HAL open science

Multidimensional Binary Vector Assignment Problem: Standard, Structural and Above Guarantee Parameterizations

Marin Bougeret, Guillerme Duvillié, Rodolphe Giroudeau, Rémi Watrigant

▶ **To cite this version:**

Marin Bougeret, Guillerme Duvillié, Rodolphe Giroudeau, Rémi Watrigant. Multidimensional Binary Vector Assignment Problem: Standard, Structural and Above Guarantee Parameterizations. FCT: Fundamentals of Computation Theory, Aug 2015, Gdańsk, Poland. pp.189-201, 10.1007/978-3-319-22177-9_15 . lirmm-01250997

HAL Id: lirmm-01250997

https://hal-lirmm.ccsd.cnrs.fr/lirmm-01250997v1

Submitted on 4 Sep 2019

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 4.0 International License

Discrete Mathematics and Theoretical Computer Science DMTCS vol. **19**:4, 2017, #3

arXiv:1506.03282v4 [cs.DS] 16 Dec 2017

Multidimensional Binary Vector Assignment problem: standard, structural and above guarantee parameterizations

Marin Bougeret[1] Guillerme Duvillié[1] Rodolphe Giroudeau[1] Rémi Watrigant[2]

[1] *LIRMM, Université Montpellier 2, France*
[2] *Hong Kong Polytechnic University. Computing department*

received 12th Nov. 2015, revised 5th Dec. 2016, accepted 3rd Oct. 2017.

In this article we focus on the parameterized complexity of the Multidimensional Binary Vector Assignment problem (called BMVA). An input of this problem is defined by m disjoint sets $V^1, V^2, \ldots, V^m$, each composed of n binary vectors of size p. An output is a set of n disjoint m-tuples of vectors, where each m-tuple is obtained by picking one vector from each set V^i. To each m-tuple we associate a p dimensional vector by applying the bit-wise AND operation on the m vectors of the tuple. The objective is to minimize the total number of zeros in these n vectors. BMVA can be seen as a variant of multidimensional matching where hyperedges are implicitly locally encoded via labels attached to vertices, but was originally introduced in the context of integrated circuit manufacturing.

We provide for this problem FPT algorithms and negative results (ETH-based results, **W**2-hardness and a kernel lower bound) according to several parameters: the standard parameter k (*i.e.* the total number of zeros), as well as two parameters above some guaranteed values.

Keywords: parameterized complexity, kernel, above guarantee parameterization, AND-cross composition, multidimensional binary vector assignment, wafer-to-wafer integration, locally encoded multidimensional matching

1 Introduction

1.1 Definition of the problem

In this paper, we consider the parameterized version of the MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT problem (BMVA). An input of this problem is described by m sets $V^1, V^2, \ldots, V^m$, each of these sets containing n p-dimensional binary vectors. We note $V^i = \{v_1^i, \ldots, v_n^i\}$ for all $i \in [m]$[(i)], and for all $j \in [n]$ and $r \in [p]$, we denote by $v_j^i[r] \in \{0, 1\}$ the rth component of v_j^i.

In order to define the output of the problem, we need to introduce the notion of stack. A stack $s = (v_1^s, v_2^s, \ldots, v_m^s)$ is an m-tuple of vectors such that $\forall i \in [m], v_i^s \in V^i$. The output of BMVA is a set S of

[(i)] $[m]$ stands for $\{1, \ldots, m\}$.

ISSN 1365–8050 © 2017 by the author(s) Distributed under a Creative Commons Attribution 4.0 International License

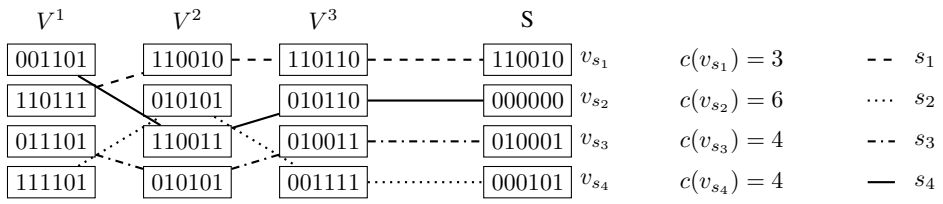

Figure 1: Example of BMVA instance with $m = 3, n = 4, p = 6$ and of a feasible solution S of cost $c(S) = 17$.

n stacks such that for all $i, j \in [m] \times [n]$, v_j^i belongs to only one stack (in that case, the stacks are said *disjoint*). An example of an instance together with a solution is depicted in Figure 1.

We are now ready to define the objective function. We define the operator $\wedge$ that, given two p-dimensional vectors u and v, computes the vector $w = (u[1] \wedge v[1], u[2] \wedge v[2], \ldots, u[p] \wedge v[p])$. We associate to any stack s a unique vector $v_s = \bigwedge_{i \in [m]} v_i^s$.

We define the cost of a binary vector v as the number of zeros in it. More formally, if v is p-dimensional, $c(v) = p - \sum_{r \in [p]} v[r]$. We extend this definition to a set of stacks $S = \{s_1, \ldots, s_n\}$ as follows: $c(S) = \sum_{j \in [n]} c(v_{s_j})$. Finally, the objective of BMVA is to obtain a set S of n disjoint stacks while minimizing $c(S)$. In the decision version of the problem, we are given an integer k, and we ask whether there exists a solution S of cost at most k. The problem is thus defined formally as follows:

Problem 1 MULTIDIMENSIONAL BINARY VECTOR ASSIGNMENT (BMVA)

Input:	m sets of n binary p-dimensional vectors, an integer k
Question:	Is there a set S of n disjoint stacks such that $c(S) \leq k$?

In order to avoid heavy notations throughout the paper, we will denote an instance of BMVA only by $\mathcal{I}[m, n, p, k]$, the notations of the sets and vectors being implicitly given as previously.

1.2 Application and related work

BMVA can be seen as a variant of multidimensional matching where hyperedges are implicitly locally encoded via labels attached to vertices. However, this kind of problem was originally introduced by Reda et al. (2009) in the context of semiconductor industry as the "yield maximization problem in wafer-to-wafer 3-D integration technology". In this context, each vector v_j^i represents a wafer, which is seen as a string of bad dies (0) and good dies (1). Integrating two wafers corresponds to superimposing the two corresponding strings. In this operation, a position in the merged string is "good" when the two corresponding dies are good, and is "bad" otherwise. The objective of Wafer-to-Wafer Integration is to form n stacks, while maximizing their overall quality, or equivalently, minimizing the number of errors (depending on the objective function). In the following, we will denote by $max \sum$ 1-BMVA the dual version of BMVA where given the same input and ouput, the objective is to maximize $np - c(S)$, the total number of ones.

The results obtained so far concerning these problems mainly concern their approximability. It is proved in Dokka et al. (2012) that when $m = 3$, BMVA is $\mathbf{NP}$-hard but admits a $\frac{4}{3}$-approximation. We can also mention Dokka et al. (2014) which provides a $f(m)$-approximation for general m, and an $\mathbf{APX}$-hardness

for $m = 3$. The main related article is Bougeret et al. (2016) where it is proved that $max \sum$ 1-BMVA has no $O(p^{1-\epsilon})$ nor $O(m^{1-\epsilon})$-approximation for any $\epsilon > 0$ unless $\mathbf{P} = \mathbf{NP}$ (even when $n = 2$), but admits a $\frac{p}{c}$-approximation algorithm for any constant $c \in \mathbb{N}$, and is $\mathbf{FPT}$ when parameterized by p (which also holds for BMVA). Notice that one of the reductions provided by Bougeret et al. (2016) is a parameter-preserving reduction from the CLIQUE problem to $max \sum$ 1-BMVA, immediately proving $\mathbf{W}1$-hardness for $max \sum$ 1-BMVA when parameterized by the objective function. This is why our motivation in this paper is to consider the parameterized complexity of BMVA. As we will see in the next section, we provide an analysis for several parameters related to this problem.

For formal definitions and detailed concepts on Fixed-Parameter Tractability, we refer to the monograph of Downey and Fellows (2013). Moreover, in order to define lower bounds on the running time of parameterized algorithms, we will rely on the *Exponential Time Hypothesis* (ETH) of Impagliazzo et al. (2001), stating that 3-SAT cannot be solved in $O^*(2^{o(n)})$ where n is the number of variables ($\mathcal{O}^*(.)$ hides polynomial terms). For more results about lower bounds obtained under ETH, we refer the reader to the survey of Lokshtanov et al. (2013).

