed in three different ways :

- $V_1 = W_1 \cup W_3 \cup W_4$ where $|W_4| \in \{K - 1, K\}$,
- $V_1 = W'_1 \cup W_3 \cup W_4$ where $W'_1 \subset W_1$, $|W'_1| = |W_1| - 1$, $|W_4| = K$,
- $V_1 = W_1 \cup W'_3 \cup W_4$ where $W'_3 \subset W_3$, $|W'_3| = |W_3| - 1$, $|W_4| = K$. Note that this composition of V_1 is only possible when W_3 is not empty, i.e. when $|V'| > 2K$.

Let $H_{1 \setminus 4} = G[U_1 \cup W_4]$ and $H_{1 \setminus 4} = G[U_1 \cup (V_1 \setminus W_4)]$. Then, for each $u^1 \in U_1$, the surplus of u^1 in H_1 is $\sigma_{H_1}(u^1) = \sigma_{H_{1 \setminus 4}}(u^1) + |N_{H_{1 \setminus 4}}(u^1)|$. Therefore, a necessary condition for $\sigma(H_1)$ to be greater than or equal to $|V'| - K - 1$ is

$$\sigma_{H_{1 \setminus 4}}(u^1) + |N_{H_{1 \setminus 4}}(u^1)| \geq |V'| - K - 1, \forall u^1 \in U_1. \quad (i)$$

We now study the effect of this necessary condition on W_4 for each possible composition of V_1 .

Case 1: $V_1 = W_1 \cup W_3 \cup W_4$, $|W_4| \in \{K - 1, K\}$. Then, $H_{1 \setminus 4} = G[U_1 \cup (W_1 \cup W_3)]$ and $\sigma_{H_{1 \setminus 4}}(u^1) = |V'| - 2K$, for each vertex $u^1 \in U_1$. Condition (i) implies that $|N_{H_{1 \setminus 4}}(u^1)| \geq (|V'| - K - 1) - (|V'| - 2K) = K - 1$, for each vertex $u^1 \in U_1$. By Observation 1, the cardinality of W_4 has to be equal to K and at least one vertex of U_1 has exactly $K - 1$ neighbors in W_4 . By Observation 2, $S = \{x \in V' \mid v_x^2 \in W_4\}$ is a stable set of size K of G' .

Case 2: $V_1 = W'_1 \cup W_3 \cup W_4$, $W'_1 \subset W_1$, $|W'_1| = |W_1| - 1$, $|W_4| = K$. Then, $H_{1 \setminus 4} = G[U_1 \cup (W'_1 \cup W_3)]$. Let $\{w\} = W_1 \setminus W'_1$ and let u be the unique neighbor of w in U_1 . Then $\sigma_{H_{1 \setminus 4}}(u) = |V'| - 2K - 1$ (as $w \notin W'_1$) and $\sigma_{H_{1 \setminus 4}}(u^1) = |V'| - 2K$, for each vertex $u^1 \in U_1 \setminus \{u\}$. Therefore, condition (i) implies that:

$$\forall u^1 \in U_1, |N_{H_{1 \setminus 4}}(u^1)| \geq \begin{cases} (|V'| - K - 1) - (|V'| - 2K - 1) = K & \text{if } u^1 = u, \\ (|V'| - K - 1) - (|V'| - 2K) = K - 1 & \text{if } u^1 \neq u. \end{cases}$$

Thus, W_4 is such that the unique neighbor of w in U_1 has exactly K neighbors in W_4 , while the other vertices of U_1 have at least $K - 1$ neighbors in W_4 . By Observation 2, $S = \{x \in V' \mid v_x^2 \in W_4\}$ is a stable set of size K of G' .

Case 3: $V_1 = W_1 \cup W'_3 \cup W_4$, $W'_3 \subset W_3$, $|W'_3| = |W_3| - 1$, $|W_4| = K$, $|V'| > 2K$. Then, $H_{1 \setminus 4} = G[U_1 \cup (W_1 \cup W'_3)]$ and $\sigma_{H_{1 \setminus 4}}(u^1) = |V'| - 2K - 1$, for each vertex $u^1 \in U_1$. Condition (i) implies that $|N_{H_{1 \setminus 4}}(u^1)| \geq (|V'| - K - 1) - (|V'| - 2K - 1) = K$, for each vertex $u^1 \in U_1$. By Observation 1, such a subset W_4 does not exist. For such a composition of V_1 , H_1 cannot be $(|V'| - K - 1)$ -CM on U_1 .

We deduce from these cases that $\sigma(H_1) \geq |V'| - K - 1$ only if $S = \{x \in V' \mid v_x^2 \in W_4\}$ is a stable set of size $|W_4| = K$ of G' .

We can conclude that if there exists a partition \mathcal{V} such that $(G, \mathcal{U}, \mathcal{V})$ is $(|V'| - K - 1)$ -MCM on \mathcal{U} , then G' contains a stable set of size K .

□

We can deduce the following corollary.

Corollary 1. *MBCMVBP is strongly NP-Hard when $|\mathcal{U}| \geq 2$.*

4. Integer linear programs for MBCMVBP

In this section, we propose two models representing the MBCMVBP. The first model, called dual based formulation, is based on the integer linear program to solve the MBCMVBP when $|\mathcal{U}| = 1$. This problem has been introduced in [8] and will be called BCMVBVP for short. The second model, called natural formulation, contains an exponential number of inequalities allowing to satisfy the property of Theorem 2, for all $U_i \in \mathcal{U}, i \in M$.

The comparison of these two models is presented in Section 5, where some possible improvements are also proposed.

4.1. Bipartite Complete Matching Vertex Blocker Problem (BCMVBVP)

We present the integer linear program given in [8] to solve BCMVBVP.

Let $x \in \{0, 1\}^{|U \cup V|}$ be defined by

$$x_u = \begin{cases} 1 & \text{if } u \in U', \\ 0 & \text{otherwise,} \end{cases} \quad \forall u \in U,$$

$$x_v = \begin{cases} 1 & \text{if } v \in N_G(U'), \\ 0 & \text{otherwise,} \end{cases} \quad \forall v \in V.$$

The BCMVBVP can be formulated as follows:

$$\min \left(\sum_{v \in V} x_v - \sum_{u \in U} x_u \right) \tag{1}$$

$$\sum_{u \in U} x_u \geq 1, \tag{2}$$

$$(IP) \quad x_u - x_v \leq 0, \quad \forall uv \in E, u \in U, v \in V, \tag{3}$$

$$x_w \in \{0, 1\}, \quad \forall w \in U \cup V. \tag{4}$$

The objective function expresses Theorem 2. Inequality (2) guarantees that the set U' is not empty. Inequalities (3) ensure that the variables x_v associated with the neighborhood of U' will be set to 1.

We denote by (P) the linear relaxation of (IP) , replacing constraint (4) by

$$x_w \leq 1, \quad \forall w \in U \cup V, \tag{5}$$

$$-x_w \leq 0, \quad \forall w \in U \cup V. \tag{6}$$

4.2. Dual based formulation

The first integer linear program is based on the dual of the model (P). If we maximize the smallest difference on each graph $H_i = G[U_i \cup V_i]$, $i \in M$, then we do not find $\sigma(H_i)$ since $\sigma(H_i)$ is given by the smallest difference. Hence, the model of MBCMVBP cannot be derived directly from (P) by maximizing the smallest surplus. We propose to use the dual formulation that can be generalized and verifies the integrality property.

Let us first study the BCMVBP. In order to convert the linear relaxation (P) of (IP) into a dual problem (D), we consider the subproblem $(P_{\tilde{u}})$ of (P) where $\tilde{u} \in U'$ and the inequality (2) of (P) is replaced by the equality $x_{\tilde{u}} = 1$:

$$\begin{aligned} & \min \left(\sum_{v \in V} x_v - \sum_{u \in U} x_u \right) \\ (P_{\tilde{u}}) \quad & y_{\tilde{u}} : \quad x_{\tilde{u}} = 1, \tag{7} \\ & y_{uv} : \quad x_v - x_u \geq 0, \quad \forall uv \in E, u \in U, v \in V, \tag{8} \\ & y_v : \quad -x_v \geq -1, \quad \forall v \in V, \tag{9} \\ & \quad \quad \quad x_w \geq 0, \quad \forall w \in (U \cup V) \setminus \{\tilde{u}\}. \tag{10} \end{aligned}$$

Note that the optimal value of (P) is the smallest optimal value of all $P_{\tilde{u}}$, $\tilde{u} \in U'$.

