APPROXIMATION ALGORITHMS FOR MULTI-DIMENSIONAL VECTOR ASSIGNMENT PROBLEMS

TRIVIKRAM DOKKA*, YVES CRAMA†, AND FRITS C.R. SPIEKSMA*

Abstract. We consider a special class of axial multi-dimensional assignment problems called *multi-dimensional vector assignment* (MVA) problems. An instance of the MVA problem is defined by m disjoint sets, each of which contains the same number n of p-dimensional vectors with nonnegative integral components, and a cost function defined on vectors. The cost of an m-tuple of vectors is defined as the cost of their component-wise maximum. The problem is now to partition the m sets of vectors into n m-tuples so that no two vectors from the same set are in the same m-tuple and so that the total cost of the m-tuples is minimized. The main motivation comes from a yield optimization problem in semi-conductor manufacturing. We consider two classes of polynomial-time heuristics for MVA, namely, *hub* heuristics and *sequential* heuristics, and we study their approximation ratio. In particular, we show that when the cost function is monotone and subadditive, hub heuristics, as well as sequential heuristics, have finite approximation ratio for every fixed m. Moreover, we establish better approximation ratios for certain variants of hub heuristics and sequential heuristics when the cost function is monotone and submodular, or when it is additive. We provide examples to illustrate the tightness of our analysis. Furthermore, we show that the MVA problem is APX-hard even for the case $m = 3$ and for binary input vectors. Finally, we show that the problem can be solved in polynomial time in the special case of binary vectors with fixed dimension p.

Key words. multi-dimensional assignment; approximability; worst-case analysis; submodularity; wafer-to-wafer integration;

AMS subject classifications. 90C27, 90C59

1. Introduction.

1.1. Problem statement. We consider a multi-dimensional assignment problem motivated by an application arising in the semi-conductor industry. Formally, the input of the problem is defined by m disjoint sets $V_1, \ldots, V_m$, where each set V_k contains the same number n of p-dimensional vectors with nonnegative integral components, and by a *cost function* $c(u) : \mathbb{Z}_+^p \to \mathbb{R}_+$. Thus, the cost function assigns a nonnegative cost to each p-dimensional vector. A (feasible) m-tuple is an m-tuple of vectors $(u^1, u^2, \ldots, u^m) \in V_1 \times V_2 \times \ldots \times V_m$, and a *feasible assignment* for $V \equiv V_1 \times \ldots \times V_m$ is a set A of n feasible m-tuples such that each element of $V_1 \cup \ldots \cup V_m$ appears in exactly one m-tuple of A. We define the component-wise maximum operator $\vee$ as follows: for every pair of vectors $u, v \in \mathbb{Z}_+^p$,

$$u \vee v = (\max(u_1, v_1), \max(u_2, v_2), \ldots, \max(u_p, v_p)).$$

Now, the cost of an m-tuple $(u^1, \ldots, u^m)$ is defined as $c(u^1 \vee \ldots \vee u^m)$ and the cost of a feasible assignment A is the sum of the costs of its m-tuples: $c(A) = \sum_{(u^1,\ldots,u^m)\in A} c(u^1 \vee \ldots \vee u^m)$.

With this terminology, the *multi-dimensional vector assignment problem* (MVA-m, or MVA for short) is to find a feasible assignment for V with minimum total cost. A case of special interest is the case when all vectors in $V_1 \cup \ldots \cup V_m$ are binary 0–1 vectors; we call this special case *binary* MVA. Finally, the *wafer-to-wafer integration*

*ORSTAT, KU Leuven, Naamsestraat 69, B-3000 Leuven, Belgium. Trivikram.Dokka@kuleuven.be, Frits.Spieksma@kuleuven.be

†QuantOM, HEC Management School, University of Liège, Rue Louvrex 14 (N1), B-4000 Liège, Belgium. Yves.Crama@ulg.ac.be

problem (WWI-m or WWI for short) arises when the cost function of the binary MVA is *additive*, meaning that $c(u) = \sum_{i=1}^{p} u_i$.

In this paper, we investigate how closely the optimal solution of MVA-m and WWI-m can be approximated by polynomial-time approximation algorithms.

EXAMPLE 1. *An instance of WWI with $m = 3, n = p = 2$ is displayed in Figure 1.1. The optimal value of the instance is equal to 2: it is achieved by assigning the first vector of V_1, the second vector of V_2, and the first vector of V_3 to the same triple, thus arriving at vector $(1, 0)$ with cost $c(1, 0) = 1$; the remaining three vectors form a second triple with cost $c(0, 1) = 1$.*

V_1	V_2	V_3
00	00	10
01	10	01

FIG. 1.1. *A WWI-3 instance with $m = 3, n = p = 2$*

1.2. Wafer-to-wafer integration and related work. The motivation for studying the WWI problem arises from the optimization of the wafer-to-wafer production process in the electronics industry. We only provide a brief description of this application; for additional details, we refer to papers by Reda, Smith and Smith [11], Taouil and Hamdioui [17], Taouil et al. [18], and Verbree et al. [19].

For our purpose, a *wafer* can be viewed as a string of elements called *dies*. Each die can be either good (operative) or bad (defective). So, a wafer can be modeled as a binary vector, where each '0' represents a good die and each '1' represents a bad die. There are m lots of wafers, say $V_1, \ldots, V_m$, and each lot contains n wafers. All wafers in a given lot are meant to have identical functionalities, were it not for the occasional occurence of defective dies during the previous production steps. The wafer-to-wafer integration process requires to form *stacks*, where a stack is obtained by "superposing" m wafers chosen from different lots; thus, a stack corresponds to a feasible m-tuple. As a result of integration, each position in the stack gives rise to a *three-dimensional stacked integrated circuit* (3D-SIC) which is 'good' only when the corresponding m entries of the selected wafers are 'good'; otherwise, the 3D-SIC is 'bad'. The yield optimization problem now consists in assigning the available wafers to n stacks so as to minimize the total number of bad 3D-SICs. Thus, the WWI problem provides a model for yield optimization.

The wafer-to-wafer yield optimization problem has recently been the subject of much attention in the engineering literature. One example is the contribution by Reda et al. [11]. These authors formulate WWI as a multi-dimensional assignment problem. A natural formulation of WWI as an integer linear programming problem turns out to be hard to solve to optimality for instances with large values of m (typical dimensions for the instances are: $3 \le m \le 10$, $25 \le n \le 75$, $500 \le p \le 1000$). On the other hand, Reda et al. [11] propose several heuristics and show that they perform well in computational experiments. Some recent work in this direction is also reported in [13, 17, 18, 19].

Our main objective in this paper is to study the approximability of the MVA problem and of the WWI problem (in the sense of [20]). Let us note at this point that the wafer-to-wafer integration problem is usually formulated in the literature as a maximization problem (since one wants to maximize the yield). However, we feel that from the approximation point of view, it is more appropriate to study its cost minimization version. Indeed, in industrial instances, the number of bad dies in each wafer is typically much less than the number of good dies. Therefore, it is more relevant to be able to approximate the (smaller) minimum cost than the (larger) maximum yield.

Since MVA is defined as a multi-dimensional assignment problem with a special cost structure, our work relates to previous publications on special classes of multi-dimensional assignment problems, such as Bandelt, Crama and Spieksma [1], Burkard, Rudolf and Woeginger [3], Crama and Spieksma [4], Dokka, Kouvela and Spieksma [6], Goossens et al. [7], Spieksma and Woeginger [15], etc. Surveys on multi-dimensional assignment problems can be found in Chapter 10 of [2] and in [14]. To the best of our knowledge, the approximability of MVA has only been previously investigated by Dokka et al. [5], who mostly focused on the case $m = 3$ with additive cost functions. The present paper extends to MVA-m and considerably strengthens the results presented in [5].

1.3. Contents of the paper. Section 2 contains a formulation of the problem as an integer program (Subsection 2.1), discusses various possible assumptions on the cost function (Subsection 2.2), describes two classes of heuristics (Subsection 2.3), and gives an overview of our results in Subsection 2.4. In Section 3, we prove that the heuristics have finite worst-case performance for every fixed m under various assumptions on the cost function c. In Section 4, we prove that the WWI-m problem is APX-hard even when $m = 3$, all input vectors are binary, and the cost function is additive. Finally, we show in Section 5 that WWI-m can be solved in polynomial time when p is fixed.

2. Problem formulation, properties, heuristics and results.

2.1. Problem formulation. As mentioned before, let $V = V_1 \times V_2 \times \ldots \times V_m$ be the set of feasible m-tuples. By a slight abuse of notations, we write $u^k \in a$ when $a = (u^1, \ldots, u^m)$ and $1 \leq k \leq m$. We also extend the definition of c to m-tuples of $\mathbb{Z}_+^{mp}$ by setting $c(u^1, \ldots, u^m) := c(u^1 \vee \ldots \vee u^m)$, and, when W is any set of m-tuples, we define $c(W) = \sum_{a \in W} c(a)$.

Let us first provide an IP-formulation of MVA-m as an m-dimensional axial assignment problem. For each $a \in V$, let x_a be a binary variable indicating whether m-tuple a is selected ($x_a = 1$) or not ($x_a = 0$) in the optimal assignment. Reda et al. [11] give the following formulation of WWI, which directly extends to MVA:

$$\begin{aligned} \text{minimize} \quad & \sum_{a \in V} c(a) x_a \\ \text{s.t.} \sum_{a : u \in a} x_a &= 1 \quad \text{for all } u \in \cup_{i=1}^m V_i, \\ x_a &\in \{0, 1\} \quad \text{for all } a \in V. \end{aligned}$$

Other formulations of MVA exist; for instance, Dokka et al. [5] propose an alternative IP-formulation that may be more effective from a computational perspective.

