Algorithmica (2021) 83:2651–2677
https://doi.org/10.1007/s00453-021-00828-5

On the Complexity of Broadcast Domination and Multipacking in Digraphs

Florent Foucaud[1,2,3] · Benjamin Gras[3,4] · Anthony Perez[3] · Florian Sikora[5]

Received: 4 May 2020 / Accepted: 9 April 2021 / Published online: 26 April 2021

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 $f : V(D) \to \mathbb{N}$ 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, and as well, by the vertex cover number. Finally, we give for both problems polynomial-time algorithms for some subclasses of acyclic digraphs.

A preliminary version of this paper was presented at the conference IWOCA'20 and appeared as [19]

Extended author information available on the last page of the article

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 [25], 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 [15] (see also [13, 16] 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 [6, 30] 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 i, there are at most i vertices from S at (directed) distance at most i 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 [6]. Equality holds for many graphs, such as strongly chordal graphs [5] and 2-dimensional square grids [1]. For undirected graphs, it is also known that $\gamma_b(G) \leqslant 2\,\mathrm{mp}\,(G) + 3$ [2] and it is conjectured that the additive constant can be removed. Consider the following computational problem.

Multipacking

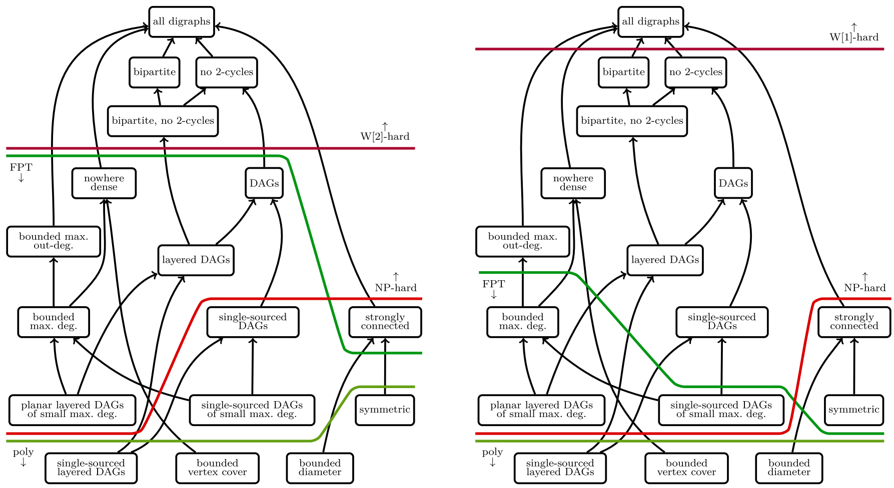
• **Input**:A digraph $D = (V, A)$, an integer $k \in \mathbb{N}$.

• **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 [23] (see also [26]) 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 [3]. An analysis of the algorithm for general undirected graphs gives quartic time when it is restricted to chordal graphs [23, 24], and a cubic-time algorithm exists for undirected strongly chordal graphs [5]. The problem is solvable in linear time on undirected interval graphs [9] and undirected trees [5, 11] (the latter was extended to undirected block graphs [24]). Note that when the dominating broadcast is required to be upper-bounded by some fixed integer $p \geqslant 2$, then the problem becomes NP-Complete [7] (for $p = 1$ this is Dominating Set).

Regarding Multipacking, to the best of our knowledge, its complexity is currently unknown, even for undirected graphs (an open question posed in [30, 31]). However, there exists a polynomial-time $(2 + o(1))$-approximation algorithm for all undirected graphs [2]. Multipacking can be solved with the same complexity as Broadcast Domination for undirected strongly chordal graphs, see [5]. Improving upon previous algorithms from [27, 30], the authors of [5] 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 later) of maximum degree 4 and maximum finite distance 2. This holds for Multipacking, even for planar layered acyclic digraphs of maximum degree 3 and maximum finite distance 2, 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 digraphs of maximum finite distance 2 or bipartite digraphs of maximum finite distance 6 without directed 2-cycles) and Multipacking is W[1]-hard (even for digraphs of maximum finite distance 3). 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 of maximum finite distance 2, 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 show that both problems are FPT when parameterized by the solution cost/solution size together with the maximum (out-)degree, and as well,



(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 (all considered digraphs are assumed to be weakly connected). 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

by the vertex cover number. Moreover it follows from a powerful meta-theorem from [22] that Broadcast Domination is FPT when parameterized by solution cost, on inputs whose underlying graphs belong to a nowhere dense class.

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.

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 $A' = \{uv \in A\ :\ u \in V' \text{ and } v \in V'\}$. We denote such an induced subgraph by $D[V']$. A directed path from a vertex p_1 to p_l is a sequence $\{p_1, \dots, p_l\}$ 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

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

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 $\{V_0, \dots, V_t\}$ 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 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 $B^+_d(v) = \{u \in V : d(v,u) \leqslant d\} \cup \{v\}$. The *eccentricity* of a vertex v in a digraph D is the largest (finite) distance between v and any vertex of D, denoted $\mathrm{ecc}(v) := \max_{u \in V} d(v,u)$. A digraph is *strongly connected* if for any two vertices u, v, there is a directed path from u to v, and *weakly connected* if its underlying undirected graph is connected. We will assume that all digraphs considered here are weakly connected (if not, each component can be considered independently). The *diameter* is the maximum directed distance $\max_{u,v \in V} d(u,v)$ between any two vertices u and v of G. If the digraph is not strongly connected, then the diameter is infinite. The *maximum finite distance* of a digraph D is the largest finite directed distance between any two vertices of G, denoted $\mathrm{mfd}(D) := \max_{u,v \in V, d(u,v) < \infty} d(u,v)$. 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* [4]. 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 [10] for details.

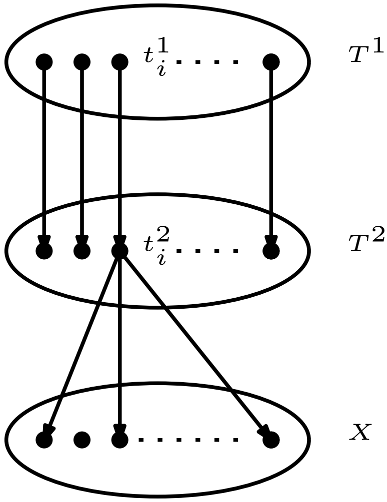


Fig. 2 Sketch of the DAG built in the construction of the proof of Theorem 1

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 *and maximum finite distance* 2.

Proof We will reduce from Exact Cover by 3-Sets, defined as follows. □

Exact Cover by 3-Sets

- **Input**: A set X of $3k$ elements (for some $k \in \mathbb{N}$), and a set $\mathcal{T} = \{t_1, \dots, t_n\}$ of triples from X.
- **Question**: Does there exist a subset $\mathcal{S}$ of k triples from $\mathcal{T}$ such that each element of X appears in (exactly) one triple in $\mathcal{S}$?

Exact Cover by 3-Sets is NP-hard even when the incidence bipartite graph of the input is planar and each element appears in at most three triples [14]. We will reduce any such instance $(X, \mathcal{T})$ of Exact Cover by 3-Sets to an instance $(D = (V', A'), k')$ of Broadcast Domination.

