On Directed Covering and Domination Problems

Tesshu Hanaka[1], Naomi Nishimura*[2], and Hirotaka Ono†[3]

1 **Department of Economic Engineering, Kyushu University, Fukuoka, Japan**
3EC15004S@s.kyushu-u.ac.jp

2 **David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, Canada**
nishi@uwaterloo.ca

3 **Department of Mathematical Informatics, Nagoya University, Nagoya, Japan**
ono@i.nagoya-u.ac.jp

Abstract

In this paper, we study covering and domination problems on directed graphs. Although undirected VERTEX COVER and EDGE DOMINATING SET are well-studied classical graph problems, the directed versions have not been studied much due to the lack of clear definitions.

We give natural definitions for DIRECTED r-IN (OUT) VERTEX COVER and DIRECTED (p,q)-EDGE DOMINATING SET as directed generations of VERTEX COVER and EDGE DOMINATING SET. For these problems, we show that (1) DIRECTED r-IN (OUT) VERTEX COVER and DIRECTED (p,q)-EDGE DOMINATING SET are NP-complete on planar directed acyclic graphs except when $r = 1$ or $(p,q) = (0,0)$, (2) if $r \geq 2$, DIRECTED r-IN (OUT) VERTEX COVER is $W[2]$-hard and $c \ln k$-inapproximable on directed acyclic graphs, (3) if either p or q is greater than 1, DIRECTED (p,q)-EDGE DOMINATING SET is $W[2]$-hard and $c \ln k$-inapproximable on directed acyclic graphs, (4) all problems can be solved in polynomial time on trees, and (5) DIRECTED $(0,1),(1,0),(1,1)$-EDGE DOMINATING SET are fixed-parameter tractable in general graphs.

The first result implies that (directed) r-DOMINATING SET on directed line graphs is NP-complete even if $r = 1$.

1998 ACM Subject Classification G.2.2 Graph Theory

Keywords and phrases directed graph, vertex cover, dominating set, edge dominating set, fixed-parameter algorithms

Digital Object Identifier 10.4230/LIPIcs.ISAAC.2017.45

1 Introduction

Covering and domination problems are well-studied problems in theory and in applications of graph algorithms, for example, VERTEX COVER [16], DOMINATING SET [16] and EDGE DOMINATING SET [24]. However, almost all of these problems are studied on undirected graphs. In particular, VERTEX COVER and EDGE DOMINATING SET on directed graphs have not been studied although there are some results on directed DOMINATING SET [11, 7, 21, 15]. This seems surprising, but maybe one reason might be that it is difficult to expand the definition naturally to directed graphs due to the unclear relationship between "direction" and "domination".

In this paper, we study directed versions of VERTEX COVER and EDGE DOMINATING SET. First, we give formal definitions of directed VERTEX COVER and directed EDGE

[*] This work is supported by the Natural Sciences and Engineering Research Council of Canada.
[†] This work is partially supported by KAKENHI, no. 26241031, 26540005, 17H01698 and 17K19960.

28th International Symposium on Algorithms and Computation (ISAAC 2017).
Editors: Yoshio Okamoto and Takeshi Tokuyama; Article No. 45; pp. 45:1–45:12
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Dominating Set. In the definitions, we consider several scenarios that reflect how the selected set influences edges via directed edges. It should be noted that the definition follows from r-Dominating Set [8, 12, 21]. These definitions are also motivated by economic network analysis. We mention applications of these problems in Section 1.2.

In a directed graph, vertex v is said to *in-cover* every incoming edge (u, v) and *out-cover* every outgoing edge (v, u) for some u. A vertex v is also said to *r-in-cover* all edges in the directed path to v of length at most r. Similarly, v is said to *r-out-cover* all edges in the directed path from v. Here, for a path $v_1, v_2, \ldots, v_\ell$, the *length* of the path is defined as the number of edges, that is, $\ell - 1$. In particular, if $r = 0$, a vertex is not considered to cover any edge. Then Directed r-In (Out) Vertex Cover is the following problem.

▶ **Definition 1.** Directed r-In (Out) Vertex Cover (r-In (Out) VC) is the problem that given a directed graph $G = (V, E)$ and two positive integers k and r, determines whether there exists a vertex subset $S \subseteq V$ of size at most k such that every edge in E is r-in (out)-covered by S. Such S is called an *r-in (out)-vertex cover.*

Furthermore, we define Directed (p, q)-Edge Dominating Set. An edge $e = (u, v)$ is said to *(p, q)-dominate* itself and all edges that vertex u p-in-covers and vertex v q-out-covers. In particular, edge (u, v) is said to *$(p, 0)$-dominate* (resp., *$(0, q)$-dominate*) itself and all edges p-in-covered by u (resp., q-out-covered by v).

Then Directed (p, q)-Edge Dominating Set is defined as follows.