1.3 Parameterizations

One of the main purposes of Fixed-Parameter Tractability is to obtain efficient algorithms when the considered parameter is small in practice. When dealing with the decision version of an optimization problem, the most natural parameter is perhaps the value of the desired solution (*e.g.* k for BMVA). Such a parameter is often referred to in the literature as the "standard parameter" of the problem. In some cases, this parameter might not be very interesting, either because it usually takes high values in practice, or because $\mathbf{FPT}$ algorithms with respect to this parameterization are trivial to find. When this happens, it is possible to obtain more interesting results by subtracting to the objective function a known lower bound of it. For instance, if one can prove that any solution of a given minimization problem is of cost at least $\mathcal{B}$, then one can ask for a solution of cost $\mathcal{B} + c$ and parameterize by c. This idea, called "above guarantee parameterization" was introduced by Mahajan et al. (2009) and first applied to MAX SAT and MAX CUT problems. It then became a fruitful line of research with similar results obtained for many other problems (among others, see Cygan et al. (2013); Gutin et al. (2007); Gutin and Yeo (2012); Mahajan et al. (2009)).

In this paper, we analyze the parameterized complexity of BMVA using three types of parameters. The first one is the standard parameter k: the number of zeros to minimize in the optimization version of the problem. Then, three natural structural parameters: m, the number of sets of the input, n, the number of vectors in each set, and p, the size of each vector. The last two parameters are above guarantee parameters.

As we said previously, we already proved in Bougeret et al. (2016) that BMVA is $\mathbf{FPT}$ parameterized by p. As we will notice in Lemma 2 that we can obtain $p \leq k$ after a polynomial pre-processing step, this implies that BMVA is also $\mathbf{FPT}$ with its standard parameter. Our idea here is to use this previous inequality in order to obtain smaller parameters. Thus, we define our first above guarantee parameter $\zeta_p = k - p$.

Finally, in order to define our last parameter, we first need to describe the corresponding lower bound $\mathcal{B}$, that will represent the maximum, over all sets of vectors, of the total number of zeros for each set. More formally, we define $\mathcal{B} = \max_{i \in [m]} c(V^i)$ where $c(V^i) = \sum_{j=1}^{n} c(v_j^i)$. Since we perform a bit-wise AND over each m-tuple, it is easily seen that any solution will be of cost at least $\mathcal{B}$. Thus, we define our last parameter $\zeta_{\mathcal{B}} = k - \mathcal{B}$.

1.4 Our results

In the next section, we present some pre-processing rules leading to a kernel of size $O(k^2m)$, and prove that even when $m = 3$ we cannot improve it to $p^{O(1)}$ unless $\mathbf{NP} \subseteq \mathbf{coNP/poly}$ (remember that BMVA is known to be $\mathbf{FPT}$ when parameterized by p). Section 3 is mainly focused on results associated with parameter $\zeta_{\mathcal{B}}$: we prove that BMVA can be solved in $O^*(4^{\zeta_{\mathcal{B}}\log(n)})$, while it is $\mathbf{W}2$-hard when parameterized by $\zeta_{\mathcal{B}}$ only, and cannot be solved in $O^*(2^{o(\zeta_{\mathcal{B}})\log(n)})$ nor in $O^*(2^{\zeta_{\mathcal{B}}o(\log(n))})$ assuming ETH. In Section 4, we focus on the parameterization by ζ_p: we show that when $n = 2$, the problem can be solved in single exponential time with this parameter, but is not in $\mathbf{XP}$ for any fixed $n \geq 3$ (unless $\mathbf{P} = \mathbf{NP}$). The reduction we use also shows that for fixed $n \in \mathbb{N}$, the problem cannot be solved in $2^{o(k)}$ (and thus in $2^{o(\zeta_{\mathcal{B}})}$) unless ETH fails, which matches the upper bound obtained in Section 3. A summary of our results is depicted in the following table.

Positive results	**Negative results**
$O(k^2m)$ kernel (Thm. 1)	no $p^{O(1)}$ kernel for $m = 3$ unless $\mathbf{NP} \subseteq \mathbf{coNP/poly}$ (Thm. 3)
$\mathcal{O}^*(4^{\zeta_{\mathcal{B}}\log(n)})$ algorithm (Thm. 4)	$\mathbf{W}2$-hard for $\zeta_{\mathcal{B}}$ only (Thm. 5) no $2^{o(\zeta_{\mathcal{B}})\log(n)}$ nor $2^{\zeta_{\mathcal{B}}o(\log(n))}$ under ETH (Thm. 6) no $2^{o(k)}$ for fixed n under ETH (Thm. 8)
$O^*(d^{\zeta_p})$ algorithm for $n = 2$ (Thm. 7)	$\mathbf{NP}$-hard for $\zeta_p = 0$ and fixed $n \geq 3$ (Thm. 8)

This article is the complete version of Bougeret et al. (2015) where some proofs were omitted due to space limitations.

2 First remarks and kernels

Let us start with two simple lemmas allowing us to bound the size of the input. Notice first that it is not always safe to create a 1-stack (*i.e.* a stack with ones on every component) when possible. Indeed, in instance $V^1 = \{\langle 111\rangle, \langle 101\rangle, \langle 011\rangle\}$, $V^2 = \{\langle 111\rangle, \langle 101\rangle, \langle 110\rangle\}$, $V^3 = \{\langle 111\rangle, \langle 011\rangle, \langle 110\rangle\}$, depicted in Figure 2, no optimal solution creates a 1-stack. However, as we will see in Lemma 1, creating 1-stacks becomes safe if $n > k$.

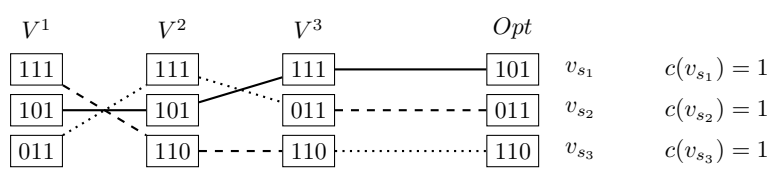

Figure 2: Example of BMVA instance such that no optimal solution creates a 1-stack.

Lemma 1. *There exists a polynomial algorithm which, given any instance $\mathcal{I}[m, n, p, k]$ of* BMVA*, either detects that $\mathcal{I}$ is a negative instance, or outputs an equivalent instance $\mathcal{I}'[m, n', p, k]$ such that $n' \leq k$.*

Proof: Let $\mathcal{I}[m, n, p, k]$ be an instance of BMVA, and suppose that $n > k$. Let us write a polynomial preprocessing rule that either detects that $\mathcal{I}$ is a no instance, or compute an equivalent instance $\mathcal{I}''[m, n'', p, k]$ with $n'' = n - 1$.

Notice first that there exists at least a 1-vector in every set V^i. If not, $\mathcal{I}$ is a no instance as any solution would be of cost at least $n > k$. It is now safe to create a 1-stack, obtaining a remaining instance $\mathcal{I}''$ with $n'' = n - 1$. Indeed, if $\mathcal{I}$ is a yes instance, then there must exist at least one 1-stack in the solution (otherwise the cost would be at least $n > k$), and thus the remaining instance is also a yes instance. As the converse is trivially true, the rule is safe. Applying it at most $n - k$ times finally leads to the desired upper bound. □

Lemma 2. *There exists a polynomial algorithm which, given any instance $\mathcal{I}[m, n, p, k]$ of* BMVA*, either detects that $\mathcal{I}$ is a negative instance, or outputs an equivalent instance $\mathcal{I}'[m, n, p', k]$ such that $p' \leq k$.*

Proof: Let $\mathcal{I}[m, n, p, k]$ be an instance of the problem, and suppose that there exists $r \in [p]$ such that for all $(i, j) \in [m] \times [n]$ we have $v^i_j[r] = 1$. In other words, the rth component of all vectors of all sets is a 1. In this case, it is clear that all vectors of any set of n stacks obtained from $\mathcal{I}$ will also contain a 1 at the rth component. Hence, we can modify $\mathcal{I}[m, n, p, k]$ into $\mathcal{I}'[m, n, p', k]$ with $p' < p$ by dropping all such components for all vectors. It is clear that this rule is safe since the cost of any solution remains unchanged, and it can be applied in polynomial time. After applying this rule, for all $r \in [p']$ there exists $(i, j) \in [m] \times [n]$ such that $v^i_j[r] = 0$. This immediately implies that the cost of any solution is at least p', and thus if $p' > k$ the algorithm detected a no instance. □

Given the two previous lemmas, we can suppose from now on that for any instance of BMVA we have $n \leq k$ and $p \leq k$. This immediately implies a polynomial kernel parameterized by k and m.