In any application of MVA, the cost function c is likely to have some structure. Indeed, in the WWI-application motivating this study, we have, as mentioned before, an *additive* cost function: $c(u) = \sum_{i=1}^p u_i$. We now list various possible assumptions on the cost function c.

2.2. Properties of the cost function c. We focus our attention on cost functions $c(u)$ satisfying one or more of the following properties:

Monotonicity: The cost function c is *monotone* if, for all $u, v \in \mathbb{Z}_+^p$ with $u \leq v$, we have $0 \leq c(u) \leq c(v)$.

Subadditivity: The cost function c is *subadditive* if, for all $u, v \in \mathbb{Z}_+^p$, we have $c(u \vee v) \leq c(u) + c(v)$.

Submodularity: The cost function c is *submodular* if, for all $u, v \in \mathbb{Z}_+^p$, we have $c(u \vee v) + c(u \wedge v) \leq c(u) + c(v)$.

(Here, $\wedge$ denotes the component-wise minimum operator:

$$u \wedge v = (\min(u_1, v_1), \min(u_2, v_2), \ldots, \min(u_p, v_p)).$$

Submodular cost functions frequently appear in the analysis of approximation algorithms for combinatorial optimization problems; for recent illustrations, see for instance [9, 16] and the references therein. The additive cost function of problem WWI actually satisfies a much stronger property than submodularity, namely:

Modularity: The cost function c is *modular* if, for all $u, v \in \mathbb{Z}_+^p$, we have $c(u \vee v) + c(u \wedge v) = c(u) + c(v)$.

It is well-known that c is modular if and only if there exist p functions $f_\ell(u_\ell)$ such that $c(u) = \sum_{\ell=1}^{p} f_\ell(u_\ell)$ (see, e.g., Theorem 2.3.3 in Simchi-Levy, Chen and Bramel [12]). For the MVA problem, therefore, assuming additivity is essentially equivalent to assuming monotonicity and modularity.

2.3. Heuristics. Consider any heuristic algorithm H for MVA-m. Following standard terminology (see, e.g., Williamson and Shmoys [20]), we say that H is a $\rho^H(m)$-*approximation* algorithm for MVA-m if H runs in polynomial time and if $\rho^H(m)$ is (an upper bound on) the *approximation ratio* of H, in the following sense: for every instance of MVA-m with optimal value c_m^{OPT}, when H returns the assignment A_m, then $c(A_m) \leq \rho^H(m) c_m^{OPT}$.

Here, we are interested in the behavior of the following *hub* and *sequential* heuristics, which all rely on the observation that MVA-2 boils down to a classical bipartite assignment (or matching) problem (see, e.g., Bandelt et al. [1] for other examples of sequential and hub heuristics).

We first describe so-called *hub* heuristics, where one particular set V_h acts as a "hub", and where a feasible solution is obtained by combining bipartite assignments constructed for each pair (V_h, V_i) into a feasible assignment; see Algorithm 1. We will also analyze a version of the single-hub heuristic called *heaviest-hub* heuristic, or H^{hhub}; here, the hub V_h is the *heaviest* set, that is, $c(V_h) \geq c(V_k)$ for $k = 1, \ldots, m$; see Algorithm 2. The idea underlying this initial condition is to make sure that all assignments will be able to take the "worst lot" into account.

Finally, since there is one feasible single-hub solution for each possible choice of the hub V_h, $h = 1, \ldots, m$, we call *multi-hub* heuristic the heuristic that outputs the best of these m solutions; see Algorithm 3.

Let us now turn to the *sequential* heuristic H^{seq} described as Algorithm 4: H^{seq} progressively builds a feasible solution H_m by optimally assigning the next set V_i to a partial solution H_{i-1}. We point out that, for WWI-m, Reda et al. [11] proposed an *iterative matching heuristic* which performed very well in their computational experiments. Algorithm 4 is a natural generalization of this iterative matching heuristic. (See also Taouil et al. [18] for a related study where sequential heuristics are called "layer-by-layer" heuristics.)

Algorithm 1 Single-hub heuristic $H^{hub}(V_h)$

{comment: $h \in \{1,\ldots,m\}$ is the index of the hub}
for $i = 1$ to m, $i \neq h$, **do**
solve an assignment problem between V_h and V_i, $i \neq h$, based on costs $c(u \vee v)$, $u \in V_h$, $v \in V_i$; call the resulting optimal assignment M_{hi}, say, $M_{hi} = \{(u_j^h, u_j^i) \mid u_j^h \in V_h, u_j^i \in V_i, j = 1,\ldots,n\}$;
end for
construct the feasible solution $M_h = \{(u_j^1, u_j^2, \ldots, u_j^m) \mid (u_j^h, u_j^i) \in M_{hi}, i = 1,2,\ldots,m, j = 1,\ldots,n\}$;
output M_h.

Algorithm 2 Heaviest-hub heuristic H^{hhub}

reindex $V_1,\ldots,V_m$ so that $c(V_1) \geq c(V_k)$ for $k = 1,\ldots,m$;
apply the single-hub heuristic $H^{hub}(V_1)$.

Observe that the order of the sets $V_1,\ldots,V_m$ is arbitrary in the sequential heuristic. We obtain a slightly more restrictive heuristic, called *heaviest-first* heuristic, or H^{heavy}, when we specify that the heaviest set is contained in the first assignment; see Algorithm 5. (A more specific version, where the sets are ordered by nonincreasing weights, was shown by Singh [13] to be computationally effective.)

Clearly, each of the above heuristics runs in polynomial time. In fact, one can measure the complexity of these heuristics by observing how many (two-dimensional) assignment problems they need to solve. The most expensive one is H^{mhub}, since it solves $O(m^2)$ assignment subproblems, whereas H^{hub}, H^{hhub}, H^{seq}, and H^{heavy} only solve $O(m)$ subproblems. Observe that the preprocessing step needed for H^{hhub} and H^{heavy} does not increase their complexity.

2.4. Overview of results. In this section we list the main results proved in our paper. First, in case, c is monotone and subadditive, *no* feasible solution can be arbitrarily far away from the optimum, as expressed by the next theorem.

THEOREM 2.1. *Every heuristic H that returns a feasible solution is an m-approximation algorithm when the cost function c is monotone and subadditive. The approximation ratio $\rho^H(m) = m$ is tight for all $m \geq 2$, even for WWI-m.*

We prove this result in Subsection 3.1. Next, we establish that the multi-hub heuristic has an approximation ratio of $\frac{m}{2}$ when c is monotone and submodular. In fact, this ratio already holds for the single-hub heuristic $H^{hub}(V_h)$ when we assume that V_h is the heaviest set.

THEOREM 2.2. *The heaviest-hub heuristic H^{hhub} and the multi-hub heuristic H^{mhub} are $\frac{m}{2}$-approximation algorithms for MVA-m when the cost function c is monotone and submodular. The approximation ratio $\rho^{hhub}(m) = \rho^{mhub}(m) = \frac{m}{2}$ is tight for all $m \geq 2$, even for binary MVA.*

We prove this result in Subsection 3.2. Also, when c is monotone and submodular, the sequential heuristic has the same worst-case approximation ratio:

THEOREM 2.3. *The sequential heuristic H^{seq} is an $\frac{m}{2}$-approximation algorithm for MVA-m when the cost function c is monotone and submodular, for every order of the sets $V_1,\ldots,V_m$. The approximation ratio $\rho^{seq}(m)) = \frac{m}{2}$ is tight for all $m \geq 2$, even for the heaviest-first heuristic and even for binary MVA.*

We prove this result in Subsection 3.3. When c is additive, a better bound can

Algorithm 3 Multi-hub heuristic H^{mhub}

for $h = 1$ to m **do**
  apply the single-hub heuristic $H^{hub}(V_h)$ to produce the feasible solution M_h;
end for
let $h^* = \arg\min_h c(M_h)$; output M_{h^*}.

Algorithm 4 Sequential heuristic H^{seq}

let $H_1 := V_1$;
for $i = 2$ to m **do**
  solve a bipartite assignment problem between H_{i-1} and V_i based on the costs $c(u^1 \vee \ldots \vee u^{i-1} \vee v)$, for all $(u^1, \ldots, u^{i-1}) \in H_{i-1}$ and $v \in V_i$; let H_i be the resulting assignment for $V_1 \times V_2 \times \ldots \times V_i$;
end for
output H_m.

be proved for the heaviest-first heuristic:

THEOREM 2.4. *The heaviest-first heuristic H^{heavy} is a $(\frac{1}{2}(m+1) - \frac{1}{4}\ln(m-1))$-approximation algorithm for MVA-m when the cost function c is additive.*

We prove this result in Subsection 3.4. Although we do not know whether the bound in Theorem 2.4 is tight, we exhibit in Section 3.5 a family of instances for which H^{heavy} displays the following behavior:

THEOREM 2.5. *There exists an infinite sequence of values of m such that the heaviest-first heuristic produces a feasible assignment with cost larger than $\frac{\sqrt{m}}{2} c_m^{OPT}$ on certain instances of WWI-m.*

This concludes the overview of our results concerning approximation ratios of the heuristics; see also Figure 2.1.

One might wonder about the precise complexity status of MVA-m, and of its special case WWI-m. The following result implies that, when restricting ourselves to polynomial-time algorithms, constant-factor approximation algorithms are the best we can hope for (unless P=NP), even for WWI-3:

THEOREM 2.6. *WWI-3 is APX-hard, even when all vectors in $V_1 \cup V_2 \cup V_3$ are 0–1 vectors with exactly two nonzero entries per vector.*

We prove this in Section 4. Finally, in case the dimension p of the vectors is fixed, we show in Section 5 that binary MVA-m can be solved in polynomial time:

THEOREM 2.7. *Binary MVA can be solved in polynomial time for each fixed p.*

3. Proofs of approximation ratios.

This section is devoted to the proofs of the approximation ratios of the heuristics described in Subsection 2.3.