For each $\tilde{u} \in U$, the dual program $(D_{\tilde{u}})$ of $(P_{\tilde{u}})$ is defined as follows:

$$\begin{aligned} & \max \left(y_{\tilde{u}} - \sum_{v \in V} y_v \right) \\ (D_{\tilde{u}}) \quad & x_{\tilde{u}} : \quad \sum_{v \in N_G(\tilde{u})} y_{\tilde{u}v} - y_{\tilde{u}} \geq 1, \tag{11} \\ & x_u : \quad \sum_{v \in N_G(u)} y_{uv} \geq 1, \quad \forall u \in U \setminus \{\tilde{u}\}, \tag{12} \\ & x_v : \quad \sum_{u \in N_G(v)} y_{uv} - y_v \leq 1, \quad \forall v \in V, \tag{13} \\ & \quad \quad \quad y_{w'} \geq 0, \quad \forall w' \in (U \cup V) \setminus \{\tilde{u}\}, \tag{14} \\ & \quad \quad \quad y_{uv} \geq 0, \quad \forall uv \in E, u \in U, v \in V \tag{15} \\ & \quad \quad \quad y_{\tilde{u}} \text{ free.} \tag{16} \end{aligned}$$

The minimum value of the $(D_{\tilde{u}})$ programs, for all $\tilde{u} \in U$, corresponds to an optimal solution of the following linear program (D):

$$\begin{aligned} & \max y \tag{17} \\ (D) \quad & x_{\tilde{u}} : \quad \sum_{v \in N_G(\tilde{u})} y_{\tilde{u}v} - y_{\tilde{u}} \geq 1, \quad \forall \tilde{u} \in U, \tag{18} \\ & x_u : \quad \sum_{v \in N_G(u)} y_{uv} \geq 1, \quad \forall \tilde{u} \in U, \forall u \in U \setminus \{\tilde{u}\}, \tag{19} \\ & x_v : \quad \sum_{u \in N_G(v)} y_{uv} - y_v \leq 1, \quad \forall \tilde{u} \in U, \forall v \in V, \tag{20} \\ & \quad \quad \quad y \leq y_{\tilde{u}} - \sum_{v \in V} y_v, \quad \forall \tilde{u} \in U, \tag{21} \\ & \quad \quad \quad y_w^{\tilde{u}} \geq 0, \quad \forall \tilde{u} \in U, \forall w \in (U \cup V) \setminus \{\tilde{u}\}, \tag{22} \\ & \quad \quad \quad y_{uv}^{\tilde{u}} \geq 0, \quad \forall \tilde{u} \in U, \forall uv \in E, u \in U, v \in V, \tag{23} \\ & \quad \quad \quad y_{\tilde{u}}^{\tilde{u}} \text{ free,} \quad \forall \tilde{u} \in U. \tag{24} \end{aligned}$$

Note that the matrix corresponding to $(D_{\tilde{u}})$ is totally unimodular (see [8]) and the coefficients of the objective function and the right hand side constraints are integers. Therefore, the optimal solution and the objective function value are integer. By the duality Theorem, optimal solutions of both programs (D) and (P) are integer. Indeed, the variable y maximizes the smallest value of each $(D_{\tilde{u}})$, $\tilde{u} \in U$, and the linear programs $(D_{\tilde{u}})$ are integer, and thus the optimal solution given by y is integer.

The program (D) allows to solve the BCMVBP.

We denote by (ID^i) the program associated with $U_i \in \mathcal{U}$. Now we extend this formulation to solve the MBCMVBP by merging the programs (ID^i) , $i \in M$. We define the binary variables

$$x_v^i = \begin{cases} 1 & \text{if } v \in V_i, \\ 0 & \text{otherwise,} \end{cases} \quad \forall v \in V \text{ and } \forall i \in M.$$

We can deduce the following integer linear program:

$$\begin{aligned} & \max y \\ & \sum_{v \in N_{H_i}(\tilde{u})} y_{uv}^{\tilde{u}} - y_{\tilde{u}}^{\tilde{u}} \geq 1, \quad \forall i \in M, \forall \tilde{u} \in U_i, & (25) \\ & \sum_{v \in N_{H_i}(u)} y_{uv}^{\tilde{u}} \geq 1, \quad \forall i \in M, \forall \tilde{u} \in U_i, \forall u \in U_i \setminus \{\tilde{u}\}, & (26) \\ & \sum_{u \in N_{H_i}(v)} y_{uv}^{\tilde{u}} - y_v^{\tilde{u}} \leq 1, \quad \forall i \in M, \forall v \in V, \forall \tilde{u} \in U_i, & (27) \\ & y^i \leq y_{\tilde{u}}^{\tilde{u}} - \sum_{v \in V} y_v^{\tilde{u}}, \quad \forall i \in M, \forall \tilde{u} \in U_i, & (28) \\ (ID') \quad & y \leq y^i, \quad \forall i \in M, & (29) \\ & y_{uv}^{\tilde{u}} \leq \mu x_v^i \quad \forall i \in M, \forall \tilde{u} \in U_i, \forall uv \in E, u \in U_i, v \in V, & (30) \\ & y_v^{\tilde{u}} \leq \mu x_v^i \quad \forall i \in M, \forall \tilde{u} \in U_i, \forall v \in V, & (31) \\ & \sum_{i \in M} x_v^i = 1 \quad \forall v \in V, & (32) \\ & y_w^{\tilde{u}} \geq 0, \quad \forall i \in M, \forall \tilde{u} \in U_i, \forall w \in (U_i \cup V) \setminus \{\tilde{u}\}, & (33) \\ & y_{uv}^{\tilde{u}} \geq 0, \quad \forall i \in M, \forall \tilde{u} \in U_i, \forall uv \in E, u \in U_i, v \in V, & (34) \\ & y_{\tilde{u}}^{\tilde{u}} \text{ free,} \quad \forall i \in M, \forall \tilde{u} \in U_i, & (35) \\ & x_v^i \in \{0, 1\}, \quad \forall i \in M, \forall v \in V, & (36) \end{aligned}$$

where μ is a big constant value.

Inequalities (25) – (28) extend (18) – (21), (29) is derived from the objective function (17) and (30) – (32) ensure the partitioning. Indeed, $\forall v \in V$, $i \in M$, if $x_v^i = 1$, then all variables y associated with v in (30) – (31) are free, thanks to the constant μ . Otherwise, all these values are equal to 0 and thus are not considered in the subset V_i . A solution where all x_i are fixed to 0 or 1 implies an integer value for all variables.

The following proposition can improve the dual formulation by reducing the number of variables and inequalities.

Proposition 1. *Let $\tilde{u}_1, \tilde{u}_2 \in U$ and let $(P_{\tilde{u}_1})$ and $v(P_{\tilde{u}_2})$ be the values of $(P_{\tilde{u}_1})$ and $(P_{\tilde{u}_2})$, respectively. If $N_G(\tilde{u}_1) \subset N_G(\tilde{u}_2)$ then $v(P_{\tilde{u}_1}) \leq v(P_{\tilde{u}_2})$.*

Proof. For all $U' \subseteq U$ containing \tilde{u}_1 and \tilde{u}_2 , we have

$$|N_G(U')| - |U'| = |N_G(U' \setminus \{\tilde{u}_1\})| - |U'| \leq |N_G(U' \setminus \{\tilde{u}_1\})| - |U' \setminus \{\tilde{u}_1\}|.$$

Let U'_2 be a subset associated to the optimal solution of $(P_{\tilde{u}_2})$. The inequality above implies that $\tilde{u}_1 \in U'_2$.

Let U'_1 be any subset of U containing \tilde{u}_1 but not \tilde{u}_2 . By constructing a subset $U'_2 = (U'_1 \setminus \{\tilde{u}_1\}) \cup \{\tilde{u}_2\}$ for each U'_1 , we obtain all subsets of U containing \tilde{u}_2 but not \tilde{u}_1 . As $N_G(\tilde{u}_1) \subseteq N_G(\tilde{u}_2)$, the inequality $|N_G(U'_1)| - |U'_1| \leq |N_G(U'_2)| - |U'_2|$ holds for all U'_1 , where $U'_2 = (U'_1 \setminus \{\tilde{u}_1\}) \cup \{\tilde{u}_2\}$. According to Theorem 2, $v(P_{\tilde{u}_1}) \leq v(P_{\tilde{u}_2})$.

□

We denote by \tilde{U} the set of vertices $\tilde{u} \in U$ such that there does not exist any $\tilde{u}' \in U$ verifying $N_G(\tilde{u}') \subset N_G(\tilde{u})$. Considering only the variables associated with \tilde{U} in the dual formulation is sufficient. The impact of this property will be analyzed in the experimental study of Section 5.3.1.

4.3. Natural formulation

In this section, we propose an alternative formulation using a smaller number of variables and without the big constant μ .

4.3.1. Integer linear program formulations

Recall that our goal is to find a partition $\mathcal{V} = \{V_1, \dots, V_m\}$ of V such that $(G, \mathcal{U}, \mathcal{V})$ is $\sigma(G, \mathcal{U}, \mathcal{V})$ -MCM on \mathcal{U} .

We now present the integer linear program formulated to find \mathcal{V} . Let $x_v^i, \forall v \in V, \forall i \in M$, be defined as in Section 4.2, and let $z \in \mathbb{N}$ be the value of $\sigma(G, \mathcal{U}, \mathcal{V})$. The expressed model follows:

max z

$$\sum_{i \in M} x_v^i = 1, \quad \forall v \in V, \tag{37}$$

$$(P') \quad \sum_{v \in N_G(U')} x_v^i - z \geq |U'|, \quad \forall i \in M, \forall U' \subseteq U_i, \tag{38}$$

$$x_v^i \in \{0, 1\}, \quad \forall v \in V, \forall i \in M, \tag{39}$$

$$z \in \mathbb{N}, \tag{40}$$

where equalities (37) ensure that each vertex $v \in V$ belongs to only one set of the partition \mathcal{V} and inequalities (38) verify Theorem 2 on each subgraph $H_i = G[U_i \cup V_i], i \in M$.