Theorem 1. BMVA *admits a kernel with $O(k^2 m)$ bits.*

Let us now turn to the main result of this section. To complement Theorem 1, we show that even when $m = 3$, we cannot obtain a polynomial kernel with the smaller parameter p under some classical complexity assumptions (notice however that the existence of a polynomial kernel in k only is still open when m is not fixed). Notice also that as BMVA was known to be $\mathbf{FPT}$ when parameterized by p, proved in Bougeret et al. (2016), it was a natural question to ask for a polynomial kernel.

In order to establish kernel lower bounds, we use the concept of AND-cross-composition of Bodlaender et al. (2014), together with the recently proved AND-conjecture of Drucker (2015). In the following, a parameterized problem is a subset of $\Sigma^* \times \mathbb{N}$, where Σ^* is the set of words over some finite alphabet Σ.

Definition 1 (Polynomial equivalence relation according to Bodlaender et al. (2014)). An equivalence relation $\mathcal{R}$ on Σ^* is called a polynomial equivalence relation if the two following conditions hold:

- There is an algorithm that given two strings $x, y \in \Sigma^*$, decides whether x and y belong to the same equivalence class in $(|x| + |y|)^{O(1)}$ time.
- For any finite set $S \subseteq \Sigma^*$, the equivalence relation $\mathcal{R}$ partitions the elements of S into at most $(max_{x \in S}|x|)^{O(1)}$ classes.

Definition 2 (AND-cross-composition according to Bodlaender et al. (2014)). Let $L \subseteq \Sigma^*$ be a set and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L AND-cross-composes into Q if there

is a polynomial equivalence relation $\mathcal{R}$ and an algorithm which, given t strings belonging to the same equivalence class of $\mathcal{R}$, computes an instance $(x^*, k^*) \in \Sigma^* \times \mathbb{N}$ in time polynomial in $\sum_{i=1}^{t} |x_i|$ such that:

- $(x^*, k^*) \in Q \Leftrightarrow x_i \in L$ for all $i \in \{1, \ldots, t\}$
- k^* is bounded by a polynomial in $max_{i=1}^{t}|x_i| + \log t$

Theorem 2 (Drucker (2015))**.** *If some set $L \subseteq \Sigma^*$ is* $\mathbf{NP}$*-hard and L AND-cross-composes into a parameterized problem Q, then there is no polynomial kernel for Q unless* $\mathbf{NP} \subseteq \mathbf{coNP/poly}$.

Theorem 3. *Even for $m = 3$,* BMVA *parameterized by p does not admit a polynomial kernel unless* $\mathbf{NP} \subseteq \mathbf{coNP/poly}$.

Proof: The proof is an AND-cross-composition inspired by the $\mathbf{NP}$-hardness reduction for BMVA provided by Dokka et al. (2012). More precisely, we cross-compose from a sequence of instances of 3-DIMENSIONAL PERFECT MATCHING. According to Definition 2 and Theorem 2, this will imply the desired result. 3-DIMENSIONAL PERFECT MATCHING is formally defined as follows:

Problem 2 3-Dimensional Perfect Matching

Input: Three sets X, Y and Z of size n, a set of hyperedges $S \subseteq X \times Y \times Z$

Question: Does there exist a subset $S' \subseteq S$ such that:

- for all $e, e' \in S'$ with $e = (x, y, z)$ and $e' = (x', y', z')$, we have $x \neq x'$, $y \neq y'$ and $z \neq z'$ (that is, S' is a matching)
- $|S'| = n$ (that is, S' is perfect)

Let $(X_1, Y_1, Z_1, S_1), \ldots, (X_t, Y_t, Z_t, S_t)$ be a sequence of t equivalent instances of 3-DPM, with respect to the following polynomial equivalence relation: (X, Y, Z, S) and (X', Y', Z', S') are equivalent if $|X| = |X'|$ (and thus $|Y| = |Y'| = |Z| = |Z'| = |X|$), and $|S| = |S'|$. In the following we denote by n the cardinality of the sets X_i (and equivalently the sets Y_i and Z_i), and by m the cardinality of the sets S_i. Moreover, for all $i \in \{1, \ldots, t\}$ we define $X_i = \{x_{i,1}, \ldots, x_{i,n}\}, Y_i = \{y_{i,1}, \ldots, y_{i,n}\}, Z_i = \{z_{i,1}, \ldots, z_{i,n}\}$, and $S_i = \{s_{i,1}, \ldots, s_{i,m}\}$. We also assume that $t = 2^q$ for some $q \in \mathbb{N}$ (if it is not the case, we add a sufficiently number of dummy yes-instances).

In the following we construct three sets (X^*, Y^*, Z^*) of nt vectors each: $X^* = \{x^*_{i,j}\}_{i=1,\ldots,t}^{j=1,\ldots,n}$, $Y^* = \{y^*_{i,j}\}_{i=1,\ldots,t}^{j=1,\ldots,n}$ and $Z^* = \{z^*_{i,j}\}_{i=1,\ldots,t}^{j=1,\ldots,n}$, where each vector is composed of $p^* = m + 2mq$ components. Let us first describe the first m components of each vector. For all $i \in \{1, \ldots, t\}$, $j \in \{1, \ldots, n\}$ and $k \in \{1, \ldots, m\}$ we set:

$$x^*_{i,j}[k] = \begin{cases} 1 & \text{if the hyperedge } s_{i,k} \text{ contains } x_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

$$y^*_{i,j}[k] = \begin{cases} 1 & \text{if the hyperedge } s_{i,k} \text{ contains } y_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

$$z^*_{i,j}[k] = \begin{cases} 1 & \text{if the hyperedge } s_{i,k} \text{ contains } z_{i,j} \\ 0 & \text{otherwise} \end{cases}$$

Then, for all $i \in \{1, \ldots, t\}$, we append two vectors b_i and $\bar{b_i}$ to all vectors $\{x^*_{i,j}\}_{j=1,\ldots,n}$, $\{y^*_{i,j}\}_{j=1,\ldots,n}$ and $\{z^*_{i,j}\}_{j=1,\ldots,n}$. The vector b_i is composed of mq coordinates, and is defined as the binary representation of the integer i, where each bit is duplicated m times. Finally, $\bar{b_i}$ is obtained by taking the complement of b_i (*i.e.* replacing all zeros by ones, and conversely) as depicted in Figure 3. It is now clear that each vector $x^*_{i,j}$ (resp. $y^*_{i,j}$, $z^*_{i,j}$) is composed of $p^* = m + 2mq$ coordinates. Thus, the parameter of the input instance is a polynomial in n, m and $\log t$ whereas the total size of the instance is a polynomial in the size of the sequence of inputs, as required in cross-compositions. It now remains to prove that (X^*, Y^*, Z^*) contains an assignment of cost $k^* = nt(mq + m - 1)$ if and only if for all $i \in \{1, \ldots, t\}$, S_i contains a perfect matching S'_i.

- Suppose that for all $i \in \{1, \ldots, t\}$ we have a perfect matching $S'_i \subseteq S_i$. W.l.o.g. suppose that $S'_i = \{s_{i,1}, \ldots, s_{i,n}\}$. Then, for each $j \in \{1, \ldots, n\}$, we have $s_{i,j} = (x_{i,j_1}, y_{i,j_2}, z_{i,j_3})$ for some $j_1, j_2, j_3 \in \{1, \ldots, n\}$. We assign x^*_{i,j_1} with y^*_{i,j_2} and z^*_{i,j_3}. It is easy to see that the cost of this triple is $m - 1 + mq$. Indeed, they all have a one at the jth coordinate, corresponding to the jth hyperedge of S_i (and this is the only shared one, since we can suppose that all hyperedges are pairwise distinct), and they all contain the same vectors b_i and $\bar{b_i}$. Summing up for all instances, we get the desired solution value.