▶ **Definition 2.** Directed (p, q)-Edge Dominating Set ((p, q)-EDS) is the problem that given a directed graph $G = (V, E)$, one positive integer k, and two non-negative integers p, q, determines whether there exists an edge subset $K \subseteq E$ of size at most k such that every edge is (p, q)-dominated by K. Such K is called a *(p, q)-edge dominating set.*

The undirected Edge Dominating Set problem is Dominating Set on (undirected) line graphs. We can see the same relationship between Directed $(0, 1)$-Edge Dominating Set and Dominating Set on directed line graphs. For a directed graph, a *directed line graph* is defined as follows:

▶ **Definition 3** ([18]). A *directed line graph* of $G = (V, E)$ is $L(G) = (E, E_2)$ such that

$$E_2 = \{((x, y), (z, w)) | (x, y), (z, w) \in E \land y = z\}.$$

It is obvious that a directed $(0, 1)$-edge dominating set on a directed graph G corresponds to a (directed) dominating set on the line graph of G. Furthermore, Directed $(1, 1)$-Edge Dominating Set corresponds to undirected Dominating Set on an underlying undirected graph of a directed line graph. These relations imply that our definition of Directed (p, q)-Edge Dominating Set is quite natural from the viewpoint of the line graph operation.

One interesting aspect of directed versions, but not undirected versions, is the asymmetry of the problem structures. For Directed r-In Vertex Cover, a vertex v in-covers only (u, v) when $r = 1$. Thus, a 1-in vertex cover is the set of all vertices whose in-degree is at least one. Therefore, it is trivial that Directed 1-In (Out) Vertex Cover is solvable in linear time, while undirected Vertex Cover is NP-complete. On the other hand, Directed $(1, 1)$-Edge Dominating Set, in a sense, corresponds to (undirected) Edge Dominating Set. For the optimization version, Edge Dominating Set is equivalent to Minimum Maximal Matching [24]. However, Directed $(1, 1)$-Edge Dominating Set does not necessarily correspond to matching on the undirected graphs underlying directed graphs due to the asymmetry of domination.

Table 1 Our results for graph classes. NP-c and $W[2]$-h stand for NP-complete and $W[2]$-hard, respectively.

Graph class	Tree	Planar DAG of bounded degree	DAG	General
1-In (Out) VC	-	-	-	$O(n)$
r-In (Out) VC ($r \geq 2$)	$O(n^4)$	NP-c	$W[2]$-h	$W[2]$-h
$(0,1),(1,0)$-EDS	$O(n^4)$	NP-c	NP-c	$2^{O(k)}n$
$(1,1)$-EDS	$O(n^4)$	NP-c	NP-c	$2^{O(k)}n$
(p,q)-EDS (p or $q \geq 2$)	$O(n^4)$	NP-c	$W[2]$-h	$W[2]$-h

For Directed (p,q)-Edge Dominating Set, there exists another source of asymmetry. That is, we can consider the case in which p and q are different. In the case in which $(p,q) = (0,1)$, edge (u,v) dominates itself and edges out-covered by v. Although Directed $(0,1)$-Edge Dominating Set is similar to Directed 1-Out Vertex Cover, surprisingly, it is NP-complete on directed acyclic graphs.

1.1 Our Contributions

Table 1 shows our results. In this paper, we first give hardness results for Directed r-In (Out) Vertex Cover and Directed (p,q)-Edge Dominating Set on restricted graphs, even on directed acyclic planar graphs of bounded degree. The hardness on directed acyclic graphs implies that we cannot design parameterized algorithms with respect to *directed treewidth* [19] and *DAG-width* [2] unless P=NP. The fact that Directed $(0,q)$-Edge Dominating Set is NP-complete even if $q = 1$ implies that r-Dominating Set on directed line graphs is NP-complete even if $r = 1$. Moreover, we prove that Directed r-In (Out) Vertex Cover is $W[2]$-hard and $c \ln k$-inapproximable on directed acyclic graphs when $r \geq 2$, and Directed (p,q)-Edge Dominating Set is $W[2]$-hard and $c \ln k$-inapproximable on directed acyclic graphs when either p or q is greater than 1. These results hold even if there are no multiple edges or loops.

On the other hand, we obtain algorithms for certain cases, including algorithms for all problems when restricted to trees, for any values of p, q, and r. The interplay among distance, direction, and domination results in a complex dynamic programming solution, running in $O(n^4)$ time. Because an edge can either dominate or be dominated by edges outside of a subtree depending on how it is directed, at each step of the algorithm we need to maintain extensive information not only about the subtree itself but also potential outside influence.

We show that Directed $(0,1),(1,0),(1,1)$-Edge Dominating Set is fixed-parameter tractable with respect to k. In particular, we give $2^{O(k)}n$-time algorithms. We emphasize that the running time of these algorithms is single exponential in k and linear in n. Moreover, our fixed-parameter algorithms are based on dynamic programming on a tree decomposition. Thus, we also show that Directed $(0,1),(1,0),(1,1)$-Edge Dominating Set can be solved in linear time on graphs whose underlying undirected graphs have bounded treewidth. Note that given a directed graph G and its underlying undirected graph G^*, the directed treewidth of G is no greater than its DAG-width which, in turn, is no greater than the treewidth of G^* [2].

1.2 Motivation and Application

As practical motivation, a number of network models employ directed graphs. For example, directed graphs are used to represent economic networks in which vertices correspond to industries and edges correspond to transactions of money or materials between industries [22, 17].

Recently, economists have used graph algorithms to analyze these economic networks in terms of graph structures in order to find critical industries and transactions [20, 23]. Based on the analyses, they discuss which kinds of economic policies should be adopted, and so on. However, there are some problems. Such analyses in economics are based on undirected graph algorithms instead of directed graph algorithms; they first transform directed graphs to undirected graphs, and then apply undirected graph algorithms to the graphs thus obtained. This is because there are many more results on graph optimization on undirected graphs than on directed graphs. Of course, such substitute algorithms might extract some information from the processed graph, but some important information is definitely lost. For example, when we would like to find a critical transaction in an economic network, the edge direction is clearly essential.

