# On the Complexity of Broadcast Domination and Multipacking in Digraphs

Florent Foucaud[1,2], Benjamin Gras[2,3], Anthony Perez[2](✉), and Florian Sikora[4]

[1] Univ. Bordeaux, Bordeaux INP, CNRS, LaBRI, UMR5800, 33400 Talence, France
[2] Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, 45067 Orléans, France
anthony.perez@univ-orleans.fr
[3] Universität Trier, Fachbereich IV, Informatikwissenschaften, 54296 Trier, Germany
[4] Univ. Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016 Paris, France

**Abstract.** We study the complexity of the two dual covering and packing distance-based problems Broadcast Domination and Multipacking in digraphs. A *dominating broadcast* of a digraph $D$ is a function $f : V(D) \rightarrow \mathbb{N}$ such that for each vertex $v$ of $D$, there exists a vertex $t$ with $f(t) > 0$ having a directed path to $v$ of length at most $f(t)$. The cost of $f$ is the sum of $f(v)$ over all vertices $v$. A *multipacking* is a set $S$ of vertices of $D$ such that for each vertex $v$ of $D$ and for every integer $d$, there are at most $d$ vertices from $S$ within directed distance at most $d$ from $v$. The maximum size of a multipacking of $D$ is a lower bound to the minimum cost of a dominating broadcast of $D$. Let Broadcast Domination denote the problem of deciding whether a given digraph $D$ has a dominating broadcast of cost at most $k$, and Multipacking the problem of deciding whether $D$ has a multipacking of size at least $k$. It is known that Broadcast Domination is polynomial-time solvable for the class of all undirected graphs (that is, symmetric digraphs), while polynomial-time algorithms for Multipacking are known only for a few classes of undirected graphs. We prove that Broadcast Domination and Multipacking are both NP-complete for digraphs, even for planar layered acyclic digraphs of small maximum degree. Moreover, when parameterized by the solution cost/solution size, we show that the problems are respectively W[2]-hard and W[1]-hard. We also show that Broadcast Domination is FPT on acyclic digraphs, and that it does not admit a polynomial kernel for such inputs, unless the polynomial hierarchy collapses to its third level. In addition, we show that both problems are FPT when parameterized by the solution cost/solution size together with the maximum out-degree. Finally, we give for both problems polynomial-time algorithms for some subclasses of acyclic digraphs.

**Keywords:** Broadcast domination · Dominating set · Multipacking · Directed graphs · Parameterized complexity

L. Gąsieniec et al. (Eds.): IWOCA 2020, LNCS 12126, pp. 264–276, 2020.
https://doi.org/10.1007/978-3-030-48966-3_20

## 1 Introduction

We study the complexity of the two dual problems Broadcast Domination and Multipacking in digraphs. These concepts were previously studied only for undirected graphs (which can be seen as *symmetric* digraphs, where for each arc $(u, v)$, the symmetric arc $(v, u)$ exists). Unlike most standard packing and covering problems, which are of local nature, these two problems have more global features since the covering and packing properties are based on arbitrary distances. This difference makes them algorithmically very interesting.

**Broadcast Domination.** Broadcast domination is a concept modeling a natural covering problem in telecommunication networks: imagine we want to cover a network with transmitters placed on some nodes, so that each node can be reached by at least one transmitter. Already in his book in 1968 [20], Liu presented this concept, where transmitters could broadcast messages but only to their neighboring nodes. It is however natural that a transmitter could broadcast information at distance greater than one, at the price of some additional power (and cost). In this setting, for a given non-zero integer cost $d$, a transmitter placed at node $v$ covers all nodes within radius $d$ from its location. If the network is directed, it covers all nodes with a directed path of length at most $d$ from $v$. For a feasible solution, the function $f : V(G) \to \mathbb{N}$ assigning its cost to each node of the graph $G$ (a cost of zero means the node has no transmitter placed on it) is called a *dominating broadcast* of $G$, and the total cost $c_f$ of $f$ is the sum of the costs of all vertices of $G$. The *broadcast domination number* $\gamma_b(G)$ of $G$ is the smallest cost of a dominating broadcast of $G$. When all costs are in $\{0, 1\}$, this notion coincides with the well-studied Dominating Set problem. The concept of broadcast domination was introduced in 2001 (for undirected graphs) by Erwin in his doctoral dissertation [13] (see also [11,12] for some early publications on the topic), in the context of advertisement of shopping malls – which could nowadays be seen as targeted advertising via "influencers" in social networks. Note that in these contexts, directed arcs make sense since the advertisement or the influence is directed towards someone. The associated computational problem is as follows.

Broadcast Domination
- **Input**: A digraph $D = (V, A)$, an integer $k \in \mathbb{N}$.
- **Question**: Does there exist a dominating broadcast of $D$ of cost at most $k$?