- Conversely, first remark that in any assignment, the cost of every triple $(x^*_{i_1,j_1}, y^*_{i_2,j_2}, z^*_{i_3,j_3})$ is at least $m - 1 + mq$, and let us prove that this bound is tight when (1) all elements are chosen within the same instance, *i.e.* $i_1 = i_2 = i_3 = i$, and (2) this triple corresponds to an element of S_i, *i.e.* $(x_{i,j_1}, y_{i,j_2}, z_{i,j_3}) \in S_i$. Indeed, suppose first that $i_1 \neq i_2$. Then, since the binary representation of i_1 and i_2 differs on at least one bit, it is clear that the resulting vector is of cost at least $m(q+1) > mq + m - 1$. Now if $i_1 = i_2 = i_3 = i$, then the result is straightforward, since at most one hyperedge of S_i can contain $x^*_{i_1,j_1}$, $y^*_{i_2,j_2}$ and $z^*_{i_3,j_3}$. Finally, using the same arguments as previously, we can easily deduce a perfect matching $S'_i \subseteq S_i$ for each $i \in \{1, \ldots, t\}$, and the result follows.

□

3 Parameterizing according to $\zeta_{\mathcal{B}}$

In this section, we present an **FPT** algorithm when parameterized by $\zeta_{\mathcal{B}}$ and n (recall that both $\zeta_{\mathcal{B}}$ and n are smaller parameters than the standard one k, since $k = \mathcal{B} + \zeta_{\mathcal{B}}$ and $n \leq k$ in any reduced instance). Notice first that it is easy to get an $\mathcal{O}^*(2^{\zeta_{\mathcal{B}}(log(n)+log(p))})$ algorithm. Indeed, by considering a set $i \in [m]$ where $c(V^i) = \mathcal{B}$, and guessing the positions of the $\zeta_{\mathcal{B}}$ new zeros (among np possible positions) that will appear in an optimal solution, we can actually guess in $\mathcal{O}^*((np)^{\zeta_{\mathcal{B}}})$ the vectors $\{v_{s^*_j}\}$ of an optimal solution, and it remains to check in polynomial time that every V^j can be "matched" to $\{v_{s^*_j}\}$. Now we show how to get rid of the $\log(p)$ term in the exponent.

Theorem 4. BMVA *can be solved in* $\mathcal{O}^*(4^{\zeta_{\mathcal{B}} \log(n)})$.

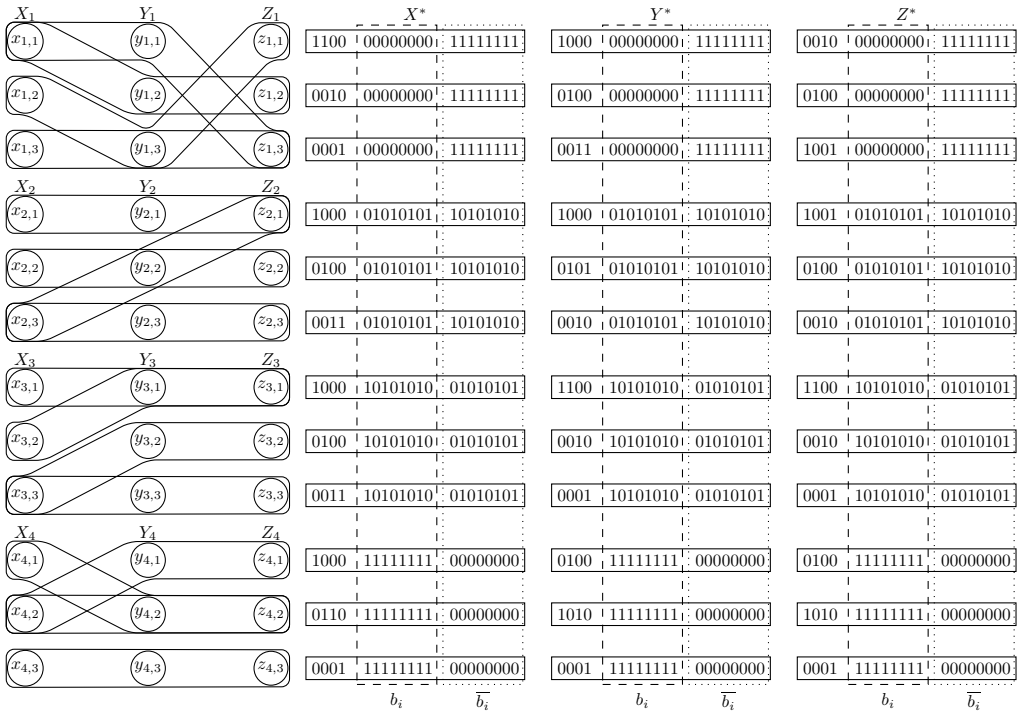

Figure 3: Example of Cross Composition construction from four equivalent instances of 3-Dimensional Perfect Matching $(X_1, Y_1, Z_1, \{(x_{1,1}, y_{1,1}, z_{1,3}), (x_{1,1}, y_{1,2}, z_{1,2}), (x_{1,2}, y_{1,3}, z_{1,3}), (x_{1,3}, y_{1,3}, z_{1,3})\})$, $(X_2, Y_2, Z_2, \{(x_{2,1}, y_{2,1}, z_{2,1}), (x_{2,2}, y_{2,2}, z_{2,2}), (x_{2,3}, y_{2,3}, z_{2,3}), (x_{2,3}, y_{2,2}, z_{2,1})\})$, $(X_3, Y_3, Z_3, \{(x_{3,1}, y_{3,1}, z_{3,1}), (x_{3,2}, y_{3,1}, z_{3,1}), (x_{3,3}, y_{3,2}, z_{3,2}), (x_{3,3}, y_{3,3}, z_{3,3})\})$, $(X_4, Y_4, Z_4, \{(x_{4,1}, y_{4,2}, z_{4,2}), (x_{4,2}, y_{4,1}, z_{4,1}), (x_{4,2}, y_{4,2}, z_{4,2}), (x_{4,3}, y_{4,3}, z_{4,3})\})$ into an instance of bMVA with $m = 3,\ n = 12$ and $p = 20$.

Proof: Let $\mathcal{I}[m, n, p, k]$ be an instance of our problem and, w.l.o.g., suppose that V^1 is a set whose number of zeros reaches the upper bound $\mathcal{B}$, *i.e.* $c(V^1) = \mathcal{B}$. The algorithm consists in constructing a solution by finding an optimal assignment between V^1 and $V^2, \ldots, V^m$, successively.

We first claim that we can decide in polynomial time whether there is an assignment between V^1 and V^2 which does not create any additional zero.

To that end, we create a bipartite graph G with bipartization (A, B), $A = \{a_1, \ldots, a_n\}$, $B = \{b_1, \ldots, b_n\}$, and link a_{j_1} and b_{j_2} for all $(j_1, j_2) \in [n] \times [n]$ iff assigning vector $v^1_{j_1}$ from V^1 and vector $v^2_{j_2}$ from V^2 does not create any additional zero in V^1 ($v^1_{j_1} \wedge v^2_{j_2} = v^1_{j_1}$).

If a perfect maching can be found in G, then we can safely delete the set V^2 and continue. In order to avoid heavy notations, we consider this first step as a polynomial pre-processing, and we re-label V^i into V^{i-1} for all $i \in \{3, \ldots, m\}$ (and m is implicitly decreased by one).

In the following, we suppose that the previous pre-processing step cannot apply (*i.e.* there is no perfect matching in G). Intuitively, in this case any assignment (including an optimal one) between V^1 and V^2 must lead to at least one additional zero in V^1. In this case, we perform a branching to guess one couple of vectors from $V^1 \times V^2$ which will induce such an additional zero. More formally, we branch on every couple $(j_1, j_2) \in [n] \times [n]$, and create a new instance as a copy of $\mathcal{I}$ in which $v^1_{j_1}$ is replaced by $v^1_{j_1} \wedge v^2_{j_2}$. This operation increases $c(V^1)$ by at least one, and thus $\mathcal{B}$ by at least one as well. If we denote by $\mathcal{I}'$ this new instance, we can check that a solution of cost at most k for $\mathcal{I}'$ will immediately imply a solution of cost at most k for $\mathcal{I}$, as $\mathcal{I}'$ is constructed from $\mathcal{I}$ by adding some zeros. The converse is also true as one assignment we enumerate corresponds to one from an optimal solution.