3.1. Monotone and subadditive costs: feasible solutions.

Here, we first establish some properties of feasible solutions depending on various assumptions on the cost function c. Consider a feasible assignment A_m for $V_1 \times \ldots \times V_m$, and let A_k denote the restriction of this assignment to $V_1 \times \ldots \times V_k$, for all $k \leq m$. Denote by

Algorithm 5 Heaviest-first heuristic H^{heavy}

reindex $V_1, \ldots, V_m$ so that $\max(c(V_1), c(V_2)) \geq c(V_k)$ for $k = 1, \ldots, m$;
apply the sequential heuristic.

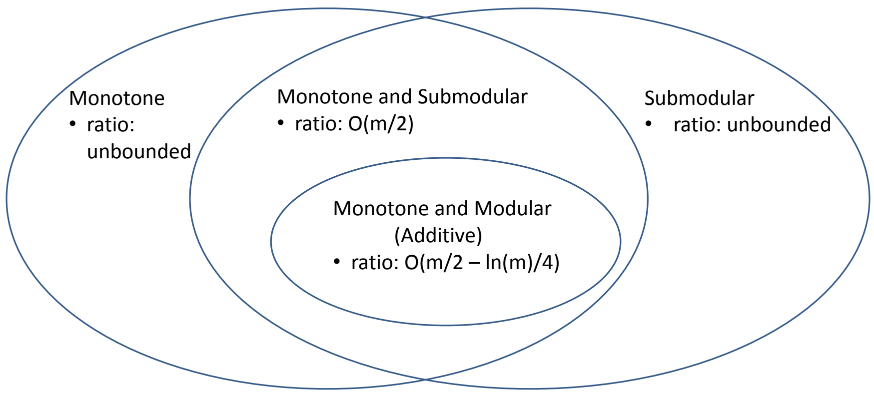

FIG. 2.1. *Overview of approximability results for monotone and submodular cost functions*

c_k^{OPT} the optimal value of the restricted instance $V_1 \times \ldots \times V_k$.

LEMMA 3.1. *If the cost function c is monotone and if A_m is a feasible assignment, then, for all $i \leq k \leq m$,*

$$c(V_i) \leq c_k^{OPT} \leq c(A_k) \leq c(A_m). \tag{3.1}$$

Proof. Obvious. ☐

LEMMA 3.2. *If the cost function c is subadditive and if A_m is a feasible assignment, then*

$$c(A_m) \leq c(A_{m-1}) + c(V_m). \tag{3.2}$$

Proof. Assume without loss of generality that the j^{th} m-tuple of A_m is $(u_j^1, \ldots, u_j^m)$ (that is, the j^{th} m-tuple in the assignment contains the j^{th} vector of V_i for each i). Then,

$$\begin{aligned} c(A_m) &= \sum_{j=1}^{n} c(u_j^1 \vee \ldots \vee u_j^m) \\ &\leq \sum_{j=1}^{n} c(u_j^1 \vee \ldots \vee u_j^{m-1}) + \sum_{j=1}^{n} c(u_j^m) \\ &= c(A_{m-1}) + c(V_m). \end{aligned}$$

☐

These two lemmas allow us to prove:

THEOREM 2.1. *Every heuristic H that returns a feasible solution is an m-approximation algorithm when the cost function is monotone and subadditive. The approximation ratio $\rho^H(m) = m$ is tight for all $m \geq 2$, even for WWI-m.*

Proof. The statement holds for $m = 1$. Then, using Eq. (3.2) from Lemma 3.2, Eq. (3.1) from Lemma 3.1, and induction on m, we obtain that every feasible solution A_m satisfies

$$\begin{aligned} c(A_m) &\le c(A_{m-1}) + c(V_m) \\ &\le (m-1)\, c^{OPT}_{m-1} + c^{OPT}_m \\ &\le m\, c^{OPT}_m. \end{aligned}$$

To see that the bound is tight, let $p = 1$, $n = m$, $V_i = \{1, 0, \ldots, 0\}$ for $i = 1, \ldots, m$, and $c(u) = u$ for all $u \in \mathbb{R}$. The cost function is obviously additive, hence this is an instance of WWI-m. The worst feasible assignment yields $\{1, 1, \ldots, 1\}$ with cost m, whereas the optimal assignment has cost 1. □

Thus, Theorem 2.1 implies that every heuristic has bounded worst-case performance (for fixed m) under the assumption that c is monotone and subadditive. On the other hand, if we relax either of the assumptions on c, then even the heaviest-hub and heaviest-first sequential heuristics do not have bounded approximation ratios on WWI-3, as shown by the following examples.

EXAMPLE 2. *For any p, we denote by $\mathbf{0}$, $\mathbf{1}$, and e_i, respectively, the all-zero, all-one, and i-th unit vector of $\mathbb{Z}^p$.*

Let $p = 3$, $V_1 = \{e_1, \mathbf{0}\}$, $V_2 = \{\mathbf{0}, e_2\}$, $V_3 = \{\mathbf{1}, \mathbf{0}\}$, and $c(u) = u_1 + u_2 + u_3 - 3\min(u_1, u_2, u_3)$. This cost function is nonnegative, subadditive (and even submodular), but not monotone since $1 = c(e_1) \not\le c(\mathbf{1}) = 0$, while $e_1 \le \mathbf{1}$. The optimal solution for this instance is $\{(e_1, e_2, \mathbf{1}), (\mathbf{0}, \mathbf{0}, \mathbf{0})\}$ with cost 0. Since $c(V_1) = c(V_2) > c(V_3)$, the heaviest-first heuristic could match V_1, V_2 to produce $\{(e_1, \mathbf{0}), (\mathbf{0}, e_2)\}$, then V_3 to produce $\{(e_1, \mathbf{0}, \mathbf{1}), (\mathbf{0}, e_2, \mathbf{0})\}$ with cost 1. Heaviest-hub can produce the same solution.

A similar observation applies when c is not subadditive: let $p = 3$, $V_1 = \{e_1, \mathbf{0}\}$, $V_2 = \{\mathbf{0}, e_2\}$, $V_3 = \{e_3, e_3\}$, and $c(u) = u_1 + u_2 + M\min(u_1, u_2, u_3)$ with $M > 0$. This cost function is nonnegative, monotone (and supermodular), but not subadditive since $c(1, 0, 1) = c(0, 1, 1) = 1$ and $c(1, 1, 1) = 2 + M$. The optimal solution is $\{(e_1, \mathbf{0}, e_3), (\mathbf{0}, e_2, e_3)\}$ with cost 2. Note that $c(V_1) = c(V_2) = 1$, $c(V_3) = 0$; hence, heaviest-first could match V_1, V_2 to produce $\{(e_1, e_2), (\mathbf{0}, \mathbf{0})\}$, then V_3 to produce $\{(e_1, e_2, e_3), (\mathbf{0}, \mathbf{0}, e_3)\}$ with cost $M + 2$. So, the performance of heaviest-first (and similarly, heaviest-hub) is unbounded for this instance.

3.2. Monotone and submodular costs: hub heuristics. The proof of Theorem 2.1 easily implies that the ratio $\rho^{hub}(m) = m-1$ is valid for the solution produced by any single-hub heuristic when the cost function is monotone and subadditive (simply start the induction with $m = 2$ in the proof). This ratio is actually tight: To see this, consider an arbitrary instance $\mathcal{I}$ of MVA-$(m-1)$, and extend it with an additional set V_m consisting of n zero vectors. With V_m as the hub, $H^{hub}(V_m)$ can produce any feasible solution of $\mathcal{I}$. Hence, Theorem 2.1 establishes the tightness of the bound.

We are going to show next that, for heaviest-hub and multi-hub heuristics, better approximation ratios can be established when we assume that the cost function is monotone and submodular.

In the sequel, we frequently assume without loss of generality, as in the proof of Lemma 3.2, that the j^{th} m-tuple of A_m is $(u^1_j, \ldots, u^m_j)$. Under this assumption, we now derive inequalities that are valid for every feasible assignment A_m.