4.3.2. Polyhedral study

To simplify the polyhedral study without modifying the optimal solution, we relax the constraints (37) as follows:

$$\sum_{i \in M} x_v^i \leq 1, \quad \forall v \in V. \tag{41}$$

Let $P(G, \mathcal{U})$ be the convex hull of the solutions of program (P') , that is,

$$P(G, \mathcal{U}) = \text{conv}(\{(x, z) \in \{0, 1\}^{|V| \times m} \times \mathbb{N} \mid x, z \text{ satisfy (38) and (41)}\}).$$

Replacing inequalities (37) by (41) implies that some vertices may not be assigned to any set of the partition \mathcal{V} . The set of these vertices is denoted by \check{V}_0 . Thus, a solution of $P(G, \mathcal{U})$ is a couple $(\check{\mathcal{V}}, z')$ where $\check{\mathcal{V}} = \{\check{V}_0, \check{V}_1, \dots, \check{V}_m\}$ is a partition of V and z' is an integer such that $z' \leq \sigma(G, \mathcal{U}, \check{\mathcal{V}} \setminus \check{V}_0)$. The incidence vector $x_{\check{v}} \in \{0, 1\}^{|V| \times m}$ of the partition $\check{\mathcal{V}}$ is then defined by

$$x_{\check{v}}^i = \begin{cases} 1 & \text{if } \check{v} \in \check{V}_i \\ 0 & \text{otherwise,} \end{cases} \quad \forall \check{v} \in \check{V}_i, \forall i \in M.$$

Note that, for all $\check{v} \in \check{V}_0, \sum_{i=1}^m x_{\check{v}}^i = 0$.

Hypothesis 1. For all $v \in V$, there exists a matching in $G \setminus \{v\}$.

This hypothesis can be verified in polynomial time. If the hypothesis does not hold, then $z = 0$.

Proposition 2. *The polytope $P(G, \mathcal{U})$ is full dimensional if and only if a 1-MCM solution exists.*

Proof.

(\Rightarrow) If a 1-MCM solution does not exist, then $z = 0$ and the polytope is not full dimensional.

(\Leftarrow) We suppose that $P(G, \mathcal{U})$ is contained in a hyperplane defined by the linear equation

$$ax + bz = \alpha \tag{42}$$

where $a = (a_v^i, v \in V, i \in M) \in \mathbb{R}^{|V| \times m}$, $b \in \mathbb{R}$ and $\alpha \in \mathbb{R}$. We show that $a = 0$, $b = 0$ and $P(G, \mathcal{U})$ cannot be included in the hyperplane (42), since it is not empty. As there exists a 1-MCM solution, there exists a solution where $z = 1$. We denote this solution by $(\check{V}, 1)$. It is clear that $(\check{V}, 0)$ is also valid for $P(G, \mathcal{U})$ and thus verifies (42). Then we deduce $b = 0$.

Let $v \in V$. There exists a solution $(\check{V}_v, 0)$ which is valid for $P(G, \mathcal{U})$ where $\check{V}_v = \{\check{V}_{v,0}, \check{V}_{v,1}, \dots, \check{V}_{v,m}\}$ and $v \in \check{V}_{v,0}$. Otherwise, a 1-MCM solution does not exist.

Let $i \in M$ and let $\check{V}_{v,i}$ be the partition obtained from \check{V}_v by moving the vertex v from $\check{V}_{v,0}$ to $\check{V}_{v,i}$. Then, the solution $(\check{V}_{v,i}, 0)$ is also valid for $P(G, \mathcal{U})$. We deduce that, for all $i \in M$, $a^{\check{V}_v} = a^{\check{V}_{v,i}}$. This equality holds for any vertex $v \in V$, we can thus conclude that $a = 0$ and hence $P(G, \mathcal{U})$ is full dimensional. \square

Proposition 3. *If a 1-MCM solution does not exist, then $\dim(P(G, \mathcal{U})) = |V| \times m$.*

Proof. Clearly, we have the equality $z = 0$. Now we prove that all equalities of $P(G, \mathcal{U})$ are equivalent to $z = 0$. We suppose that $P(G, \mathcal{U})$ is contained in a hyperplane defined by the linear equation

$$ax = \alpha \tag{43}$$

where $a = (a_v^i, v \in V, i \in M) \in \mathbb{R}^{|V| \times m}$ and $\alpha \in \mathbb{R}$. We show that $a = 0$ and that $P(G, \mathcal{U})$ cannot be included in the hyperplane (43), since it is not empty.

Let $v \in V$, $i \in M$ and let $\check{V}_v, \check{V}_{v,i}$ be defined as in the proof of Proposition 2. By Hypothesis 1, there exists a solution $(\check{V}_v, 0)$ which is valid for $P(G, \mathcal{U})$, where $v \in \check{V}_{v,0}$. The solution $(\check{V}_{v,i}, 0)$ is also valid for $P(G, \mathcal{U})$ and we deduce that, for all $i \in M$, $a^{\check{V}_v} = a^{\check{V}_{v,i}}$. Thus, $a = 0$ and $\dim(P(G, \mathcal{U})) = |V| \times m$. \square

In the following, for a better understanding of the proofs, we consider that a 1-MCM solution exists. The proofs can be adapted for the case when no 1-MCM solution exists.

Proposition 4. *The inequality (41) associated with $v \in V$ defines a facet of $P(G, \mathcal{U})$.*

Proof. Let us denote by $ax + bz \leq \alpha$ the inequality (41) associated with $v \in V$. Let $a'x + b'z \leq \beta$ be a facet defining an inequality of $P(G, \mathcal{U})$ such that $\{x \in P(G, \mathcal{U}) : ax + bz = \alpha\} \subseteq \{x \in P(G, \mathcal{U}) : a'x + b'z = \beta\}$. We show that $a' = \rho a$ and $b' = \rho b$ for some $\rho \in \mathbb{R}$.

As a 1-MCM solution exists, there exists a solution where $z = 1$. We denote this solution by $(\check{V}, 1)$. It is clear that $(\check{V}, 0)$ is also valid for $P(G, \mathcal{U})$ and thus verifies (42). Then, we deduce $b = 0$.

Let $v' \in V \setminus \{v\}$ and $i \in M$. By hypothesis, there exists a solution $(\check{V}_{v'}, 0)$ which is valid for $P(G, \mathcal{U})$ where $v' \in \check{V}_{v',0}$. It is clear that the solution $(\check{V}_{v',i}, 0)$ is also valid for $P(G, \mathcal{U})$. We deduce $a^{\check{V}_{v'}} = a^{\check{V}_{v',i}}$ for all $i \in M$. Thus, $a_{v'}^i = 0$ for all $i \in M$ and $v' \in V \setminus \{v\}$.

Furthermore, there also exists a valid solution $(\check{V}'_v, 0)$ for $P(G, \mathcal{U})$ where $v \in \check{V}'_{v,0}$. It is clear that the solution $(\check{V}'_{v,i}, 0)$ is also valid for $P(G, \mathcal{U})$. We deduce that $a^{\check{V}'_v} = a^{\check{V}'_{v,i}}$ for all $i, j \in M$. We set $a_v^i = \rho$ and we deduce $a_v^i = a_v^j = \rho$ for all $i, j \in M$. \square

Proposition 5. *The inequality (38) associated with $i \in M$ and $U' \subseteq U_i$ defines a facet of $P(G, \mathcal{U})$ if and only if*

1. *if $U' \subset U_i$, then there does not exist $U'' \subseteq U_i$ where $U' \subset U''$ such that $|N_G(U')| - |U'| \geq |N_G(U'')| - |U''|$,*
2. *there does not exist $U'' \subseteq U_i$ where $|U''| = |U'|$ such that $N_G(U'') \subset N_G(U')$,*

3. $G[U' \cup N_G(U')]$ is connected,

4. there exists a partition $\check{V} = \{\check{V}_0, \dots, \check{V}_m\}$ and $k > 0$ such that $(G, \mathcal{U}, \check{V})$ is k -MCM on \mathcal{U} , satisfying $|\check{V}_i \cap N_G(U')| = |U'| + k$.

Proof.

(\Rightarrow)

1. Suppose there exists $U'' \subseteq U_i$, where $U' \subset U''$, such that $|N_G(U')| - |U'| \geq |N_G(U'')| - |U''|$. Then, the inequality (38) associated with i and U'' can be expressed as follows:

$$\sum_{v \in N_G(U')} x_v^i - z + \sum_{v \in N_G(U'') \setminus N_G(U')} x_v^i \geq |U'| + |U'' \setminus U'|.$$

The inequality (38) associated with i and U' is a linear combination of the latter inequality and the trivial inequalities

- $-x_v^i \geq -1$, for all $v \in N_G(U'') \setminus N_G(U')$,
- $0 \geq -1$, $|U'' \setminus U'| - |N_G(U'') \setminus N_G(U')|$ times.

2. If there exists $U'' \subseteq U_i$ where $|U''| = |U'|$ such that $N_G(U'') \subset N_G(U')$, then the inequality (38) associated with i and U' is a linear combination of the inequality (38) associated with i and U'' and the trivial inequality $x_v^i \geq 0$ for all $v \in N_G(U') \setminus N_G(U'')$.

3. Suppose that $G[U' \cup N_G(U')]$ is not connected. Let $U_1 \subset U'$ such that $G[U_1 \cup N_G(U_1)]$ is a connected component. In this case, the inequality (38) associated with i and U' is a linear combination of the inequality (38) associated with i and U_1 , the inequality (38) associated with i and $U' \setminus U_1$ and the trivial inequality $z \geq 0$.

4. Suppose that there does not exist a partition $\check{V} = \{\check{V}_0, \dots, \check{V}_m\}$ such that $(G, \mathcal{U}, \check{V})$ is k -MCM on \mathcal{U} , satisfying $|\check{V}_i \cap N_G(U')| = |U'| + k$. Then, $|\check{V}_i \cap N_G(U')| > |U'| + k$, for all k -MCM solutions $(G, \mathcal{U}, \check{V} = \{\check{V}_0, \dots, \check{V}_m\})$. We deduce that the inequality $\sum_{v \in N_G(U')} x_v^i - z \geq |U'| + 1$ is valid and dominates the inequality (38) associated with i and U' .