As the value of $\mathcal{B}$ in this branching increases by at least one while we still look for a solution of cost k, this implies that this branching will be applied at most $\zeta_{\mathcal{B}}$ times. Summing up, we have one polynomial pre-processing and one branching of size n^2 which will be applied at most $\zeta_{\mathcal{B}}$ times. The total running time of this algorithm is thus bounded by $\mathcal{O}^*(4^{\zeta_{\mathcal{B}} \log(n)})$.

□

Despite its simplicity, we now show that, when considering each parameter (n and $\zeta_{\mathcal{B}}$) separately, this algorithm is the best we can hope for (whereas the existence of an $\mathcal{O}^*(2^k)$ algorithm is still open). Indeed, we first show in Theorem 6 that the linear dependence in $\zeta_{\mathcal{B}}$ and $log(n)$ in the exponent is necessary (unless ETH fails), and also that we cannot hope for an $\mathbf{FPT}$ algorithm parameterized by $\zeta_{\mathcal{B}}$ only unless $\mathbf{FPT} = \mathbf{W}2$ (Theorem 5). Finally, as we will see in the next section (Theorem 8), this result is matched by a $2^{o(\zeta_{\mathcal{B}})}$ lower bound when $n \in \mathbb{N}$ is fixed. We now present a reduction from the HITTING SET problem which produces an instance of BMVA.

Problem 3 Hitting Set	
Input:	m subsets $R_1, ..., R_m$ of $[n]$, and an integer k
Question:	Is there a set R of k elements of $[n]$ such that $R \cap R_i \neq \emptyset$ for any $i \in [m]$?

Lemma 3. *There is a polynomial reduction from* HITTING SET *to* BMVA *that given an instance composed of m subsets of $[n]$ and an integer k, constructs an instance of* BMVA $\mathcal{I}[m', n', p', k']$ *such that $n' = n$ and $\zeta_{\mathcal{B}} = k$.*

Proof: Let $R_1, ..., R_m$ be subsets of $[n]$, and $k \in \mathbb{N}$. We construct m sets $V^1, ..., V^m$ of n vectors each, where, for all $i \in [m]$ we have $V^i = \{v^i_1, ..., v^i_n\}$, each vector being composed of n components. For all

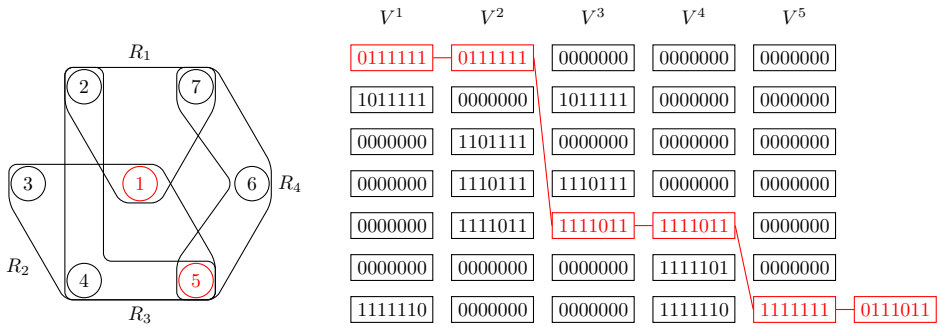

Figure 4: Example of reduction from an instance of Hitting Set consisting of four subsets of $[n = 7] : R_1 = \{1, 2, 7\}, R_2 = \{1, 3, 4, 5\}, R_3 = \{2, 4, 5\}, R_4 = \{5, 6, 7\}$, and an integer $k = 2$, to an instance of BMVA with $m' = 5, n' = n, p = n$.

$i \in [m]$ and all $j \in [n]$, if $j \in R_i$, then the vector v^i_j is composed of ones everywhere except at the jth component. If $j \notin R_i$, then v^i_j is a 0-vector (*i.e.* a vector with zero in every component). We also add a set V^* composed of $(n-1)$ 0-vectors and one 1-vector as depicted in Figure 4.

For this constructed instance, it is clear that $\mathcal{B} = n(n-1)$ because of the set V^*. In other words, any assignment will lead to a solution with $(n-1)$ 0-vectors, and thus with at least $n(n-1)$ zeros. We will actually show that this instance has a solution with $n(n-1)+k$ zeros if and only if $R_1, ..., R_m$ has a hitting set of size k. By the foregoing, we only need to focus on the only vector of each set which is assigned to the 1-vector of V^*.

$\Rightarrow$ Let $J \subseteq [n]$ be a hitting set of size k. By the definition of a hitting set, for all $i \in [m]$, there exists $j_i \in J \cap R_i$. Thus, for all $i \in [m]$, we select the vector $v^i_{j_i}$ from the set V^i to be assigned to the 1-vector of V^*. By construction, this vector has only one zero at the j_i^{th} component, which implies that the conjunction of all such vectors $\bigwedge_{i=1}^{m} v^j_{j_i}$ will have a 1 everywhere except at the components corresponding to J. We thus have the desired number of zeros in our solution.

$\Leftarrow$ Conversely, for each $i \in [m]$, let $j_i \in [n]$ be the vector from V^i which is assigned to the 1-vector of V^*. Since the resulting conjunction of all these vectors has only k zeros, $v^i_{j_i}$ cannot be a 0-vector, and we thus have $j_i \in R_i$. Using the same arguments as previously, $\{u_{j_i}\}_{i \in [m]}$ corresponds to a hitting set of $R_1, ..., R_m$ of size k.

As we seen previously, $\mathcal{B} = n(n-1)$ for the obtained instance, $k' = n(n-1)+k$, (which implies $\zeta_{\mathcal{B}} = k$), and the size of all sets is n, as desired. □

As we can see, the reduction is parameter-preserving for $\zeta_{\mathcal{B}}$. From the **W**2-hardness of HITTING SET provided by Downey and Fellows (2013), we have the following:

Theorem 5. BMVA *is* **W**2*-hard when parameterized by* $\zeta_{\mathcal{B}}$.

As said previously, we also use this reduction to show the following result:

Theorem 6. BMVA *cannot be solved in* $O^*(2^{o(\zeta_{\mathcal{B}})\log(n)})$ *nor* $O^*(2^{\zeta_{\mathcal{B}} o(\log(n))})$*, unless ETH fails.*

Proof: To show this, we use a the previous hardness result, but using a constrained version of the HITTING SET problem obtained by Lokshtanov et al. (2011), where the element set is $[k] \times [k]$ and can thus be seen as a table with k rows and k columns:

Problem 4 $k \times k$ Hitting Set

Input:	An integer k, and $R_1, ..., R_t \subseteq [k] \times [k]$
Question:	Is there a set R containing exactly one element from each row such that $R \cap R_i \neq \emptyset$ for any $i \in [t]$?

Lokshtanov et al. (2011) show that assuming ETH this problem cannot be solved in $2^{o(k\log(k))}n^{O(1)}$ (whereas a simple brute force solves it in $O^*(2^{k\log(k)})$). Notice that we can modify the question of this problem by dropping the constraint that S contains at least one element from each row. Indeed, let us add to the instance a set of k sets $\{R'_1, \ldots, R'_k\}$, where R'_i contains all elements of row i for $i \in [k]$. Now, finding a (classical) hitting set of size k on this modified instance is equivalent to finding a solution of size k for the original instance of $k \times k$-HITTING SET. Moreover, it is easy to check that a $2^{o(k\log(k))}n^{O(1)}$ algorithm for this relaxed problem would also contradict ETH. To summarize, we know that unless ETH fails, there is no $2^{o(k\log(k))}n^{O(1)}$ algorithm for the classical HITTING SET problem, even when the ground set has size k^2. This allows us to perform the reduction of Lemma 3 on these special instances, leading to an instance $\mathcal{I}[m', n', p', k']$ with associated parameter $\zeta_{\mathcal{B}}$ such that $\zeta_{\mathcal{B}} = k$ and $n' = k^2$. Suppose now that there exists an algorithm for BMVA running in $2^{o(\zeta_{\mathcal{B}})\log(n)}(k+m+n+p)^{O(1)}$. Using the reduction above, we would be able to solve the instance of $k \times k$-HITTING SET in $2^{o(k)\log(k^2)}n^{O(1)}$, and thus in $2^{o(k\log(k))}n^{O(1)}$, which would violate ETH. A similar idea also rules out any algorithm running in $2^{\zeta_{\mathcal{B}} o(\log(n))}$ under ETH. □