LEMMA 3.3. *If the cost function c is monotone and submodular, and if A_m is a feasible assignment such that the j^{th} m-tuple of A_m is $(u^1_j, \ldots, u^m_j)$, then, for all*

$k \in \{1, \dots, m-1\}$,

$$c(A_m) \leq c(A_{m-1}) + \sum_{j=1}^{n} c(u_j^k \vee u_j^m) - c(V_k) \tag{3.3}$$

$$\leq c(A_{m-1}) + c(V_m) - \sum_{j=1}^{n} c(u_j^k \wedge u_j^m) \tag{3.4}$$

$$\leq c(A_{m-1}) + c(V_m). \tag{3.5}$$

Proof.

$$c(A_m) = \sum_{j=1}^{n} c(u_j^1 \vee \dots \vee u_j^m) \tag{3.6}$$

$$\leq \sum_{j=1}^{n} c(u_j^1 \vee \dots \vee u_j^{m-1}) + \sum_{j=1}^{n} c(u_j^k \vee u_j^m)$$

$$- \sum_{j=1}^{n} c((u_j^1 \vee \dots \vee u_j^{m-1}) \wedge (u_j^k \vee u_j^m)) \tag{3.7}$$

$$\leq \sum_{j=1}^{n} c(u_j^1 \vee \dots \vee u_j^{m-1}) + \sum_{j=1}^{n} c(u_j^k \vee u_j^m) - \sum_{j=1}^{n} c(u_j^k) \tag{3.8}$$

$$\leq \sum_{j=1}^{n} c(u_j^1 \vee \dots \vee u_j^{m-1}) + \sum_{j=1}^{n} c(u_j^m) - \sum_{j=1}^{n} c(u_j^k \wedge u_j^m) \tag{3.9}$$

$$\leq \sum_{j=1}^{n} c(u_j^1 \vee \dots \vee u_j^{m-1}) + \sum_{j=1}^{n} c(u_j^m) \tag{3.10}$$

where (3.6) is by definition of the cost function, (3.7) holds by submodularity applied to $u = u_j^1 \vee \dots \vee u_j^{m-1}$ and $v = u_j^k \vee u_j^m$ for each j, (3.8) follows by monotonicity (since $u_j^k \leq (u_j^1 \vee \dots \vee u_j^{m-1}) \wedge (u_j^k \vee u_j^m)$), (3.9) by submodularity applied to $u = u_j^k$, $v = u_j^m$, and (3.10) by nonnegativity of c. Inequalities (3.8), (3.9), (3.10) are equivalent to (3.3), (3.4), (3.5), respectively. □

We can now prove:

THEOREM 2.2. *The heaviest-hub heuristic H^{hhub} and the multi-hub heuristic H^{mhub} are $\frac{m}{2}$-approximation algorithms for MVA-m when the cost function c is monotone and submodular. The approximation ratio $\rho^{hhub}(m) = \rho^{mhub}(m) = \frac{m}{2}$ is tight for all $m \geq 2$, even for binary MVA.*

Proof. We prove the theorem by induction on m. The result is trivial when $m = 2$. For larger values of m, assume as in the description of Algorithm 2 that V_1 is the heaviest set, let $H_m = M_1$ be the solution found by the heaviest-hub heuristic H^{hhub}, and let H_{m-1} be the restriction of this assignment H_m to $W = V_1 \times \dots \times V_{m-1}$.

We consider now two cases. Assume first that $c(V_1) \leq \frac{1}{2} c_m^{OPT}$. Applying (3.5) to the assignment M_1, we obtain

$$c(H_m) \leq c(H_{m-1}) + c(V_m). \tag{3.11}$$

Since H_{m-1} results from applying the heaviest-hub heuristic (with heaviest hub V_1)

to $W = V_1 \times \ldots \times V_{m-1}$, we have by induction and by monotonicity of c :

$$c(H_{m-1}) \leq \rho^{hhub}(m-1)\, c^{OPT}(W) \leq \frac{1}{2}(m-1)\, c_m^{OPT} \tag{3.12}$$

where $c^{OPT}(W)$ is the cost of an optimal assignment on W.

Finally, using the assumption that $c(V_1) \leq \frac{1}{2} c_m^{OPT}$, we conclude from (3.11)–(3.12) that

$$c(H_m) \leq (\frac{m-1}{2} + \frac{1}{2})\, c_m^{OPT} = \frac{m}{2}\, c_m^{OPT}.$$

Consider next the case where $c(V_1) \geq \frac{1}{2} c_m^{OPT}$. Assume, without loss of generality, that the j^{th} vector of H_m is $(u_j^1, \ldots, u_j^m)$. Then, by Eq. (3.3):

$$c(H_m) \leq c(H_{m-1}) + \sum_{j=1}^{n} c(u_j^1 \vee u_j^m) - c(V_1).$$

With $M_{1,m}$ denoting the optimal matching of V_1 and V_m as in Algorithm 1, we find: $\sum_{j=1}^{n} c(u_j^1 \vee u_j^m) = c(M_{1,m}) \leq c_m^{OPT}$. Thus,

$$\begin{aligned} c(H_m) &\leq c(H_{m-1}) + c(M_{1,m}) - c(V_1) \\ &\leq \rho^{hhub}(m-1)\, c_{m-1}^{OPT} + c_m^{OPT} - \frac{1}{2} c_m^{OPT} \\ &\leq (\frac{m-1}{2} + \frac{1}{2})\, c_m^{OPT} \\ &= \frac{m}{2}\, c_m^{OPT}. \end{aligned}$$

This proves that the approximation ratio $\rho^{hhub}(m) = \frac{1}{2} m$ is valid for the heaviest-hub heuristic H^{hhub} and hence, for the multi-hub heuristic H^{mhub} as well.

To prove that the ratio is tight, consider the function $r_2(u) = f(\sum_{i=1}^{p} u_i)$, where $f : \mathbb{R} \to \mathbb{R}$ is defined by $f(x) = x$ when $x \leq 2$, and $f(x) = 2$ when $x \geq 2$. Since f is monotone nondecreasing and concave, it follows easily that r_2 is monotone and submodular on $\mathbb{Z}_+^p$ (see, e.g., Theorem 2.3.6 in Simchi-Levy et al. [12]). (When u is a binary vector, $r_2(u)$ is the rank function of the uniform matroid of rank 2.)

Now, let $p = n = m$, $V_i = \{e_i, \mathbf{0}, \ldots, \mathbf{0}\}$ for $i = 1, \ldots, m$, and $c(u) = r_2(u)$. By symmetry, any of the sets V_i can be chosen as the heaviest set, and the multi-hub heuristic delivers a solution with the same cost as the heaviest-hub heuristic. In particular, it is easy to see that multi-hub can produce the assignment H_m in which e_i is matched with $m-1$ zero vectors, for all i. The resulting assignment H_m has cost m, whereas the optimal solution assigns $(e_1, \ldots, e_m)$ to the same tuple, and has cost $r_2(e_1 \vee \ldots \vee e_m) = 2$. ◻

Let us observe that the submodulariy assumption is necessary in Theorem 2.2, as shown by the following example.

EXAMPLE 3. *Let* $m = 3$, $n = 2$, $p = 3$, $V_1 = \{e_1, e_1\}$, $V_2 = V_3 = \{e_2, e_3\}$, *and* $c(u) = \max(u_1, u_2, u_3) + \min(u_1, u_2, u_3)$. *This cost function can be checked to be subadditive, but not submodular. The optimal solution is* $\{(e_1, e_2, e_2), (e_1, e_3, e_3)\}$, *with cost* 2. *However, using* V_1 *as a hub, heaviest-hub may find the solution* $\{(e_1, e_2, e_3), (e_1, e_3, e_2)\}$ *with cost* $4 > \frac{m}{2}$. *Multi-hub may fail in the same way.*

3.3. Monotone and submodular costs: sequential heuristics. Let us now turn to the analysis of sequential heuristics. It follows again from the proof of Theorem 2.1 that the performance ratio of any sequential heuristics is bounded by $m-1$ when the cost function is monotone and subadditive. Under the stronger submodularity assumption, we can establish a better bound:

THEOREM 2.3. *The sequential heuristic H^{seq} is an $\frac{m}{2}$-approximation algorithm for MVA-m when the cost function c is monotone and submodular, for every order of the sets $V_1, \ldots, V_m$. The approximation ratio $\rho^{seq}(m)) = \frac{m}{2}$ is tight for all $m \geq 2$, even for the heaviest-first heuristic and even for binary MVA.*

Proof. Let H_m be a feasible assignment for V found by the sequential heuristic. We prove the theorem by induction on m. The result is trivial when $m = 2$. For larger values of m, we distinguish among two cases as in the proof of the previous theorem. Assume first that $c(V_{m-1}) \leq \frac{1}{2}c_m^{OPT}$. Then, consider the partial assignment $A_{m-2,m}$ that is obtained by assigning optimally V_m to H_{m-2} (independently of V_{m-1}). Let H_m^* be the concatenation of H_{m-1} and $A_{m-2,m}$ (that is, H_m^* assigns V_{m-1} to H_{m-2} as in H_{m-1}, and V_m to H_{m-2} as in $A_{m-2,m}$). Note that $H_{m-1}^* = H_{m-1}$; therefore, $c(H_m) \leq c(H_m^*)$ since, by definition, the sequential heuristic assigns V_m optimally to H_{m-1}. Applying (3.5) to the assignment H_m^*, we obtain

$$c(H_m) \leq c(H_m^*) \leq c(A_{m-2,m}) + c(V_{m-1}). \tag{3.13}$$

Since $A_{m-2,m}$ results from applying the sequential heuristic to $W = V_1 \times \ldots \times V_{m-2} \times V_m$, we have by induction:

$$c(A_{m-2,m}) \leq \rho^{seq}(m-1)\, c^{OPT}(W) \leq \frac{1}{2}(m-1)\, c_m^{OPT} \tag{3.14}$$

where $c^{OPT}(W)$ is the cost of an optimal assignment on W.

Finally, using the assumption that $c(V_{m-1}) \leq \frac{1}{2}c_m^{OPT}$, we conclude from (3.13)–(3.14) that

$$c(H_m) \leq (\frac{m-1}{2} + \frac{1}{2})\, c_m^{OPT} = \frac{m}{2}\, c_m^{OPT}.$$

Assume now alternatively that $c(V_{m-1}) \geq \frac{1}{2}c_m^{OPT}$. Let $M_{m-1,m}$ be an optimal matching of V_{m-1} with V_m, and consider the assignment H_m^+ obtained by concatenating H_{m-1} with $M_{m-1,m}$. Assume, without loss of generality, that the j^{th} vector of H_m^+ is $(u_j^1, \ldots, u_j^m)$. Then, by definition of H_m, $c(H_m) \leq c(H_m^+)$ and by Eq. (3.3):

$$c(H_m) \leq c(H_m^+) \leq c(H_{m-1}) + \sum_{j=1}^{n} c(u_j^{m-1} \vee u_j^m) - c(V_{m-1}). \tag{3.15}$$

Moreover, $\sum_{j=1}^{n} c(u_j^{m-1} \vee u_j^m) = c(M_{m-1,m}) \leq c_m^{OPT}$. Thus, we derive

$$\begin{aligned} c(H_m) &\leq c(H_{m-1}) + c(M_{m-1,m}) - c(V_{m-1}) \\ &\leq \rho^{seq}(m-1)\, c_{m-1}^{OPT} + c_m^{OPT} - \frac{1}{2}c_m^{OPT} \\ &\leq (\frac{m-1}{2} + \frac{1}{2})\, c_m^{OPT} \\ &= \frac{m}{2}\, c_m^{OPT}. \end{aligned}$$

This establishes the validity of the approximation ratio $\rho^{seq}(m) = \frac{1}{2}m$.