The theoretical motivation is a relationship between directed Dominating Set and Directed (p, q)-Edge Dominating Set. As we mentioned above, Directed $(0, 1)$-Edge Dominating Set is directed Dominating Set on directed line graphs and Directed $(1, 1)$-Edge Dominating Set is undirected Dominating Set on an underlying undirected graph of a directed line graph. Directed line graphs are well-studied for DNA sequencing and have some useful properties and characterizations [18, 3]. As for combinatorial problems on graphs, (directed) Hamiltonian Path on directed line graphs can be solved in time $O(n^2 + m^2)$ [4] while Hamiltonian Path on undirected line graphs is NP-complete [1]. Therefore, some directed problems could be easier than the undirected versions on line graphs. Unfortunately, however, our results show that directed Dominating Set and the distance version, that is, directed r-Dominating Set, remain NP-complete even on directed line graphs.

1.3 Related problems

One of the most famous covering problems is Vertex Cover. This is a classical NP-complete problem on undirected graphs, but known to be fixed-parameter tractable [6]. In terms of graph parameters, the size of the minimum vertex cover of G is called the *vertex cover number* of G. For any graph, it is easily seen that vertex cover number is greater than or equal to the treewidth [14].

Edge Dominating Set is the problem that given an undirected graph $G = (V, E)$ and an integer k, determines whether there exists a set of edges X of size at most k such that any edge in $E \setminus X$ has at least one incident edge in X. This problem is NP-complete even on bipartite, planar, and bounded degree graphs [24], but fixed-parameter tractable in general [13]. As we have seen, the Edge Dominating Set problem is equivalent to Dominating Set on line graphs. Moreover, the (optimization) Edge Dominating Set problem is equivalent to Minimum Maximal Matching [24].

Dominating Set is a classical domination problem. This problem is known to be $\Omega(\log n)$-inapproximable, but $O(\log n)$-approximable by a simple greedy algorithm on general graphs [9]. With respect to parameterized complexity, Dominating Set is $W[2]$-complete, unlike Vertex Cover and Edge Dominating Set [10]. Therefore, this problem is well-studied on restricted graphs. Recently, Dawar et al. [8] and Drange et al. [12] considered fixed-parameter tractability and the existence of problem kernels for some sparse graph classes. Their results include the distance version, that is, r-Dominating Set. This approach was generalized to directed graphs because the directed Dominating Set problem is also $W[2]$-complete [21].

The remainder of this paper is organized as follows. In Section 2, we first give basic terminology, notions, and definitions. In Section 3, we show the hardness results of the problems. In Section 4, we give polynomial-time algorithms on trees and fixed-parameter algorithms on general graphs. We prove many theorems (including lemmas) in this paper, but several of the proofs are omitted due to space limitations.

2 Preliminaries

In this section, we give notation and definitions. Let $G = (V, E)$ be a directed graph where $|V| = n$ and $|E| = m$. A vertex u is called an *in-neighbor* of v if there exists an edge (u, v) and a vertex w is called an *out-neighbor* of v if there exists an edge (v, w). Moreover, the sets of in (out)-neighbors of v are denoted by $N^{in}(v)$ (resp., $N^{out}(v)$). The number of in (out)-neighbor vertices of v is called the *in (out)-degree* and denoted by $indeg(v) := |N^{in}(v)|$ (resp., $outdeg(v) := |N^{out}(v)|$).

For two vertices u, v, the *distance* from u to v is defined as the number of edges in the shortest path from u to v, denoted by $dist(u, v)$. A vertex u such that $dist(u, v)$ is at most r is called an *r-in-neighbor* of v and a vertex w such that $dist(v, w)$ is at most r is called an *r-out-neighbor* of v. The sets of r-in (out)-neighbors of v are denoted by $N_r^{in}(v)$ (resp., $N_r^{out}(v)$). Note that $N_r^{in}(v) = N^{in}(v)$ and $N_r^{out}(v) = N^{out}(v)$ when $r = 1$.

In an undirected graph G^*, a set of edges such that no edges share an endpoint is called a *matching*. Furthermore, a matching is *maximal* if no proper superset is a matching. An edge dominating set is the edge set E' such that every edge in $E \setminus E'$ is adjacent to at least one edge in E'. Therefore, a maximal matching is an edge dominating set. As a typical design tool of parameterized algorithms, we make use of the tree decomposition and treewidth in this paper. We denote the *treewidth* of G^* by $\mathbf{tw}(G^*)$. For formal definitions of treewidth and tree decomposition, see [5], for example.

A directed graph G is called a *directed acyclic graph* (*DAG*) if G has no directed cycle and a *planar graph* if it can be embedded in the plane without any edges crossing. We mention results on such restricted graphs.

3 Hardness results

In this section, we discuss the hardness of Directed r-In (Out) Vertex Cover and Directed (p, q)-Edge Dominating Set.

3.1 Directed (0, 1), (1, 0)-Edge Dominating Set

We first show that Directed $(0, 1), (1, 0)$-Edge Dominating Set is NP-complete. Although Directed $(0, 1)$-Edge Dominating Set is very similar to 1-Out Vertex Cover, there is a large gap in terms of time complexity.

