New Results on Directed Edge Dominating Set

Rémy Belmonte

University of Electro-Communications, Chofu, Tokyo, 182-8585, Japan

Tesshu Hanaka

Department of Information and System Engineering, Chuo University, Tokyo, Japan

Ioannis Katsikarelis

Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243
LAMSADE, 75016, Paris, France

Eun Jung Kim[1]

Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243
LAMSADE, 75016, Paris, France

Michael Lampis[2]

Université Paris-Dauphine, PSL Research University, CNRS, UMR 7243
LAMSADE, 75016, Paris, France

Abstract

We study a family of generalizations of Edge Dominating Set on directed graphs called Directed (p, q)-Edge Dominating Set. In this problem an arc (u, v) is said to dominate itself, as well as all arcs which are at distance at most q *from* v, or at distance at most p *to* u.

First, we give significantly improved FPT algorithms for the two most important cases of the problem, $(0, 1)$-dEDS and $(1, 1)$-dEDS (that correspond to versions of Dominating Set on line graphs), as well as polynomial kernels. We also improve the best-known approximation for these cases from logarithmic to constant. In addition, we show that (p, q)-dEDS is FPT parameterized by $p + q + \mathrm{tw}$, but W-hard parameterized just by tw, where tw is the treewidth of the underlying graph of the input.

We then go on to focus on the complexity of the problem on tournaments. Here, we provide a complete classification for every possible fixed value of p, q, which shows that the problem exhibits a surprising behavior, including cases which are in P; cases which are solvable in quasi-polynomial time but not in P; and a single case ($p = q = 1$) which is NP-hard (under randomized reductions) and cannot be solved in sub-exponential time, under standard assumptions.

2012 ACM Subject Classification Theory of computation → Parameterized complexity and exact algorithms

Keywords and phrases Edge Dominating Set, Tournaments, Treewidth

Digital Object Identifier 10.4230/LIPIcs.MFCS.2018.67

Funding Partially supported by JSPS and MAEDI under the Japan-France Integrated Action Program (SAKURA), Project GRAPA 38593YJ.

[1] The author was partially supported by the ANR grant "ESIGMA" (ANR-17-CE40-0028).
[2] The author was partially supported by the ANR grant "ESIGMA" (ANR-17-CE40-0028).

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).
Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 67; pp. 67:1–67:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Table 1 Complexity status for various values of p and q: on general digraphs.

Param.	p, q	FPT / W-hard	Kernel	Approximability
k	$p+q \leq 1$	$2^{O(k)}$ [22] $\to 2^k$ [Thm.3]	$O(k)$ vertices [Thm.8]	3-apprx [Thm.4]
	$p = q = 1$	$2^{O(k)}$ [22] $\to 9^k$ [Thm.2]	$O(k^2)$ vertices [Thm.7]	8-apprx [Thm.5]
	$\max\{p, q\} \geq 2$	W[2]-hard [22]	-	no $o(\ln k)$-approx [22]
tw	any p, q	W[1]-hard [Thm.11]	-	-
tw+p+q	any p, q	FPT [Thm.12]	unknown	-

1 Introduction

Edge Dominating Set (EDS) is a classical graph problem, equivalent to Minimum Dominating Set on line graphs. Despite the problem's prominence, EDS has until recently received very little attention in the context of directed graphs. In this paper we investigate the complexity of a family of natural generalizations of this classical problem to digraphs, building upon recent work [22].

One of the reasons that EDS has not so far been well studied in digraphs is that there are several natural ways in which the undirected version can be generalized. For example, seeing as EDS is exactly Dominating Set in line graphs, one could define Directed EDS as (Directed) Dominating Set in line digraphs [23]. In this formulation, an arc (u, v) dominates all arcs (v, w); however (v, w) does not dominate (u, v). Another natural way to define the problem would be to consider Dominating Set on the underlying graph of the line digraph, so as to maximize the symmetry of the problem, while still taking into account the directions of arcs. In this formulation, (u, v) dominates arcs coming out of v and arcs coming into u, but not other arcs incident on u, v.

A unifying framework for studying such formulations was recently given in [22], which defined (p, q)-dEDS for any two non-negative integers p, q. In this setting, an arc (u, v) dominates every other arc which lies in a directed path of length at most q that begins at v, or lies in a directed path of length at most p that ends at u. In other words, (u, v) dominates arcs in the forward direction up to distance q, and in the backward direction up to distance p. The interest in defining the problem in such a general manner is that it allows us to capture at the same time Directed Dominating Set on line digraphs ($(0, 1)$-dEDS), Dominating Set on the underlying graph of the line digraph ($(1, 1)$-dEDS), as well as versions corresponding to r-Dominating Set in the line digraph. We thus obtain a family of optimization problems on digraphs, with varying degrees of symmetry, all of which crucially depend on the directions of arcs in the input digraph.

Our contribution. In this paper we advance the state of the art on the complexity of Directed (p, q)-Edge Dominating Set on two fronts.[3]

First, we study the complexity and approximability of the problem in general. The problem is NP-hard for all values of p, q (except $p = q = 0$), even for planar bounded-degree DAGs [22], so it makes sense to study its parameterized complexity and approximability. We show that its two most natural cases, $(1, 1)$-dEDS and $(0, 1)$-dEDS, admit FPT algorithms with running times 9^k and 2^k respectively, where k is the size of the optimal solution. These algorithms significantly improve upon the FPT algorithms given in [22], which uses the fact

[3] We note that in the remainder we always assume that $p \leq q$, as in the case where $p > q$ we can reverse the direction of all arcs and solve (q, p)-dEDS.

Table 2 Complexity status for various values of p and q: on tournaments.