The example given in the proof of Theorem 2.2 proves that the approximation ratio $\rho^{seq}(m) = \frac{m}{2}$ is tight even for the heaviest-first heuristic. □

As a side-remark, the worst-case example used in the proof of Theorem 2.2 and of Theorem 2.3 shows that, for monotone submodular instances of binary MVA-m, the same ratio $\frac{m}{2}$ is tight for the (expensive) combined heuristic that results by successively running the single-hub heuristic and the sequential heuristic for all possible choices of the hub and for all possible permutations of the sets $V_1, \ldots, V_m$. Also, Example 3 shows that the submodularity assumption is necessary in Theorem 2.3.

We return in Section 3.5 to a discussion of the approximation ratio of sequential heuristics for the more restrictive WWI-m problem.

3.4. Additive costs: heaviest-first heuristic. In this section, we explicitly rely on the assumption that the cost function is additive, i.e., $c(u) = \sum_{\ell=1}^{p} u_\ell$, and we derive an improved approximation ratio for the heaviest-first heuristic. We first establish a series of preliminary results.

3.4.1. Preliminary results for additive cost functions. If the j^{th} m-tuple of an arbitrary assignment A_m is $u_j = (u_j^1, \ldots, u_j^m)$, then, for all $j = 1, \ldots, n$

$$c(u_j) = \sum_{\ell=1}^{p} (u_{j\ell}^1 \vee \ldots \vee u_{j\ell}^m).$$

Thus,

$$\begin{aligned} c(A_m) &= \sum_{j=1}^{n} c(u_j) \\ &= c(A_{m-1}) + c(V_m) - \sum_{j=1}^{n} \sum_{\ell=1}^{p} \left((u_{j\ell}^1 \vee \ldots \vee u_{j\ell}^{m-1}) \wedge u_{j\ell}^m \right). \end{aligned}$$

For each j, ℓ, let $k(j,\ell)$ be an (arbitrary) index $k \in \{1, \ldots, m-1\}$ such that

$$u_{j\ell}^{k(j,\ell)} = u_{j\ell}^1 \vee \ldots \vee u_{j\ell}^{m-1}.$$

For each j, k, let $L(j,k) = \{\ell : k(j,\ell) = k\}$ (roughly speaking, $L(j,k)$ is the set of coordinates ℓ for which the maximum of $u_{j\ell}^1, \ldots, u_{j\ell}^{m-1}$ is attained in set V_k). Then,

$$\begin{aligned} c(A_m) &= c(A_{m-1}) + c(V_m) - \sum_{j=1}^{n} \sum_{\ell=1}^{p} \left(u_{j\ell}^{k(j,\ell)} \wedge u_{j\ell}^m \right) \\ &= c(A_{m-1}) + c(V_m) - \sum_{j=1}^{n} \sum_{k=1}^{m-1} \sum_{\ell \in L(j,k)} (u_{j\ell}^k \wedge u_{j\ell}^m). \end{aligned} \tag{3.16}$$

Consider now the quantity $Q = \sum_{j=1}^{n} \sum_{k=1}^{m-1} \sum_{\ell \in L(j,k)} (u_{j\ell}^k \wedge u_{j\ell}^m)$. Intuitively, $c(V_m) - Q$ in Eq. (3.16) represents the amount by which the cost of the partial solution A_{m-1} increases when the set V_m is appended to this partial solution: so, Q can be viewed as the amount of $c(V_m)$ that is "covered" by $V_1, \ldots, V_{m-1}$.

Clearly, there exists an index $k^* \in \{1, \ldots, m-1\}$ such that

$$\sum_{j=1}^{n} \sum_{\ell \in L(j,k^*)} (u_{j\ell}^{k^*} \wedge u_{j\ell}^m) \geq \frac{1}{m-1} Q$$

(there is a set V_{k^*} that, by itself, covers at least the fraction $\frac{1}{m-1}Q$ of the amount of $c(V_m)$ that is covered by $V_1, \ldots, V_{m-1}$ together).

Assume now that A_m is an optimal assignment: $c(A_m) = c_m^{OPT}$. Denote by H_m the assignment produced by a sequential heuristic which optimally matches the partial assignment H_{m-1} with V_m, and denote by H_{m,k^*} the assignment obtained by concatenating H_{m-1} with the assignment $\{(u_j^{k^*}, u_j^m) : j = 1, \ldots, n\}$ extracted from the optimal solution A_m. Clearly, $c(H_m) \leq c(H_{m,k^*})$. Inequality (3.4) implies that

$$\begin{aligned} c(H_m) &\leq c(H_{m,k^*}) \\ &\leq c(H_{m-1}) + c(V_m) - \sum_{j=1}^{n}\sum_{\ell=1}^{p}(u_{j\ell}^{k^*} \wedge u_{j\ell}^m) \\ &\leq c(H_{m-1}) + c(V_m) - \sum_{j=1}^{n}\sum_{\ell \in L(j,k^*)}(u_{j\ell}^{k^*} \wedge u_{j\ell}^m) \\ &\leq c(H_{m-1}) + c(V_m) - \frac{1}{m-1}Q. \end{aligned}$$

Using the definition (3.16) of Q, we obtain

$$c(H_m) \leq c(H_{m-1}) + c(V_m) - \frac{1}{m-1}(c(A_{m-1}) + c(V_m) - c_m^{OPT}) \tag{3.17}$$

$$= c(H_{m-1}) + \frac{m-2}{m-1}c(V_m) + \frac{1}{m-1}(c_m^{OPT} - c(A_{m-1})). \tag{3.18}$$

Note that the inequality (3.17)-(3.18) is valid for any sequential heuristic. But we are going to apply it next to the analysis of the heaviest-first heuristic.

3.4.2. A bound for the heaviest-first heuristic. As described in Algorithm 5, the heaviest-first heuristic arises when the first assignment contains the heaviest set V_i. Here, we assume with loss of generality that V_1 is the heaviest set.

We let $\mathcal{H}^{odd}(m) = \sum_{k=1}^{m} \frac{1}{2k-1}$. Then $\mathcal{H}^{odd}(m) = \mathcal{H}(2m-1) - \frac{1}{2}\mathcal{H}(m-1)$, where $\mathcal{H}(m) = \sum_{k=1}^{m}\frac{1}{k}$ is the harmonic function. It is well-known that $\ln(m+1) \leq \mathcal{H}(m) \leq 1 + \ln m$ for all $m \geq 1$. Thus, the function $\mathcal{H}^{odd}$ grows like $\frac{1}{2}\ln(m)$ and $\mathcal{H}^{odd}(m) \geq \frac{1}{2}\ln(m)$.

THEOREM 2.4. *The heaviest-first heuristic H^{heavy} is a $(\frac{1}{2}(m - \mathcal{H}^{odd}(m-1)+1)$-approximation algorithm for MVA-m when the cost function c is additive. Thus, $\rho^{heavy}(m) \leq \frac{1}{2}(m - \mathcal{H}^{odd}(m-1)+1) \leq \frac{1}{2}(m+1) - \frac{1}{4}\ln(m-1)$.*

Proof. Let H_m be the solution found by the heaviest-first heuristic. The proof proceeds by induction, starting with $m = 2$ and $\rho^{heavy}(2) = 1$.

Consider first the case where $c(V_1) \leq \frac{m-1}{2m-3}c_m^{OPT}$. By induction,

$$c(H_{m-1}) \leq \rho^{heavy}(m-1)\, c_{m-1}^{OPT},$$

where c_{m-1}^{OPT} is the cost of the optimal assignment for $V_1 \times \ldots \times V_{m-1}$. Let A_m again be an optimal assignment for $V_1 \times \ldots \times V_m$. Clearly, $c_{m-1}^{OPT} \leq c(A_{m-1})$. Using this in (3.17) together with $c(V_m) \leq c(V_1) \leq \frac{m-1}{2m-3}c_m^{OPT}$ yields

$$\begin{aligned} c(H_m) &\leq \rho^{heavy}(m-1)c(A_{m-1}) + (\frac{m-2}{m-1})(\frac{m-1}{2m-3})c_m^{OPT} + \frac{1}{m-1}(c_m^{OPT} - c(A_{m-1})) \\ &\leq \rho^{heavy}(m-1)\big(c(A_{m-1}) + c_m^{OPT} - c(A_{m-1})\big) + \frac{m-2}{2m-3}\,c_m^{OPT} \\ &\leq \big(\rho^{heavy}(m-1) + \frac{m-2}{2m-3}\big)\,c_m^{OPT}. \end{aligned}$$

The alternative case is when $c(V_1) \geq \frac{m-1}{2m-3} c_m^{OPT}$. Repeat the analysis leading to Eq. (3.15) in the second part of the proof of Theorem 2.3, but this time with V_1 replacing V_{m-1}. From there,

$$\begin{aligned} c(H_m) &\leq c(H_{m-1}) + c(M_{1,m}) - c(V_1) \\ &\leq \rho^{heavy}(m-1)\, c_{m-1}^{OPT} + c_m^{OPT} - \frac{m-1}{2m-3}\, c_m^{OPT} \\ &\leq \big(\rho^{heavy}(m-1) + \frac{m-2}{2m-3}\big)\, c_m^{OPT}. \end{aligned}$$

Altogether, we obtain the recurrence equation:

$$\rho^{heavy}(m) = \rho^{heavy}(m-1) + \frac{m-2}{2m-3}.$$

To analyze this relation, let $r_m = m - 2\rho^{heavy}(m)$. Then, $r_m - r_{m-1} = \frac{1}{2m-3}$, so that $r_m = r_2 + \sum_{k=2}^{m-1} \frac{1}{2k-1}$. Since $r_2 = 0$, $r_m = \mathcal{H}^{odd}(m-1) - 1$. □

The tightness of the bound established in Theorem 2.4 is discussed in Section 3.5.