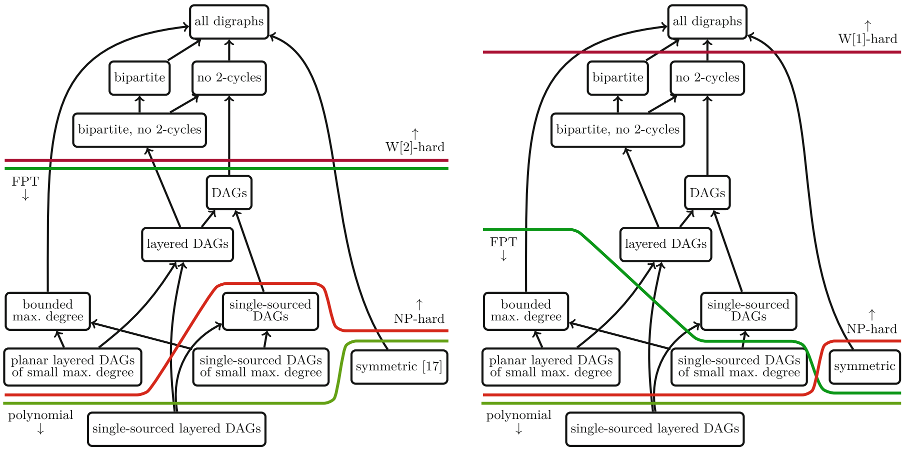
**Multipacking.** The dual notion for Broadcast Domination, studied from the linear programming viewpoint, was introduced in [5,24] and called *multipacking*. A set $S$ of vertices of a (di)graph $G$ is a *multipacking* if for every vertex $v$ of $G$ and for every possible integer $d$, there are at most $d$ vertices from $S$ at (directed) distance at most $d$ from $v$. The *multipacking number* $\mathrm{mp}(G)$ of $G$ is the maximum size of a multipacking in $G$. Intuitively, if a graph $G$ has a multipacking $S$, any dominating broadcast of $G$ will require to have cost at least $|S|$ to cover the vertices of $S$. Hence the multipacking number of $G$ is a lower bound to its broadcast domination number [5]. Equality holds for many graphs, such as strongly chordal graphs [4]. Consider the following computational problem.

MULTIPACKING
- **Input**: A digraph $D = (V, A)$, an integer $k \in \mathbb{N}$.
- **Question**: Does there exist a multipacking $S \subseteq V$ of $D$ of size at least $k$?

**Known Results.** In contrast with most graph covering problems, which are usually NP-hard, Heggernes and Lokshtanov designed in [17] (see also [21]) a sextic-time algorithm for BROADCAST DOMINATION in undirected graphs. This intriguing fact has motivated research on further algorithmic aspects of the problem. For general undirected graphs, no faster algorithm than the original one is known. A quintic-time algorithm exists for undirected series-parallel graphs [2]. An analysis of the algorithm for general undirected graphs gives quartic time when it is restricted to chordal graphs [17,18], and a cubic-time algorithm exists for undirected strongly chordal graphs [4]. The problem is solvable in linear time on undirected interval graphs [7] and undirected trees [4,9] (the latter was extended to undirected block graphs [18]).

Regarding MULTIPACKING, to the best of our knowledge, its complexity is currently unknown, even for undirected graphs (an open question posed in [24,25]). However, there exists a polynomial-time $(2 + o(1))$-approximation algorithm for all undirected graphs [1]. MULTIPACKING can be solved with the same complexity as BROADCAST DOMINATION for undirected strongly chordal graphs, see [4]. Improving upon previous algorithms from [22,24], the authors of [4] give a simple linear-time algorithm for undirected trees.

**Our Results.** We study BROADCAST DOMINATION and MULTIPACKING for directed graphs (digraphs), which form a natural setting for not necessarily symmetric telecommunication networks. In contrast with undirected graphs, we show that BROADCAST DOMINATION is NP-complete, even for planar layered acyclic digraphs (defined afterwards) of maximum degree 4. This holds for MULTIPACKING, even for planar layered acyclic digraphs of maximum degree 3, or acyclic digraphs with a single source and maximum degree 5. Moreover, when parameterized by the solution cost/solution size, we prove that BROADCAST DOMINATION is W[2]-hard (even for bipartite digraphs without directed 2-cycles) and MULTIPACKING is W[1]-hard. On the positive side, we show that BROADCAST DOMINATION is FPT on acyclic digraphs (DAGs for short) but does not admit a polynomial kernel for layered DAGs, unless the polynomial hierarchy collapses to its third level. Moreover, we show that both BROADCAST DOMINATION and MULTIPACKING are polynomial-time solvable for layered DAGs with a single source. We also design FPT algorithms for both problems when parameterized by the solution cost/solution size together with the maximum out-degree. The resulting complexity landscape is represented in Fig. 1. We start with some definitions in Sect. 2. We prove our results for BROADCAST DOMINATION in Sect. 3. The results for MULTIPACKING are presented in Sect. 4. We conclude in Sect. 5. Due to lack of space, we omit some proofs that can be found in the full version of the paper [14].



(a) Complexity landscape of Broadcast Domination

(b) Complexity landscape of Multipacking

**Fig. 1.** Complexity landscape of Broadcast Domination and Multipacking for some classes of digraphs. An arc from class $A$ to class $B$ indicates that $A$ is a subset of $B$. Parameterized complexity results are for parameter solution cost/solution size.