Range of p, q	Complexity
$p = q = 1$	NP-hard [Thm. 13], FPT [Thm. 2], polynomial kernel [Thm. 7]
$p = 2$ or $q = 2$	Quasi-P-time [Thm. 25], W[2]-hard [Thm 24]
remaining cases	P-time [Thm. 26 and 27]

that the treewidth (of the underlying graph of the input) is at most $2k$ and runs dynamic programming over a tree-decomposition of width at most $10k$, obtained by the algorithm of [5]. The resulting running-time estimate for the algorithm of [22] is thus around 25^{10k}. Though both of our algorithms rely on standard branching techniques, we make use of several non-trivial ideas to obtain reasonable bases in their running times. We also show that both of these problems admit polynomial kernels. These are the only cases of the problem which may admit such kernels, since the problem is W-hard for all other values of p, q [22].

Furthermore, we give an 8-approximation for $(1, 1)$-dEDS and a 3-approximation for $(0, 1)$-dEDS. We recall that [22] showed an $O(\log n)$-approximation for general values of p, q, and a matching logarithmic lower bound for the case $\max\{p, q\} \geq 2$. Therefore our result completes the picture on the approximability of the problem by showing that the only two currently unclassified cases belong in APX.

Finally, we consider the problem's complexity parameterized by the treewidth of the underlying graph and show that, even though the problem is FPT when all of p, q, tw are parameters, it is in fact W[1]-hard if parameterized only by tw. (See Table 1).

Our second, and perhaps main contribution in this paper is an analysis of the complexity of the problem on tournaments, which are one of the most well-studied classes of digraphs (see Table 2). One of the reasons for focusing on this class is that the complexity of Dominating Set has a peculiar status on tournaments, as it is solvable in quasi-polynomial time, W[2]-hard, but neither in P nor NP-complete (under standard assumptions). Here we provide a *complete classification* of the problem which paints an even more surprising picture. We show that (p, q)-dEDS goes from being in P for $p + q \leq 1$; to being APX-hard and unsolvable in $2^{n^{1-\epsilon}}$ under the (randomized) ETH for $p = q = 1$; to being equivalent to Dominating Set on tournaments, hence NP-intermediate, quasi-polynomial-time solvable, and W[2]-hard, when one of p and q equals 2; and finally to being polynomial-time solvable again if $\max\{p, q\} \geq 3$ and neither p nor q equals 2. We find these results surprising, because few problems demonstrate such erratic complexity behavior when manipulating their parameters and because, even though in many cases the problem does seem to behave like Dominating Set, the fact that $(1, 1)$-dEDS becomes significantly harder shows that the problem has interesting complexity aspects of its own. The most technical part of this classification is the reduction that establishes the hardness of $(1, 1)$-dEDS, which makes use of several *randomized* tournament constructions, which we show satisfy certain desirable properties with high probability; as a result our reduction itself is randomized.

Due to space restrictions, some of our proofs are omitted here.

Related Work. On undirected graphs Edge Dominating Set, also known as Maximum Minimal Matching, is NP-complete even on bipartite, planar, bounded degree graphs as well as other special cases [34, 24]. It can be approximated within a factor of 2 [19] (or better in some special cases [8, 29, 2]), but not a factor better than 7/6 [9] unless P=NP. The problem has been the subject of intense study in the parameterized and exact algorithms community [32], producing a series of improved FPT algorithms [17, 3, 18, 30]; the current best is given in [25]. A kernel with $O(k^2)$ vertices and $O(k^3)$ edges is also known [21].

For (p,q)-dEDS, [22] shows the problem to be NP-complete on planar DAGs, in P on trees, and W[2]-hard and $c \ln k$-inapproximable on DAGs if $\max\{p,q\} > 1$. The same paper gives FPT algorithms for $\max\{p,q\} \leq 1$. Their algorithm performs DP on a tree-decomposition of width w in $O(25^w)$, and uses the fact that $w \leq 2k$, and the algorithm of [5] to obtain a decomposition of width $10k$.

Dominating Set is known not to admit an $o(\log n)$-approximation [12, 27], and to be W[2]-hard and unsolvable in time $n^{o(k)}$ under the ETH [13, 10]. The problem is significantly easier on tournaments, as the optimal is always at most $\log n$, hence there is a trivial $n^{O(\log n)}$ (quasi-polynomial)-time algorithm. It remains, however, W[2]-hard [14]. The problem thus finds itself in an intermediate space between P and NP, as it cannot have a polynomial-time algorithm unless FPT=W[2], and it cannot be NP-complete under the ETH (as it admits a quasi-polynomial time algorithm). The generalization of Dominating Set where vertices dominate their r-neighborhood has also been well-studied in general [7, 11, 15, 26]. This problem is much easier on tournaments for $r \geq 2$, as the size of the solution is always a constant [4].

2 Definitions and Preliminaries

Graphs and domination. We use standard graph-theoretic notation. If $G = (V,E)$ is a graph, $S \subseteq V$ a subset of vertices and $A \subseteq E$ a subset of edges, then $G[S]$ denotes the subgraph of G induced by S, while $G[A]$ denotes the subgraph of G that includes A and all its endpoints. We let $V = A \dot\cup B$ denote the disjoint set union of A and B. For a vertex $v \in V$, the set of neighbors of v in G is denoted by $N_G(v)$, or simply $N(v)$, and $N_G(S) := (\bigcup_{v\in S} N(v)) \setminus S$ will be written as $N(S)$. We define $N[v] := N(v) \cup \{v\}$ and $N[S] := N(S) \cup S$. Depending on the context, we use (u,v) for $u,v \in V$ to denote either an undirected edge connecting two vertices u,v, or an *arc* (a directed edge) with *tail* u and *head* v. An *incoming* (resp. *outgoing*) arc for vertex v is an arc whose head (resp. tail) is v.