We create V' by taking two copies T^1, T^2 of $\mathcal{T}$ and one copy of X. More precisely, we let $T^j = \{t_i^j : 1 \leqslant i \leqslant n\}$ for $j \in \{1, 2\}$. We now add an arc from a vertex $t_i^1 \in T^1$ to its corresponding vertex t_i^2 in T^2, and from a vertex $t_i^2 \in T^2$ to all elements of X that are contained in t_i in $(X, \mathcal{T})$. See also Fig. 2. Formally:

$$A' = \{t_i^1 t_i^2 : 1 \leqslant i \leqslant n\} \bigcup \{t_i^2 x : x \in t_i, 1 \leqslant i \leqslant n\}$$

The construction can be done in polynomial time, and there is no cycle in D: arcs go either from T^1 to T^2 or from T^2 to X. Hence D is a layered DAG with three layers and thus, maximum finite distance 2. In fact D is obtained from the bipartite incidence graph of $(X, \mathcal{T})$ (which is planar and of maximum degree 3) reproduced on the vertices of $T^2 \cup X$, by adding pendant vertices (those from T^1) to those of T^2, orienting the arcs as required. Thus, the maximum degree of D is 4 and D is planar.

Claim 2 *The instance $(X, \mathcal{T})$ is a YES-instance if and only if the digraph D has a dominating broadcast of cost $k' = n + k$.*

Proof $\Rightarrow$ Given a solution $\mathcal{S}$ of $(X, \mathcal{T})$, set $f(t_i^1) = 2$ for all $t_i \in \mathcal{S}$, $f(t_i^1) = 1$ for each of the $n-k$ remaining vertices of T^1 and $f(v) = 0$ for all vertices of T^2 and X. For every vertex $t_i^2 \in T^2$, we have $d(t_i^1, t_i^2) = 1$. Similarly, for every vertex $x \in X$, $d(t_i^1, x) \leqslant 2$ holds for the vertex t_i^1 such that t_i is in $\mathcal{S}$ and contains x in $(X, \mathcal{T})$. Since every vertex t_i^1 of T^1 satisfies $f(t_i^1) \geqslant 1$, it is covered by itself, and it follows that f is a dominating broadcast of cost $n + k$.

$\Leftarrow$ Let us now consider the case where we are given a dominating broadcast for D of cost $n + k$. Note that since the maximum finite distance is 2, we can assume $f : V' \to \{0, 1, 2\}$. Remark that the vertices of T^1 are n sources. Therefore, any broadcast needs to set $f(t_i^1) \geqslant 1$ for each $t_i^1 \in T^1$, and this covers all vertices of T^1 and T^2. It remains to cover vertices of X with a cost of k, which can be done by setting $f(t_i^1) = 2$ for some vertices of T^1 and $f(t_j^2) = 1$ for some vertices of T^2. Notice that it is never useful to set $f(x) = 1$ for some vertex $x \in X$ as setting an additional cost of 1 to any $f(t_i^2)$ such that $t_i^2 \in A'$ is always better. Hence, the corresponding set of triples is a valid cover of $(X, \mathcal{T})$. (And it is an exact cover because there are $3k$ elements covered by k triples.) □

We next give two parameterized reductions for Broadcast Domination.

Theorem 3 *Broadcast Domination parameterized by solution cost k is W[2]-hard, even on digraphs of maximum finite distance 2, and on bipartite digraphs without directed 2-cycles of maximum finite distance 6.*

Proof We provide two reductions from the W[2]-hard Multicolored Dominating Set problem [8], defined as follows. □

Multicolored Dominating Set

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

We first provide a reduction that gives digraphs with directed 2-cycles.

Construction 1. We build an instance $(D = (V', A'), k')$ of Broadcast Domination as follows. To obtain the vertex set V', we duplicate V into two sets V^1

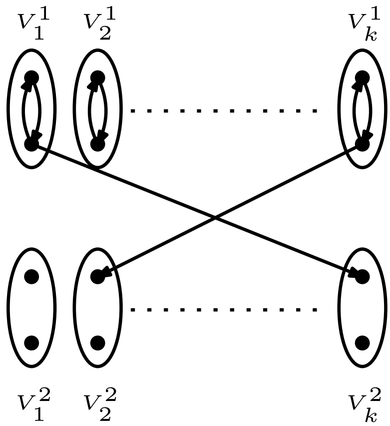


Fig. 3 Sketch of the built digraph D in the first reduction of the proof of Theorem 3

and V^2. Following the partition of V into k sets, we let $V^1 = \{V_1^1, \dots, V_k^1\}$ and $V^2 = \{V_1^2, \dots, V_k^2\}$. We then add every possible arc within V_i^1 ($1 \leqslant i \leqslant k$), and an arc from a vertex v in V^1 to each vertex of V^2 corresponding to a vertex from the closed neighborhood of v in G. Altogether, $V' = V^1 \cup V^2$. Finally, we set $k' = k$. See Figure 3 for an illustration. Clearly $\mathrm{mfd}\,(D) = 2$.

Claim 4 *The graph G has a multicolored dominating set of size k if and only if the digraph D has a dominating broadcast of cost k.*

Proof $\Rightarrow$ Let $S \subseteq V$ be a multicolored dominating set of size k of G. We claim that setting $f(v) = 1$ for every vertex v of V^1 such that the corresponding vertex v of G is in S, yields a dominating broadcast of cost k. To see this, notice that each vertex $v \in V_i^1$ ($1 \leqslant i \leqslant k$) with cost 1 covers V_i^1. Now, since these vertices of cost 1 form a dominating set in G, they cover the vertices of V^2 corresponding to their closed neighborhood in G, and hence f is a dominating broadcast.

$\Leftarrow$ Assume now that D has a dominating broadcast f of cost k. Notice first that any set V_i^1 ($1 \leqslant i \leqslant k$) must contain a vertex v such that $f(v) \geqslant 1$. Since f has cost k, this means that for every vertex $w \in V^2$, $f(w) = 0$. It follows that one needs to cover the vertices of V^2 using k vertices in V^1, which can be done only if there is a multicolored dominating set of size k in G. □

We now give a similar but more involved construction, which gives bipartite instances of maximum finite distance 6 and no directed 2-cycles.

Construction 2 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. Following the partition of V into k sets, for $0 \leqslant i \leqslant 3$, we let $V^i = \{V_1^i, \dots, V_k^i\}$. For a vertex $v \in V$, for $0 \leqslant i \leqslant 3$ its copy in V^i is denoted v^i. We assume that

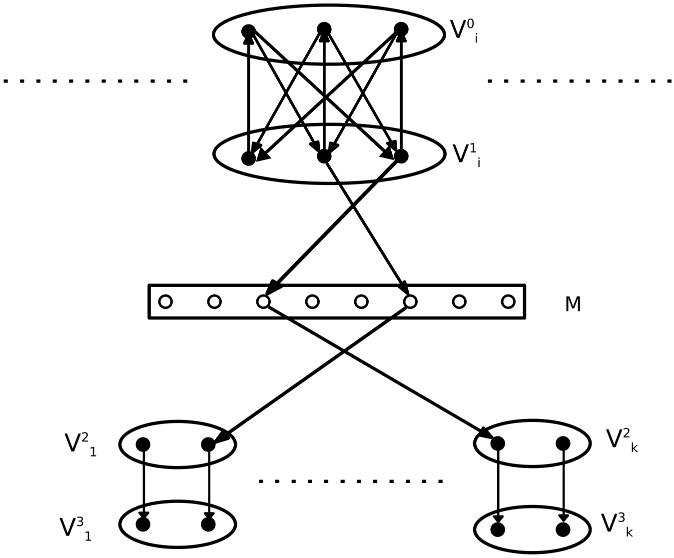


Fig. 4 Sketch of the built digraph D' in the second reduction of the proof of Theorem 3

$|V_i| \geqslant 2$, 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 $k' = 3k$. The construction is illustrated in Fig. 4. It is clear that $\mathrm{mfd}\,(D') = 6$ (shortest paths of length 6 exist from vertices of V^0 to vertices of V^3, but no longer shortest paths exist). The digraph has clearly no directed 2-cycles, and is bipartite with sets $V^0 \cup M \cup V^3$ and $V^1 \cup V^2$.