(\Leftarrow) Let us denote by $ax + bz \leq \alpha$ the inequality (38) associated with $i \in M$ and $U' \subseteq U_i$. Let $a'x + b'z \leq \beta$ be a facet defining an inequality of $P(G, \mathcal{U})$ such that $\{x \in P(G, \mathcal{U}) : ax + bz = \alpha\} \subseteq \{x \in P(G, \mathcal{U}) : a'x + b'z = \beta\}$. We show that $a' = \rho a$ and $b' = \rho b$ for some $\rho \in \mathbb{R}$.

Conditions 1–3 ensure that there exists a matching A such that $|A \cap G[U', V']| = |U'|$, for all $V' \subseteq N_G(U')$ where $|V'| = |U'|$. Let $v \in V \setminus N_G(U')$. By Hypothesis 1, there exists a complete matching A on U where v is not covered by A . The solution associated with A is denoted by $(\check{V}_v, 0)_A$. It is clear that the solutions $(\check{V}_{v,j}, 0)_A$, where $j \in M$, are valid. All these solutions are valid for $P(G, \mathcal{U})$ and verify the inequality (38) associated with U' to the equality. We deduce $a^{\check{V}_v} = a^{\check{V}_{v,j}}$, for all $j \in M$. Thus, for all $V' \subseteq N_G(U')$ where $|V'| = |U'|$, we have $a_v^{j'} = 0$, for all $j \in M$ and $v \in V \setminus N_G(U')$.

Let $v \in N_G(U')$. By Hypothesis 1, there exists a complete matching A on U where v is not covered by A . The solution associated with A is denoted by $(\check{V}_v, 0)_A$. It is clear that the solutions $(\check{V}_{v,j}, 0)_A$, where $j \in M \setminus \{i\}$, are valid. All these solutions are valid for $P(G, \mathcal{U})$ and verify the inequality (38) associated with U' to the equality. We deduce $a^{\check{V}_v} = a^{\check{V}_{v,j}}$, for all $j \in M \setminus \{i\}$. Thus, we have $a_v^{j'} = 0$, for all $j \in M \setminus \{i\}$ and $v \in N_G(U')$.

Let $v_1, v_2 \in N_G(U')$. By Hypothesis 1, there exists a complete matching A_1 without v_1 and with v_2 . As $G[U', N_G(U')]$ is connected, there exists an alternative path between v_1 and v_2 . We define the complete matching A_2 using this alternative path. The solutions $(\check{V}_{v_1}, 0)_{A_1}$ and $(\check{V}_{v_2}, 0)_{A_2}$ associated respectively to A_1 and A_2 are valid for $P(G, \mathcal{U})$ and verify the inequality (38) associated with U' to the equality. We deduce $a^{\check{V}_{v_1}} = a^{\check{V}_{v_2}}$. Thus, $a_{v_1}^{i'} = a_{v_2}^{i'} = \rho$, for all $v_1, v_2 \in N_G(U')$.

By condition 4, there exists a partition $\check{V} = \{\check{V}_0, \dots, \check{V}_m\}$ such that $(G, \mathcal{U}, \check{V})$ is k -MCM on \mathcal{U} , satisfying $|\check{V}_i \cap N_G(U')| = |U'| + k$. Let $v \in \check{V}_i$ and $\check{V}' = \{\check{V}_0 \cup \{v\}, \check{V}_i \setminus \{v\}, (\check{V}_j)_{j \in M \setminus \{0,i\}}\}$. The solutions (\check{V}', k) and $(\check{V}', k - 1)$ are valid for $P(G, \mathcal{U})$ and verify the inequality (38) associated with U' to the equality. We deduce $a^{\check{V}'} + b^{\check{V}'} = a^{\check{V}''} + b^{\check{V}''}$. Thus, $a_v^{i'} = -b$ and so $b' = -\rho$. \square

In the following lemma, we show that condition 4 does not imply condition 1. Moreover, condition 4 is stronger than conditions 2 and 3, **but these two conditions** are easier to implement, as verifying condition 4 is NP-hard (since the k -MCM property must be checked).

Lemma 1. *Let us consider the inequality (38) associated with $i \in M$ and $U' \subseteq U_i$. If there exists a partition $\check{V} = \{\check{V}_0, \dots, \check{V}_m\}$ and $k > 0$ such that $(G, \mathcal{U}, \check{V})$ is k -MCM on \mathcal{U} , satisfying $|\check{V}_i \cap N_G(U')| = |U'| + k$, then*

1. *if $U' \subset U_i$ then there can exist $U'' \subseteq U_i$ where $U' \subset U''$ such that $|N_G(U')| - |U'| \geq |N_G(U'')| - |U''|$,*
2. *there does not exist $U'' \subseteq U_i$ where $|U''| = |U'|$ such that $N_G(U'') \subset N_G(U')$,*
3. *$G[U' \cup N_G(U')]$ is connected.*

Proof.

1. If $N_G(U') \subset \check{V}_i$ then, by hypothesis, we have $|\check{V}_i \cap N_G(U')| = |N_G(U')| = |U'| + k$. Let $U'' \subseteq U_i$, where $U' \subset U''$ and let $|N_G(U'')| = |N_G(U')| + \ell_1$ and $|U''| = |U'| + \ell_2$. We have $|N_G(U'')| - |U''| = |N_G(U')| - |U'| + \ell_1 - \ell_2$. By choosing $\ell_1 = \ell_2$, we obtain $|N_G(U')| - |U'| = |N_G(U'')| - |U''| = k$.
2. We suppose that there exists a subset $U'' \subseteq U_i$, where $|U''| = |U'|$, such that $N_G(U'') \subset N_G(U')$. Then, $|N_G(U'')| < |N_G(U')|$, so we have $|U''| + k = |U'| + k = |\check{V}_i \cap N_G(U')| > |\check{V}_i \cap N_G(U'')|$ and so $k > |\check{V}_i \cap N_G(U'')| - |U''|$ which leads to a contradiction with the extended Hall theorem: $k \leq |\check{V}_i \cap N_G(U'')| - |U''|$.
3. The graph $G[U' \cup (N_G(U') \cap \check{V}_i)]$ is connected, otherwise $|\check{V}_i \cap N_G(U')| - |U'| > k \times \ell$, where ℓ is the number of connected subgraphs in U' . We deduce that $G[U' \cup N_G(U')]$ is connected.

□

4.3.3. Valid inequalities

Recall that inequalities (38) give an upper bound on z when considering each subset $U_i \in \mathcal{U} = \{U_1, \dots, U_m\}$ separately. We now extend this family of inequalities by considering these subsets two by two, simultaneously. We illustrate the interest of this approach in the following example. Let $U_s, U_t \in \mathcal{U}$, $s \neq t$, and suppose that $|U_s| = |U_t| = 2$ and $N_G(U_s) = N_G(U_t)$ with cardinality 4. Using inequalities (38), z is bounded by 2. However, it is straightforward that $z = 0$.

For any subsets $U'_s \subseteq U_s$ and $U'_t \subseteq U_t$, the set of common neighbors of U'_s and U'_t is denoted by $N_G(U'_s U'_t) = N_G(U'_s) \cap N_G(U'_t)$. Let us consider $\tilde{N}_G(U'_s) = N_G(U'_s) \setminus N_G(U'_s U'_t)$ the set of vertices belonging exclusively to the neighborhood of U'_s . In the same way, we define $\tilde{N}_G(U'_t) = N_G(U'_t) \setminus N_G(U'_s U'_t)$.

Let us denote by

$$\tilde{k}_{\max} = \max \left\{ |\tilde{N}_G(U'_s)| - |U'_s|, |\tilde{N}_G(U'_t)| - |U'_t| \right\} \quad \text{and} \quad \tilde{k}_{\min} = \min \left\{ |\tilde{N}_G(U'_s)| - |U'_s|, |\tilde{N}_G(U'_t)| - |U'_t| \right\}$$

the upper bounds of the optimal solution for U_s and U_t when considering only the exclusive neighborhoods. Then, it is clear that the upper bound k_{sup} of the optimal solutions is less than or equal to $\tilde{k}_{\min} + |N_G(U'_s U'_t)|$. In order to improve k_{sup} , we complete $\tilde{N}_G(U'_t)$ and/or $\tilde{N}_G(U'_s)$ with some vertices of the common neighborhood $N_G(U'_s U'_t)$. Two cases have to be studied:

1. $\tilde{k}_{\max} - \tilde{k}_{\min} \geq |N_G(U'_s U'_t)|$: only one of the two subsets, $\tilde{N}_G(U'_s)$ or $\tilde{N}_G(U'_t)$, can be completed with the common vertices in such a way that $k_{\text{sup}} = \tilde{k}_{\min} + |N_G(U'_s U'_t)|$.
2. $\tilde{k}_{\max} - \tilde{k}_{\min} < |N_G(U'_s U'_t)|$: first, one of the two subsets, $\tilde{N}_G(U'_s)$ or $\tilde{N}_G(U'_t)$, can be completed with $\tilde{k}_{\max} - \tilde{k}_{\min}$ common vertices. Then, the remaining common vertices can complete both $\tilde{N}_G(U'_s)$ and $\tilde{N}_G(U'_t)$ equally. We deduce that $k_{\text{sup}} = \tilde{k}_{\max} + \left(|N_G(U'_s U'_t)| - (\tilde{k}_{\max} - \tilde{k}_{\min}) \right) / 2$.