4 Parameterizing according to ζ_p

We now consider the problem parameterized by $\zeta_p = k - p$ (recall that $p \leq k$). Notice that one motivation of this parameterization is the previous reduction of Lemma 3 from HITTING SET. Indeed, when applied for $n = 2$, it reduces an instance of VERTEX COVER to an instance of BMVA with $k = p + \zeta_p$ where ζ_p is equal to the size of the vertex cover. Our intuition is confirmed by the following result: we show that when parameterized by ζ_p, the problem is indeed $\mathbf{FPT}$ when $n = 2$ (Theorem 7). We complement this by showing that for any $n \geq 3$, it becomes $\mathbf{NP}$-hard when $\zeta_p = 0$ (Theorem 8), and is thus even not in $\mathbf{XP}$. The reduction we use even proves that for any fixed $n \geq 3$, the problem cannot be solved in $2^{o(k)}$ (and thus in $2^{o(\zeta_{\mathcal{B}})}$) unless ETH fails, while the algorithm of Theorem 4 runs in $O^*(2^{O(\zeta_{\mathcal{B}})})$. In the following, n-BMVA denotes the problem BMVA where the size of all sets is fixed to some constant $n \in \mathbb{N}$.

4.1 Positive result for $n = 2$

In this subsection, we prove that 2-BMVA is $\mathbf{FPT}$ parameterized by ζ_p. To do so, we reduce to the ODD CYCLE TRANSVERSAL problem (OCT for short). In this problem, given a graph $G = (V, E)$ and an integer $c \in \mathbb{N}$, the objective is to decide whether there exists a partition (X, S_1, S_2) of V with $|X| \leq c$ such that S_1 and S_2 are independent sets.

We first introduce a generalized version of OCT, called BIP-OCT. In this problem, we are given a set of vertices V, an integer c, and a set of m pairs $(A_1, B_1), ..., (A_m, B_m)$ with $A_i, B_i \subseteq V$ for all $i \in [m]$ and $A_i \cap B_i = \emptyset$. Informally, each pair (A_i, B_i) can be seen as a complete bipartite subgraph. The output of BIP-OCT is described by a partition (X, S_1, S_2) of V such that for any $i \in [m]$, either ($A_i \setminus X \subseteq S_1$ and

$B_i \setminus X \subseteq S_2$) or ($A_i \setminus X \subseteq S_2$ and $B_i \setminus X \subseteq S_1$). The question is whether there exists such a partition with $|X| \leq c$. As we can see, if all A_i and B_i are singletons (and thus form edges), then BIP-OCT corresponds to OCT. Notice that in the following, the considered parameter of OCT and BIP-OCT will always be the standard parameter, *i.e.* c. We first show that there is a linear parameter-preserving reduction from 2-BMVA parameterized by ζ_p to BIP-OCT, and then that there is also a linear parameter-preserving transformation from BIP-OCT to OCT.

Lemma 4. *There is a linear parameter-preserving reduction from* 2-BMVA *parameterized by* ζ_p *to* BIP-OCT.

Proof: Let $\mathcal{I}[m, 2, p, p + \zeta_p]$ be an instance of 2-BMVA (*i.e.* in which every set contains only two vectors), and let us construct an instance $\mathcal{I}'$ of BIP-OCT, such that $\mathcal{I}$ has a solution of cost $p + \zeta_p$ iff $\mathcal{I}'$ has a solution of size ζ_p. Notice first that we can suppose that for any $i \in [m]$ and any $r \in [p]$, we cannot have both $v_1^i[r] = 0$ and $v_2^i[r] = 0$ as otherwise any stack s from any solution would have $v_s[r] = 0$, and thus we could safely remove such a component r from the instance (and decrease k and p by one).

Let the vertex set of $\mathcal{I}'$ be $[p]$. Then, for all $i \in [m]$, let us define $A_i = \{r | v_1^i[r] = 0\}$, and $B_i = \{r | v_2^i[r] = 0\}$ as depicted in Figure 5. By the foregoing, and as required in an instance of bip-OCT, we have $A_i \cap B_i = \emptyset$. Let us prove that $\mathcal{I}$ has a solution of cost $p + \zeta_p$ iff $\mathcal{I}'$ has a solution (X, S_1, S_2) with $|X| \leq \zeta_p$.

$\Rightarrow$ Let $S = \{s_1, s_2\}$ be a solution of $\mathcal{I}$ of cost $p + \zeta_p$. Let $X = \{r | v_{s_1}[r] = v_{s_2}[r] = 0\}$, $S_1 = \{r | v_{s_1}[r] = 0 \text{ and } v_{s_2}[r] = 1\}$, and $S_2 = \{r | v_{s_1}[r] = 1 \text{ and } v_{s_2}[r] = 0\}$. Notice that (X, S_1, S_2) forms a partition of $[p]$ (as we cannot have a r_0 with $v_{s_1}[r_0] = v_{s_2}[r_0] = 1$, as this would imply that all the nm vectors have $v[r_0] = 1$, and such cooordinates have been removed from the instance in Lemma 2), and $|X| = \zeta_p$. It remains to prove that (X, S_1, S_2) is a feasible solution of $\mathcal{I}'$. Let $i \in [m]$. Without loss of generality, let us suppose that v_1^i has been added to s_1 and v_2^i has been added to s_2. Let $r \in A_i \setminus X$. Since $r \in A_i$, we have $v_1^i[r] = 0$, and thus $v_{s_1}[r] = 0$. Since $r \notin X$, we have $v_{s_2}[r] = 1$. Thus, $r \in S_1$, which proves $A_i \setminus X \subseteq S_1$. Similarly, we can prove that $B_i \setminus X \subseteq S_2$.

$\Leftarrow$ Let (X, S_1, S_2) be a solution of $\mathcal{I}'$ with $|X| \leq \zeta_p$. Let s_1 be such that $v_{s_1}[r] = 0$ iff $r \in X$ or $r \in S_1$, and let s_2 be such that $v_{s_2}[r] = 0$ iff $r \in X$ or $r \in S_2$. It remains to prove that the solution $S = \{s_1, s_2\}$ is feasible, which immediately implies that its cost is $p + \zeta_p$. Let $i \in [m]$. Without loss of generality, let us suppose that $A_i \setminus X \subseteq S_1$ and $B_i \setminus X \subseteq S_2$. We now claim that v_1^i can be assigned to s_1 and v_2^i can be assigned to s_2 without creating any new zero. To do so, let us show that for all $r \in [p]$, we have $v_1^i[r] = 0 \implies v_{s_1}[p] = 0$ (resp. $v_2^i[r] = 0 \implies v_{s_2}[p] = 0$). Indeed, let $r \in [p]$ such that $v_1^i[r] = 0$. Then by construction, it means that $r \in A_i$. Thus, by definition of the solution (X, S_1, S_2), it means that either $r \in X$ or $r \in S_1$, which implies $v_{s_1}[r] = 0$ as desired. Similar arguments show that $v_2^i[r] = 0 \implies v_{s_2}[p] = 0$ for all $r \in [p]$. □

Lemma 5. *There is a linear parameterized reduction from* BIP-OCT *to* OCT.