## 2 Preliminaries

**Directed Graphs.** We mainly consider digraphs, usually denoted $D = (V, A)$[1], where $V$ is the set of vertices and $A$ the set of arcs. For an arc $uv \in A$, we say that $v$ is an *out-neighbor* of $u$, and $u$ an *in-neighbor* of $v$. Given a subset of vertices $V' \subseteq V$, we define the digraph *induced* by $V'$ as $D' = (V', A')$ where $A' = \{uv \in A : \ u \in V' \textit{ and } v \in V'\}$. We denote such an induced subdigraph by $D[V']$. A directed path from a vertex $p_1$ to $p_l$ is a sequence $\{p_1, \ldots, p_l\}$ such that $p_i \in V$ and $p_i p_{i+1} \in A$ for every $1 \leqslant i < l$. When $p_1 = p_l$, it is a directed cycle. A digraph is *acyclic* whenever it does not contain any directed cycle as an induced subgraph. An acyclic digraph is called a *DAG* for short. The *(open) out-neighborhood* of a vertex $v \in V$ is the set $N^+(v) = \{u \in V : vu \in A\}$, and its *closed out-neighborhood* is $N^+[v] = N^+(v) \cup \{v\}$. We define similarly the open and closed in-neighborhoods of $v$ and denote them by $N^-(v)$ and $N^-[v]$, respectively. A *source* is a vertex $v$ such that $N^-(v) = \emptyset$. For the sake of readability, we always mean out-neighborhood when speaking of the *neighborhood* of a vertex. A DAG $D = (V, A)$ is *layered* when its vertex set can be partitioned into $\{V_0, \ldots, V_t\}$ such that $N^-(V_0) = \emptyset$ and $N^+(V_t) = \emptyset$ (vertices of $V_0$ and $V_t$ are respectively called *sources* and *sinks*), and $uv \in A$ implies that $u \in V_i$ and $v \in V_{i+1}$, $0 \leqslant i < t$. A *single-sourced layered DAG* is a layered DAG with only one source, that is, satisfying $|V_0| = 1$. A digraph is *bipartite* or *planar* if its underlying undirected graph has the corresponding property. Every layered

[1] Our reductions will also use undirected graphs, denoted $G = (V, E)$ with $V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$.

digraph is bipartite. Given two vertices $u$ and $v$, we denote by $d(u, v)$ the length of a shortest directed path from $u$ to $v$. For a vertex $v \in V$ and an integer $d$, we define the *ball of radius $d$ centered at $v$* by $B_d^+(v) = \{u \in V : \ d(v, u) \leqslant d\} \cup \{v\}$. Consider a dominating broadcast $f : V(D) \to \mathbb{N}$ on $D$. The set of *broadcast dominators* is defined as $V_f = \{v \in V : \ f(v) > 0\}$. For any set $S \subseteq V$ of vertices of $D$, we define $f(S)$ as the value $f(S) = \sum_{u \in S} f(u)$.

**Parameterized Complexity.** A *parameterized problem* is a decision problem together with a *parameter*, that is, an integer $k$ depending on the instance. A problem is *fixed-parameter tractable* (FPT for short) if it can be solved in time $f(k) \cdot |I|^c$ for an instance $I$ of size $|I|$ with parameter $k$, where $f$ is a computable function and $c$ is a constant. Given a parameterized problem $P$, a *kernel* is a function which associates to each instance of $P$ an equivalent instance of $P$ whose size is bounded by a function $h$ of the parameter. When $h$ is a polynomial, the kernel is said to be *polynomial*. An *FPT-reduction* between two parameterized problems $P$ and $Q$ is a function mapping an instance $(I, k)$ of $P$ to an instance $(f(I), g(k))$ of $Q$, where $f$ and $g$ are computable in FPT time with respect to parameter $k$, and where $I$ is a YES-instance of $P$ if and only if $f(I)$ is a YES-instance of $Q$. When moreover $f$ can be computed in polynomial time and $g$ is polynomial in $k$, we say that the reduction is a *polynomial time and parameter transformation* [3]. Both reductions can be used to derive conditional lower bounds: if a parameterized problem $P$ does not admit an FPT algorithm (resp. a polynomial kernel) and there exists an FPT-reduction (resp. a polynomial time and parameter transformation) from $P$ to a parameterized problem $Q$, then $Q$ is unlikely to admit an FPT algorithm (resp. a polynomial kernel). Both implications rely on certain standard complexity hypotheses; we refer the reader to the book [8] for details.

## 3 Complexity of Broadcast Domination

### 3.1 Hardness Results

**Theorem 1.** Broadcast Domination *is NP-complete, even for planar layered DAGs of maximum degree* 4.

**Theorem 2.** Broadcast Domination *parameterized by solution cost $k$ is W[2]-hard, even on bipartite digraphs without directed* 2*-cycles.*

*Proof (sketch).* We provide a reduction from the W[2]-hard Multicolored Dominating Set problem [6].

Multicolored Dominating Set
- **Input**: A graph $G = (V, E)$ with $V$ partitioned into sets $\{V_1, \ldots, V_k\}$, $k \in \mathbb{N}$.
- **Question**: Does there exist a dominating set $S$ of $G$ s.t. $|S \cap V_i| = 1$ for $1 \leqslant i \leqslant k$?