A common vertex $v \in N_G(U'_s U'_t)$ cannot be used to complete $\tilde{N}_G(U'_s)$ or $\tilde{N}_G(U'_t)$ if it is assigned to a subset V_i in a solution, where $i \in M \setminus \{s, t\}$. Then, k_{sup} will be decreased by at least $1/2$. Let $\ell = s$ if $|\tilde{N}_G(U'_s)| - |U'_s| < |\tilde{N}_G(U'_t)| - |U'_t|$, let $\ell = t$ otherwise. A vertex $v \in \tilde{N}_G(U'_\ell)$ cannot be used to complete $\tilde{N}_G(U'_s)$ or $\tilde{N}_G(U'_t)$ if it is assigned to a subset V_i in a solution, where $i \in M \setminus \{s, t\}$. Then, k_{sup} will be decreased by 1. Thus the following inequalities are valid for all U'_s and U'_t :

$$z + \frac{1}{2} \sum_{v \in N_G(U'_s U'_t)} (1 - x_v^s - x_v^t) + \sum_{v \in N_G(U'_\ell)} (1 - x_v^s - x_v^t) \leq k_{sup}. \tag{44}$$

5. Implementation and experimental results

We first present the strategies used to solve our problems, explain how instances have been produced, and then analyze the obtained results.

5.1. Solving strategies

For the exact solution, we use a Branch-and-Cut algorithm where the separation strategies are based on inequalities (38) and (44). We also propose some improvements for the first one and explain how our Branch-and-Cut algorithm is managed.

5.1.1. Separation algorithm

Given a solution $(x^*, z^*) \in \mathbb{R}_+^{|V| \times m} \times \mathbb{R}_+$, the *separation problem* for a family of inequalities consists in determining whether (x^*, z^*) satisfies these inequalities. If it does not, it consists in finding one of these inequalities being violated by (x^*, z^*) . An algorithm solving this problem is called a separation algorithm associated with these inequalities.

Separation of inequalities (38). The number of inequalities (38) in (P') is exponential. Hence, a **polynomial-time** separation algorithm is necessary to allow the use of these inequalities inside a cutting plane algorithm. The equivalence between separation and optimization in combinatorial optimization [5] implies that the linear relaxation of the problem (P') can be solved in polynomial time. Let $(x^*, z^*) \in \mathbb{R}_+^{|V| \times m} \times \mathbb{R}_+$ be a solution of the linear relaxation. For all $i \in M$, we have to find a set $U'_i \subseteq U_i$ such that $\sum_{v \in N_G(U'_i)} x_v^{i*} - z^* < |U'_i|$.

We decompose the problem into m subproblems. Each subproblem is associated with a single set $U_i \subseteq U$. In the following, we consider the graph $G[U_i \cup V]$. Let $w : v \rightarrow \mathbb{R}_+$ be the weight function, where $w(v) = x_v^{i*}$ for all $v \in V$. The separation problem consists in solving the integer linear program (P') with the objective function $z' = \min(\sum_{v \in V} w(v)x_v - \sum_{u \in U_i} x_u)$. Let U'_i be given by the optimal solution of (P') . If $z' < z^*$, the inequality $\sum_{v \in N_G(U'_i)} x'_v - z \geq |U'_i|$ is violated when respecting (x^*, z^*) . In this case, we add the inequality associated with i and U'_i to the model. Otherwise, all inequalities (38) are verified.

As the added constraint might not define a facet, we propose some improvements according to conditions 1 – 3 of Proposition 5.

Improvement 1. If the resulting U'_i does not respect condition 1, there exists a set \tilde{U}'_i where $\sum_{v \in N_G(\tilde{U}'_i) \setminus N_G(U'_i)} w(v) = |\tilde{U}'_i|$ and $|N_G(\tilde{U}'_i) \setminus N_G(U'_i)| = |\tilde{U}'_i|$, implying $w(v) = 1$ for all $v \in N_G(\tilde{U}'_i) \setminus N_G(U'_i)$. Thus, we aim to maximize $|\tilde{U}'_i|$. We ensure the maximality of U'_i by subtracting $\sum_{u \in U} \epsilon x_u$ from the objective function, where ϵ is a small value. Figure 4a shows an example of valid \tilde{U}'_i allowing to enforce the inequality (38) associated with $U'_i = U''_i \setminus \tilde{U}'_i$. Indeed, the inequality associated to U'_i is a linear combination of the one associated to U''_i and some trivial inequality.

Improvement 2. If the resulting U'_i does not respect condition 2, then there exists a set $U''_i \subseteq U_i$ with $|U''_i| = |U'_i|$ such that $N_G(U''_i) \subset N_G(U'_i)$. Then, $\sum_{v \in N_G(U'_i) \setminus N_G(U''_i)} w(v) = 0$. Thus, we aim to minimize $|N_G(U'_i)|$. We ensure the minimality of $N_G(U'_i)$ by adding $\sum_{v \in V} \epsilon x_v$ to the objective function, where ϵ is a small value. Figure 4b shows two sets U'_i and U''_i of same size, where the inequality (38) associated with U''_i strengthen the inequality (38) associated with U'_i . Indeed, U''_i shows that the maximum surplus is 0, whereas U'_i has a surplus of value 1.

Improvement 3. If $G[U'_i \cup N_G(U'_i)]$ is not connected, then, for each connected component, if the induced inequality is violated, it is added to the model. In Figure 4c, the inequality associated to U'_i is a linear combination of the inequalities (38) associated with $U''_{i,1}$ and $U''_{i,2}$.

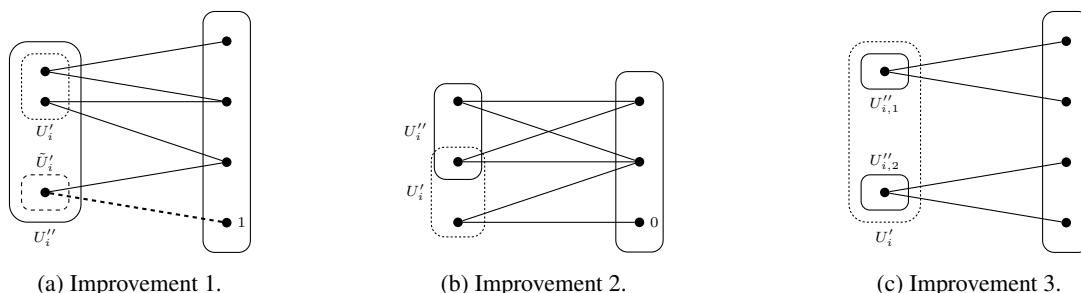


Figure 4: Improvements of the separation algorithm.

Separation of inequalities (44). We propose a heuristic for solving the separation problem associated with the inequality. Let $M' \subseteq M$ and $\mathcal{U}' = \{U'_i, i \in M'\}$ be the set of sets found by the previous algorithm. For all couples $(U'_i, U'_j) \in \mathcal{U}'^2$, if the inequality (44) associated with this couple is violated, then it is added to the model.

5.1.2. Branch-and-Cut algorithm

In this section, we present a Branch-and-Cut algorithm for solving (P') . We aim to address the algorithmic applications of the model and the theoretical results presented in the previous sections. To start the optimization, we consider the linear program with the objective function

$$\max\{z \mid x \in [0, 1]^{m \times |V|}, z \in \mathbb{N}_+, x \text{ satisfies (37)}\}.$$

An important task in the Branch-and-Cut algorithm is to determine whether an optimal solution of the linear relaxation of the MBCMVBP is feasible. An optimal solution x^* of the linear relaxation is feasible for the MBCMVBP if x^* is integer and x^* satisfies inequalities (38). Thus, whether or not x^* is feasible for MBCMVBP can be verified in polynomial time. If not, the Branch-and-Cut algorithm uses the inequalities (38) and (44) and their separations are successively performed. We note that all inequalities are global (*i.e.* valid in all the Branch-and-Cut tree) and several inequalities may be added at each iteration. Testing the two versions of the algorithm shows that the integer **vector-based** separation outperforms the fractional **vector-based** one. Thus, only the integer **vector-based** separation method is used to separate vectors $(x^*, z^*) \in \mathbb{N}_+^{|V| \times m} \times \mathbb{N}_+$.

5.2. Instances

Recall that our study was motivated by real-life applications. We are especially focused on human resources problems, such as people assignment to time shifts (hospitals, factories, etc.) according to their **own** skills.

The following parameters have been used to produce our randomly generated instances for the MBCMVBP:

- m : the number of slots (2 or 5),
- n_U : the size of U_i (10, 25, 50, 75),
- n_V : the size of V ($1.25 \times n_U$, $1.5 \times n_U$, $1.75 \times n_U$),
- d : the density of the graph (10%, 20%, or 40% of n_U).

As the resources are often limited regarding the jobs, we have chosen the values of n_V not too big relatively to the values of n_U . The density of the graph expresses the multi-skilling degree of people and depends on the studied domain. This value is a key feature of the generated instances, being of great importance on execution times. With these parameters, the graph instances could have been **randomly generated**. Nevertheless, to ensure not to generate trivial instances, we are using the following strategy. The instance generator first builds a matching. Then, the obtained graph is randomly completed using two different generators.

First generator. We consider a uniform degree distribution on V . This generator completes the graph by randomly adding edges to ensure the uniformity of the distribution.

Second generator. We consider a global density $(\frac{|E|}{|U| \times |V|})$ and the generator completes the graph by randomly adding edges to ensure the global density.

For a better understanding of the results, we present the experiments on instances with a uniform partition of U : $|U_i| = |U|/m$ for all $i \in M$. We randomly generated 5 instances for each set (m, n_U, n_V, d) of parameters, and thus obtained 700 different instances.

All the tests have shown that these two strategies **do not** have any influence on the obtained results. Thus, for the sake of clarity, we only present results on the instances given by the first generator.

5.3. Results

All the models of BCMVBP and MBCMVBP are implemented in Java using CPLEX 12.6 solver, on an Intel core i5 running at 3.4 GHz, with 8 GB RAM.