Proof: Let $\mathcal{I} = (V, \{A_i, B_i\}_{i \in [m]}, c)$ be an instance of BIP-OCT. Let us construct a graph $G' = (V', E')$ which contains an odd cycle transversal of size c if and only if $\mathcal{I}$ has a solution of size c for BIP-OCT. Observe first that we cannot simply set $V' = V$ and $E' = \bigcup_{i \in [m], a \in A_i, b \in B_i} \{a, b\}$. Indeed, if for example $A_1 = \{2, \ldots, n\}$, $B_1 = \{1\}$, $A_2 = \{2, \ldots, \frac{n}{2}\}$ and $B_2 = \{\frac{n}{2} + 1, \ldots, n\}$, defining G' as above would lead to an odd cycle transversal of size one, as removing only vertex $\{1\}$ makes the graph bipartite with bipartization (A_2, B_2). However, this solution is not feasible for BIP-OCT as $A_1 \setminus X = A_1$, and $A_1 \not\subseteq A_2$

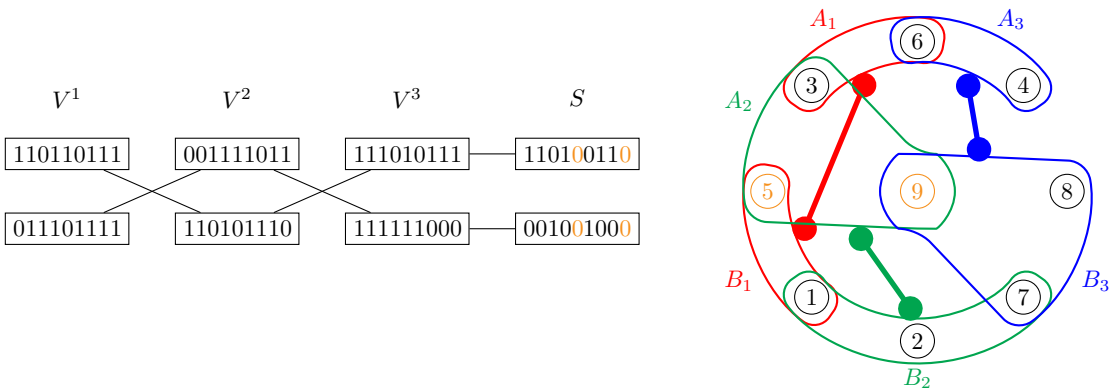

Figure 5: Example of reduction from an instance of 2-BMVA, with $n = 2, m = 3, p = 9$, admitting a solution of cost $p + 2$, to an instance of BIP-OCT, with $|V| = p$, $A_1 = \{3, 6\}, B_1 = \{1, 5\}, A_2 = \{2, 3, 7\}, B_2 = \{3, 5, 9\}, A_3 = \{4, 6\}, B_3 = \{7, 8, 9\}$, admitting the partition $(\{5, 9\}, \{3, 4, 6\}, \{1, 2, 7, 8\})$ as solution of cost 2.

and $A_1 \not\subseteq B_2$. Intuitively, we have to prevent solutions of G' from splitting sets $A_i \setminus X$ (and $B_i \setminus X$) between the two parts of the bipartization. To do so, we will construct G' as described above, and then we "augment" each bipartite graph by adding $c + 1$ new vertices on each side. More formally, we start by setting $V' = V$ as said before, and for all $i \in [m]$, we create two sets of $c + 1$ new vertices A'_i, B'_i. We then set $E' = \bigcup_{i \in [m], a \in A_i \cup A'_i, b \in B_i \cup B'_i} \{a, b\}$.

Let us now prove that $\mathcal{I}$ contains a solution of size c for BIP-OCT if and only if G' contains an odd cycle transversal of size c.

$\Rightarrow$ Let (X, S_1, S_2) be an optimal solution of $\mathcal{I}$. We define a partial solution X', S'_1, S'_2 of G' by setting $X' = X$ and $S'_l = S_l$ for $l \in \{1, 2\}$ (the solution is partial in the sense that it remains to assign vertices of $A'_i \cup B'_i$, for all $i \in [m]$). Let $i \in [m]$. If $A_i \setminus X = \emptyset$ and $B_i \setminus X = \emptyset$, then we add (arbitrarily) A'_i to S'_1 and B'_i to S'_2. Otherwise, if $A_i \setminus X \neq \emptyset$ and is added to S_l, we add A'_i to S'_l and B'_i to $S'_{l'}$ with $l, l' \in \{1, 2\}, l' \neq l$, and if $B_i \setminus X \neq \emptyset$ and is added to S_l, we add B'_i to S'_l and A'_i to $S'_{l'}$ with $l, l' \in \{1, 2\}, l' \neq l$.

This new solution has the same size ($|X'| = |X|$) and we claim that it is an odd cycle transversal of G'. Indeed, let us check that any edge $\{u, v\} \in E'$ such that $\{u, v\} \cap X' = \emptyset$ is not entirely contained in a S'_l. If $\{u, v\}$ is an edge of a complete bipartite of $\mathcal{I}$, *i.e.* if there exists i such that $u \in A_i$ and $v \in B_i$, then by definition of the solution (X, S_1, S_2) it is straightforward that u and v are not both in S'_1 nor in S'_2. Otherwise, if $\{u, v\}$ is adjacent to one or two of the new vertices, let i be such that $u \in A'_i$. If $v \in B'_i$, then the solution is valid as A'_i and B'_i are never added to the same set S'_l, $l \in \{1, 2\}$. Otherwise, we necessarily have $v \in B_i$. Let $l \in \{1, 2\}$ be such that $B_i \setminus X$ (which is not empty) has been added to S'_l. In this case A_i (and thus u) has been added to $S'_{l'}$, with $l' \neq l$.

$\Leftarrow$ Let (X', S'_1, S'_2) be an optimal solution of G'. For any $i \in [m]$, let $\tilde{A}_i = (A_i \cup A'_i) \setminus X'$ and $\tilde{B}_i = (B_i \cup B'_i) \setminus X'$. A first observation is that $\tilde{A}_i \neq \emptyset$ and $\tilde{B}_i \neq \emptyset$ as $|A_i \cup A'_i| = |B_i \cup B'_i| > c$ and $|X'| \leq c$. A second observation is that for any u and $v \in \tilde{A}_i$, u and v are in the same set S'_l for some $l \in \{1, 2\}$. Indeed, suppose by contradiction that $u \in S'_1$ and $v \in S'_2$. As $\tilde{B}_i \neq \emptyset$, there exists $b \in \tilde{B}_i$ and

$l \in \{1,2\}$ such that $b \in S'_l$. As all the edges of the complete bipartite subgraph on $(A_i \cup A'_i, B_i \cup B'_i)$ belong to E', we have $\{u,b\} \in E'$ and $\{v,b\} \in E'$, and thus S'_l contains both endpoints of an edge of E', which is a contradiction. In the same way, we can prove that for any $i \in [m]$, and any u and $v \in \tilde{B}_i$, u and v are in the same set S'_l for some $l \in \{1,2\}$.

Thus, according to the two previous observations, for any $i \in [m]$ we can define $\lambda_{\tilde{A}_i} \in \{1,2\}$ and $\lambda_{\tilde{B}_i} \in \{1,2\}$ such that $\tilde{A}_i \subseteq S'_{\lambda_{\tilde{A}_i}}$ and $\tilde{B}_i \subseteq S'_{\lambda_{\tilde{B}_i}}$, with $\lambda_{\tilde{A}_i} \neq \lambda_{\tilde{B}_i}$.

Let us now define $X = X' \cap V$, $S_1 = S'_1 \cap V$, and $S_2 = S'_2 \cap V$, and check that this is a valid solution of $\mathcal{I}$. Let $i \in [m]$. Observe first that $A_i \setminus X \subseteq \tilde{A}_i$, and thus either $A_i \setminus X = \emptyset$, or $A_i \setminus X \subseteq S_{\lambda_{\tilde{A}_i}}$. As the same fact also holds for $B_i \setminus X$, and as $\lambda_{\tilde{A}_i} \neq \lambda_{\tilde{B}_i}$, the constraint ($A_i \setminus X \subseteq S_1$ and $B_i \setminus X \subseteq S_2$) or ($A_i \setminus X \subseteq S_2$ and $B_i \setminus X \subseteq S_1$) is respected, and the solution is feasible, which concludes the proof. □

As ODD CYCLE TRANSVERSAL can be solved in $O^*(2.3146^c)$, proved by Lokshtanov et al. (2014), and since our parameters are exactly preserved in our two reductions, we obtain the following result:

Theorem 7. 2-BMVA *can be solved in* $O^*(d^{\zeta_p})$ *where* $d \leq 2.3146$ *is such that* OCT *can be solved in* $O^*(d^c)$.

4.2 Negative results for $n \geq 3$

We now complement the previous result by proving that the problem is intractable with respect to the parameter ζ_p for larger values of n.