**Construction.** We build an instance $(D' = (V', A'), k')$ of Broadcast Domination as follows. To obtain the vertex set $V'$, we multiplicate $V$ into four sets $V^0$, $V^1$, $V^2$ and $V^3$ and we will have a set $M$ of *subdivided* vertices. The set $V^0 \cup V^1$ will induce an oriented complete bipartite graph, while $V^2 \cup V^3$ will induce a matching. For a vertex $v \in V$, $0 \leqslant i \leqslant 3$, its copy in $V^i$ is denoted $v^i$. We assume that $|V_i| \geqslant 2$, since otherwise one must take the only vertex in $V_i$. For each $1 \leqslant i \leqslant k$ we then add the following arcs:

- for every pair $v, w$ of distinct vertices of $V_i$, we add an arc from $v^0$ to $w^1$;
- for every $v \in V_i$, we add an arc from $v^1$ to $v^0$;
- for every $v \in V_i$, we add an arc from $v^2$ to $v^3$.

Moreover, for every edge $vw$ in $G$, we add an arc from $v^1$ to $w^2$, and we subdivide it once. The set of all subdivision vertices is called $M$. Finally, we set $k' = 3k$. Digraph $D'$ has no directed 2-cycles, and is bipartite with sets $V^0 \cup M \cup V^3$ and $V^1 \cup V^2$. One can check that $G$ has a multicolored dominating set of size $k$ if and only if $D'$ has a dominating broadcast of cost $3k$. □

### 3.2 Complexity and Algorithms for (Layered) DAGs

We now address the special cases of (layered) DAGs. Note that Dominating Set remains W[2]-hard on DAGs by a reduction from [23, Theorem 6.11.2]. In contrast, we now give an FPT algorithm for Broadcast Domination on DAGs that counterbalances the W[2]-hardness result.

**Theorem 3.** Broadcast Domination *parameterized by solution cost $k$ is FPT for DAGs.*

*Proof.* The proof relies on the following proposition, which is reminiscent of a stronger statement of Dunbar et al. [11] for undirected graphs (stating that there always exists an optimal dominating broadcast where each vertex is covered exactly once, which is false for digraphs).

**Proposition 4.** *For any digraph $D = (V, A)$, there exists an* optimal *dominating broadcast such that every broadcast dominator is covered by itself only.*

*Proof.* Let $f$ be an optimal dominating broadcast of $D$, and assume there exists two vertices $u, v \in V$ such that $f(v) \geqslant 1$ and $f(u) \geqslant d(u, v)$. In this case, $v$ is covered by both $u$ and itself. Notice that $d(u, v) + f(v) > f(u)$, since otherwise setting $f(v)$ to 0 would result in a better dominating broadcast. We claim that setting $f(u)$ to $d(u, v) + f(v)$ and $f(v)$ to 0 yields an optimal dominating broadcast $f_u$. Notice that since $d(u, v) + f(v) > f(u)$, any vertex covered by $u$ in $f$ is still covered in $f_u$. Similarly, any vertex covered by $v$ in $f$ is now covered by $u$ in $f_u$. Finally, we have $f(u) + f(v) \geqslant f_u(u) + f_u(v)$ since $f_u(u) = d(u, v) + f(v) \leqslant f(u) + f(v)$ and $f_u(v) = 0$, implying that the cost of $f_u$ is at most the cost of $f$. □

We can now prove Theorem 3. Let $D = (V, A)$ be a DAG. We consider the set $V_0$ of sources of $D$. Observe that for every $s \in V_0$, $f(s) \geqslant 1$ must hold. In particular, this means that $|V_0| \leqslant k$ (otherwise we return NO). We provide a branching algorithm based on this simple observation and on Proposition 4. We start with an initial broadcast $f$ consisting of setting $f(s) = 1$ for every vertex $s$ in $V_0$. At each step of the branching algorithm, we let $N_f = \cup_{v \in V_f} B^+_{f(v)}(v)$ be the set of currently covered vertices, and we consider the digraph $D_f = D[V \setminus N_f]$. Notice that $D_f$ is acyclic and hence contains a source $u$. Since every vertex of $N_f \setminus V_f$ is covered, we may assume by Proposition 4 that in the sought optimal solution, $u$ is only covered by itself or by a vertex in $V_f$. This means that one needs to branch on at most $k+1$ distinct cases: either setting $f(u) = 1$, or increasing the cost of one of its at most $k$ broadcasting ancestors in $V_f$. At every branching, the parameter $k$ decreases by 1, which ultimately gives an $O^*(2^{k \log k})$-time algorithm and completes the proof of Theorem 3. □

We will now complement the previous result by a negative one, which can be proved using a reduction from Hitting Set, defined as follows.

Hitting Set
- **Input**: A *universe* $U$ of elements, a collection $\mathcal{F}$ of subsets of $U$, an integer $k \in \mathbb{N}$.
- **Question**: Does there exist a *hitting set* $S$ of size $k$, that is, a set of $k$ elements from $U$ such that each set of $\mathcal{F}$ contains an element of $S$?

Hitting Set has no polynomial kernel when parameterized by $k+|U|$, unless the polynomial hierarchy collapses to its third level [10, Theorem 5.1].