To show this, we introduce a variant of the SAT problem. Let $(X, \mathcal{C})$ be an instance I of SAT, where $X = \{x_1, x_2, \ldots, n\}$ is the set of variables and $\mathcal{C} = \{C_1, C_2, \ldots, C_m\}$ is the set of clauses. We consider a bipartite graph $G_I = (X \cup \mathcal{C}, E)$, where $E = \{\{x, C\} \mid x \in X, C \in \mathcal{C}$ such that $x \in C$ or $\bar{x} \in C\}$. An instance I of SAT is called *planar* if G_I is planar. Much is known concerning the planar version of SAT. For example, 3SAT is known to be NP-complete even if the instance is restricted to being planar. The restricted version of 3SAT is called Planar 3SAT.

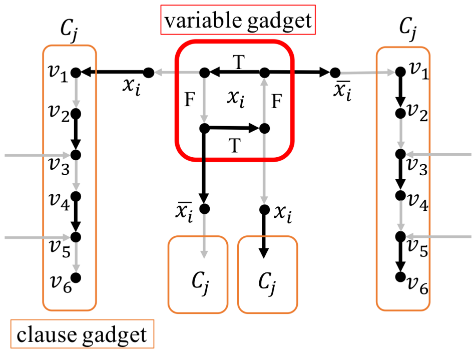

Figure 1 Constructed graph of the reduction from 3SAT to $(0,1)$-EDS

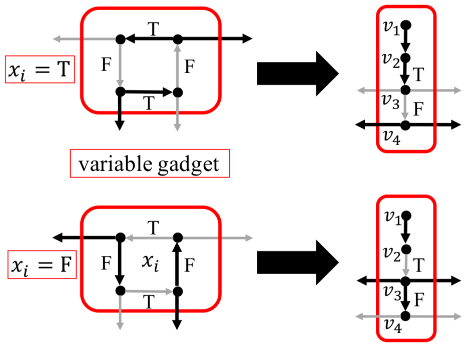

Figure 2 Replacing a cycle by a directed path for a variable's gadget

Here, we consider another restriction of Planar 3SAT. In the restricted instances, each literal appears at most twice, that is, $\forall y \in X \cup \bar{X} : |\{C \in \mathcal{C} \mid y \in C\}| \leq 2$. Instead, the size of each clause is relaxed to be not exactly three but at most three. We call this version Planar At-Most3SAT(L2).

▶ **Lemma 4.** Planar At-Most3SAT(L2) *is NP-complete.*

By using Lemma 4, we can obtain Theorem 5.

▶ **Theorem 5.** Directed $(0,1),(1,0)$-Edge Dominating Set *is NP-complete on directed planar graphs such that indeg*(v) + *outdeg*$(v) \leq 3$ *holds for any vertex.*

Proof. We only consider Directed $(0,1)$-Edge Dominating Set as the other proof is similar. This problem is clearly in NP. Thus, we show the hardness. The reduction is from Planar At-Most3SAT(L2).

Let n be the number of variables, m be the number of clauses, and l be the number of literals in an input Φ for Planar At-Most3SAT(L2). Then, we construct a graph as in Figure 1. First, we create n cycles of length four corresponding to the variables in Φ and m paths of length five corresponding to the clauses in Φ. For a variable's gadget, if we include the two horizontal edges in the $(0,1)$-edge dominating set, it corresponds to setting the variable to true in Φ. Otherwise, we include the two vertical edges, which corresponds to setting the variable to false. Note that the size of a minimum $(0,1)$-edge dominating set for a cycle of length four is two. In Figure 1, thick lines represent that they are included in the solution (we use the same convention in the other figures).

We connect each clause gadget to the variable gadgets corresponding to the literals in the clause, as follows. For $v_1, v_2, \ldots, v_6$ the vertices in the clause gadget, each of v_1, v_3, and v_5 is connected by a path of length two, called a *linking path*, to one of the vertices in a variable gadget. We can observe that there are l linking paths in the constructed graph. For each variable, there are at most two occurrences of true literals and at most two of false literals. Because the variable gadget has four vertices corresponding to literals, by connecting each vertex in the variable gadget to a clause gadget, for any vertex v in the constructed graph, *indeg*(v) + *outdeg*$(v) \leq 3$.

Finally, we conclude this proof by obtaining the following lemma.

▶ **Lemma 6.** *An input* Φ *for* Planar At-Most3SAT(L2) *has a satisfying truth assignment if and only if there exists a* $(0,1)$*-edge dominating set of size* $2n + l + 2m$ *in a constructed graph.* ◀