3.5. Bad instances for additive cost functions. In this section we complement the previous results by showing that hub and sequential algorithms can perform rather poorly even when the cost function is additive. (Recall that for monotone submodular nonadditive functions, the bounds in Theorem 2.2 and Theorem 2.3 were already shown to be tight for all $m \geq 2$, even for the multi-hub and for the heaviest-first heuristic.)

Let us first consider the case $m = 3$. For MVA-3 with additive costs, Dokka et al. [5] established the validity and the tightness of the bounds established in Theorem 2.3 and Theorem 2.4, respectively. To see the former, observe that tightness of the bound $\rho^{seq}(3) = \frac{3}{2}$ follows from the instance depicted in Figure 1.1: indeed, for this instance, $c^{OPT} = 2$, whereas the sequential heuristic might find a solution with value 3.

To see that $\rho^{heavy}(3) = \frac{4}{3}$, consider the instance with $p = 3$, $V_1 = \{e_1, e_2, \mathbf{0}\}$, $V_2 = \{e_3, e_2, \mathbf{0}\}$, $V_3 = \{e_1, \mathbf{0}, e_3\}$. Its optimal value is $c^{OPT} = 3$, whereas H^{heavy} might produce first $H_2 = \{(e_1, e_3), (e_2, e_2), (\mathbf{0}, \mathbf{0})\}$, then $H_3 = \{(e_1, e_3, e_1), (e_2, e_2, \mathbf{0}), (\mathbf{0}, \mathbf{0}, e_3)\}$, with $c(H_3) = 4$.

An obvious improvement to heuristics H^{seq} and H^{heavy} would be to run H^{seq} for all possible permutations of the sets $V_1, \ldots, V_m$ in the first step, then to retain the best of the $m!$ feasible solutions found (see Bandelt et al. [1], Crama and Spieksma [4] for related "multiple-pass" heuristics). Interestingly, when $m = 3$, it follows again from the previous example that this multiple-pass heuristic (which involves solving six bipartite matching problems) has the same worst-case ratio as H^{heavy} (which only solves two matching problems). This observation also entails that the ratio $\rho(3) = \frac{4}{3}$ is tight for the iterative matching algorithm of Reda et al. [11].

Let us now turn to the general case $m \geq 3$ for additive cost functions. The ratio $\rho^{hhub}(m) = \frac{m}{2}$ is tight in this case for the heaviest-hub heuristic, as illustrated by the following example: Let $p = 2$ and $n = m$, let V_1 contain e_1 and let $V_2, \ldots, V_m$ contain e_2; all other vectors are $\mathbf{0}$. Then, $c^{OPT} = 2$ but the heaviest-hub heuristic may yield $c(H^{hub}(V_1)) = m$. For multi-hub, on the other hand, Dokka et al. [5] give an example showing that the performance ratio of the heuristic may be as bad a $\frac{m}{4}$, whereas Theorem 2.2 only proves the upper bound $\frac{m}{2}$. We do not know the exact approximation ratio of multi-hub for additive cost functions.

Dokka et al. [5] observed that the worst-case approximation ratio of the sequential heuristic can grow as fast as $\Omega(\sqrt{m})$ for certain instances with additive cost functions. We now strengthen this result by establishing a lower bound of the same order for the heaviest-first heuristic.

THEOREM 2.5. *There exists an infinite sequence of values of m such that the heaviest-first heuristic produces a feasible assignment with cost larger than $\frac{\sqrt{m}}{2} c_m^{OPT}$ on certain instances of WWI-m.*

Proof. Fix an arbitrary positive integer r. We are going to describe an instance of WWI-m with $m = r^2 + 1$ and $n = p = 2r$. In order to simplify the description of the instance, we label the input sets from V_0 to V_{r^2}. We write v_{ij} to denote the j^{th} vector of set V_i, $i = 0, \ldots, r^2$, $j = 1, \ldots, 2r$. The construction of the sets $V_0, V_1 \ldots, V_{r^2}$ is as follows. (An instance with $r = 3$ is displayed in Figure 3.1, and the corresponding heuristic and optimal solutions are illustrated in Figure 3.2.)

- In V_0, $v_{0j} = e_j$ for $j = 1, \ldots, r$, and $v_{0j} = \mathbf{0}$ for $j = r+1, \ldots, 2r$.
- For $i > 0$, write $i = (k-1)r + \ell$ with $k, \ell \in \{1, \ldots, r\}$. Then, in V_i,
  - for $j = 1, \ldots, r$, $v_{ij} = e_j$ if $j \neq \ell$ and $v_{i\ell} = \mathbf{0}$;
  - for $j = r+1, \ldots, 2r$, $v_{ij} = \mathbf{0}$ if $j \neq r+k$ and $v_{i,r+k} = e_{r+k}$.

For this instance, the optimal cost equals $2r$: for $j = 1, \ldots, 2r$, the j^{th} tuple of the optimal assignment simply collects all vectors e_j (note that there is at most one such e_j in each set V_i).

However, the heaviest-first heuristic may find a solution with cost $r^2 + r$ as follows: First, note that $c(V_i) = r$ for all i, so that H^{heavy} may consider the sets $V_0, V_1 \ldots, V_{r^2}$ in that order. When matching V_1 to V_0, H^{heavy} may assign the $(r+1)^{st}$ vector of V_1 to the first vector of V_0. In the next $r-1$ assignment stages, it assigns the $(r+1)^{st}$ vector of V_i $(i = 2, \ldots, r)$ to the tuple containing the i^{th} vector of V_0. Then, in the next r assignments, H^{heavy} assigns the $(r+2)^{nd}$ vector of V_i $(i = r+1, \ldots, 2r)$ to the tuple $i = 1, \ldots, r$ containing the i^{th} vector of V_0. Proceeding in this way yields a solution with cost $r^2 + r$. ∎

V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8	V_9
100000	000000	100000	100000	000000	100000	100000	000000	100000	100000
010000	010000	000000	010000	010000	000000	010000	010000	000000	010000
001000	001000	001000	000000	001000	001000	000000	001000	001000	000000
000000	000100	000100	000100	000000	000000	000000	000000	000000	000000
000000	000000	000000	000000	000010	000010	000010	000000	000000	000000
000000	000000	000000	000000	000000	000000	000000	000001	000001	000001

FIG. 3.1. *A bad instance for the heaviest-first heuristic with $r = 3$, $m = 10$*

OPT	H_{heavy}
100000	100111
010000	010111
001000	001111
000100	000000
000010	000000
000001	000000

FIG. 3.2. *Cost of optimal solution is 6; cost returned by H^{heavy} is 12*

4. WWI-3 is hard to approximate. As mentioned earlier, Reda et al. [11] have observed that WWI-m is NP-hard for $m \geq 3$. An explicit proof is found in Dokka et al. [5]. Our objective is now to strengthen this result by showing that WWI-3 does not admit a polynomial-time approximation scheme, unless P=NP.

We shall describe a reduction from 3-*bounded maximum* 3-*dimensional matching* (MAX-3DM-3) to WWI-3. An instance of MAX-3DM-3 consists of three pairwise disjoint sets X, Y, Z such that $|X| = |Y| = |Z| = q$, and of a set of triples $S \subseteq X \times Y \times Z$ such that every element of $X \cup Y \cup Z$ appears in at most three triples of S; let $|S| = s$. A *matching* in S is a subset $S' \subseteq S$ such that no element of $X \cup Y \cup Z$ appears in two triples of S'. The goal of the MAX-3DM-3 problem is to find a matching of maximum cardinality in S.

Kann [8] showed that MAX-3DM-3 is APX-hard. An instance of MAX-3DM-3 is called a *perfect instance* if its optimal solution consists of q triples that cover all elements of $X \cup Y \cup Z$ (that is, if S contains a feasible assignment). Petrank [10] proved that perfect instances of MAX-3DM-3 are hard to approximate, and that the existence of a polynomial-time approximation scheme for perfect instances would imply P=NP.

Now, consider an arbitrary perfect instance I' of MAX-3DM-3. We build a corresponding instance I of WWI-3 by using the gadget depicted in Figure 4.1, as explained next.