Claim 5 *The graph G has a multicolored dominating set of size k if and only if the digraph D' has a dominating broadcast of cost* $3k$.

Proof $\Rightarrow$ Let $S \subseteq V$ be a multicolored dominating set of size k of G. We claim that setting $f(v^1) = 3$ for every vertex v^1 of V^1 such that $v \in S$ yields a dominating broadcast of cost $3k$. To see this, notice first that each such vertex belonging to V_i^1, $1 \leqslant i \leqslant k$, covers the whole set $V_i^0 \cup V_i^1$ and all the vertices of M with an in-neighbor in V_i^1. Now, each vertex v^1 with $v \in S$ covers (at distance 3) each vertex w^2 and w^3 of $V^2 \cup V^3$ such that w is in the closed neighborhood of v in G. Since S is dominating, f is thus a dominating broadcast.

⇐ Assume now that D' has a dominating broadcast f of cost $3k$. First, we claim that for every i with $1 \leqslant i \leqslant k$, we need a total cost of 3 for the vertices in $V_i^0 \cup V_i^1$. Indeed, for a vertex $v \in V_i$,f if $f(v^0) = 2$, v^0 does not cover V^1. If $f(v^1) = 2$, no vertex w^0 with $w \neq v$ and $w \in V_i$ is covered. Clearly, we cannot cover the vertices of $V_i^0 \cup V_i^1$ with two vertices broadcasting at cost 1. Thus, we can assume that there is a total cost of exactly 3 on the vertices of $V_i^0 \cup V_i^1$ for $1 \leqslant i \leqslant k$, and each vertex v of $V^2 \cup V^3 \cup M$ satisfies $f(v) = 0$. We now prove that there exists a vertex v of $V_i^0 \cup V_i^1$, $1 \leqslant i \leqslant k$ such that $f(v) = 3$. First, since a vertex v^1 of V_i^1 with $f(v^1) = 2$ does not cover the vertices of V_i^0 (except for v^0), it is not possible to cover $V_i^0 \cup V_i^1$ with a cost of 1 on another vertex. Similarly, since a vertex v^0 of V_i^0 with $f(v^0) = 2$ does not cover v^1, an additional cost of 1 cannot cover v^1 and all vertices of M that are out-neighbors of vertices in V_i^1. Similarly, we cannot have three vertices with a broadcasting cost of 1 each. Thus, there is a vertex of $V_i^0 \cup V_i^1$ with a broadcast cost of 3. Notice that it cannot be a vertex of V_i^0, since otherwise the out-neighbors of V_i^1 in M are not covered. Thus there is a vertex v^1 in V_i^1 with $f(v^1) = 3$. This covers, in particular, all the vertices w^2, w^3 of $V_i^2 \cup V_i^3$ such that vw is an edge in G, and no other vertex of $V_i^2 \cup V_i^3$. It follows that the set of vertices v of V such that $f(v^1) = 3$ forms a dominating set of G of size k. Thus, the proof is complete. □

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 [29, 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 6 *Broadcast Domination parameterized by solution cost k can be solved in FPT time $2^{O(k \log k)} n^{O(1)}$ time for DAGs of order n.*

The proof relies on the following proposition, which is reminiscent of a stronger statement of Dunbar et al. [13] for undirected graphs (stating that there always exists an optimal dominating broadcast where each vertex is covered exactly once, which is false for digraphs). Recall that the set of broadcast dominators is denoted V_f and contains all vertices v such that $f(v) > 0$.

Proposition 7 *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 exist 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 6.

Proof of Theorem 6 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 7. 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 7 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 6. □

We will now complement the previous result by a negative one, which can be proved using a reduction similar to the one in Theorem 1 but 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 ?

Theorem 8 *Broadcast Domination parameterized by solution cost k does not admit a polynomial kernel even on layered DAGs of maximum finite distance* 2, *unless the polynomial hierarchy collapses to its third level.*

Proof We provide a reduction from Hitting Set. It is shown in [12, Theorem 5.1] that if Hitting Set admits a polynomial kernel when parameterized by $|U|+k$ (a variant called Small Universe Hitting Set), then the polynomial hierarchy collapses to its third level.

We do the same reduction as the one from Exact Cover by 3-Sets from Theorem 1, except that the set T of triples is replaced by U and the set X of elements is replaced by $\mathcal{F}$. We again obtain a DAG with three layers and maximum finite distance 2. The solution cost for the instance of Broadcast Domination is set to $|U|+k$, and the proof of validity of the reduction is the same.

Since this is clearly a polynomial time and parameter transformation, the result follows. □

We now show that Broadcast Domination can be solved in polynomial time on special kinds of DAGs.

Theorem 9 *Broadcast Domination is linear-time solvable on single-sourced layered DAGs.*

Proof Let $D = (V, A)$ be a single-sourced layered DAG with layers $\{V_0, \dots, V_t\}$. For the sake of readability, sets V_i with $\left|V_i\right| = 1$ are denoted by $\{s_i\}$, for $0 \leqslant i \leqslant t$.

Our algorithm relies on the following structural properties of some optimal dominating broadcasts for single-sourced layered DAGs. □

Claim 10 *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* $\left|V_j\right| = 1$.

Proof Let f be an optimal dominating broadcast of D having the properties of Proposition 7.

Property (*i*). Let $0 \leqslant i < j \leqslant t$ be indices such that s_i covers all layers up to V_{j-1}, where j is the smallest index such that $\left|V_j\right| \geqslant 2$ and $f(V_j) > 0$. Notice that i exists since $f(s_0) \geqslant 1$. If j does not exist, then we are done. We hence assume j is well-defined. By the choice of i, we know that $f(s_i) = d(s_i, V_{j-1}) = j - i - 1$. Let v_j^1 and v_j^2 be two vertices of V_j. We first consider the case where $\left|V_f \cap V_j\right| = 1$ and assume w.l.o.g. that $f(v_j^1) \geqslant 1$. This means that v_j^2 must be covered by s_i, which in turn covers v_j^1, which is impossible by the choice of i (and the definition of f). We thus have $\left|V_j \cap V_f\right| \geqslant 2$, and assume that $f(v_j^1) \geqslant 1$ and $f(v_j^2) \geqslant 1$. Assume first that $j = t$. In that case, s_i covers all vertices in $\cup_{a=i}^{t} V_{a-1}$, and hence setting $f(v_j^1) = f(v_j^2) = 0$ and increasing $f(s_i)$ by 1 leads to a dominating broadcast of smaller cost, a contradiction.

We thus assume $j < t$. We claim that the dominating broadcast f_i defined by setting:

$$\begin{cases} f_i(s_i) & = f(s_i) + max\{f(v_j^1), f(v_j^2)\} + 1 \\ f_i(v_j^1) & = 0 \\ f_i(v_j^2) & = 0 \\ f_i(v) & = f(v) \qquad \forall\, v \neq \{s_i, v_j^1, v_j^2\} \end{cases}$$

is optimal. Notice first that $c_{f_i}(V) \leqslant c_f(V)$. Now, every vertex covered by both v_j^1 and v_j^2 is covered by s_i: indeed, since s_i corresponds to a layer with a single vertex, it has a directed path of length $d(s_i, v_{j-1}) + max\{f(v_j^1), f(v_j^2)\} + 1$ to every vertex covered by both v_j^1 and v_j^2, which are thus still covered.