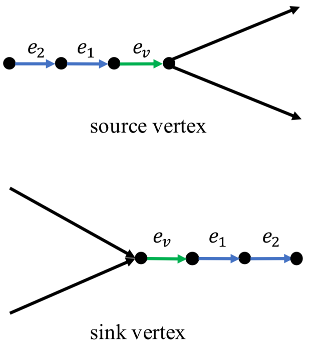

Figure 3 Vertex gadgets for source and sink in the reduction to $(1, 1)$-EDS

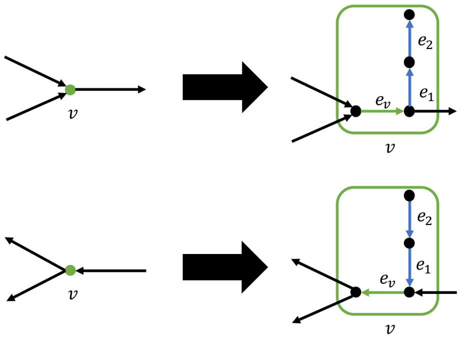

Figure 4 Vertex gadgets for other vertices in the reduction to $(1, 1)$-EDS

By replacing each variable gadget by a path v_1, v_2, v_3, v_4 of length three and connecting vertex v_3 and true literals in a clause, and vertex v_4 and false literals (see Figure 2), we can also show that Directed $(0, 1)$-Edge Dominating Set is NP-complete on directed acyclic planar graphs of bounded degree. Note that edge (v_1, v_2) is contained in any $(0, 1)$-edge dominating set. Moreover, including edge (v_2, v_3) in the $(0, 1)$-edge dominating set corresponds to setting the variable to true and including edge (v_3, v_4) corresponds to setting the variable to false.

▶ **Corollary 7.** Directed $(0, 1), (1, 0)$-Edge Dominating Set *is NP-complete on directed acyclic planar graphs such that* $indeg(v) + outdeg(v) \leq 4$ *holds for any vertex.*

3.2 Directed (1, 1)-Edge Dominating Set

As for Directed $(1, 1)$-Edge Dominating Set, we obtain a stronger result in terms of a degree constraint. To show this, we first introduce a variant of planar graphs. A graph is *planar almost cubic* if it is planar, there are exactly two vertices of degree two, and the degree of all other vertices is three. We show that Vertex Cover remains NP-complete on planar almost cubic graphs.

▶ **Lemma 8.** Vertex Cover *on planar almost cubic graphs is NP-complete.*

By using Lemma 8, we show the following theorem.

▶ **Theorem 9.** Directed $(1, 1)$-Edge Dominating Set *is NP-complete on directed acyclic planar graphs such that* $indeg(v) + outdeg(v) \leq 3$ *holds for any vertex.*

Proof. Since Directed $(1, 1)$-Edge Dominating Set clearly belongs to NP, we prove the hardness. We show a reduction from Vertex Cover on planar almost cubic graphs. Suppose that we are given an instance (G, k) of Vertex Cover. For an undirected planar almost cubic graph G, we choose two vertices with degree two in G as source and sink vertices. We then arrange each vertex in a horizontal line such that the two vertices of degree two become ends of the line and orient every edge from left to right. Note that there exist exactly one source vertex such that the in-degree is zero and out-degree is two and exactly one sink vertex such that the in-degree is two and out-degree is zero. For other vertices v, it holds that $indeg(v) = 1$ and $outdeg(v) = 2$ or $indeg(v) = 2$ and $outdeg(v) = 1$. Each oriented edge corresponding to an edge in G is called an *original edge*.

Next, we attach paths of length three to the source vertex and the sink vertex as a vertex gadget as in Figure 3. Moreover, we replace any other vertex by a path of length three consisting of e_v, e_1, e_2 as in Figure 4. An edge e_v in G' corresponds to vertex v in G. Let G' be the constructed graph. Since we only replace vertices in G by paths, G' remains planar and acyclic and for any vertex v in G', $indeg(v) + outdeg(v) \leq 3$. Then the following lemma completes the proof.

▶ **Lemma 10.** *An instance (G, k) of* Vertex Cover *is a yes-instance if and only if an instance $(G', n+k)$ of* Directed $(1,1)$-Edge Dominating Set *is a yes-instance.* ◀

We also obtain the following result on the distance-generalized version.

▶ **Corollary 11.** Directed (p, q)-Edge Dominating Set *is NP-complete on directed acyclic planar graphs such that $indeg(v) + outdeg(v) \leq 3$ holds for any vertex when $p, q \geq 1$.*

3.3 Distance generalization

In this subsection, we consider the distance-generalized versions as with Corollary 11. We first show that Directed r-In (Out) Vertex Cover and Directed $(0, q), (p, 0)$-Edge Dominating Set are NP-complete on directed acyclic planar graphs of bounded degree.

▶ **Theorem 12.** *When r, p and q are greater than 1,* Directed r-In (Out) Vertex Cover *and* Directed $(0, q), (p, 0)$-Edge Dominating Set *are NP-complete on directed acyclic planar graphs such that $indeg(v) + outdeg(v) \leq 4$ holds for any vertex v.*

From Theorems 5 and 12, we can conclude directed r-Dominating Set on directed line graphs is NP-complete.

▶ **Corollary 13.** *The (directed) r-*Dominating Set *problem is NP-complete on directed line graphs even if $r = 1$.*

Finally, we show that Directed r-In (Out) Vertex Cover and Directed (p, q)-Edge Dominating Set are $W[2]$-hard on directed acyclic graphs by a reduction from Set Cover, which is $W[2]$-complete and $\Omega(\log n)$-inapproximable [10, 9].

▶ **Theorem 14.** Directed r-In (Out) Vertex Cover *is $W[2]$-hard on directed acyclic graphs when $r \geq 2$.* Directed (p, q)-Edge Dominating Set *is $W[2]$-hard on directed acyclic graphs when $p \geq 2$ or $q \geq 2$. For these problems, there is no polynomial-time $c \ln k$-approximation algorithm for any constant $c < 1$ unless P=NP, where k is the size of an optimal solution, though they can be approximated within ratio $O(\log n)$ by a greedy algorithm.*

4 Algorithms

In this section, we give polynomial-time algorithms for Directed r-In (Out) Vertex Cover and Directed (p, q)-Edge Dominating Set on trees and fixed-parameter algorithms for Directed $(0, 1), (1, 0), (1, 1)$-Edge Dominating Set on general graphs.