The instance I consists of three sets V_X, V_Y, V_Z, each of cardinality $q + 3s$. Each element e of each V_k, $k \in \{X, Y, Z\}$, is a 0-1 vector of length $6q + 4s$ containing exactly two nonzero elements. So, we can view each e as an edge in an undirected graph $G = (U, A)$ where U is a vertex set with cardinality $6q + 4s$ and A can be identified with $V_X \cup V_Y \cup V_Z$. The elements of U are

- x_1, x_2 for each $x \in X$
- y_1, y_2 for each $y \in Y$
- z_1, z_2 for each $z \in Z$
- x_t, y_t, z_t and u_t for each triple $t \in S$

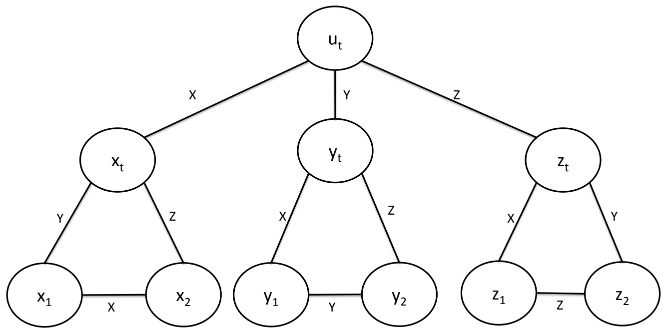

FIG. 4.1. *Gadget*

and the edges in $A = V_X \cup V_Y \cup V_Z$ are

- $(x_1, x_2) \in V_X$ for each $x \in X$ (element edges)
- $(y_1, y_2) \in V_Y$ for each $y \in Y$ (element edges)
- $(z_1, z_2) \in V_Z$ for each $z \in Z$ (element edges)
- $(x_1, x_t) \in V_Y$, $(x_2, x_t) \in V_Z$, $(y_1, y_t) \in V_X$, $(y_2, y_t) \in V_Z$, $(z_1, z_t) \in V_X$, $(z_2, z_t) \in V_Y$, for each $t \in S$ (gadget edges)
- $(x_t, u_t) \in V_X$, $(y_t, u_t) \in V_Y$, $(z_t, u_t) \in V_Z$ for each $t \in S$ (gadget edges).

We say that an element of V_X (V_Y, V_Z) is an X-edge (Y-edge, Z-edge). The subgraph induced by all gadget edges associated with a same triple t is called the *gadget* associated with t and is denoted by $g(t)$. Note that $g(t)$ contains three element edges.

Observe that a *feasible triple* for WWI-3 consists of an X-edge, a Y-edge and a Z-edge. A feasible triple of edges $T \subseteq A$ defines (and can be identified with) a subgraph (U_T, T) of G, where U_T is the subset of vertices covered by T. The cost of T is $|U_T|$. Note that a feasible triple is either

- a triangle K_3 with cost 3, or
- a claw $K_{1,3}$, or a path P_4, with cost 4, or
- disconnected with cost either 5 or 6.

We say for short that T is connected if (U_T, T) is connected.

A feasible assignment for I is a collection of $q+3s$ feasible triples covering all edges of G. We now collect some properties of feasible assignments for further reference.

LEMMA 4.1. *Let $M \subseteq V_X \times V_Y \times V_Z$ be a feasible assignment for I, with $|M| = q + 3s$.*
(1) M contains at most $3q$ triangles.
(2) The cost of M (and hence, the optimal value of I) is at least $q + 12s$.
(3) If the cost of M is $q + 12s$, then M contains $3q$ triangles, $3s - 2q$ additional connected triples, and no disconnected triples.
(4) If the cost of M is equal to $q + 12s + r$ $(r \geq 0)$, then M contains at least $3q - r$ triangles and at most r disconnected triples.

Proof. (1) We say M *covers* A', with $A' \subseteq A$, if all edges in A' are contained in M. Observe that M contains the same number of edges as A (namely, $3q + 9s$ edges), and hence, since M covers A, each edge of A must be covered exactly once. In particular, each element edge can be covered by at most one triangle, which implies that there are at most $3q$ triangles in M.

(2) The cost of M is equal to $3c_3 + 4c_4 + 5c_5 + 6c_6$, where c_k is the number of triples with cost equal to k, and $c_3 + c_4 + c_5 + c_6 = |M|$. There holds:

$$c(M) = 3c_3 + 4c_4 + 5c_5 + 6c_6 \tag{4.1}$$
$$\geq 3c_3 + 4(|M| - c_3 - c_5 - c_6) + 5(c_5 + c_6) \tag{4.2}$$
$$= -c_3 + (c_5 + c_6) + 4|M|. \tag{4.3}$$

Since $c_3 \leq 3q$ and $c_5 + c_6 \geq 0$, Eq. (4.3) implies that the cost of M is at least $-(3q) + 4(q + 3s) = q + 12s$.

(3) The previous reasoning shows that the cost of $|M|$ can be equal to $q + 12s$ only if $c_3 = 3q$ and $c_5 + c_6 = 0$.

(4) Intuitively, every missing triangle and every disconnected triple increases the cost of M by at least one unit with respect to the lower bound $q + 12s$, as expressed by the inequality (4.3). More formally, if $c_3 < 3q - r$, then Eq. (4.3) leads to

$$3c_3 + 4c_4 + 5c_5 + 6c_6 > -(3q - r) + 4|M| = q + 12s + r. \tag{4.4}$$

Similarly, if $c_5 + c_6 > r$, then Eq. (4.3) together with $c_3 \leq 3q$ imply

$$3c_3 + 4c_4 + 5c_5 + 6c_6 > -3q + r + 4|M| = q + 12s + r. \tag{4.5}$$

□

We are now ready to establish the relation between the solutions of I and I'

LEMMA 4.2. *If I' is a perfect instance of MAX-3DM-3, then the optimal value of I is $q + 12s$.*

Proof. If $t \in S$ is in the perfect matching, then use three triangles and the claw centered at u_t in the associated gadget $g(t)$. Otherwise, use three claws centered at x_t, y_t and z_t, respectively. Clearly, in the constructed solution for WWI-3 there are only triangles and claws with exactly $3q$ triangles. Hence, by Lemma 4.1 it follows that the cost of the solution is $q + 3s$. □

The converse statement will follow from Lemma 4.3 hereunder, with $\delta \geq 0$.

LEMMA 4.3. *Let $\delta \geq 0$ be a real number. If instance I has a feasible solution with cost at most $q + 12s + \delta q$, then instance I' possesses a matching with size at least $(1 - 6\delta)q$.*

Proof. Consider a feasible solution M for instance I with cost at most $q+12s+\delta q$. We call a gadget *damaged* (by M) if :
(Type (g)) at least one of its gadget edges is in a disconnected triple of M, or
(Type (e)) one of its element edges is not included in a triangle of M.
Equivalently, a gadget is *undamaged* if all its gadget edges are in connected triples of M and if all its element edges are in triangles of M.

We call an element edge *damaged* (by M) if it is not included in a triangle of M, or it is in a triangle contained in a damaged gadget. Equivalently, an element edge is *undamaged* if it is in a triangle contained in an undamaged gadget.

It follows from Lemma 4.1 that M contains at least $3q - \delta q$ triangles. Thus, at most δq element edges are not included in triangles.

Note that if an edge is damaged, then it is contained in a damaged gadget. Since I' is an instance of MAX-3DM-3, each element edge occurs in at most three gadgets. In particular, each damaged element edge can damage at most three gadgets, so that there are at most $3\delta q$ damaged gadgets of type (e).

Furthermore, Lemma 4.1 also implies that at most δq triples can be disconnected; these triples contain at most $3\delta q$ gadget edges, which can damage at most $3\delta q$ gadgets (damaged gadgets of type (g)).

Since each damaged gadget may yield at most three damaged element edges of type (ii), we find that, altogether there are at most $18\delta q$ damaged element edges, which leaves at least $3(1-6\delta)q$ undamaged element edges.
counting of δq potential damaged edges of type (i).)

Now, the main element of the proof of the lemma is the following claim:

CLAIM 4.4. *Every undamaged element edge, say (x_1, x_2), is in a triangle (x_1, x_2, x_t) from some undamaged gadget $g(t)$. We claim that the other two element edges in $g(t)$ are also included in triangles from $g(t)$.*

(Proof of claim.) To see this, consider one of the other element edges in $g(t)$, say (y_1, y_2). Since $g(t)$ is undamaged, (y_1, y_2) must be covered by a triangle contained in a gadget $g(t')$. Assume by contradiction that $t \neq t'$ (otherwise, we are done).

Again because $g(t)$ is undamaged, the X-edge (y_1, y_t) is in a connected triple T, which must necessarily contain the Y-edge (y_t, u_t) (indeed, at vertex y_1, (y_1, y_t) is only incident to X-edges and to the Y-edge (y_1, y_2) which is already covered by a triangle in $g(t')$; so, T must contain either the Y-edge (y_t, u_t) or the Z-edge (y_2, y_t); but the latter case implies the former one).

The previous reasoning applies similarly to (y_2, y_t), so that the claw $\{(y_1, y_t), (y_2, y_t), (y_t, u_t)\}$ must be in M.

This implies, in turn, that (x_t, u_t) and (z_t, u_t) must be in the same triple, which can only contain (z_2, z_t) as Y-edge. Thus, (z_1, z_t) must be in a triple together with (z_1, z_2), contradicting the hypothesis that (z_1, z_2) is undamaged. (End of claim.)

Hence the $3(1-6\delta)q$ undamaged element edges can be divided into groups of three that correspond to $(1-6\delta)q$ undamaged gadgets. Then the corresponding $(1-6\delta)q$ triples in instance I' form a matching. □

We are now ready for the main result of this section.