Property (*ii*). Suppose that f satisfies Property (i). Assume there exists two vertices s_i and s_j with $0 \leqslant i < i+1 < j \leqslant t$ such that $f(s_i) \geqslant d(s_i, s_j)$. In other words, vertex s_i covers vertex s_j. Consider that i is chosen to be minimum with this property. Notice that since f fulfills the properties of Proposition 7, we have $f(s_j) = 0$. We distinguish two cases:

- If $f(s_i) > d(s_i, s_j)$, consider the dominating broadcast f_i obtained from f by setting $f_i(s_i) = d(s_i, s_j) - 1$ and $f_i(s_j) = f(s_i) - d(s_i, s_j)$. Notice that every vertex covered by s_i in f is still covered in f_i: indeed, s_i covers all vertices up to V_{j-1}, and vertices in higher layers are now covered by s_j, which covers itself. By construction, we have:

$$
\begin{aligned}
c_{f_i}(V) &= c_{f_i}(V \setminus \{s_i, s_j\}) + f_i(s_i) + f_i(s_j)\\
&= c_{f_i}(V \setminus \{s_i, s_j\}) + d(s_i, s_j) - 1 + f(s_i) - d(s_i, s_j)\\
&< c_f(V \setminus \{s_i, s_j\}) + f(s_i)\\
&< c_f(V)
\end{aligned}
$$

  the last inequality holding since $f(s_j) = 0$. This leads to a contradiction since f is an optimal dominating broadcast. Thus this case does not happen.
- We may hence assume that $f(s_i) = d(s_i, s_j)$. Since f fulfills the properties of Proposition 7 and Property (*i*), V_{j+1} has to dominate itself, and thus s_{j+1} must exist, unless $j = t$. Consider the dominating broadcast f_i obtained from f by setting $f_i(s_i) = d(s_i, s_j) - 1$, $f_i(s_j) = 1 + f(s_{j+1})$ and $f_i(s_{j+1}) = 0$. If $j = t$ we consider that $f(s_{j+1}) = 0$. Notice that every vertex covered by s_{j+1} in f is covered by s_j in f_i. We have:

$$
\begin{aligned}
c_{f_i}(V) &= c_{f_i}(V \setminus \{s_i, s_j, s_{j+1}\}) + f_i(s_i) + f_i(s_j) + f_i(s_{j+1})\\
&= c_{f_i}(V \setminus \{s_i, s_j, s_{j+1}\}) + d(s_i, s_j) - 1 + f(s_{j+1}) + 1\\
&= c_f(V \setminus \{s_i, s_j, s_{j+1}\}) + f(s_i) + f(s_{j+1})\\
&= c_f(V)
\end{aligned}
$$

  the last equality holding since $f(s_j) = 0$.

We have thus obtained a dominating broadcast f_i of the same cost as f, still satisfying Property (i) and Proposition 7, but where every vertex s_l with $l \leqslant i$ satisfies (ii). If f_i still does not satisfy (ii), we reiterate this process (each time, with increasing value of i) until (ii) is satisfied for all vertices. This concludes the proof of Claim 10. □

We thus deduce a simple top-down procedure to compute an optimal dominating broadcast f. We initiate our solution by 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 10, this process leads to the construction of an optimal dominating broadcast. □

3.3 Algorithms for Structural Parameters and Structured Classes

We now give some algorithms for structural parameters and classes.

Theorem 11 *Broadcast Domination can be solved in time $\delta^\delta n^{O(\delta)}$ for digraphs of order n and diameter δ.*

Proof To solve Broadcast Domination by brute-force, we may try all the subsets of size k, and for each subset, try all possible k^k broadcast functions. But we can assume that $k \leqslant \delta$, since a single vertex with cost δ covers all the digraph. □

We next consider jointly two parameters. Recall that by Theorems 1 and 3, such a result probably does not hold for each of them individually.

Theorem 12 *Broadcast Domination parameterized by solution cost k and maximum out-degree d can be solved in FPT time $k^k 2^{d^{O(k)}} n^{O(1)}$ on digraphs of order n.*

Proof Let $(D=(V,A),k)$ be an instance of Broadcast Domination such that D has maximum out-degree d. Consider a dominating broadcast f of cost k. A vertex v with $f(v)=i>0$ covers all vertices of its ball of radius i, which has size at most $\sum_{j=0}^{i}(d-1)^j+1 \leqslant id^i+1$. Thus, if the input has more than $n=k(k+1)d^k$ vertices, we can reject. Otherwise, a simple brute-force algorithm over all possible 2^n possible subsets and, given a subset, all k^k possible broadcasts, is FPT. The result follows. □

Next, we consider the *vertex cover number* of input digraphs, that is, the smallest size of a set of vertices that intersects all arcs (or, in other words, the vertex cover number of the underlying undirected graph).

Theorem 13 *Broadcast Domination parameterized by the vertex cover number c of the input digraph of order n can be solved in FPT time $2^{c^{O(c)}} n^{O(1)}$.*

Proof Let $(D=(V,A),k)$ be an instance of Broadcast Domination and let $S \subseteq V$ be a vertex cover of D of size c. Let us partition the set $V \setminus S$ (which induces no arcs) into equivalence classes $C_1,\ldots,C_t$ according to their in- and out-neighborhoods in S: two vertices are in the same class if and only if they have the same sets of in- and out-neighbors. There are $t \leqslant 2^{2c}$ such classes.

For a given class, any broadcasting vertex out of the class either covers all vertices in the class, or none. Similarly, a vertex broadcasting at radius r inside the class covers the same set of vertices outside the class as any other vertex from the class would. Hence, we may assume that at most one selected vertex b_i per class C_i broadcasts with $f(b_i) > 1$. We can assume that the other vertices v in the class either all satisfy $f(v) = 0$ or all $f(v) = 1$ (the latter may happen if they all need to cover themselves, for example if they are all sources). Moreover, $\mathrm{mfd}\,(D) \leqslant 2c + 1$ since every shortest path is either contained in S or has to alternate between a vertex of S and one of $V \setminus S$, but cannot have repeated vertices.

Hence, for each equivalence class C_i, we have $2 \times (2c + 1)$ choices: $2c + 1$ for the value of $f(b_i)$, and two possibilities for the other vertices of C_i. Similarly, for each vertex of S, we have $2c + 1$ possible broadcast values. In total, this gives $(t + c)^{O(c)} = 2^{c^{O(c)}}$ different possible dominating broadcasts, and each of them can be checked in polynomial time. □

We next see how to apply the following powerful theorem from [22], to show that Broadcast Domination is FPT for any class of digraphs whose underlying graph is *nowhere dense*. We will not give a definition of nowhere dense graph classes, and refer to the book [28] instead. Such classes include planar graphs, graphs excluding a fixed (topological) minor, graphs of bounded degree, graph classes of bounded expansion, etc.

Theorem 14 [[22]] *Let $\mathcal{C}$ be a nowhere dense graph class. There exists ϵ such that, given as inputs a graph $G \in \mathcal{C}$ and a first-order logic graph property φ, the problem of deciding whether G satisfies φ can be solved in time $f(|\varphi|)|G|^{1+\epsilon}$, that is, it is FPT when parameterized by the length of φ.*

Corollary 15 *For every fixed nowhere dense graph class $\mathcal{C}$, Broadcast Domination parameterized by the solution cost of the input digraph is FPT for inputs whose underlying graphs are in $\mathcal{C}$.*

Proof We want to show that for fixed parameter value k of the solution cost, Broadcast Domination can be expressed in first-order logic by a formula whose length is bounded by a function of k, and apply Theorem 14.

To do so, we extend the classic approach for defining k -Dominating Set in first-order logic (see e.g. [28, Chapter 18.4]).