In a directed graph $G = (V,E)$, the set of *out-neighbors* (resp.*in-neighbors*) of a vertex v is defined as $\{u \in V : (v,u) \in E\}$ (resp. $\{u \in V : (u,v) \in E\}$) and denoted as $N^+_G(v)$ (resp. $N^-_G(v)$). Similarly as for undirected graphs, $N^+(S)$ and $N^-(S)$ respectively stand for the sets $(\bigcup_{v\in S} N^+(v)) \setminus S$ and $(\bigcup_{v\in S} N^-(v)) \setminus S$. For a subdigraph H of G and subsets $S,T \subseteq V$, we let $\delta_H(S,T)$ denote the set of arcs in H whose tails are in S and heads are in T. We use $\delta^-_H(S)$ (resp. $\delta^+_H(S)$) to denote the set $\delta_H(V \setminus S, S)$ (resp. the set $\delta_H(S, V \setminus S)$). If S is a singleton consisting of a vertex v, we write $\delta^+_H(v)$ (resp. $\delta^-_H(v)$) instead of $\delta^+_H(\{v\})$ (resp. $\delta^-_H(\{v\})$). The *in-degree* $d^-_H(v)$ (respectively *out-degree* $d^+_H(v)$) of a vertex v is defined as $|\delta^-_H(v)|$ (resp. $|\delta^+_H(v)|$), and we write $d_H(v)$ to denote $d^+_H(v) + d^-_H(v)$. We omit H if it is clear from the context. If H is $G[A]$ for some vertex or arc set of G, then we write A in place of $G[A]$. A *source* (resp. *sink*) is a vertex that has no incoming (resp. outgoing) arcs.

For integers $p,q \geq 0$, an arc $e = (u,v)$ is said to (p,q)-*dominate* itself, and all arcs that are on a directed path of length at most p to u or on a directed path of length at most q from v. The central problem in this paper is Directed (p,q)-Edge Dominating Set ((p,q)-dEDS): given a directed graph $G = (V,E)$, a positive integer k and two non-negative integers p,q, we are asked to determine whether an arc subset $K \subseteq E$ of size at most k exists, such that every arc is (p,q)-dominated by K (a (p,q)-*edge dominating set* of G).

Complexity background. We assume that the reader is familiar with the basic definitions of parameterized complexity, such as the classes FPT and W[1], as well as the Exponential Time Hypothesis (ETH, see [10]). For a problem P, we let OPT_P denote the value of its optimal solution. We also make use of standard graph width measures, such as *vertex cover number* vc, *treewidth* tw and *pathwidth* pw [10].

Tournaments. A *tournament* is a directed graph in which every pair of distinct vertices is connected by a single arc. Given a tournament T, we denote by T^{rev} the tournament obtained from T by reversing the direction of every arc. Every tournament has a *king* (sometimes also called a 2-king), i.e. a vertex from which every other vertex can be reached by a path of length at most 2. One such king is the vertex of maximum out-degree (see e.g. [4]). It is folklore that any tournament contains a *Hamiltonian path*, i.e. a directed path that uses every vertex. The Dominating Set problem can be solved by brute force in time $n^{O(\log n)}$ on tournaments, by the following lemma:

▶ **Lemma 1** ([10]). *Every tournament on n vertices has a dominating set of size $\leq \log n + 1$.*

3 Tractability

3.1 FPT algorithms

In this section, we present FPT branching algorithms for $(0,1)$-dEDS and $(1,1)$-dEDS. Both algorithms operate along similar lines, taking into consideration the particular ways available for domination of each arc.

▶ **Theorem 2.** *The $(1,1)$-dEDS problem parameterized by solution size k can be solved in time $O^*(9^k)$.*

Proof. We present an algorithm that works in two phases. In the first phase we perform a branching procedure which aims to locate vertices with positive out-degree or in-degree in the solution. The general approach of this procedure is standard (as long as there is an uncovered arc, we consider all ways in which it may be covered), and uses the fact that at most $2k$ vertices have positive in- or out-degree in the solution. However, in order to speed up the algorithm, we use a more sophisticated branching procedure which picks an endpoint of the current arc (u,v) and *completely guesses* its behavior in the solution. This ensures that this vertex will never be branched on again in the future. Once all arcs of the graph are covered, we perform a second phase, which runs in polynomial time, and by using a maximum matching algorithm finds the best solution corresponding to the current branch.

Let us now describe the branching phase of our algorithm. We construct three sets of vertices V^+, V^-, V^{+-}. The meaning of these sets is that when we place a vertex u in V^+, V^-, or V^{+-} we guess that u has (i) positive out-degree and zero in-degree in the optimal solution; (ii) positive in-degree and zero out-degree in the optimal solution; (iii) positive in-degree and positive out-degree in the optimal solution, respectively. Initially all three sets are empty. When the algorithm places a vertex in one of these sets we say that the vertex has been *marked*.

Our algorithm now proceeds as follows: given a graph $G(V,E)$ and three disjoint sets V^+, V^-, V^{+-} we do the following:

1. If $|V^+| + |V^-| + 2|V^{+-}| > 2k$, reject.
2. While there exists an arc (u,v) with both endpoints unmarked do the following and return the best solution:
   a. Call the algorithm with $V^+ := V^+ \cup \{v\}$ and other sets unchanged.
   b. Call the algorithm with $V^{+-} := V^{+-} \cup \{v\}$ and other sets unchanged.
   c. Call the algorithm with $V^- := V^- \cup \{u\}$ and other sets unchanged.
   d. Call the algorithm with $V^{+-} := V^{+-} \cup \{u\}$ and other sets unchanged.
   e. Call the algorithm with $V^+ := V^+ \cup \{u\}$, $V^- := V^- \cup \{v\}$, and V^{+-} unchanged.