5.3.1. Dual formulation

In Section 4.2, we proposed an improvement in order to reduce the size of the ILP: by replacing U with \tilde{U} in the dual formulation, only a restricted number of variables are considered (Proposition 1). We used our benchmark to investigate the efficiency of this method. Considering the high computing times of the dual formulation, we restricted our study to the 110 smallest instances of our benchmark. In Table 1, we show the four instances where the method effectively reduced the size of the ILP. The first three columns identify the instance. The computing time of the ILP without improvement (resp. with improvement) is indicated in seconds in column *CPU1* (resp. column *CPU2*). The number of nodes of the branching tree for both variants is given in columns *#Nodes1* and *#Nodes2*, respectively. The last column presents the number of nodes not being considered in the dual formulation.

$ U $	$ V $	density	<i>CPU1</i>	<i>CPU2</i>	<i>#Nodes1</i>	<i>#Nodes2</i>	$ U \setminus \tilde{U} $
50	62	10	2	2	0	0	1
50	62	10	0	1	0	0	5
50	62	10	0	0	0	0	1
50	75	10	1	2	0	0	1

Table 1: The 4/110 instances for which the improvements reduce the size of the ILP.

We can note on lines 2 and 4 that, even with a reduced sized model, the computing time is greater. This is due to the time spent to compute \tilde{U} . Moreover, as less than 4% of the instances benefit from the method, the obvious conclusion is that this improvement is useless. We aimed to generate random instances and hence did not take into account the fact that the neighborhood of a vertex of U can be included in the neighborhood of another one ¹.

5.3.2. Comparison between formulations

The goal of this section is to evaluate the dual formulation proposed in Section 4.2 and the natural formulation presented in Section 4.3. Table 2 presents some results on these formulations without improvement. Each line represents five randomly generated instances, sharing the same parameters. Our benchmark is restricted to the smallest instances, allowing to solve at least one instance over five, in less than one hour.

The first three columns identify the instance ($|U|$, $|V|$, graph density). In column *CPU1* (resp. column *CPU2*) is indicated the average computing times over the five instances, in seconds, of the dual formulation (resp. the natural formulation). In columns *#Nodes1* and *#Nodes2* are given the average number of nodes of the branching tree for both variants, respectively. The last two columns (*opt1* and *opt2*) present the ratio of instances solved in less than one hour.

¹In a real-life application, it could correspond to the fact that some people are more experienced than others in a given area, thus having a wider range of skills.

$ U $	$ V $	density	CPU1	CPU2	#Nodes1	#Nodes2	opt1	opt2
50	62	10	14	0	494.6	0	5/5	5/5
50	62	20	429.8	1.4	254.4	9.8	5/5	5/5
50	62	40	1428	0	566.4	0	5/5	5/5
50	75	10	2.8	0.6	0	0	5/5	5/5
50	75	20	123.8	2.6	553.2	4.6	5/5	5/5
50	75	40	1926.6	1.6	708.8	2.6	4/5	5/5
50	87	10	3.6	0.6	18.6	0	5/5	5/5
50	87	20	379	3	330.4	11.8	5/5	5/5
50	87	40	2298.4	4	462.8	1562.6	2/5	5/5
100	125	10	1241.8	4	44.2	6	4/5	5/5
100	125	20	3234.2	19.6	69.4	11480.6	2/5	5/5

Table 2: Comparison of the two formulations.

With a computing time limited to one hour for each instance, the dual formulation only solves 47 instances over 55, whereas the natural formulation allows solving all instances. Furthermore, the computing times of the natural formulation is much faster. Indeed, the longest computing time is 50 seconds. The standard deviation of the number of nodes in the branching tree for the dual formulation is smaller than in the natural formulation. As the natural formulation leads to models considering fewer variables, the computing times and memory amount needed to solve are better than the ones for the dual formulation. We can notice that instances (50,87,40) and (100,125,20) are the most difficult to solve using the dual formulation (see Table 2). Indeed, the number of nodes generated by the dual formulation is smaller than the number of nodes generated by the natural one. The optimality has not been reached in the allotted time for three instances out of five. Thus only a few numbers of nodes have been explored in the branching tree. Furthermore, the number of nodes in the branching tree is larger for these instances with the natural formulation, due to the separation algorithm. As mentioned in subsection 5.1.2, only integer vectors are separated, implying that lots of nodes might be explored in the branching tree in order to find the best solution. Remark that these instances are also the ones having many edges. Nevertheless, the execution times remain small for these instances.

5.3.3. Natural formulation and improvements tests

We now focus on the natural formulation and the different improvements proposed in Section 5.1.1.

First, we present an analysis of computing times for some instances. We have excluded instances having less than 50 vertices, for which the computing times are too tiny to expect any improvement. The execution time results are represented by grouped bar charts in Figures 6, 7 and 8 for various densities and graph sizes.

Each instance has been solved using inequalities (38), and adding one or a combination of the following variants:

- *imprv0*: with no improvement,
- *imprv1*: with improvement 1,
- *imprv2*: with improvement 2,
- *imprv3*: with improvement 3,
- *imprv4*: using inequalities (44).

The results are presented in Figure 6 (for 2 slots), in Figure 7 (for 5 slots) and in Figure 8 (for large instances and 5 slots). On the left side, we show the results obtained by testing each improvement separately; on the right side, we combined the different improvements.

Let us start with some general observations without focusing on the improvements. For the overall benchmark, we can notice that for a given couple (density, degree), the computing times are growing as a function of the size of the instances, as expected. We can also observe in Figure 5 that the difficulty is related to the triplet (number of slots, density, $|V|$) for instances with the same $|U|$ value. We present the computing times for 2 slots and $|U| = 100$. As it is the case for many algorithms designed for optimization problems, the most difficult instances are not the very small

or very large ones. Indeed, we notice that the time is first increasing until reaching a specific density and then it is decreasing. We can also remark that the density of the hardest instances increases with their size.

We can explain these observations as follows. For low and high densities, the surplus values of the subsets of U are homogeneous (small values for low densities and large values for high densities). Thus, only a limited number of inequalities is generated, hence reducing the computing times. The hardest instances are those where the variations of the surplus (on the power set of U) is the most important.

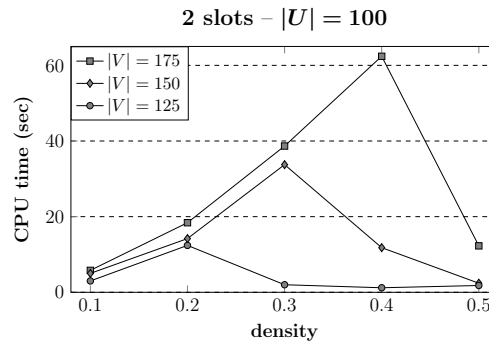


Figure 5: Execution time results for 2 slots and different densities.

Considering the results for single improvements, we can notice that *imprv3* has no real influence on the computing times for our benchmark. Indeed, most of the subgraphs produced by the separation procedure are connected. Otherwise, to be considered as a valid solution for the separation procedure, all components of these subgraphs must have the same surplus. The way the instances are generated rarely enables the application of *imprv3*.

Observations on 2 slots. Let us observe the efficiency of each improvement independently (see the left side of Figure 6). We notice that with a small density (0.1), the optimal solutions are quickly found by the linear program (P'), and the improvements have no significant impact. *Imprv4* can lead to a reduction of computing times with degree 15, but the difference remains very small. Medium density (0.2) instances lead to the same observations when the number of edges is small. *Imprv4* has a stronger impact on larger instances (degree 30). For high density (0.4), *imprv1* and *imprv2* allow to enforce inequalities (38), thus giving better results for larger instances when the degree is equal to 60. The efficiency of *imprv4* is limited to large instances, needing high computing times. Indeed, for these instances, a lot of inequalities (38) are generated. As our separation heuristic is based on the set of vertices associated with inequalities (38), inequalities (44) are more likely to be generated by our heuristic.

Considering combined improvements (see the right side of Figure 6), *imprv3* not being relevant, we studied the combination of the three other ones. As seen earlier, it is better not to use any improvement for small densities. For medium densities, none of the combinations achieves to give better computing times than *imprv4*. For example, combining *imprv1* and *imprv2* decreases the performance for small and medium densities because of the presence of ϵ in the objective function of the separation algorithm. Indeed, ϵ is subtracted when applying *imprv1* and added when applying *imprv2*. Then, if k is small, one can cancel the benefit of the other.

Combinations show some efficiency for high density graphs, but none of them are suitable for all instances.

Observations on 5 slots. On 5 slots (see Figure 7), the efficiency of the improvements is limited to instances of larger size. Moreover, one can remark that *imprv1*, *imprv2* and *imprv4* are more efficient for five slots than for 2 slots, when dealing with high-density instances. Indeed, the number of inequalities is greater for 5 slots (5 choose 2). For small and medium densities, the different improvements never achieve to give better computing times, for the same reasons as for 2 slots. The combinations of the improvements do not go any further except for high-density instances. The results concerning larger instances are presented in Figure 8, Tables 3 and 4. Note that large instances of density 0.4 are more difficult to solve than those of density 0.5. When considering density 0.5, we note that *imprv4* is efficient for a few instances for which the time lost by generating the inequalities without a good impact is weak. Unfortunately, for the instances with lots of nodes, checking the improvement needs a lot of time, and their impact is not enough to

compensate the time used to generate the inequalities. For density 0.4, the instances are hard, as mentioned before, and none of the improvements is efficient.