We will use the property $dp(x, y, i)$, stating that there is a directed path from x to y of length at most i. This can be expressed in first-order logic for fixed i. To this end, we state that either $x = y$, or there is an arc from x to y, or there is a directed path of length 2 from x to y (i.e. there exists a vertex z, x has an arc to z, and z, an arc to y), $\cdots$ or there exists a directed path of length i from x to y.

Let $V_1, \dots, V_k$ denote the sets of broadcast dominators of a potential dominating broadcast f, where V_i contains the vertices broadcasting at radius i. The union $V_f = \bigcup_{i=1}^{k} V_i$ has size at most k, and since k is considered to be fixed, we can "guess" the size of each set V_i. To this end, we let $v_i^1, \dots, v_i^k$ be the potential vertices of V_i. For a given partition Π of V_f into sets $V_1, \dots, V_k$, we can express the fact that a given vertex x is dominated by f as the formula $dom_{\Pi}(x, v_1^1, \dots, v_k^k)$, which is composed of the conjunction of all formulae of type $dp(v_i^j, x, i)$, where in $\Pi, 1 \leqslant j \leqslant |V_i|$.

Now, given the set $\Pi_1, \ldots, \Pi_t$ of all partitions of V_f into sets $V_1, \ldots, V_k$ (note that $t \leqslant k^k$), the first-order formula for Broadcast Domination is given as

$$\exists v_1^1 \ldots \exists v_k^k, \left(\forall x \in G, dom_{\Pi_1}(x, v_1^1, \ldots, v_k^k)\right) \vee \ldots \vee \left(\forall x \in G, dom_{\Pi_t}(x, v_1^1, \ldots, v_k^k)\right).$$

□

We remark that Corollary 15 does not imply Theorem 12, indeed there are digraph classes of bounded maximum out-degree whose underlying graphs do not form a nowhere dense class of graphs. For example, every d-degenerate graph can be oriented so as to have maximum out-degree at most d. Indeed, a graph is *d-degenerate* if its vertices can be ordered $v_1, \ldots, v_n$ such that for $2 \leqslant i \leqslant n$, v_i has at most d neighbors among $v_1, \ldots, v_{i-1}$. Thus, orienting every edge v_iv_j with $i < j$ from v_j to v_i produces a digraph of maximum out-degree at most d. However, for every d, the class of d-degenerate graphs is not nowhere dense [22, 28].

4 Complexity of Multipacking

We will need the following results to prove our results for Multipacking.

Lemma 16 *Let $D = (V, A)$ be a digraph. There exists a multipacking of maximum size containing every source of D.*

Proof Let $D = (V, A)$ and let $S \subseteq V$ be a multipacking of D of size at least k.

Assume there exists a source $s \in V$ that does not belong to S. We say that a vertex $v \in V$ is *full* w.r.t. S whenever there exists an integer $p > 0$ such that $|B_p^+(v) \cap S| = p$. Assume first that s is not full w.r.t. S. In that case, one can safely add s to the multipacking S and obtain a new solution of size at least k. Hence, we now consider the case where s is full. Notice that if s is full at distance 1 (i.e. $\left|B_1^+(s) \cap S\right| = 1$), then the set $(S \setminus \{u\}) \cup \{s\}$ is a multipacking of size at least k (recall that s is a source), and thus we are done.

We hence assume that this is not the case. Let $1 \leqslant i \leqslant \operatorname{ecc}(s)$ be the smallest integer such that $|B_i^+(s) \cap S| < i$ and $|B_{i+1}^+(s) \cap S| = i + 1$. Notice that $|N^+[s] \cap S| = 0$, since otherwise s would be full at distance 1. In particular, since s is full at distance $i + 1$, this means that $|B_{i+1}^+(s) \cap S| \geqslant 2$. Let u be any vertex of $B_{i+1}^+(s) \cap S$. We claim that the set $S' = (S \setminus \{u\}) \cup \{s\}$ is a multipacking of D. First, it is clear that $|S'| = |S|$. Now, since s is a source and $|N^+(s) \cap S| = 0$, adding s to the multipacking cannot violate the constraint for any vertex $v \in V$. Similarly, removing a vertex from a multipacking cannot create any new constraint, hence the result follows. □

The following lemma is the central result of both our polynomial-time algorithm (Theorem 24) and NP-completeness reduction (Theorem 20).

Lemma 17 *Let $D = (V, A)$ be a single-sourced layered DAG with layers $V_0, V_1, \dots, V_t$. There 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 $\left|S \cap V_j\right| \geqslant 2$, and $i < j$ be the largest index such that $\left|S \cap V_i\right| = 0$. 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 $\left|S \cap V_{i-1}\right| = 1$. 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 18 *Multipacking is NP-complete, even for planar layered DAGs of maximum degree 3 and maximum finite distance 2.*

Proof We provide a reduction from the NP-complete Independent Set problem [20], which remains NP-complete on planar cubic graphs [21].

Independent Set

- **Input**: A graph $G = (V, E)$, an integer $k \in \mathbb{N}$.
- **Question**: Does there exist an independent set of G of size at most k ?

The construction of the instance $(D = (V', A'), k')$ of Multipacking is done by setting $V' = E_1 \cup E_2 \cup V$ where $E_1 = \{e_1^1, \dots, e_m^1\}$ and $E_2 = \{e_1^2, \dots, e_m^2\}$ are two copies of E. We add an arc $e_i^1 e_i^2$ for every $1 \leqslant i \leqslant m$, and two arcs from e_i^2 to the corresponding vertices u and v in V (where $e_i = uv$).

Formally:

$$A' = \{e_i^1 e_i^2 \ : \ 1 \leqslant i \leqslant m\} \cup \{e_i^2 u, \ e_i^2 v \ : \ 1 \leqslant i \leqslant m \ and \ e_i = uv\}$$

It is clear here that D is a layered DAG with three layers and thus, $\mathrm{mfd}\,(D) = 2$.

This reduction can also be seen as follows: given any instance of Independent Set, we subdivide each edge uv by adding a new vertex w with $wu, wv \in A$ and a pending source seeing w. Doing so, most properties of the given instance (such as planarity and maximum degree) are preserved. One can see that the graph G has an independent set of size k if and only if the digraph D has a multipacking of size $k' = m + k$.

$\Rightarrow$ Let S be an independent set of G of size k, and let $S' = E_1 \cup S$. First, S' is of size $m + k$. Then, for any $e_1 \in E_1$, $|N^+[e_1] \cap S'| = 1$ and $|B_2^+(e_1) \cap S'| \leqslant 2$ hold since S is an independent set. By similar arguments, $|N^+[e_2] \cap S'| \leqslant 1$ holds for any $e_2 \in E_2$, and thus no vertex of E_2 can have two out-neighbors in S. All other vertices of D are sinks (i.e. with empty out-neighborhood), so the multipacking property is trivially satisfied for them. Thus S' is a multipacking of D of size $m + k$.

$\Leftarrow$ Let S be a multipacking of maximum size in D such that $|S| \geqslant m + k$. Each vertex of E_1 is a source of D, so by Lemma 16 we can assume that $E_1 \subseteq S$ and then $E_2 \cap S = \emptyset$. So $S \setminus E_1 \subseteq V$, and its size is at least k. Assume S contains two vertices u, v of V that are adjacent in G, then $\left|N^+[e_i^2] \cap S\right| \geqslant 2$ with $e_i^2 = uv$, which contradicts the fact that S is a multipacking of D. Thus $S \setminus E_1$ is an independent set of G of size at least k. □

Remark 19 Multipacking can be solved in time $O^*(2^n)$ by trying all subsets of vertices as a solution. By observing that the reduction of Theorem 18 from Independent Set is linear and that it is unlikely to obtain a subexponential algorithm for Independent Set under the ETH [2] [18, Corollary 11.10], a subexpontential algorithm is also unlikey for Multipacking under the ETH.