It is not hard to see that Step 1 is correct as as $|V^+| + |V^-| + 2|V^{+-}|$ is a lower bound on the sum of the degrees of all vertices in the optimal and therefore cannot surpass $2k$.

Branching Step 2 is also correct: in order to cover (u, v) the optimal solution must either take an arc coming out of v (2a,2b), or an arc coming into u (2c,2d), or, if none of the previous cases apply, it must take the arc itself (2e).

Once we have applied the above procedure exhaustively, all arcs of the graph have at least one marked endpoint. We say that an arc (u, v) with $u \in V^- \cup V^{+-}$, or with $v \in V^+ \cup V^{+-}$ is covered. We now check if the graph contains an uncovered arc (u, v) with exactly one marked endpoint. We then branch by considering all possibilities for its other endpoint. More precisely, if $u \in V^+$ and v is unmarked, we branch into three cases, where v is placed in V^+, or V^-, or V^{+-} (and similarly if v is the marked endpoint). This branching step is also correct, since the degree specification for the currently marked endpoint does not dominate the arc (u, v), hence any feasible solution must take an arc incident on the other endpoint.

Once the above procedure is also applied exhaustively we have a graph where all arcs either have both endpoints marked, or have one endpoint marked but in a way that if we respect the degree specifications the arc is guaranteed to be covered. What remains is to find the best solution that agrees with the specifications of the sets V^+, V^-, V^{+-}.

We first add to our solution S all arcs $\delta(V^+, V^-)$, i.e. all arcs (u, v) such that $u \in V^+$ and $v \in V^-$, since there is no other way to dominate these arcs. We then define a bipartite graph $H = (V^+ \cup V^{+-}, V^- \cup V^{+-}, \delta(V^+ \cup V^{+-}, V^- \cup V^{+-}))$. That is, H contains all vertices in V^+ along with a copy of V^{+-} on one side, all vertices of V^- and a copy of V^{+-} on the other side and all arcs in E with tails in $V^+ \cup V^{+-}$ and heads in $V^- \cup V^{+-}$. We now compute a minimum edge cover of this graph, that is, a minimum set of edges that touches every vertex. This can be done in polynomial time by finding a maximum matching and then adding an arbitrary incident edge for each unmatched vertex. It is not hard to see that a minimum edge cover of this graph corresponds exactly to the smallest $(1, 1)$ edge dominating set that satisfies the specifications of the sets V^+, V^-, V^{+-}.

To see that the running time of our algorithm is $O^*(9^k)$ we observe that there are two branching steps: either we have an arc (u, v) with both endpoints unmarked; or we have an arc with exactly one unmarked endpoint. In both cases we measure the decrease of the quantity $\ell := 2k - (|V^+| + |V^-| + |V^{+-}|)$. The first case produces two instances with $\ell' := \ell - 1$ (2a,2c), and three instances with $\ell' := \ell - 2$. We therefore have the recurrence $T(\ell) \leq 2T(\ell - 1) + 3T(\ell - 2)$ which gives $T(\ell) \leq 3^\ell$. For the second case, we have three branches, all of which decrease ℓ, therefore we also have $T(\ell) \leq 3^\ell$ in this case. Taking into account that, initially $\ell = 2k$ we get a running time of at most $O^*(9^k)$. ◀

▶ **Theorem 3.** *The* $(0, 1)$-dEDS *problem parameterized by solution size* k *can be solved in time* $O^*(2^k)$.

3.2 Approximation algorithms

We present here constant-factor approximation algorithms for $(0, 1)$-dEDS, and $(1, 1)$-dEDS. Both algorithms appropriately utilize a maximal matching.

▶ **Theorem 4.** *There are polynomial-time* 3*-approximation algorithms for* $(0, 1)$-dEDS.

▶ **Theorem 5.** *There is a polynomial-time* 8*-approximation algorithm for* $(1, 1)$-dEDS.

Proof. Let $G = (V, E)$ be an input directed graph. We partition V into (S, R, T) so that S and T are the sets of sources and sinks respectively, and $R = V \setminus S \setminus T$. We construct an $(1, 1)$-edge dominating set K as follows.

1. Add the arc set $\delta(S,T)$ to K.
2. For each vertex of $v \in R \cap N^+(S)$, choose precisely one arc from $\delta^+(v)$ and add it to K.
3. For each vertex of $v \in R \cap N^-(T)$, choose precisely one arc from $\delta^-(v)$ and add it to K.
4. Let $G' = (R, E')$ be the subdigraph of G whose arc set consists of those arcs not $(1,1)$-dominated by K thus far constructed. Let M be a maximal matching in (the underlying graph of) G'. Let M^- and M^+ be respectively the tails and heads of the arcs in M. To K, we add all arcs of M, an arc of $\delta^-_G(v)$ for every $v \in M^-$, and also an arc of $\delta^+_G(v)$ for every $v \in M^+$.

Clearly, the algorithm runs in polynomial time. In particular, for any vertex v considered at Step 2-4, both $\delta^+(v)$ and $\delta^-(v)$ are non-empty and choosing an arc from a designated set is always possible. We show that K is indeed an $(1,1)$-edge dominating set. Suppose that an arc (u,v) is not $(1,1)$-dominated by K. As the first, second and third step of the construction ensures that any arc incident with $S \cup T$ is $(1,1)$-dominated, we know that (u,v) is contained in the subdigraph G' constructed at step 4. For $(u,v) \notin M$ and M being a maximal matching, one of the vertices u, v must be incident with M. Without loss of generality, we assume v is incident with M (and the other cases are symmetric). If $v \in M^-$, then clearly the arc $e \in M$ whose tail coincides with v would $(1,0)$-dominate (u,v), a contradiction. If $v \in M^+$, then the outgoing arc of v added to K at step 4 would $(1,0)$-dominate (u,v), again reaching a contradiction. Therefore, the constructed set K is a solution to $(1,1)$-dEDS.