**Theorem 5.** Broadcast Domination *parameterized by solution cost $k$ does not admit a polynomial kernel even on layered DAGs, unless the polynomial hierarchy collapses to its third level.*

**Theorem 6.** Broadcast Domination *is polynomial-time solvable on single-sourced layered DAGs.*

*Proof (sketch).* Let $D = (V, A)$ be a single-sourced layered DAG with layers $\{V_0, \ldots, V_t\}$. For the sake of readability, sets $V_i$ such that $|V_i| = 1$ are denoted $s_i$, for $0 \leqslant i \leqslant t$. Due to lack of space, the following claim is given without proof.

*Claim 1.* There always exists an optimal dominating broadcast $f$ of $D$ such that:

(i) $V_f \subseteq \bigcup_{i=0}^{t} s_i$
(ii) every $s_i \in V_f$, $0 \leqslant i \leqslant t$, covers exactly $B^+_l(s_i)$, where $l = j - i - 1$ and $j$ is the smallest index such that $j \geqslant i+2$ and $|V_j| = 1$.

We thus deduce a simple top-down procedure to compute an optimal dominating broadcast $f$. We start with setting $i = 0$. While there remain uncovered vertices, we let $f(s_i) = j - i - 1$ for the smallest value $j$ such that $s_j$ exists and $j \geqslant i+2$. In other words, $s_i$ will cover all vertices below it, until the closest vertex of the set $\bigcup_{j=0}^{t} s_j$ that is not a neighbour of $s_i$. We then carry on by setting $i = j$. By Claim 1, this leads to $f$ being an optimal dominating broadcast. □

We finally give an FPT algorithm for two parameters. By Theorems 1 and 2, such a result probably does not hold for each of them individually.

**Theorem 7.** Broadcast Domination *parameterized by solution cost $k$ and maximum out-degree is FPT.*

## 4 Complexity of Multipacking

We will need the following results to prove our results for Multipacking. The first one was proved for undirected graphs in [16].

**Lemma 8.** *Let $D = (V, A)$ be a digraph with a shortest directed path of length $3k - 3$ vertices. Then, $D$ has a multipacking of size $k$.*

*Proof.* It suffices to select every third vertex on the path. □

**Lemma 9.** *Let $D = (V, A)$ be a digraph. There always exists a multipacking of maximum size containing every source of $D$.*

The following lemma is the central result of both our polynomial-time algorithm (Theorem 14) and NP-completeness reduction (Theorem 12).

**Lemma 10.** *Let $D = (V, A)$ be a single-sourced layered DAG. There always exists a multipacking $S \subseteq V$ of maximum size such that for every $1 \leqslant i \leqslant t$, $|S \cap V_i| \leqslant 1$.*

*Proof.* Let $S \subseteq V$ be a multipacking of $D$ of maximum size. By definition of a multipacking, considering each ball centered at the source $s$, the following holds for every $1 \leqslant i \leqslant t$:

$$\left|S \cap \cup_{j=0}^{i} V_j\right| \leqslant i \tag{1}$$

We will prove the result inductively, by locally modifying $S$ in a top-down manner until it has the desired property. Let $j \geqslant 2$ be the smallest index such that $|S \cap V_j| \geqslant 2$, and $i < j$ be the largest index such that $|S \cap V_i| = 0$. Notice that $i$ is well-defined due to (1). Moreover, let $s_j^1$ and $s_j^2$ be two vertices of $S \cap V_j$.

**Case 1.** We assume first that $i = j - 1$. Let $u_i^1$ and $u_i^2$ be vertices of $V_i$ such that $u_i^1 s_j^1$ and $u_i^2 s_j^2$ belong to $A$ (note that in a layered DAG every non-source vertex has a predecessor in the previous layer). Since $S$ is a multipacking, we have $u_i^1 \neq u_i^2$ and neither $u_i^1$ nor $u_i^2$ is adjacent to both $s_j^1$ and $s_j^2$. Moreover, a vertex $s_{i-1}$ in $S \cap V_{i-1}$ cannot be adjacent to both $u_i^1$ and $u_i^2$, since otherwise we would have $\left|B_2^+(s_{i-1}) \cap S\right| > 2$. Moreover by minimality of the index $j$, there is at most one vertex of $S$ in $V_{i-1}$. Assuming w.l.o.g. that $u_i^1$ has no predecessor in $S$, the set $(S \setminus \{s_j^1\}) \cup \{u_i^1\}$ is a multipacking having the same size than $S$.

**Case 2.** We now consider the case where $i < j-1$. First, we will prove that there is a vertex $v_i$ in $V_i$ with no in-neighbor in $S$. If $S \cap V_{i-1} = \emptyset$, any vertex of $V_i$ can be chosen as vertex $v_i$. Otherwise, by choice of $j$ we have $|S \cap V_{i-1}| = 1$. Assume $S \cap V_{i-1} = \{s_{i-1}\}$. We claim that $s_{i-1}$ is not adjacent to every vertex of $V_i$.