[2] The Exponential Time Hypothesis (ETH) assumes that there is no algorithm solving 3-SAT in time $2^{o(n)}$, where n is the number of variables in the formula.

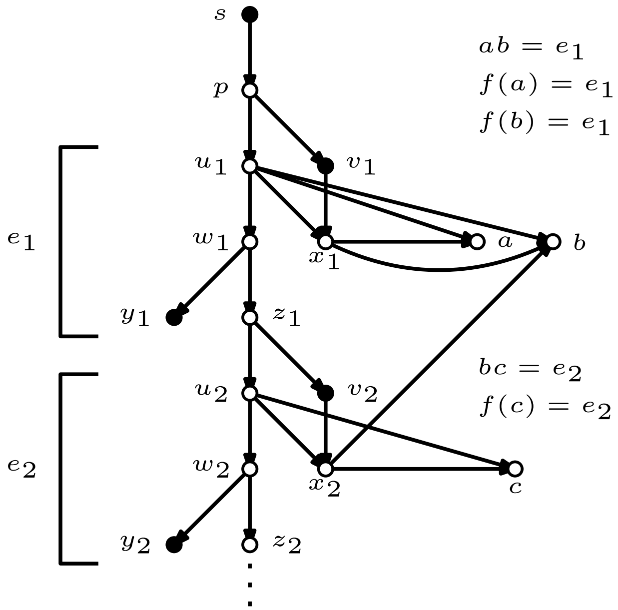


Fig. 5 Sketch of the construction in the proof of Theorem 20 for edges $e_1 = ab$ and $e_2 = bc$ with $f(a) = f(b) = e_1$ and $f(c) = e_2$

Theorem 20 *Multipacking is NP-complete on single-sourced DAGs of maximum degree* 5.

Proof We provide a reduction from Independent Set problem [20], which remains NP-complete for cubic graphs [21]. We define the function $f : V \to E$ such that for $v \in V$, $f(v) = e_i$ if and only if e_i is the first edge in which v appears (recall that $E = \{e_1, \dots, e_m\}$). We create the digraph $D = (V', A)$ as follows (see Fig. 5):

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

□

Claim 21 *The graph G has an independent set of size k if and only if the digraph D has a multipacking of size $k' = k + 2m + 1$.*

Proof ⇒ Let S be an independent set of size k of G. We set $S' = \{s\} \cup \{v_i, y_i \;:\; 1 \leqslant i \leqslant m\} \cup S$. We need to show that S' is a multipacking of D. Notice first that S' contains exactly $2m + k + 1$ vertices. The vertices s and p

satisfy the multipacking property since there is at most one vertex of S' at distance exactly i from both these vertices for any i (and there is no vertex of S' at distance 1 from s and none at distance 0 from p). Each vertex of V and each vertex y_i trivially satisfies the multipacking property since they are sinks. For $1 \leqslant i \leqslant m$, notice that x_i cannot have two out-neighbors in S' since S is an independent set. Hence, x_i and v_i satisfy the multipacking property, since for the latter $B^+_d(v_i) = \{v_i, x_i, u, v\}$ where d is the maximum finite distance in D, $uv = e_i$, and $N^+[v_i] = \{v_i, x_i\}$. Moreover, one can see that w_i satisfies the multipacking property if and only if z_i satisfies it and that z_i satisfies the multipacking property if and only if u_{i+1} satisfies it (z_m is a sink and hence satisfies the multipacking property). We can notice that $B^+_d(u_i) = B^+_d(w_i) \cup \{x_i, u_i\} \cup V(e_i)$. We have $\left|S \cap (\{x_i, u_i\} \cup V(e_i))\right| \leqslant 1$, and the fact that for every other vertex t of $B^+_d(u_i)$, $d(u_i, t) = d(w_i, t) + 1$. So if w_i satisfies the property, then u_i also does. This means that z_{i-1} satisfies it, and thus that w_{i-1} does as well. Using this, and the fact that z_m satisfies the property, we get by induction that for every i, $\{u_i, w_i, z_i\}$ satisfy the property.

$\Leftarrow$ Let S be a multipacking of size k' of D. First, notice that if M is a multipacking of any digraph H, then for any subdigraph H' of H, $M \cap V(H')$ is a multipacking of H'. Notice also that $H = D[V' \setminus V]$ is a single-sourced layered DAG. Let S' be a multipacking of H of maximum size. Using Lemma 17, we can assume that S' contains at most one vertex per layer. For any given $1 \leqslant i \leqslant m$, we are going to prove that for $W_i = \{u_i, v_i, w_i, x_i, y_i, z_i\}$, $\left|S' \cap W_i\right| \leqslant 2$. We can see that $S' \cap \{u_i, v_i\}$ is either empty (which is sufficient to conclude since there remain only two distinct non-empty layers of D' in W_i), or $S' \cap \{u_i, v_i\} = u_i$ (then $S' \cap \{w_i, x_i\} = \emptyset$, which again is enough to conclude), or $S' \cap \{u_i, v_i\} = v_i$. In the latter case, either $S' \cap \{w_i, x_i\} = w_i$, which implies that $S' \cap \{y_i, z_i\} = \emptyset$ or $S' \cap \{w_i, x_i\} = \emptyset$. In both cases, we get that $\left|S' \cap W_i\right| \leqslant 2$. One can also easily see that both s and p cannot be together in S'. Thus, the maximum size of a multipacking of D' is $2m + 1$.

Thus $|S \cap (V' \setminus V)| \leqslant 2m + 1$, and $|S \cap V| \geqslant k$. We also know that for $a, b \in S \cap V$, $ab \notin E$, otherwise there would exist an edge $e_i = ab$ and thus $N^+[x_i] \cap S$ would be of size at least 2. So we can conclude that $S \cap V$ is an independent set of G of size at least k. □

This completes the proof. □

Theorem 22 *Multipacking parameterized by solution size k is W[1]-hard, even on digraphs of maximum finite distance 3.*

Proof We provide an FPT-reduction from Multicolored Independent Set, which is W[1]-hard when parameterized by k [10]. □

Multicolored Independent Set

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

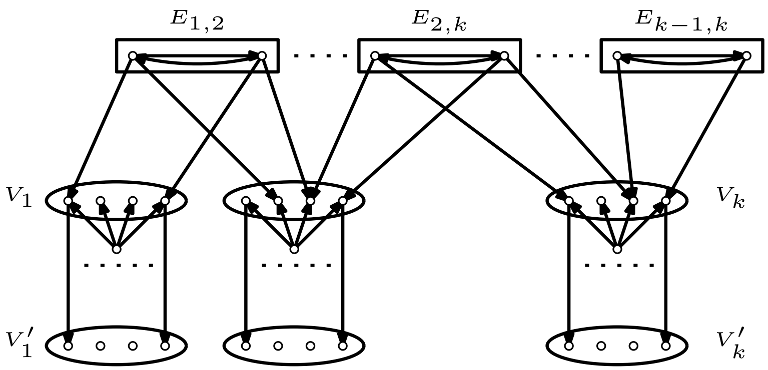


Fig. 6 Sketch of the construction of the digraph D in the proof of Theorem 22

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 $E_{i,j}$, $1 \leqslant i < j \leqslant k$ 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 $\{s_1, \dots, s_k\}$ 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.

See Fig. 6 for an illustration.

Claim 23 *The graph G has a multicolored independent set of size k if and only if the digraph D has a multipacking of size* $k' = 2k + \binom{k}{2}$.