To prove the claimed approximation ratio, we first note that $\delta(S,T)$ is contained in any (optimal) solution because any arc of $\delta(S,T)$ can be $(1,1)$-dominated only by itself. Note that these arcs do not $(1,1)$-dominate any other arcs of G. Further, we have $|R \cap N^+(S)| \leq OPT_{(1,1)dEDS} - |\delta(S,T)|$ because in order to $(1,1)$-dominate any arc of the form (s,r) with $s \in S$ and $r \in R$, one must take at least one arc from $\{(s,r)\} \cup \delta^+(r)$. Since the collection of sets $\{(s,r) : s \in S\} \cup \delta^+(r)$ are disjoint over all $r \in R \cap N^+(S)$, the inequality holds. Likewise, it holds that $|R \cap N^-(T)| \leq OPT_{(1,1)dEDS} - |\delta(S,T)|$. In order to $(1,1)$-dominate the entire arc set M, one needs to take at least $|M|/2$ arcs. This is because an arc e can $(1,1)$-dominate at most two arcs of M. That is, we have $|M|/2 \leq OPT_{(1,1)dEDS} - |\delta(S,T)|$ Therefore, it is $|K| \leq |\delta(S,T)| + |R \cap N^+(S)| + |R \cap N^-(T)| + 3|M| \leq 8OPT_{(1,1)dEDS}$. ◀

3.3 Polynomial kernels

We give polynomial kernels for $(1,1)$-dEDS and $(0,1)$-dEDS. We first introduce a relation between the vertex cover number and the size of a minimum $(1,1)$-edge dominating set, shown in [22] and then proceed to show a quadratic-vertex/cubic-edge kernel for $(1,1)$-dEDS.

▶ **Lemma 6** ([22])**.** *Given a directed graph G, let G^* be the undirected underlying graph of G, $vc(G^*)$ be the vertex cover number of G^*, and K be a minimum $(1,1)$-edge dominating set in G. Then $vc(G^*) \leq 2|K|$.*

▶ **Theorem 7.** *There exists an $O(k^2)$-vertex/$O(k^3)$-edge kernel for $(1,1)$-dEDS.*

Proof. Given a directed graph G, we denote the underlying undirected graph of G by G^*. Let K be a minimum $(1,1)$-edge dominating set and $\text{vc}(G^*)$ be the size of a minimum vertex cover in G^*. First, we find a maximal matching M in G^*. If $|M| > 2k$, we conclude this is a no-instance by Lemma 6 and the well-known fact that $|M| \leq \text{vc}(G^*)$ [20]. Otherwise, let S be the set of endpoints of edges in M. Then S is a vertex cover of size at most $4k$ for the underlying undirected graph of G and $V \setminus S$ is an independent set.

We next explain the reduction step. For each $v \in S$, we arbitrarily mark the first $k+1$ tail vertices of incoming arcs of v with "in" (or all, if the in-degree of v is $\leq k$) and also arbitrarily the first $k+1$ head vertices of outgoing arcs of v with "out" (or all, if the out-degree of v is

$\leq k$). After this marking, if there exists a vertex $u \in V \setminus S$ without marks "in", "out", we can delete it. We next show correctness. First, we can observe that if some $v \in S$ has more than $k+1$ incoming arcs, they must be dominated by an outgoing arc of v. Similarly, if $v \in S$ has more than $k+1$ outgoing arcs, they must be dominated by an incoming arc of v. This means that every arc incident on an unmarked vertex u must be dominated because each vertex v in S adjacent to u has at least $(k+1)$ incoming arcs other than (u, v), or $(k+1)$ outgoing arcs other than (v, u), due to the fact that u is unmarked. Moreover, for an incoming (resp. outgoing) arc of u, there exists an outgoing (resp. incoming) arc of $v \in S$ that dominates all arcs dominated by the incoming (resp. outgoing) arc of u except for arcs incident on u. Thus we need not include any arc incident on u in the solution. By the reduction step, we obtain the reduced graph.

From the above, the size of an independent set, being the subset of $V \setminus S$, is bounded by $4k \cdot 2(k+1) = 8k^2 + 8k$, following the reduction step. Thus, the number of vertices in the reduced graph is at most $4k + 8k^2 + 8k = 8k^2 + 12k$. Moreover, there exist at most $4k \cdot (8k^2 + 12k) = 32k^3 + 48k^2$ arcs between the sets of the vertex cover and the independent set. Therefore, the number of arcs in the reduced graph is at most $\binom{4k}{2} + 32k^3 + 48k^2 = 32k^3 + 56k^2 - 2k$. ◀

Using a more strict relation between vc and the size of a minimum $(0, 1)$-edge dominating set, we obtain a linear-vertex/quadratic-edge kernel for $(0, 1)$-dEDS.