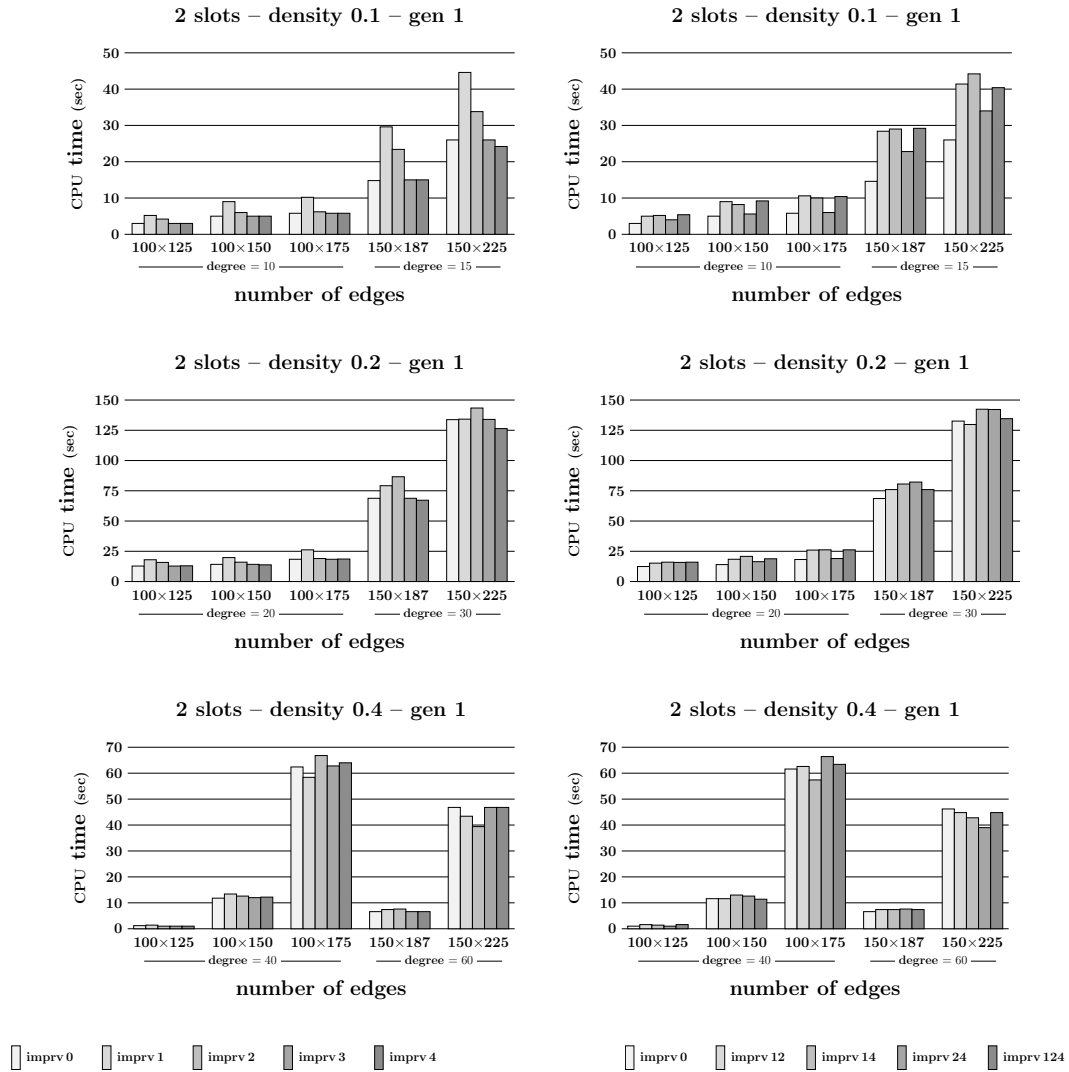


Figure 6: Execution time results for 2 slots.

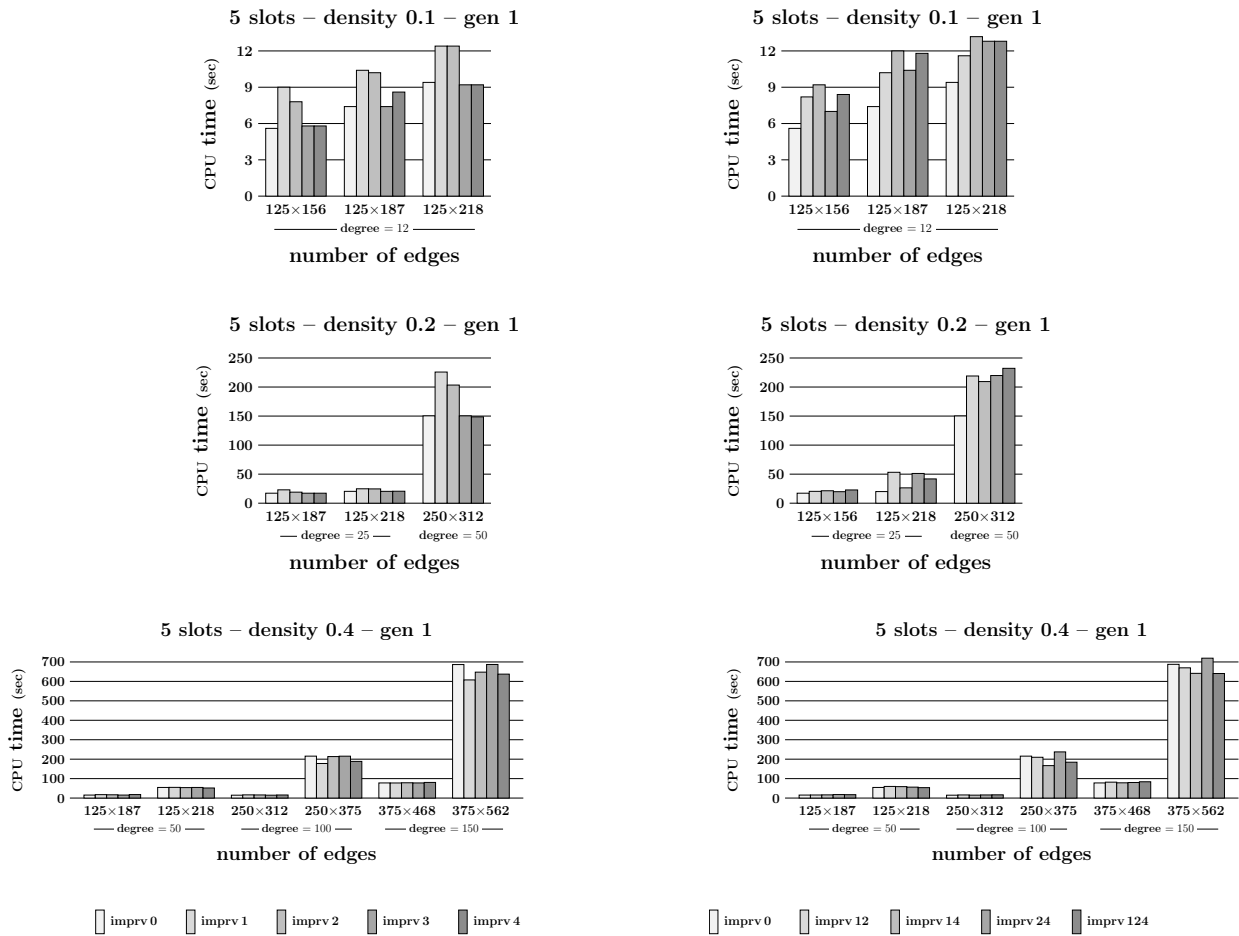


Figure 7: Execution time results for 5 slots.

In this end of section, we focus our analysis on the biggest instances of our tests; details are presented in Tables 3 and 4. The entries in these tables are:

- δ_v : the degree of all $v \in V$ (given by the first generator)
- $|U|$: the size of U ,
- $|V|$: the size of V ,
- imprv : the associated improvement,
- CPU : the total CPU time in seconds,
- $Ct_{(38)}$: the number of generated inequalities (38),
- $Ct_{(44)}$: the number of generated inequalities (44),
- No : the number of generated nodes in the Branch-and-Cut or Branch-and-Bound tree.

Table 3 shows the results for instances generated using density 0.4. After focusing on computing times in previous analysis, we now study the number of generated inequalities and the size of search space. We first note that the number of nodes in the branching tree (given in the last column) is not correlated with the improvements proposed in the previous section. Indeed, the number of nodes is similar in most cases, but we also have some instances showing **significant** variations. Furthermore, we can notice that the number of nodes is relatively small **regarding** the size of instances. This shows the efficiency of our algorithm in terms of memory usage. Secondly, our algorithm finds an exact solution using only a few inequalities (less than 250), which implies good CPU and memory performances. The results also show that our improvements do not systematically reduce the number of constraints, whereas improvements 1

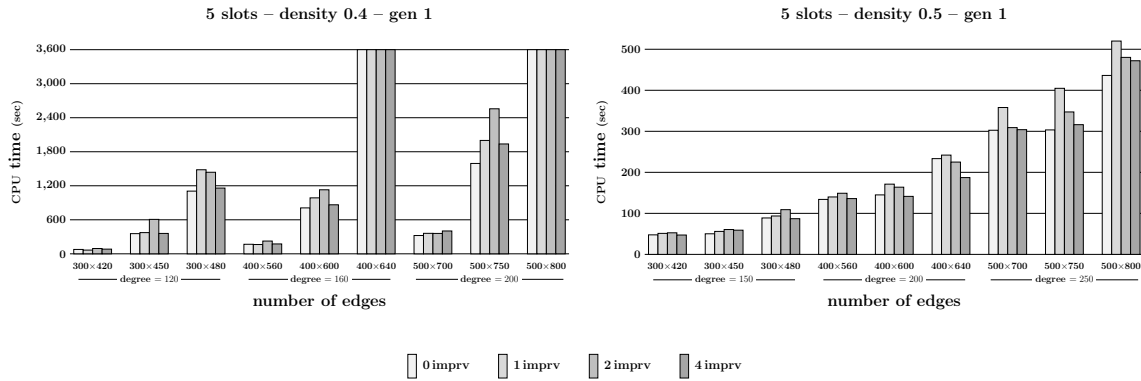


Figure 8: Execution time results for large instances on 5 slots.

and 2 theoretically should strengthen the inequalities $Ct_{(38)}$.