Proof $\Rightarrow$ Let $S = \{u_1, \dots, u_k\}$ be an independent set of G of size k such that $u_i \in V_i$ for every $1 \leqslant i \leqslant k$. Let $S' \subseteq V'$ be a set that contains exactly one arbitrary vertex $e^*_{i,j}$ for every set $E_{i,j}$ $(1 \leqslant i < j \leqslant n)$, together with each vertex of V_i' corresponding to each vertex u_i of S. Finally, add $\{s_1, \dots, s_k\}$ to S'. We claim that S' is the sought multipacking of D. To see this, notice first that $|S'| = 2k + \binom{k}{2}$ by construction. Moreover, every vertex contains at most one vertex from S' in its closed out-neighborhood. We now prove that every vertex $e_{i,j} \in E_{i,j}$ contains at most two vertices from S' in $B_2^+(e_{i,j})$. Assume for a contradiction this is not the case; then, apart from $e^*_{i,j}$, there are two other vertices a and b in $B_2^+(e_{i,j})$. We have that $a \in V_i'$ and $b \in V_j'$. By construction, this means that ab is an edge of G, contradicting the fact that S is an independent set. Finally, since every vertex s_i $(1 \leqslant i \leqslant k)$ has vertices from only one set V_i' in its distance 2 neighborhood, and since S is a multicolored set, the result follows. The only vertices for which checking their distance 3 neighborhood is needed are vertices from $E_{i,j}$ for every $1 \leqslant i < j \leqslant n$. One can notice that for any $e_{i,j} \in E_{i,j}$, $B_3^+(e_{i,j}) \subseteq E_{i,j} \cup V_i' \cup V_j' \cup V_i \cup V_j$, which contains at most 3 vertices of S' since $|S' \cap (V_i' \cup V_j' \cup V_i \cup V_j)| = 2$ and $|S' \cap E_{i,j}| = 1$ by construction.

$\Leftarrow$ Assume that D has a multipacking $S' \subseteq V'$ of size $k' = 2k + \binom{k}{2}$. By Lemma 16, we can assume that S' contains $\{s_1, \dots, s_k\}$. In particular, this means that $S' \cap V_i = \emptyset$ for every $1 \leqslant i \leqslant k$. Moreover, at distance 2, we have $|S' \cap V'_i| \leqslant 1$ for $1 \leqslant i \leqslant k$ since otherwise there would be three vertices from S' in $B_2^+(s_i)$, for some vertex s_i. Moreover, for $1 \leqslant i < j \leqslant n$, $|E_{i,j} \cap S'| \leqslant 1$ since $E_{i,j}$ is a bi-directed clique. Thus, by the size of S', the only possibilities are to pick exactly one vertex in each set V'_i and one vertex $e_{i,j}$ in each set $E_{i,j}$. This can be done only if there exists a multicolored independent set of size k in G: otherwise one would have to select two vertices $a \in V_i$ and $b \in V_j$, $i \neq j$ such that $ab \in E$, which in turn would imply that the vertex from $E_{i,j}$ corresponding to the edge ab has three vertices in its distance-2 neighborhood (namely $e_{i,j}$, a and b).

Thus, the proof is complete. $\square$

4.2 Algorithms

Next, we present a linear-time algorithm.

Theorem 24 *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 17, 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 thus give a polynomial-time bottom-up procedure to find such a set of vertices. 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 will be 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 multipacking of maximum size, 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. Notice that 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 $A_t = V_t$. $\square$

Claim 25 *Let S be a multipacking with at most one vertex per layer. Then S satisfies*:

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

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. □

Notice that Claim 25 gives one less vertex than what Lemma 17 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. Since the sets of active and universal vertices can be constructed by standard graph searching, the whole algorithm takes $O(|V| + |A|)$ time. □

We now give algorithms for structural parameters. We next give a simple algorithm for digraphs of bounded diameter.

Theorem 26 *Multipacking can be solved in time $n^{O(\delta)}$ for digraphs of order n and diameter δ.*

Proof To solve Multipacking by brute-force, we may try all the subsets of size k, and for each subset, check its validity. But in a YES-instance, we have $k \leqslant \delta$, since any ball of radius δ contains all vertices. □

The next algorithm considers jointly two parameters. Recall that by Theorems 18 and 22, such a result is unlikely to hold for each of them individually.[3]

Theorem 27 *Multipacking parameterized by solution size k and maximum degree d can be solved in FPT time $2^{O(kd^k)} + O(d^k n)$ for digraphs of order n.*

Proof Let $(D = (V, A), k)$ be an instance of Multipacking such that D has maximum degree d.

First, we try to find a packing of at least k pairwise disjoint balls of radius k in D (here the undirected distance in the underlying graph of D is considered). If such a packing P exists, then the set S of k centers of the balls of P is a valid solution to Multipacking. Indeed, for every integer $i \leqslant k$, for each vertex v of D, there is at most one vertex of S at directed distance at most i from v. We can solve this problem by reducing to Set Packing, defined as follows.

Set Packing

- **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 packing S of size k, that is, a set of k subsets from $\mathcal{F}$ that are pairwise disjoint ?

---

[3] Note that in the conference version of this paper [19], we have claimed the same algorithm for the maximum out-degree, instead of the maximum degree. However, this algorithm was based on an incorrect claim (Lemma 8 in [19]).

For the reduction, we let $U = V(D)$ and $\mathcal{F}$ be the family of all balls of radius k of D. This set system can be computed in time $O(d^k n)$ using n breadth-first searches. Indeed, each ball of radius k in D has size at most $\sum_{i=0}^{k}(d-1)^i + 1$, which is $O(d^k)$. It is known that Set Packing can be solved in FPT time $2^{O(\Delta k)} + n$ [17], where Δ is the maximum size of a set in $\mathcal{F}$; here $\Delta = O(d^k)$. Hence, applying this reduction gives us a running time of $2^{O(kd^k)} + O(d^k n)$.

If the answer of the previous algorithm is YES, we accept. Otherwise, consider a hypothetical maximum-size packing P of balls of radius k in D: we know that P has size at most $k-1$. Let S be the set of centers of the balls in P. Now, every vertex of D that is not inside a ball in P, is at (undirected) distance at most $2k$ from some vertex in S (otherwise, we could select such a vertex as the center of an additional ball of radius k, and obtain a packing P' of larger size than P, contradicting the maximality of P). Thus D can be covered by $|P| \leqslant k-1$ balls of radius $2k$, and so there are $n \leqslant (k-1)\sum_{i=0}^{2k}(d-1)^i + 1$ vertices in D, which is $d^{O(k)}$. A brute-force algorithm take time $n^{O(k)}$, which is thus $d^{O(k^2)}$, and this is subsumed by the term $2^{O(kd^k)}$ from the running time of the first part. □

Next, we consider the vertex cover number, already considered for Theorem 13.

Theorem 28 *Multipacking parameterized by the vertex cover number c of the input digraph of order n can be solved in FPT time $2^{2^{O(c)}} n^{O(1)}$.*

Proof Let $(D = (V, A), k)$ be the input of Multipacking and let S be a vertex cover of D of size c. As for Theorem 13, we partition the set $V \setminus S$ (which induces no arcs) into equivalence classes $C_1, \dots, C_t$ according to their in- and out-neighborhoods in S. There are $t \leqslant 2^{2c}$ such classes.