4.1 Algorithms on Trees

We solve Directed (p, q)-Edge Dominating Set by dynamic programming on a graph G for which the underlying undirected graph is a tree, which we can root at an arbitrary vertex; henceforth we use $\hat{G}$ to denote such a rooted tree. When we use the terms *parent*, *child*, *ancestor*, and *descendant*, we are referring to the relationships between vertices in $\hat{G}$.

We first extend the definition of distance to specify distances between vertices and edges. For an edge $e = (u, v)$ and vertices w and x, we define $dist(w, e)$ to be $dist(w, u)$ and $dist(e, x)$ to be $dist(v, x)$. Moreover, for two edges $e = (u, v)$ and $f = (x, y)$, we define $dist(e, f)$ to be $dist(v, x)$. An edge e *i-in-dominates* (or just *in-dominates*) all edges f such that $dist(f, e) \leq i$ and an edge e *j-out-dominates* (or just *out-dominates*) all edges f such that $dist(e, f) \leq j$. In a directed path containing edges e and f, the edges (not including e and f) traversed along the path are *between* e and f. If there are k edges between e and f, then e $(k + 1)$-out-dominates f and f $(k + 1)$-in-dominates e.

In $\hat{G}$, we use T_v to denote the subtree rooted at the vertex v, and $G[T_v]$ to denote the subgraph of (the directed graph) G induced on the vertices in T_v. We call $G[T_v]$ the *subtree of G rooted at v* and use $conn(v)$ to denote the edge connecting v to its parent, if it has one. We refer to a vertex v as an *out-vertex* if $conn(v)$ is directed from v to its parent and a *in-vertex* if $conn(v)$ is directed from v's parent to v. If v is the root of $\hat{G}$, it is neither an out-vertex nor an in-vertex. We use $same(v)$ and $diff(v)$ to denote the sets of children of v that are out-vertices and in-vertices, respectively, if v is an out-vertex and that are in-vertices and out-vertices, respectively, if v is an in-vertex. Furthermore, we use $ST(v)$ to denote the set of subtrees rooted at vertices in $same(v)$ and $DT(v)$ to denote the set of subtrees rooted at vertices in $diff(v)$; these are considered to be two different *types of subtrees.* In addition, we use C_s to denote the set of edges between v and vertices in $same(v)$, and C_d to denote the set of edges between v and vertices in $diff(v)$; just as there are two types of subtrees, we consider these set to constitute two types of *connecting edges.*

Our dynamic-programming algorithm processes vertices in an order such that a vertex v is processed after all its descendants, where we use information about the subtrees rooted at the children of v to determine how to dominate edges in $G[T_v]$. We store not only the sizes of edge dominating sets, but also the sizes of edge dominating sets defined in terms of their *reach* and *deficit*, which are measures of the impact of edges inside a subtree in the domination of edges outside the subtree and the impact of edges outside a subtree in the domination of edges inside the subtree.

To see how edges in subtrees rooted at children of v can have an impact on each other, suppose v has two children w and x such that w is an out-vertex and x is a in-vertex. Furthermore, consider an edge e_w in $G[T_w]$ such that $dist(e_w, w) = i$ and an edge e_x in $G[T_x]$ such that $dist(x, e_x) = j$. We can form a directed path that starts at e_w and traverses the edges (w, v) and (v, x) to end at e_x. The number of edges between e_w and e_x is $i+j+2$, which means that e_w $(i + j + 3)$-out-dominates e_x and that e_x $(i + j + 3)$-in-dominates e.

To determine the reach of a set of edges K in $G[T_v]$, we first determine the shortest distance i from an edge in K to v, if v is an out-vertex, or the shortest distance i from v to an edge in K, if v is an in-vertex. When v is an endpoint of an edge in K (that is, $i = 0$), that edge will be able to q-out-dominate an edge outside of $G[T_v]$, if v is an out-vertex, or p-in-dominate an edge outside of $G[T_v]$, if v is an in-vertex. We thus define $maxreach(v) = q$ for each out-vertex v and $maxreach(v) = p$ for each in-vertex v. More generally, we define the *reach of K beyond $G[T_v]$* to be $maxreach(v) - i$.

To measure which edges depend on outside edges for domination, we define the *deficit of K within $G[T_v]$* to be maximum over $dist(e, v)$ (resp., $dist(v, e)$) over all edges e in $G[T_v]$ not (p, q)-dominated by any edge in K, for v an out-vertex (resp., in-vertex). Since the edge between v and its parent is the outside edge that can cover the largest deficit, we set $maxdeficit(v) = p$ for v an out-vertex and $maxdeficit(v) = q$ for v a in-vertex. We refer to all edges e with $dist(e, v) \leq d$ (resp., $dist(v, e) \leq d$) to be *edges of deficit of most d* in $G[T_v]$, for v an out-vertex (resp., an in-vertex). Should an edge outside a subtree have sufficient reach to dominate all edges of deficit at most d, we will say that the edge *covers the deficit.*

Using these concepts, we say that a set of edges K is a *reach-r-deficit-d edge dominating set* for $G[T_v]$ if the reach of K beyond $G[T_v]$ is r, and K (p,q)-dominates $G[T_v \backslash J]$ where J is the set of edges of deficit at most d in $G[T_v]$. In our algorithm, we use $D[v,r,d]$ to store the minimum number of edges in a reach-r-deficit-d edge dominating set for $G[T_v]$.