Theorem 8. *For any fixed* $n \geq 3$, n-BMVA *is not in* $\mathbf{XP}$ *when parameterized by* ζ_p *(unless* $\mathbf{P} = \mathbf{NP}$*), and cannot be solved in* $2^{o(k)}$ *(unless ETH fails).*

Proof: Let $\chi \geq 3$. We present a reduction from χ-COLORING. Given a graph $G = (V,E)$, this problem consists to ask for a mapping $f : V \longrightarrow [\chi]$ such that for all $\{u,v\} \in E$ we have $f(u) \neq f(v)$. Let $E = \{e_1, ..., e_{m_G}\}$ and $V = [n_G]$. Let us construct an instance $\mathcal{I}$ of n-BMVA with $n = \chi$, $p = n_G$, $m = m_G$ and such that G admits a χ-coloring iff I has a solution of cost p (*i.e.* $\zeta_p = 0$). To each edge $e_i = \{u,v\} \in E$, $i \in [m_G]$, we associate a set V^i with $|V^i| = \chi$, where:

- v^i_1 represents the vertex u, that is $v^i_1[u] = 0$ and $v^i_1[r] = 1$ for any $r \in [n_G]$, $r \neq u$,
- v^i_2 represents the vertex v, that is $v^i_2[v] = 0$ and $v^i_2[r] = 1$ for any $r \in [n_G]$, $r \neq v$,
- for all $j \in \{3, \ldots, \chi\}$, v^i_j is a 1-vector, *i.e.* it has a 1 at every component.

An example of this construcion is depicted in Figure 6. Let us now prove that G admits a χ-coloring iff $\mathcal{I}$ has a solution of cost $p = n_G$.

$\Rightarrow$ Let $S_j \subseteq V$, $j \in [\chi]$ be the χ color classes (notice that the S_j are pairwise disjoint, some of them may be empty, and $\bigcup_{j \in [\chi]} S_j = V$). To each S_j we associate a stack s_j such that $v_{s_j}[r] = 0$ iff $r \in S_j$. It remains to prove that the solution $S = \{s_1, \ldots, s_\chi\}$ is feasible, as its cost is exactly p by construction. Let us consider a set V^i where v^i_1 (resp. v^i_2) represents a vertex u (resp. v). As $\{u,v\}$ is an edge of G, we know that u and v have two different colors, *i.e.* that $u \in S_j$ and $v \in S_{j'}$, for some $j, j' \in [\chi]$ with $j \neq j'$. Thus, we can add v^i_1 to stack s_j, v^i_2 to stack $s_{j'}$, and the $\chi - 2$ other v^i_j ($j \geq 3$) in an arbitrary way. Since the only 0 in v^i_1 (resp. v^i_2) is at the uth (resp. v^{th}) component, we have $v^i_1 \wedge v_{s_j} = v_{s_j}$ (resp. $v^i_2 \wedge v_{s_{j'}} = v_{s_{j'}}$), which proves that S is feasible.

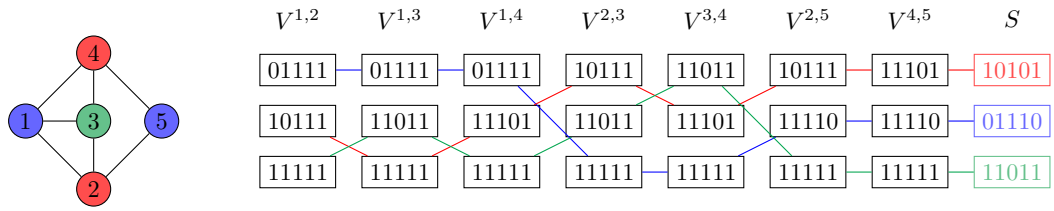

Figure 6: Example of reduction from a positive instance of χ-COLORING, with $\chi = 3$, $V = [5], E = \{\{1,2\},\{1,3\},\{1,4\},\{2,3\},\{3,4\},\{2,5\},\{4,5\}\}$, to an instance of BMVA with $m = |E| = 6$, $n = \chi = 3, p = |V| = 5$ admitting a solution of cost p.

$\Leftarrow$ Let $S = \{s_1, \ldots, s_\chi\}$ be the stacks of an optimal solution. For $j \in [\chi]$, let $S_j = \{r \in [p] | v_{s_j}[r] = 0\}$. Notice that $\bigcup_{j=1}^{\chi} S_j = V$, and as I is of cost p, all the S_j are pairwise disjoints and form a partition of V. Moreover, as for any $i \in [m]$, v_1^i and v_2^i have been assigned to different stacks, the corresponding vertices have been assigned to different colors, and thus each S_j induces an independent set, which completes the reduction.

It is known, thanks to Impagliazzo et al. (2001) that there is no $2^{o(|V|)}$ algorithm for deciding whether a graph $G = (V, E)$ admits a χ-COLORING, for any $\chi \geq 3$ (under ETH). As we can see, the value of the optimal solution for n-BMVA in the previous reduction equals the number of vertices in the instance of χ-COLORING, which proves that n-BMVA cannot be solved in $2^{o(k)}$ for any $n \geq 3$.

□

Finally, remark that as for the parameterization by p, one could ask if BMVA is **FPT** when parameterized by the first lower bound $\mathcal{B}$. However, we can see in the previous reduction that we obtain a graph with $\mathcal{B} = 2$, and thus the problem is even not in **XP** unless **P** = **NP**.

5 Conclusion

In this article, we presented some negative and positive results for a multidimensional binary vector assignment problem in the framework of parameterized complexity. Notice that neither lower bounds of Theorem 6 nor Theorem 8 are able to rule out an algorithm running in $O^*(2^k)$ (when n is part of the input), hence the existence of such an algorithm seems a challenging open problem. Another interesting question concerns the improvement of the $O(k^2 m)$ kernel of Theorem 1 by getting rid of the parameter m: does BMVA admit a polynomial kernel when parameterized by k only?

References

Bodlaender, H. L., Jansen, B. M., and Kratsch, S. (2014). Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305.

Bougeret, M., Boudet, V., Dokka, T., Duvillié, G., and Giroudeau, R. (2016). On the complexity of wafer-to-wafer integration. *Discrete Optimization*, 22:255–269.

Bougeret, M., Duvillié, G., Giroudeau, R., and Watrigant, R. (2015). Multidimensional binary vector assignment problem: Standard, structural and above guarantee parameterizations. In *Fundamentals of*

Computation Theory - 20th International Symposium, FCT 2015, Gdańsk, Poland, August 17-19, 2015, Proceedings, pages 189–201.

Cygan, M., Pilipczuk, M., Pilipczuk, M., and Wojtaszczyk, J. O. (2013). On multiway cut parameterized above lower bounds. *ACM Transactions on Computation Theory (TOCT)*, 5(1):3:1–3:11.

Dokka, T., Bougeret, M., Boudet, V., Giroudeau, R., and Spieksma, F. C. (2012). Approximation algorithms for the wafer to wafer integration problem. In *Approximation and Online Algorithms - 10th International Workshop, WAOA 2012, Ljubljana, Slovenia, September 13-14, 2012, Revised Selected Papers*, pages 286–297.

Dokka, T., Crama, Y., and Spieksma, F. C. (2014). Multi-dimensional vector assignment problems. *Discrete Optimization*, 14:111–125.

Downey, R. G. and Fellows, M. R. (2013). *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer.

Drucker, A. (2015). New limits to classical and quantum instance compression. *SIAM J. Comput.*, 44(5):1443–1479.

Gutin, G., Rafiey, A., Szeider, S., and Yeo, A. (2007). The linear arrangement problem parameterized above guaranteed value. *Theory of Computing Systems*, 41(3):521–538.

Gutin, G. and Yeo, A. (2012). The multivariate algorithmic revolution and beyond. chapter Constraint Satisfaction Problems Parameterized Above or Below Tight Bounds: A Survey, pages 257–286. Springer-Verlag, Berlin, Heidelberg.

Impagliazzo, R., Paturi, R., and Zane, F. (2001). Which problems have strongly exponential complexity? *Journal of Computer and System Sciences*, 63:512–530.

Lokshtanov, D., Marx, D., and Saurabh, S. (2011). Slightly superexponential parameterized problems. In *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 760–776.

Lokshtanov, D., Marx, D., and Saurabh, S. (2013). Lower bounds based on the exponential time hypothesis. *Bulletin of EATCS*, 3(105).

Lokshtanov, D., N.S., N., and Ramanujan, M. (2014). Faster parameterized algorithms using linear programming. *ACM Transactions on Algorithms (TALG)*, 11(2):15.

Mahajan, M., Raman, V., and Sikdar, S. (2009). Parameterizing above or below guaranteed values. *Journal of Computer and System Sciences*, 75(2):137–153.

Reda, S., Smith, G., and Smith, L. (2009). Maximizing the functional yield of wafer-to-wafer 3-d integration. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 17(9):1357–1362.