THEOREM 2.6. *WWI-3 is APX-hard even when all vectors in $V_X \cup V_Y \cup V_Z$ are 0–1 vectors with exactly two nonzero entries per vector.*

Proof. When we apply the reduction to a perfect instance I' of MAX-3DM-3, Lemma 4.2 yields $c^{OPT}(I) = q + 12s$ for the resulting instance I of WWI-3. A $(1+\epsilon)$-approximation algorithm for WWI-3 would imply that we can compute, in polynomial time, a solution of I with objective value at most equal to

$$(1+\epsilon)\, c^{OPT}(I) \leq q + 12s + 37\epsilon\, q$$

(here we have used $s \leq 3q$). Then Lemma 4.3 (with $\delta = 37\epsilon$) implies the existence of a matching of size at least $(1-222\epsilon)$ for instance I', and this matching can be found in polynomial time. Hence, a PTAS for WWI-3 would imply a PTAS for any perfect instance of 3-bounded MAX-3DM. □

5. Binary inputs and fixed p. In this section we consider again the *binary* MVA problem, that is, the special case of MVA where all vectors in $V_1 \cup \ldots \cup V_m$ are binary. We want to argue that the binary MVA problem can be solved in polynomial time when p is fixed.

For an instance of the binary MVA problem, we let, as in Theorem 2.5, v_{ij} denote the j^{th} vector in set V_i, $j = 1, \ldots, n$, $i = 1, \ldots, m$. Let $b_1, \ldots b_{2^p}$ be all distinct 0-1 vectors of length p, arbitrarily ordered, and consider a feasible m-tuple $(u^1, \ldots, u^m)$. We say that $(u^1, \ldots, u^m)$ is of *type t* if $u^1 \vee \ldots \vee u^m = b_t$.

We construct a mixed integer formulation of MVA featuring variables x_t:

$$x_t = \text{number of } m\text{-tuples of type } t \text{ in the assignment}, \; t = 1, \ldots, 2^p.$$

We also need assignment variables: for each $i = 1, \ldots, m; j = 1, \ldots, n; t = 1, \ldots, 2^p$,

$$z^i_{jt} = 1 \text{ if } v_{ij} \text{ is assigned to an } m\text{-tuple of type } t.$$

The formulation is now:

$$\min \sum_{t=1}^{2^p} c(b_t) x_t \tag{5.1}$$

$$\sum_{j:\; b_t \geq v_{ij}} z^i_{jt} = x_t \quad \text{for each } t = 1, \ldots, 2^p, i = 1, \ldots, m, \tag{5.2}$$

$$\sum_{t:\; b_t \geq v_{ij}} z^i_{jt} = 1 \quad \text{for each } j = 1, \ldots, n, i = 1, \ldots, m, \tag{5.3}$$

$$x_t \quad \text{integer} \quad \text{for each } t = 1, \ldots, 2^p, \tag{5.4}$$

$$z^i_{jt} \geq 0 \quad \text{for each } j = 1, \ldots, n, t = 1, \ldots, 2^p, i = 1, \ldots, m. \tag{5.5}$$

The objective function (5.1) minimizes the total cost. Constraints (5.2)-(5.3) are the familiar transportation constraints. Notice further that integrality of x_t implies integrality of z^i_{jt}.

LEMMA 5.1. *Formulation (5.1)-(5.5) is a correct formulation of the binary MVA problem.*

Proof. Consider a feasible solution of the binary MVA problem. This solution prescribes, for each binary vector v_{ij} in each set V_i, whether this vector should be assigned to an m-tuple of type t. This determines the x_t and z^i_{jt} values, which clearly satisfy constraints (5.2)-(5.5).

Conversely, consider x_t, z^i_{jt} values that satisfy (5.2)-(5.5). One can construct a feasible solution of MVA-m as follows: (1) Create a set X containing a copy of vector b_t for each $x_t > 0$. (2) For each $i = 1, \ldots, m$, construct a bipartite graph $G = (V_i \cup X, E)$ where vector v_{ij} of V_i is connected with vector b_t of X if $v_{ij} \leq b_t$. The values x_t, z^i_{jt} define a feasible solution of the transportation problem with supply equal to 1 for each vertex in V_i and demand equal to x_t for vertex t in X. (3) Construct m-tuples of vectors by assigning m vectors – one from each V_i – to the same m-tuple if they all are matched to same vector in X in the solution of the transportation problem (there may be several ways of performing this step; however, any way suffices). This yields a feasible solution of the MVA problem with value at most equal to $\sum_{t=1}^{2^p} c(b_t) x_t$. Hence, the optimal value of (5.1)-(5.5) is equal to the optimal value of the MVA problem. □

THEOREM 2.7. *Binary MVA can be solved in polynomial time for each fixed p.*

Proof. Lemma 5.1 shows that formulation (5.1)-(5.5) is correct. This formulation involves 2^p integer variables x_t, $O(mn2^p)$ continuous variables z^i_{jt}, and $O(m2^p + mn)$ constraints. When we fix p, this results in a fixed number of integer variables x_t, each of which takes at most $n + 1$ distinct values. Therefore, in order to find an optimal solution it is enough to check the feasibility of (5.2)-(5.5) for $O(n^{2^p})$ assignments of values to the x_t variables, and to choose the solution with the minimum cost. □

6. Conclusions. In this paper, we have considered the multi-dimensional vector assignment problem MVA-m and we have analyzed the performance of several polynomial-time heuristics for this problem in terms of their worst-case approximation ratio. We have also proved that the problem is APX-hard, even when $m = 3$. Among the main questions that remain open at this stage, let us mention the following ones:

1. What is the exact approximation ratio of the multi-hub heuristic in case of additive costs? We know that it lies between $m/4$ and $m/2$.

2. What is the exact approximation ratio of the heaviest-first sequential heuristic in case of additive costs? We know that it lies between $\Omega(\sqrt{m})$ and $O(m - \ln m)$.

3. Does there exist a polynomial-time algorithm with constant (i.e., independent of m) approximation ratio for MVA-m?

Acknowledgments. The project leading to these results was partially funded by OT Grant OT/07/015 and by the Interuniversity Attraction Poles Programme initiated by the Belgian Science Policy Office, Grant P7/36.

REFERENCES

[1] H. Bandelt, Y. Crama, and F.C.R. Spieksma, *Approximation algorithms for multi-dimensional assignment problems with decomposable costs*, Discrete Applied Mathematics, 49 (1994), pp. 25–50.

[2] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems*, SIAM, Philadelphia, 2009.

[3] R. Burkard, R. Rudolf, and G. Woeginger, *Three-dimensional axial assignment problems with decomposable cost coefficients*, Discrete Applied Mathematics, 65 (1996), pp. 123–139.

[4] Y. Crama and F.C.R. Spieksma, *Approximation algorithms for three-dimensional assignment problems with triangle inequalities*, European Journal of Operational Research, 60 (1992), pp. 273–279.

[5] T. Dokka, M. Bougeret, V. Boudet, R. Giroudeau, and F.C.R. Spieksma, *Approximation algorithms for the wafer to wafer integration problem*, in Approximation and Online Algorithms, Thomas Erlebach and Giuseppe Persiano, eds., vol. 7846 of Lecture Notes in Computer Science, 2013, pp. 286–297.

[6] T. Dokka, A. Kouvela, and F.C.R. Spieksma, *Approximating the multi-level bottleneck assignment problem*, Operations Research Letters, 40 (2012), pp. 282–286.

[7] D. Goossens, S. Polyakovskiy, F.C.R. Spieksma, and G. Woeginger, *Between a rock and a hard place: the two-to-one assignment problem*, Mathematical Methods of Operations research, 2 (2012), pp. 223–237.

[8] V. Kann, *Maximum bounded 3-dimensional matching is MAX SNP-complete*, Information Processing Letters, 37 (1991), pp. 27–35.

[9] C. Koufogiannakis and N. Young, *Greedy Δ-approximation algorithm for covering with arbitrary constraints and submodular cost*, Algorithmica, 66 (2013), pp. 113–152.

[10] E. Petrank, *The hardness of approximation: gap location*, Computational Complexity, 4 (1994), pp. 133–157.

[11] S. Reda, L. Smith, and G. Smith, *Maximizing the functional yield of wafer-to-wafer integration*, IEEE Transactions on VLSI Systems, 17 (2009), pp. 1357–1362.

[12] D. Simchi-Levi, X. Chen, and J. Bramel, *The Logic of Logistics: Theory, Algorithms, and Applications for Logistics and Supply Chain Management*, Springer Series in Operations Research, Springer, New York, second ed., 2005.

[13] E. Singh, *Wafer ordering heuristic for iterative wafer matching in W2W 3D-SIC's with diverse die yields*, in 3D-Test, First IEEE International Workshop on Testing Three-Dimensional Stacked Integrated Circuits, 2010. Poster 10.

[14] F.C.R. Spieksma, *Multi-index assignment problems: complexity, approximation, applications*, in L. Pitsoulis and P. Pardalos, eds., Nonlinear Assignment Problems, Algorithms and Applications (Kluwer Academic Publishers, (2000), pp. 1–12.

[15] F.C.R. Spieksma and G. Woeginger, *Geometric three-dimensional assignment problems*, European Journal of Operational Research, 91 (1996), pp. 611–618.

[16] Z. Svitkina and L. Fleischer, *Submodular approximation: Sampling-based algorithms and lower bounds*, SIAM Journal on Computing, 40 (2011), pp. 1715–1737.

[17] M. Taouil and S. Hamdioui, *Layer redundancy based yield improvement for 3D wafer-to-wafer stacked memories*, IEEE European Test Symposium, (2011), pp. 45–50.

[18] M. Taouil, S. Hamdioui, J. Verbree, and E. Marinissen, *On maximizing the compound yield for 3D wafer-to-wafer stacked ICs*, in IEEE International Test Conference, IEEE, ed., 2010, pp. 183–192.

[19] J. Verbree, E. Marinissen, P. Roussel, and D. Velenis, *On the cost-effectiveness of matching repositories of pre-tested wafers for wafer-to-wafer 3D chip stacking*, IEEE European Test Symposium, (2010), pp. 36–41.

[20] D. Williamson and D. Shmoys, *The Design of Approximation Algorithms*, Cambridge University Press, New York, 2011.