When processing a vertex v, we determine $D[v,r,d]$ for values of r and d in the ranges $0 \leq r \leq \mathit{maxreach}(v)$ and $0 \leq d \leq \mathit{maxdeficit}(v)$. For the base cases, for each leaf v in $\hat{G}$, we set $D[v,r,d] = 0$ for all values of r and d. To determine the value of $D[v,r,d]$, we will consider all possible options for adding edges between v and its children to K, a reach-r-deficit-d edge dominating set for $G[T_v]$, as the choice of edges of K in the subtrees rooted at the children of v will be represented by already-computed table entries.

The computation of the table entries depends on the following lemmas.

▶ **Lemma 15.** *The reach of K beyond $G[T_v]$ is $\mathit{maxreach}(v)$ if and only if $K \cap C_s \neq \emptyset$.*

▶ **Lemma 16.** *If $K \cap C_s = \emptyset$, the reach of K beyond $G[T_v]$ is one less than the maximum over all vertices $u \in \mathit{same}(v)$ of the reach of K restricted to $G[T_u]$.*

▶ **Lemma 17.** *For any child u of v, $\mathit{conn}(u)$ covers a deficit of $\mathit{maxdeficit}(u)$ in $G[T_u]$.*

▶ **Lemma 18.** *For any child u of v, if $\mathit{conn}(u)$ is not included in K, then the maximum possible deficit within $G[T_v]$ that can be covered by K is $\mathit{maxdeficit}(u) - 1$.*

▶ **Lemma 19.** *For any child u of v, if $\mathit{conn}(u)$ is not included in K, the deficit in $G[T_v]$ will be covered by any single connecting edge of the opposite type. Thus, if $K \cap C_d \neq \emptyset$, $d = 0$.*

The complete proofs of Theorems 20 and 21 are omitted.

▶ **Theorem 20.** *There is an algorithm that solves* Directed (p,q)-Edge Dominating Set *on trees in $O(n^4)$-time.*

▶ **Theorem 21.** *There is an algorithm that solves* Directed r-In (Out) Vertex Cover *on trees in $O(n^4)$-time.*

4.2 Fixed-Parameter Algorithm for Directed Edge Dominating Set

In this subsection, we give a $2^{O(k)}n$-time algorithm for Directed $(1,1)$-Edge Dominating Set. First, we obtain the following lemmas and theorem.

▶ **Lemma 22.** *Given a directed graph G, let G^* be the underlying undirected graph of G and s be the minimum size of* Directed $(1,1)$-Edge Dominating Set *on G. Then the following inequality holds: $\mathbf{tw}(G^*) \leq 2s$.*

Proof. Let G^* be an undirected graph, $\mathbf{tw}(G^*)$ be the treewidth of G^*, and $\mathbf{vc}(G^*)$ be the size of minimum vertex cover. Then we have $\mathbf{tw}(G^*) \leq \mathbf{vc}(G^*)$ [14]. Let M^* be a minimum maximal matching in G^*. A minimum $(1,1)$-edge dominating set in G is an (not necessarily minimum) edge dominating set in G^*. If not, there is an edge not dominated by the $(1,1)$-edge dominating set in G. Moreover, for any edge dominating set D in undirected graphs, $|D| \geq |M^*|$ holds because a minimum maximal matching is a minimum edge dominating set [24]. Therefore, $s \geq |M^*|$ holds. On the other hand, we have a well-known result that for any maximal matching M, $\mathbf{vc}(G^*) \leq 2|M|$ [16]. Moreover, we already know that $\mathbf{tw}(G^*) \leq \mathbf{vc}(G^*)$ holds. Finally, we can obtain $\mathbf{tw}(G^*) \leq 2s$. ◀

▶ **Lemma 23.** *Given a directed graph G, let G^* be the underlying undirected graph of G. Then given a tree decomposition of G^* of width at most ℓ, there exists an algorithm that solves* Directed $(1,1)$-Edge Dominating Set *in $25^\ell \ell^{O(1)} n$-time.*

▶ **Theorem 24** ([5]). *There exists an algorithm that, given an n-vertex graph G and an integer ℓ, in time $2^{O(\ell)}n$ either outputs that the treewidth of G is larger than ℓ, or constructs a tree decomposition of G of width at most $5\ell + 4$.*

Finally, Directed $(1, 1)$-Edge Dominating Set can be solved in the following time.

▶ **Theorem 25.** *Given an instance (G, k) of* Directed $(1, 1)$-Edge Dominating Set*, it can be solved in $2^{O(k)}n$-time.*

Proof. Given an instance (G, k), we first determine whether the treewidth of G^* is at most $2k$ in $2^{O(k)}n$-time by using Theorem 24. If $\mathbf{tw}(G^*) > 2k$, we conclude that it is a no-instance by Lemma 22. Otherwise, we use the $25^{\ell}\ell^{O(1)}n$-time algorithm based on a tree decomposition of width at most $10k + 4$ obtained by Theorem 24. Therefore, the total running time is $2^{O(k)}n + 25^{10k+4}(10k + 4)^{O(1)}n = 2^{O(k)}n$. ◀

Thus, Directed $(1, 1)$-Edge Dominating Set is fixed-parameter tractable with respect to k. We emphasize that the running time of this algorithm is single exponential in k and linear in n. In the same way, we can prove Directed $(0, 1), (1, 0)$-Edge Dominating Set is fixed-parameter tractable with respect to k.