▶ **Theorem 8.** *There exists an $O(k)$-vertex/$O(k^2)$-edge kernel for $(0, 1)$-dEDS.*

4 W[1]-hardness by treewidth

In this section we characterize the complexity of (p, q)-dEDS parameterized by treewidth. Our main result is that, even though the problem is FPT when parameterized by $p+q+\text{tw}$, it becomes W[1]-hard if parameterized only by tw. The algorithm is based on standard dynamic programming techniques, while for hardness we reduce from the k-Multicolored Clique problem, which is defined as follows: given a graph $G = (V, E)$, with V partitioned into k independent sets $V = V_1 \uplus \cdots \uplus V_k$, $|V_i| = n, \forall i \in [1, k]$, we are asked to find a subset $S \subseteq V$, such that $G[S]$ forms a clique with $|S \cap V_i| = 1, \forall i \in [1, k]$. The problem k-Multicolored Clique is well-known to be W[1]-complete [16].

Construction. Given an instance $[G = (V, E), k]$ of k-Multicolored Clique, with $V = \bigcup_{\forall i \in [1,k]} V_i$ and $V_i = \{v_0^i, \ldots, v_{n-1}^i\}$ we will construct an instance $[G' = (V', E'), \text{tw}(G')]$ of (p, q)-dEDS parameterized by the treewidth of the underlying undirected graph, with $p = q = 2n$, as follows. We first make k *main cycles* on n vertices $V_i' = \{u_0^i, \ldots, u_{n-1}^i\}$, $\forall i \in [1, k]$, each corresponding to a set $V_i \subseteq V$ and we associate each vertex $v_l^i \in V_i$ with the arc (u_l^i, u_{l+1}^i) from cycle V_i' (its *corresponding* arc). Let $\bar{E}$ be the set of *non-edges* between vertices from different sets from G, i.e. the set of all pairs $(v_l^i, v_o^j) \notin E$.

For each $(v_l^i, v_o^j) \in \bar{E}$ with $i < j$, we will create the following *cross-gadget* $\hat{C}_{l,o}^{i,j}$: we first make five new vertices $a_{l,o}^{i,j}$, $b_{l,o}^{i,j}$, $c_{l,o}^{i,j}$, $d_{l,o}^{i,j}$ and $e_{l,o}^{i,j}$ and then add arcs from $a_{l,o}^{i,j}$ and $c_{l,o}^{i,j}$ to $e_{l,o}^{i,j}$ and from $e_{l,o}^{i,j}$ to $b_{l,o}^{i,j}$ and $d_{l,o}^{i,j}$. We let set $Q_{l,o}^{i,j}$ contain all four of these arcs and refer to them as the *cross-arcs*. We also add both arcs between $a_{l,o}^{i,j}$ and $c_{l,o}^{i,j}$, as well as both arcs between $b_{l,o}^{i,j}$ and $d_{l,o}^{i,j}$. These are referred to as the *flip-arcs*. Finally, we add a path of length $4n-2$ from $b_{l,o}^{i,j}$ to $a_{l,o}^{i,j}$ and a path of length $4n-2$ from $d_{l,o}^{i,j}$ to $c_{l,o}^{i,j}$ (on $4n-3$ new vertices each). We call these the *long paths*.

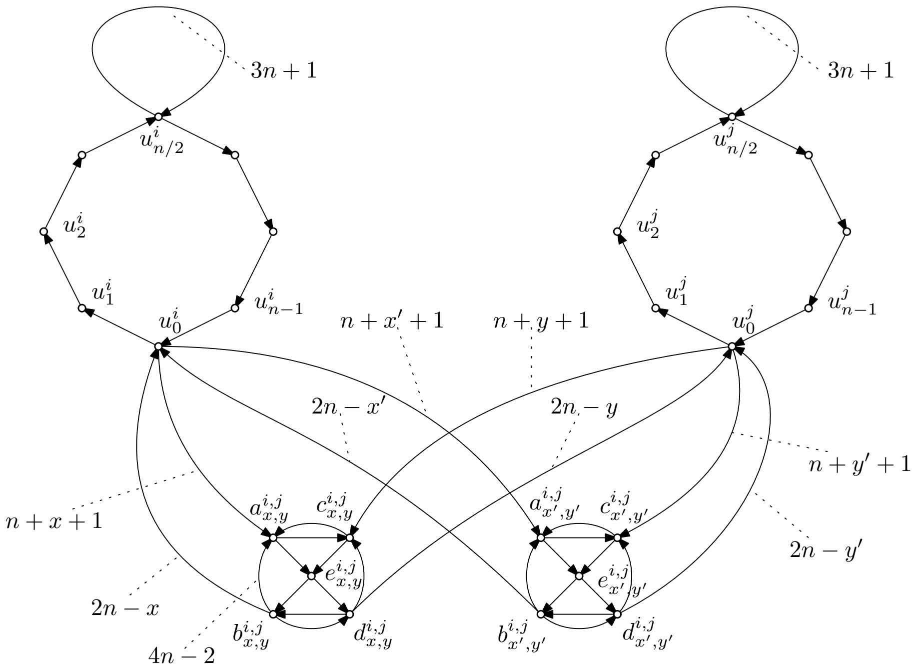

Figure 1 An example of our construction (even n). Dotted lines show the length of each path.

To connect each gadget to the main cycles, we then add a path of length $n+l+1$ (with $n+l$ new vertices) from u^i_0 to $a^{i,j}_{l,o}$ and a path of length $2n-l$ (with $2n-l-1$ new vertices) from $b^{i,j}_{l,o}$ to u^i_0. We also add a path of length $n+o+1$ from u^j_0 to $c^{i,j}_{l,o}$ and a path of length $2n-o$ from $d^{i,j}_{l,o}$ to u^j_0.

Finally, in order to ensure any $(2n,2n)$-edge dominating set will select at least one arc from each of the k main cycles, we will attach a *guard cycle* to each *middle* vertex of each V'_i: the middle vertex of V'_i is $u^i_{n/2}$ and we attach a cycle of length $3n+1$ to it.[4] This concludes our construction and Figure 1 provides an illustration. Clearly, the construction requires polynomial time.