Assume for a contradiction that this is the case. This means that $s_{i-1}$ is within distance $j-(i-1)$ of every vertex contained in $\cup_{l=i}^{j} V_l$. By the choice of indices $i$ and $j$ we know that $\cup_{l=i}^{j} V_l$ contains at least $j-(i-1)$ vertices from $S$, which in turn implies that $\left|B^+_{j-(i-1)}(s_{i-1}) \cap S\right| = j-(i-1)+1$, contradicting (1). Thus, there is a vertex $v_i$ in $V_i$ that has no in-neighbor in $S$. Now, we know by choices of $i$ and $j$ that $|S \cap V_p| = 1$ for $i < p < j$. Hence the set $(S \setminus \{s_{i+1}\}) \cup \{v_i\}$, where $\{s_{i+1}\} = S \cap V_{i+1}$, is a multipacking of $D$ having the same size than $S$. By iterating the above argument, we end up with $i = j-1$, in which case we can apply the argument from Case 1. Overall, after each iteration of Case 1, $j$ strictly increases. The procedure terminates when the value of $j$ reaches $t$. □

## 4.1 Hardness Results

**Theorem 11.** Multipacking *is NP-complete, even for planar layered DAGs of maximum degree* 3.

**Theorem 12.** Multipacking *is NP-complete on single-sourced DAGs of maximum degree* 5.

*Proof (sketch).* We provide a reduction from the NP-complete Independent Set problem [15]. We define the function $f : V(G) \to E$ such that for $v \in V(G), f(v) = e_i$ if and only if $e_i$ is the first edge in which $v$ appears (recall that $E = \{e_1, \ldots, e_m\}$). We create the digraph $D = (V', A)$ as follows (see Fig. 2):

$$\begin{aligned} V' =& \{u_i, v_i, w_i, x_i, y_i, z_i : \ 1 \leqslant i \leqslant m\} \cup V \cup \{s, p\} \\ A =& \{u_i w_i, u_i x_i : \ 1 \leqslant i \leqslant m\} \cup \{v_i x_i : \ 1 \leqslant i \leqslant m\} \cup \{w_i y_i, w_i z_i : \ 1 \leqslant i \leqslant m\} \bigcup \\ & \{z_i u_{i+1}, z_i v_{i+1} : \ 1 \leqslant i \leqslant m-1\} \cup \{x_i u, x_i v : \ 1 \leqslant i \leqslant m \text{ and } e_i = uv\} \bigcup \\ & \{u_i u : \ 1 \leqslant i \leqslant m \text{ and } f(u) = e_i\} \cup \{sp, pu_1, pv_1\} \end{aligned}$$

To conclude, one can see that $G$ has an independent set of size $k$ if and only if $D$ has a multipacking of size $k' = k + 2m + 1$. □

**Theorem 13.** Multipacking *parameterized by solution size $k$ is W[1]-hard.*

*Proof (sketch).* We provide an FPT-reduction from Multicolored Independent Set, which is W[1]-hard when parameterized by $k$ [8].

Multicolored Independent Set
- **Input**: A graph $G = (V, E)$ with $V$ partitioned into sets $\{V_1, \ldots, V_k\}$, $k \in \mathbb{N}$.
- **Question**: Does there exist an independent set $S$ of $G$ s.t. $|S \cap V_i| = 1$ for $1 \leqslant i \leqslant k$?

**Construction.** We construct an instance $(D = (V', A'), k')$ of Multipacking as follows. We consider the bipartite incidence graph of $G$, that is we add $V \cup E$ to $V'$. To construct $A'$, we add an arc from a vertex $e \in E$ to a vertex $v \in V$ if and only if $e$ contains $v$. We next group vertices of $E$ into $\binom{k}{2}$ sets $E_{i,j}$, $1 \leqslant i < j \leqslant k$

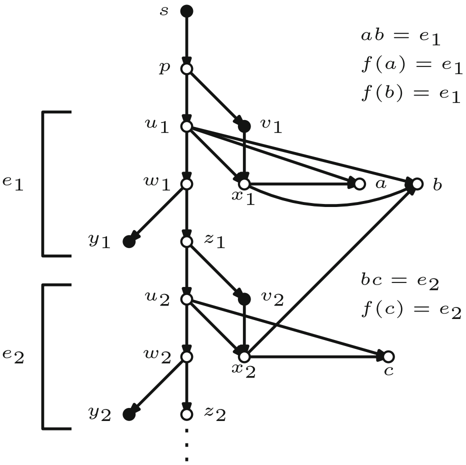


**Fig. 2.** Sketch of the construction in the proof of Theorem 12 for edges $e_0 = ab$ and $e_1 = bc$ with $f(a) = f(b) = e_0$ and $f(c) = e_1$.

according to the colors of their corresponding endpoints, and add every possible arc within each set $E_{i,j}$. We next duplicate the vertices of each set $V_i$ into a set $V_i'$ such that there is an arc from each vertex $v_i \in V_i$ to its corresponding copy $v_i'$ in $V_i'$. Finally, we add $k$ vertices $\{s_1, \ldots, s_k\}$ such that there is an arc from $s_i$ to every vertex of $V_i$. Notice in particular that the maximum finite distance is 3. To conclude, one can see that the graph $G$ has a multicolored independent set of size $k$ if and only if the digraph $D$ has a multipacking of size $k' = 2k + \binom{k}{2}$. □