▶ **Theorem 26.** *Given an instance (G, k) of* Directed $(0, 1), (1, 0)$-Edge Dominating Set*, it can be solved in $2^{O(k)}n$-time.*

References

1 Alan A. Bertossi. The edge hamiltonian path problem is NP-complete. *Information Processing Letters*, 13(4):157–159, 1981.

2 Dietmar Berwanger, Anuj Dawar, Paul Hunter, Stephan Kreutzer, and Jan Obdržálek. The dag-width of directed graphs. *Journal of Combinatorial Theory, Series B*, 102(4):900–923, 2012.

3 Jacek Blazewicz, Alain Hertz, Daniel Kobler, and Dominique de Werra. On some properties of DNA graphs. *Discrete Applied Mathematics*, 98(1):1–19, 1999.

4 Jacek Blazewicz, Marta Kasprzak, Benjamin Leroy-Beaulieu, and Dominique de Werra. Finding hamiltonian circuits in quasi-adjoint graphs. *Discrete Applied Mathematics*, 156(13):2573–2580, 2008.

5 Hans L. Bodlaender, Pål Grønås Drange, Markus S. Dregi, Fedor V. Fomin, Daniel Lokshtanov, and Michał Pilipczuk. A $c^k n$ 5-approximation algorithm for treewidth. *SIAM Journal on Computing*, 45(2):317–378, 2016.

6 Jonathan F. Buss and Judy Goldsmith. Nondeterminism within P^*. *SIAM Journal on Computing*, 22(3):560–572, 1993.

7 Miroslav Chlebík and Janka Chlebíková. Approximation hardness of dominating set problems in bounded degree graphs. *Information and Computation*, 206(11):1264–1275, 2008.

8 Anuj Dawar and Stephan Kreutzer. Domination problems in nowhere-dense classes. In *Proceedings of IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS2009)*, pages 157–168. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2009.

9 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-sixth Annual ACM Symposium on Theory of Computing (STOC2014)*, pages 624–633. ACM, 2014.

10 Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness I: Basic results. *SIAM Journal on Computing*, 24(4):873–921, 1995.

11 Rodney G. Downey and Michael R. Fellows. Parameterized computational feasibility. In *Proceedings of Feasible Mathematics II*, pages 219–244. Birkhäuser Boston, 1995.

12 Pål Grønås Drange, Markus Dregi, Fedor V. Fomin, Stephan Kreutzer, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, Felix Reidl, Fernando Sánchez Villaamil, Saket Saurabh, Sebastian Siebertz, and Somnath Sikdar. Kernelization and sparseness: the case of dominating set. In *Proceedings of 33rd Symposium on Theoretical Aspects of Computer Science (STACS 2016)*, pages 31:1–31:14. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

13 Henning Fernau. edge dominating set: Efficient enumeration-based exact algorithms. In *Proceedings of Parameterized and Exact Computation: Second International Workshop (IWPEC2006)*, pages 142–153. Springer Berlin Heidelberg, 2006.

14 Jiří Fiala, Petr A. Golovach, and Jan Kratochvíl. Parameterized complexity of coloring problems: Treewidth versus vertex cover. *Theoretical Computer Science*, 412(23):2513–2523, 2011.

15 Robert Ganian, Petr Hliněný, Joachim Kneis, Alexander Langer, Jan Obdržálek, and Peter Rossmanith. Digraph width measures in parameterized algorithmics. *Discrete Applied Mathematics*, 168:88–107, 2014.

16 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman & Co., New York, NY, USA, 1979.

17 Tesshu Hanaka, Shigemi Kagawa, Hirotaka Ono, and Keiichiro Kanemoto. Finding environmentally critical transmission sectors, transactions, and paths in global supply chain networks. *Energy Economics*, 2017. (in press).

18 Frank Harary and Robert Z. Norman. Some properties of line digraphs. *Rendiconti del Circolo Matematico di Palermo*, 9(2):161–168, 1960.

19 Thor Johnson, Neil Robertson, P.D. Seymour, and Robin Thomas. Directed tree-width. *Journal of Combinatorial Theory, Series B*, 82(1):138–154, 2001.

20 Shigemi Kagawa, Sangwon Suh, Klaus Hubacek, Thomas Wiedmann, Keisuke Nansai, and Jan Minx. CO_2 emission clusters within global supply chain networks: Implications for climate change mitigation. *Global Environmental Change*, 35:486–496, 2015.

21 Stephan Kreutzer and Siamak Tazari. Directed nowhere dense classes of graphs. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA2012)*, pages 1552–1562. Society for Industrial and Applied Mathematics, 2012.

22 Michael Lahr and Erik Dietzenbacher. *Input-Output Analysis: Frontiers and Extensions.* Palgrave Macmillan UK, 2001.

23 Omar Rifki, Hirotaka Ono, and Shigemi Kagawa. The robustest clusters in the input–output networks: global CO_2 emission clusters. *Journal of Economic Structures*, 6(1):3, 2017.

24 Mihalis Yannakakis and Fanica Gavril. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics*, 38(3):364–372, 1980.