▶ **Lemma 9.** *If G has a k-multicolored clique of size k, then G' has a $(2n,2n)$-edge dominating set of size $|\bar{E}|+k$.*

▶ **Lemma 10.** *If G' has a $(2n,2n)$-edge dominating set of size $|\bar{E}|+k$, then G has a k-multicolored clique of size k.*

▶ **Theorem 11.** *The (p,q)-dEDS problem is W[1]-hard parameterized by the treewidth of the input graph.*

▶ **Theorem 12.** *The (p,q)-dEDS problem can be solved in time $O^*((p+q)^{O(tw)})$ on graphs of treewidth at most tw.*

Proof (Sketch). The proof relies on standard techniques (Dynamic Programming over tree decompositions), so we only sketch the details here. Our algorithm maintains a table for each node of the given tree decomposition, indexed by a set of *state-assignments* to all vertices in

[4] We assume, without loss of generality, that n is even as we can always add a dummy vertex to each subset V_i.

the bag, each entry of which contains the minimum number of selected arcs from the node's terminal subgraph for the state of each vertex to be justified, i.e. for the partial solution described by this set of states to be valid. The state of each vertex in the bag describes its distance to the closest endpoint of a selected arc, i.e. it either has a path of length at most p to the tail of a selected arc, or the head of a selected arc has a path of length at most q to the vertex in question. We also use "promise" states signifying that the partial solution has not yet selected the arc that will be closest to some vertex, by doubling the amount of states we use. It is not hard to see that using such a state representation, we can compute the values of all partial solutions for the problem over the nodes of the tree decomposition in time polynomial on the table's size: the states of introduced vertices must match the distances in the node's subgraph, all partial solutions involving a forgotten vertex must be compared over all its states to retain the minimum, while for join nodes, the state of a vertex must match the "promise" state for the same vertex in the other branch of the join for the partial solutions to be accurately extended. In this way we can check the values of potential global solutions in the table of the root node of the tree decomposition. ◀

5 On Tournaments

A complete complexity classification for the problems (p, q)-dEDS is presented in this section. For $p = q = 1$, the problem is NP-hard under a randomized reduction while being amenable to an FPT algorithm and polynomial kernelization due to the results of Sections 3.1 and 3.3. The hardness reduction is given in Subsection 5.1. When $p = 2$ or $q = 2$, the complexity status of (p, q)-dEDS is equivalent to Dominating Set on tournaments and is discussed in Subsection 5.2. In the remaining cases, when $p + q \leq 1$, or $\max\{p, q\} \geq 3$ while neither of them equals 2, the problems turn out to be in P (Subsection 5.3).

5.1 Hard: when $p = q = 1$

We present a randomized reduction from Independent Set to $(1, 1)$-dEDS. Our reduction preserves the size of the instance up to polylogarithmic factors; as a result it shows that $(1, 1)$-dEDS does not admit a $2^{n^{1-\epsilon}}$ algorithm, under the randomized ETH. Furthermore, our reduction preserves the optimal value, up to a factor $(1 - o(1))$; as a result, it shows that $(1, 1)$-dEDS is APX-hard under randomized reductions.

Before moving on, let us give a high-level overview of our reduction. The first step is to reduce Independent Set to Almost Induced Matching, the problem of finding the maximum set of vertices that induce a graph of maximum degree 1. Our reduction produces an instance of Almost Induced Matching that has several special properties, notably producing a bipartite graph $G = (A, B, E)$. The basic strategy will be then to construct a tournament $T = (V', E')$, where $V' = A \cup B \cup C$, where C is a set of new vertices. All edges of E will be directed from A to B, non-edges of E will be directed from B to A, and all other edges will be set randomly. This intuitively encodes the structure of G in T. The idea is now that a solution S in G (that is, a set of vertices of G that induces a graph with maximum degree 1) will correspond to an edge dominating set in T where all vertices except those of S will have total degree 2, and the vertices of S will have total degree 1. In particular, vertices of $S \cap A$ will have out-degree 1 and in-degree 0, and vertices of $S \cap B$ will have in-degree 1 and out-degree 0.

The random structure of the remaining arcs of the tournament T is useful in two respects: in one direction, given the solution S for G, it is easy to deal with vertices that have degree 1 in $G[S]$: we select the corresponding arc from A to B in T. For vertices of degree 0 however, we are forced to look for edge-disjoint paths that will allow us to achieve our degree goals. Such paths are guaranteed to exist if C is random and large enough. In the other direction, given a good solution in T, we would like to guarantee that, because the internal structure of A, B, and C is chaotic, the only way to obtain a large number of vertices with low degree is to place those with in-degree 0 in A, and those with out-degree 0 in B.

▶ **Theorem 13.** $(1,1)$-dEDS *on tournaments cannot be solved in polynomial time, unless NP ⊆ BPP. Furthermore,* $(1,1)$-dEDS *is APX-hard under randomized reductions, and does not admit an algorithm running in time* $2^{n^{1-\epsilon}}$ *for any* ϵ*, unless the randomized ETH is false.*

We first reduce the INDEPENDENT SET problem on cubic graphs to the following intermediate problem called ALMOST INDUCED MATCHING, commonly known as MAXIMUM DISSOCIATION NUMBER in the literature [33, 31]. A subgraph of G induced on a vertex set $S \subseteq V$ is called an *almost induced matching*, if every vertex $v \in S$ has degree ≤ 1 in $G[S]$.

▶ **Definition 14.** The problem ALMOST INDUCED MATCHING (AIM) takes as input an undirected graph $G = (V, E)$. The goal is to find an almost induced matching having the maximum number of vertices.