By Lemma 16, we can assume that all sources belong to an optimal solution. Consider any class C_i. Its vertices are either all sources, or none of them are. If they are not sources, they all have a common in-neighbor, and thus at most one vertex of C_i can belong to a multipacking. It is not important which one is selected, since all vertices in C_i are twins. We may thus simply try all possibilities of selecting at most one vertex per class C_i, and all possibilities of selecting vertices of S. Thus, there are $2^{t+c} = 2^{2^{O(c)}}$ potential multipackings of D containing all sources. Each of them can be checked in polynomial time. This is an FPT algorithm. □

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, we managed to solve some questions for Broadcast Domination, that we leave 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. Also, Broadcast Domination is FPT for nowhere dense graphs, as it can be expressed in first-order logic. However, it is not clear whether this holds for Multipacking. It is also unknown whether Multipacking is NP-hard on undirected graphs, as asked in [30, 31].

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.

We note that in most of our hardness reductions, the maximum finite distance is very small (which helps us to control the problems at hand), but the actual diameter is infinite (as our digraphs are not strongly connected). It seems a challenging question to derive hardness results for strongly connected digraphs, which can be seen as an intermediate class between the two extremes that are undirected graphs, and DAGs.

We have also shown that both problems are FPT when parameterized by the vertex cover number. What about smaller parameters such as tree-width or DAG-width?

Finally, can our FPT algorithms for both problems parameterized by the solution cost/solution size and maximum (out-)degree be strengthened to a polynomial kernel?

Acknowledgements We thank the four referees for their careful reviewing. In particular, thanks to the referee who pointed out that Lemma 8 from the conference version [19] is wrong: this has led to the corrected version of Theorem 27. Florent Foucaud 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). Florian Sikora is partially funded by the project ESIGMA (ANR-17-CE23-0010).

References

1. Beaudou, L., Brewster, R C.: On the multipacking number of grid graphs. *Discrete Mathematics & Theoretical Computer Science*, 21:#23 (2019)
2. Beaudou, Laurent, Brewster, Richard C.: Foucaud, Florent: Broadcast domination and multipacking: bounds and the integrality gap. The Australasian Journal of Combinatorics **71**(1), 86–97 (2019)
3. Blair, Jean 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*, volume 169 of *Congressus Numerantium*, pages 55–77 (2004)
4. Bodlaender, H. L., Thomassé, S., Yeo, A.: Kernel bounds for disjoint cycles and disjoint paths. Theoretical Comput. Sci. **412**(35), 4570–4578 (2011)
5. Brewster, R. C., MacGillivray, G., Yang, F.: Broadcast domination and multipacking in strongly chordal graphs. Discret. Appl. Math. **261**, 108–118 (2019)
6. 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)
7. Cáceres, José., Hernando, C., Mora, M.è., Pelayo, Ignacio M., Puertas, María L.: General bounds on limited broadcast domination. *Discrete Mathematics & Theoretical Computer Science*, 20(2) (2018)
8. Casel, K.: Resolving conflicts for lower-bounded clustering. In *13th International Symposium on Parameterized and Exact Computation, IPEC 2018, August 20-24, 2018, Helsinki, Finland*, pages 23:1–23:14 (2018)
9. Ruei-Yuan, C., Sheng-Lung, P.: A linear-time algorithm for broadcast domination problem on interval graphs. In *Proceedings of the 27th Workshop on Combinatorial Mathematics and Computation Theory*, pages 184–188. Providence University Taichung, Taiwan (2010)
10. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D.: Dániel, M., Michal P., Pilipczuk, M., Saurabh, S., : Parameterized Algorithms. Springer, Cham (2015)

11. Dabney, J., Dean, B. C., Hedetniemi, S. T.: A linear-time algorithm for broadcast domination in a tree. Networks **53**(2), 160–169 (2009)
12. Dom, M., Lokshtanov, D., Saurabh, S.: Kernelization lower bounds through colors and ids. ACM Transactions Algorithms **11**(2), 13:1-13:20 (2014)
13. Dunbar, J. E., Erwin, D. J., Haynes, T. W., Hedetniemi, S. M., Hedetniemi, S. T.: Broadcasts in graphs. Discret. Appl. Math. **154**(1), 59–75 (2006)
14. Dyer, Martin E., Frieze, Alan M.: Planar 3DM is NP-complete. J. Algorithms **7**, 174–184 (1986)
15. Erwin, D. J.: *Cost domination in graphs*. PhD thesis, Western Michigan University (2001)
16. Erwin, D. J.: Dominating broadcasts in graphs. Bulletin of the ICA **42**, 89–105 (2004)
17. Fellows, M. R., Knauer, C. N., Naomi, R., Prabhakar, R., Frances A., S., Ulrike, T., Dimitrios M., Whitesides, S.: Faster fixed-parameter tractable algorithms for matching and packing problems. Algorithmica **52**(2), 167–176 (2008)
18. Fomin, F.V., Kratsch, D.: *Exact Exponential Algorithms*. Texts in Theoretical Computer Science. An EATCS Series. Springer (2010)
19. Foucaud, F., Gras, B., Perez, A., Sikora, F.: On the complexity of broadcast domination and Multipacking in digraphs. In *Proceedings of the 31st International Workshop on Combinatorial Algorithms*, volume 12126 of *Lecture Notes in Computer Science*, pages 264–276 (2020)
20. Michael, R. G and David S. J.: Computers and Intractability. Freeman, San Francisco (1979)
21. Garey, M. R., Johnson, D.S., Stockmeyer, L.: Some simplified *NP*-complete graph problems. Theoretical Comput. Sci. **1**(3), 237–267 (1976)
22. Grohe, M., Kreutzer, S., Siebertz, S.: Deciding first-order properties of nowhere dense graphs. Journal of the ACM **64**(3), (2017). https://doi.org/10.1145/3051095
23. Heggernes, P., Lokshtanov, D.: Optimal broadcast domination in polynomial time. Discret. Math. **306**(24), 3267–3280 (2006)
24. Heggernes,P.,Sæther, SH: Broadcast domination on block graphs in linear time. In *Computer Science - Theory and Applications - 7th International Computer Science Symposium in Russia, CSR 2012, Nizhny Novgorod, Russia, July 3-7, 2012. Proceedings*, pages 172–183 (2012)
25. Liu, C. L.: Introduction to combinatorial mathematics. McGraw-Hill, New York (1968)
26. Lokshtanov, D.: Broadcast domination. Master's thesis, University of Bergen (2007)
27. Mynhardt, C.M., Teshima, L.E.: Broadcasts and multipackings in trees. Utilitas Mathematica **104**, 227–242 (2017)
28. Nešetřil, J., and de Mendez, P. O.: Sparsity: graphs, structures, and algorithms. Springer Publishing Company, Incorporated (2012)
29. Ordyniak, S., Kreutzer, S.: Quantitative Graph Theory. Width-measures for directed graphs and algorithmic applications, pp. 195–245. Chapman and Hall/CRC, Florida (2014)
30. Teshima, L.: Broadcasts and multipackings in graphs. Master's thesis, University of Victoria (2012)
31. Yang, F.: New results on broadcast domination and multipacking. Master's thesis, University of Victoria (2015)

Authors and Affiliations

Florent Foucaud[1,2,3] · **Benjamin Gras**[3,4] · **Anthony Perez**[3] · **Florian Sikora**[5]

✉ Florent Foucaud
florent.foucaud@uca.fr

Benjamin Gras
benjamin.gras@univ-orleans.fr

Anthony Perez
anthony.perez@univ-orleans.fr

Florian Sikora
florian.sikora@dauphine.psl.eu

1 LIMOS, CNRS UMR 6158, Université Clermont Auvergne, Aubière, France

2 Univ. Bordeaux, CNRS, Bordeaux INP, LaBRI, UMR5800, F-33400 Talence, France

3 Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, F-45067 Orléans, France

4 Universität Trier, Fachbereich 4, Informatikwissenschaften, Germany

5 Univ. Paris-Dauphine, PSL University, CNRS, LAMSADE, 75016 Paris, France