### 4.2 Algorithms

**Theorem 14.** Multipacking *can be solved in linear time on single-sourced layered DAGs.*

*Proof.* Let $D = (V, A)$ be a single-sourced layered DAG. By Lemma 10, in every single-sourced layered DAG there is a multipacking of maximum size that is a maximum-size set of vertices with at most one vertex per layer such that two chosen vertices of consecutive layers are not adjacent. We give a polynomial-time bottom-up procedure to find such a set. At each step of the procedure, a layer $V_i$ is partitioned into a set of *active* vertices and a set of *universal* ones, denoted respectively $A_i$ and $U_i$. Our goal is to select exactly one vertex in each set of active vertices. We initiate the algorithm by setting $A_t = V_t$ and $U_t = \emptyset$. Now, for every $i$ with $0 \leqslant i < t$, we set $U_i = \{u \in V_i : A_{i+1} \subseteq N^+(u)\}$ and $A_i = V_i \setminus U_i$. In other words, $U_i$ contains the vertices of layer $V_i$ that are adjacent to all active vertices of $V_{i+1}$. During the procedure, if some layer $V_i$ satisfies $A_i = \emptyset$, we let $A_{i-1} = V_{i-1}$ and repeat this process until $V_0$ is reached.

To construct a maximum multipacking, we start from $V_0$, and for each $0 \leqslant i \leqslant t$ we pick a vertex $s_i$ in each non-empty set $A_i$ of active vertices. Every time a vertex $s_i$ is picked, we remove its closed neighborhood from $D$. By construction, every time a vertex $s_i$ is picked, there exists a vertex $s_{i+1} \in A_{i+1}$ such that $s_i s_{i+1}$ does not belong to $A$ (otherwise $s_i$ would belong to $U_i$). To prove the optimality of our algorithm, let $0 \leqslant i < t$ be such that $A_i = \emptyset$, and $j > i$ be the smallest integer greater than $i$ such that $V_j = A_j$. Such a $j$ exists since $A_t = V_t$.

*Claim 2.* Let $S$ be a multipacking with at most one vertex per layer. Then:

$$\left|S \cap \cup_{k=i}^{j} V_k\right| \leqslant j - i \tag{2}$$

*Proof.* Let $S$ be an optimal multipacking with at most one vertex per layer. Assume by contradiction that $\left|S \cap \bigcup_{k=i}^{j} V_k\right| > j - i + 1$, and call $s_k$ the vertex in $V_k \cap S$ for every $i \leqslant k \leqslant j$. We know that $s_i \in U_i$, and since every vertex in $A_{i+1}$ is an out-neighbor of $s_i$, then $s_{i+1} \in U_{i+1}$. By induction, for every $i \leqslant k \leqslant j$, we have $s_k \in U_k$, but $U_j = \emptyset$ by choice of $j$, leading to a contradiction.

Note that Claim 2 gives one less vertex than what Lemma 10 implies, and that it is the value reached by our algorithm, since for $i \leqslant k \leqslant j$, the only layer with $U_k = V_k$ is $V_i$. The sets of active and universal vertices can be constructed by standard graph searching, so the whole algorithm takes $O(|V| + |A|)$ time. □

**Theorem 15.** Multipacking *parameterized by solution size $k$ and maximum out-degree $d$ is FPT.*

*Proof.* Let $(D = (V, A), k)$ be an instance of Multipacking such that $D$ has maximum out-degree $d$. By Lemma 8, if $D$ has a shortest directed path of length $3k - 3$, we can accept the input (this can be checked in polynomial time). Thus, we can assume that the length of any shortest path is at most $3k-2$. If a vertex $u$ has a directed path to a vertex $v$, we say that $u$ *absorbs* $v$, and a set $S$ of vertices is *absorbing* if every vertex in $D$ is absorbed by some vertex of $S$. If $D$ has a set of $k$ vertices, no two of which are absorbed by some common vertex (*e.g.* a set of $k$ sources), we can accept, since this set forms a valid solution. Note that this property is satisfied by any minimum-size absorbing set $S$: indeed, if some vertex $w$ absorbs two vertices $u, v$ of $S$, we may replace them by $w$ and obtain a smaller absorbing set, a contradiction. We claim that we can find a minimum-size absorbing set in FPT time. Indeed, we can reduce this problem to Hitting Set (defined for the proof of Theorem 5) as follows. We let $U = V(D)$, and $\mathcal{F}$ contains a set $F_v$ for every vertex $v$, where $F_v$ comprises every vertex which absorbs $v$ (including $v$ itself). Because $D$ has out-degree at most $d$ and the length of any shortest path is at most $3k - 2$, every vertex of $U$ is contained in at most $d_U = \sum_{i=0}^{3k-2}(d-1)^i + 1$ sets of $\mathcal{F}$. Moreover, a set of vertices of $U = V(D)$ is a hitting set of $(U, \mathcal{F})$ if and only if it is an absorbing set of $D$. We can solve Hitting Set in FPT time when parameterized by $d_U$ and solution size $k$ [19], which proves the above claim. As mentioned before, if the obtained minimum-size absorbing set of $D$ has size at least $k$, since it forms a valid multipacking, we can accept. Otherwise, $D$ can be covered by $k-1$ balls of radius at most $3k-2$. Each ball has at most $\sum_{i=0}^{3k-2}(d-1)^i+1=d^{O(k)}$ vertices, so in total $D$ has at most $d^{O(k)}$ vertices and a brute-force algorithm is FPT. □

## 5 Conclusion

We have studied Broadcast Domination and Multipacking on various subclasses of digraphs, with a focus on DAGs. It turns out that they behave very differently than for undirected graphs. We feel that Multipacking is slightly more challenging. Indeed, some problems that we solved for Broadcast Domination are open for Multipacking. For example, it would be interesting to see whether Multipacking is FPT for DAGs, and whether it remains W[1]-hard for digraphs without directed 2-cycles. It is also unknown whether Multipacking is NP-hard on undirected graphs, as asked in [24,25]. On the other hand, we showed that Multipacking is NP-complete for single-sourced DAGs, but we do not know whether the same holds for Broadcast Domination. It is not difficult to show that both problems are FPT when parameterized by the vertex cover number. What about smaller parameters such as tree-width or DAG-width?