Table 4 shows the results for instances generated using density 0.5. We first notice that all instances are solved in less than 10 minutes. Secondly, comparing to the results of Table 3, the branching tree size is even smaller (less than 2). Furthermore, the size of the instances does not influence the number of nodes. Finally, the number of inequalities generated by the Branch-and-Cut algorithm ($Ct_{(38)}$, $Ct_{(44)}$) is always less than 25. The fact that only a small amount of them is needed to solve the instances proves the strength of our inequalities. Thus, we can conclude that our algorithm is well fitted for [high-density](#) graphs.

δ_v	$ U $	$ V $	imprv	CPU	$C_{t(38)}$	$C_{t(44)}$	No
120	300	420	0	77.2	14.2		1.6
			1	65.6	13.0		1.6
			2	93.6	14.6		2.4
			4	82.6	14.4	10.0	2.4
		450	0	355.6	62.0		40.8
			1	374.0	61.2		39.0
			2	607.4	65.8		41.6
			4	359.6	60.8	10.0	35.0
		480	0	1105.6	180.6		232.0
			1	1482.4	228.6		1173.8
			2	1438.2	203.6		427.6
			4	1159.4	193.6	10.0	233.6
160	400	560	0	168.2	11.0		1.2
			1	164.0	11.6		1.4
			2	225.0	11.8		1.2
			4	174.2	12.4	10.0	1.4
		600	0	809.8	46.4		24.2
			1	986.8	48.4		32.6
			2	1129.2	46.6		23.6
			4	864.6	46.6	10.0	23.8
		640	0	>1h	223.8		430.6
			1	>1h	225.8		459.4
			2	>1h	214.6		340.6
			4	>1h	232.6	10.0	324.6
200	500	700	0	322.6	10.6		1.2
			1	361.2	10.8		1.2
			2	359.6	10.8		1.2
			4	402.8	12.0	10.0	1.6
		750	0	1593.6	43.0		19.2
			1	1999.6	42.0		17.6
			2	2557.6	43.6		18.2
			4	1937.4	46.8	10.0	22.0
		800	0	>1h	96		69.8
			1	>1h	99		65.6
			2	>1h	93		68.8
			4	>1h	98	10.0	59.6

Table 3: 5 slots – density 0.4 – generator 1.

δ_v	$ U $	$ V $	imprv	CPU	$C_{t(38)}$	$C_{t(44)}$	No
150	300	420	0	47.6	6.8		1.4
			1	51.2	6.8		1.4
			2	52.6	6.8		1.4
			4	47.2	6.4	10.0	1.0
		450	0	50.2	9.6		1.0
			1	56.0	9.6		1.0
			2	60.6	9.6		1.0
			4	59.0	9.6	10.0	1.2
		480	0	88.8	11.0		1.2
			1	93.6	11.0		1.2
			2	109.0	11.2		1.4
			4	87.0	10.8	10.0	1.6
200	400	560	0	134.0	6.4		1.2
			1	140.0	6.4		1.2
			2	149.0	6.4		1.2
			4	136.0	7.2	10.0	1.0
		600	0	145.0	9.8		1.4
			1	171.2	9.8		1.4
			2	163.8	9.8		1.4
			4	141.4	9.2	10.0	1.0
		640	0	233.4	10.4		1.8
			1	242.0	10.2		1.2
			2	225.0	10.2		1.2
			4	187.2	10.2	10.0	1.0
250	500	700	0	302.6	6.4		1.2
			1	358.0	6.4		1.2
			2	309.0	6.4		1.2
			4	304.0	6.2	10.0	1.0
		750	0	303.4	8.2		1.2
			1	404.8	8.2		1.0
			2	347.2	8.2		1.2
			4	316.2	7.8	10.0	1.0
		800	0	436.2	10.2		1.2
			1	520.0	10.2		1.2
			2	480.2	10.2		1.4
			4	472.0	10.4	10.0	1.0

Table 4: 5 slots – density 0.5 – generator 1.

6. Conclusion

We have studied the Multiple Bipartite Complete Matching Vertex Blocker Problem (MBCMVBVP). Within the field of staff management, it can be used to identify the critical number of people below which the system becomes understaffed. The NP-completeness of the problem has been shown using a reduction from the stable set problem. Two mathematical formulations have been proposed. The first one is based on the dual formulation of a particular case and the second one (natural formulation) is based on the surplus form of Hall’s theorem. These models have been compared using randomly generated instances.

A further study has been made for the natural formulation, which has been shown as outperforming the dual one. We proposed a Branch-and-Cut algorithm to solve the natural formulation of the problem, providing good results. Based on the polyhedral approach, we derived some improvements and a separation procedure for a new family of inequalities. We tested the performance of these improvements and explained in which case they are efficient.

In future work, we aim to generalize the valid inequalities to more than two subsets and to propose a dedicated separation algorithm. Another lead is trying to take advantage of some specific graph structures to propose some ad-hoc strategies.

A variant of the problem taking into account priorities on some subsets of vertices or considering incompatibility constraints might be studied as well.

References

- [1] D. S. Altner, Ö. Ergun and N. A. Uhan, *The maximum flow network interdiction problem: valid inequalities, integrality gaps, and approximability*, *Operations Research Letters*, 38(1), 33-38, 2010.
- [2] C. Bazgan, S. Toubaline, D. Vanderpooten, *Critical edges/nodes for the minimum spanning tree problem: complexity and approximation*, *Journal of Combinatorial Optimization*, 26(1), 178-189, 2013.
- [3] J. Edmonds, *Maximum matching and a polyhedron with 0,1-vertices*, *Journal of Research of the National Bureau of Standards*, 69B, 125-130, 1965.
- [4] Michael R. Garey and David S. Johnson, *Computers and Intractability: A Guide to the Theory of Np-Completeness*, W. H. Freeman & Co., New York, NY, USA, 1979.
- [5] M. Grötschel, L. Lovász and A. Schrijver, *The ellipsoid method and its consequences in combinatorial optimization*, *Combinatorica*, 169-197, 1981.
- [6] P. Hall, *A Contribution to the Theory of Groups of Prime-Power Order*, *Proceedings of the London Mathematical Society*, 29-07, 1934.
- [7] L. Khachiyan, E. Boros, K. Borys, K. Elbassioni, V. Gurvich, G. Rudolf and J. Zhao, *On Short Paths Interdiction Problems: Total and Node-Wise Limited Interdiction*, *Theory of Computing Systems*, 43(2), 204-233, 2008.
- [8] P. Laroche, F. Marchetti, S. Martin and Z. Roka, *Bipartite Complete Matching Vertex Interdiction Problem: Application to Robust Nurse Assignment*. *IEEE International Conference on Control, Decision and Information Technologies (CoDIT'14)*, Metz, 3-5 Novembre, 2014, France. *IEEE Conference Publications*. DOI: 10.1109/CoDIT.2014.6996890, pages 182 - 187 (2014)
- [9] Y. Li and Z. Nie, *A Note on n-Critical Bipartite Graphs and Its Applications*, *COCOA 2009: Combinatorial Optimization and Applications*, LNCS 5573, 279-286, 2009.
- [10] L. Lovász and M. D. Plummer, *Matching theory*, North-Holland, 1986.
- [11] A. Schrijver, *Combinatorial Optimization, Polyhedra and Efficiency*, Springer Algorithms and Combinatorics, vol. 24, 2000.
- [12] R. Zenklusen, B. Ries, C. Picouleau, D. de Werra, M.-C. Costa and C. Bentz, *Blockers and transversals*, *Discrete Mathematics*, 309, pp. 4306-4314, 2009.
- [13] R. Zenklusen, *Matching interdiction*, *Discrete Applied Mathematics*, 158, 1676-1690, 2010.
- [14] R. Zenklusen, *A Short Note on the NP-Completeness of Node-Wise Limited Matching Interdiction*, *Technical report*, 2012.
- [15] B. Ries, C. Bentz, C. Picouleau, D. de Werra, M.-C. Costa and R. Zenklusen *Blockers and transversals in some subclasses of bipartite graphs: When caterpillars are dancing on a grid*, *Discrete Mathematics*, 310 (1), 132-146, 2010
- [16] C. Bazgan, C. Bentz, C. Picouleau and B. Ries, *Blockers for the Stability Number and the Chromatic Number*, *Graphs and Combinatorics*, 31 (1), 73-90, 2015
- [17] Ö.Y. Diner, D. Paulusma, C. Picouleau and B. Ries, *Contraction Blockers for Graphs with Forbidden Induced Paths*, Paschos V., Widmayer P. (eds) *Algorithms and Complexity. CIAC 2015. Lecture Notes in Computer Science*, 9079, 194-207, 2015
- [18] D. Paulusma, C. Picouleau and B. Ries, *Reducing the Clique and Chromatic Number via Edge Contractions and Vertex Deletions*, In: Cerulli R., Fujishige S., Mahjoub A. (eds) *Combinatorial Optimization. ISCO 2016. Lecture Notes in Computer Science*, 9849, 38-49, 2016
- [19] D. Paulusma, C. Picouleau and B. Ries, *Blocking Independent Sets for H-Free Graphs via Edge Contractions and Vertex Deletions*, In: Gopal T., Jger G., Steila S. (eds) *Theory and Applications of Models of Computation. TAMC 2017. Lecture Notes in Computer Science*, 10185, 470-483, 2017