**Acknowledgements.** FF is partially funded by the IFCAM project "Applications of graph homomorphisms" (MA/IFCAM/18/39) and by the ANR project HOSIGRA (ANR-17-CE40-0022). FS is partially supported by the project ESIGMA (ANR-17-CE23-0010).

## References

1. Beaudou, L., Brewster, R.C., Foucaud, F.: Broadcast domination and multipacking: bounds and the integrality gap. Australas. J. Comb. **71**(1), 86–97 (2019)
2. Blair, J.R.S., Heggernes, P., Horton, S., Manne, F.: Broadcast domination algorithms for interval graphs, series-parallel graphs, and trees. In: Proceedings of the 35th South-eastern International Conference on Combinatorics, Graph Theory, and Computing. Congressus Numerantium, vol. 169, pp. 55–77 (2004)
3. Bodlaender, H.L., Thomassé, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. Theor. Comput. Sci. **412**(35), 4570–4578 (2011)
4. Brewster, R.C., MacGillivray, G., Yang, F.: Broadcast domination and multipacking in strongly chordal graphs. Discrete Appl. Math. **261**, 108–118 (2019)
5. Brewster, R.C., Mynhardt, C.M., Teshima, L.E.: New bounds for the broadcast domination number of a graph. Cent. Eur. J. Math. **11**, 1334–1343 (2013)
6. Casel, K.: Resolving Conflicts for lower-bounded clustering. In: 13th International Symposium on Parameterized and Exact Computation, IPEC 2018, Helsinki, Finland, 20–24 August 2018, pp. 23:1–23:14 (2018)
7. Chang, R.Y., Peng, S.L.: A linear-time algorithm for broadcast domination problem on interval graphs. In: Proceedings of the 27th Workshop on Combinatorial Mathematics and Computation Theory, pp. 184–188. Providence University Taichung, Taiwan (2010)
8. Cygan, M., et al.: Parameterized Algorithms. Springer, Switzerland (2015). https://doi.org/10.1007/978-3-319-21275-3

9. Dabney, J., Dean, B.C., Hedetniemi, S.T.: A linear-time algorithm for broadcast domination in a tree. Networks **53**(2), 160–169 (2009)
10. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and IDs. ACM Trans. Algorithms **11**(2), 13:1–13:20 (2014)
11. Dunbar, J.E., Erwin, D.J., Haynes, T.W., Hedetniemi, S.M., Hedetniemi, S.T.: Broadcasts in graphs. Discrete Appl. Math. **154**(1), 59–75 (2006)
12. Erwin, D.J.: Dominating broadcasts in graphs. Bull. ICA **42**, 89–105 (2004)
13. Erwin, D.J.: Cost domination in graphs. Ph.D. thesis, Western Michigan University (2001)
14. Foucaud, F., Gras, B., Perez, A., Sikora, F.: On the complexity of broadcast domination and multipacking in digraphs. CoRR abs/2003.10570 (2020). https://arxiv.org/abs/2003.10570
15. Garey, M.R., Johnson, D.S.: Computers and Intractability. Freeman, San Francisco (1979)
16. Hartnell, B.L., Mynhardt, C.M.: On the difference between broadcast and multipacking numbers of graphs. Utilitas Mathematica **94**, 19–29 (2014)
17. Heggernes, P., Lokshtanov, D.: Optimal broadcast domination in polynomial time. Discrete Math. **306**(24), 3267–3280 (2006)
18. Heggernes, P., Sæther, S.H.: Broadcast domination on block graphs in linear time. In: Hirsch, E.A., Karhumäki, J., Lepistö, A., Prilutskii, M. (eds.) CSR 2012. LNCS, vol. 7353, pp. 172–183. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-30642-6_17
19. Hermelin, D., Wu, X.: Weak compositions and their applications to polynomial lower bounds for kernelization. In: SODA 2012, Kyoto, Japan, 17–19 January 2012, pp. 104–113 (2012)
20. Liu, C.L.: Introduction to Combinatorial Mathematics. McGraw-Hill, New York (1968)
21. Lokshtanov, D.: Broadcast domination. Master's thesis, University of Bergen (2007)
22. Mynhardt, C.M., Teshima, L.E.: Broadcasts and multipackings in trees. Utilitas Mathematica **104**, 227–242 (2017)
23. Ordyniak, S., Kreutzer, S.: Width-measures for directed graphs and algorithmic applications. In: Quantitative Graph Theory, pp. 195–245. Chapman and Hall/CRC (2014)
24. Teshima, L.: Broadcasts and multipackings in graphs. Master's thesis, University of Victoria (2012)
25. Yang, F.: New results on broadcast domination and multipacking. Master's thesis, University of Victoria (2015)