▶ **Theorem 15.** *[1, 10]* INDEPENDENT SET *is APX-hard on cubic graphs. Furthermore,* INDEPENDENT SET *cannot be solved in time* $2^{o(n)}$ *unless the ETH is false.*

ALMOST INDUCED MATCHING is known to be NP-complete on bipartite graphs of maximum degree 3 and on C_4-free bipartite graphs [6]. It is also NP-hard to approximate on arbitrary graphs within a factor of $n^{1/2-\epsilon}$ for any $\epsilon > 0$ [28]. The next lemma supplements the known hardness results on bipartite graphs and might be of independent interest.

▶ **Lemma 16.** ALMOST INDUCED MATCHING *is APX-hard and cannot be solved in time* $2^{o(n)}$ *under the ETH, even on bipartite graphs of degree at most 4. Furthermore, this hardness still holds if we are promised that* $OPT_{AIM} > 0.6n$ *and that there is an optimal solution* S *that includes at least* $n/20$ *vertices with degree* 0 *in* $G[S]$*.*

As we use a random construction, the following property of a uniform random tournament is useful. Intuitively, the property established in Lemma 17 states that it is impossible in a large random tournament to have two large sets of vertices X, Y such that all vertices of X have in-degree 0 and out-degree 1 in a $(1,1)$-edge dominating set, while all vertices of Y have in-degree 1 and out-degree 0.

▶ **Lemma 17.** *Let* $T = (V, E)$ *be a random tournament on the vertex set* $\{1, 2, \dots, n\}$*, in which* (i, j) *is an arc of* T *with probability* $1/2$*. Then the following event happens with high probability: for any two disjoint sets* $X, Y \subseteq V$ *with* $|X| > (\log n)^2$ *and* $|Y| > (\log n)^2$*, there exists a vertex* $x \in X$ *with at least two outgoing arcs to* Y*.*

▶ **Lemma 18.** *Let* $G = (V = A \dot\cup B \dot\cup C, E)$ *be a random directed graph with* $|A| = |B| = n$ *and* $|C| = 4n$ *such that for any pair* (x, y) *with* $\{x, y\} \cap C \neq \emptyset$ *we have exactly one arc, oriented from* x *to* y*, or from* y *to* x *with probability* $1/2$*. Let* $\ell \geq n/20$ *be a positive integer. Then with high probability, we have: for any two disjoint sets* $X \subseteq A, Y \subseteq B$ *with* $|X| = |Y| = \ell$*, there exist* ℓ *vertex-disjoint directed paths from* X *to* Y*.*

▶ **Theorem 19.** *There is a probabilistic polynomial-time algorithm computing, given an instance* G *of* ALMOST INDUCED MATCHING*, an instance* T *of* $(1,1)$-dEDS *such that with high probability:*

(i) *if* $OPT_{AIM}(G) \geq L_1$*, then* $OPT_{(1,1)dEDS}(T) \leq |V(T)| - L_1/2 + 1$*,*
(ii) *if* $OPT_{AIM}(G) < L_2 - 5(\log L_2)^2$*, then* $OPT_{(1,1)dEDS}(T) > |V(T)| - L_2/2 + 1$*.*

Proof of Theorem 13. Let G be an instance of Independent Set on cubic graphs and let G' be the instance of Almost Induced Matching obtained by the construction of Lemma 16. We set ℓ as in the reduction and observe that $OPT_{IS}(G) \geq k$ if and only if $OPT_{AIM}(G') \geq \ell$.

Let G^* be a disjoint union of $10(\log \ell)^2$ copies of G'. Then G^* is a gap instance, whose optimal solution is either at least $10\ell(\log \ell)^2$, or at most $10\ell(\log \ell)^2 - 10(\log \ell)^2 \leq L - 5(\log L)^2$, where $L := 10\ell(\log \ell)^2$. Now Theorem 19 implies that using a probabilistic polynomial-time algorithm for $(1,1)$-dEDS with two-sided bounded errors, one can correctly decide an instance of Independent Set on cubic graphs with bounded errors. We observe that the size of the instance has only increased by a poly-logarithmic factor, hence an algorithm solving the new instance in time $2^{n^{1-\epsilon}}$ would give a randomized sub-exponential time algorithm for 3-SAT.

Finally, for APX-hardness, we observe that we may assume we start our reduction from an Independent Set instance where either $OPT_{IS} \geq k$ or $OPT_{IS} < rk$, for some constant $r < 1$, and for $k = \Theta(n)$. Lemma 16 then gives an instance of Almost Induced Matching where either $OPT_{AIM} \geq L_1$ or $OPT_{AIM} \leq r' L_1 = L_2$, for some (other) constant $r' < 1$. We now use Theorem 19 to create a gap-instance of $(1,1)$-dEDS. ◀

5.2 Equivalent to Dominating Set on tournaments: $p = 2$ or $q = 2$

▶ **Lemma 20.** *On tournaments without a source, we have* $OPT_{(0,2)dEDS} \leq OPT_{DS}$.

Proof. Let $T = (V,E)$ be a tournament with no source and $D \subseteq V$ be a dominating set of T. Then let $K \subseteq E$ be a set containing one arbitrary incoming arc of every vertex in D. We claim K $(0,2)$-dominates all arcs in E: since D is a dominating set, for any vertex $u \notin D$ there must be an arc (v,u) from some $v \in D$. Thus all outgoing arcs (u,w) from such $u \notin D$ are $(0,2)$-dominated by K, as are all arcs (v,u) from $v \in D$. ◀

▶ **Lemma 21.** *Let $T = (V,E)$ be a tournament and let s be a source of T. Then $\delta^+(s)$ is an optimal (p,q)-edge dominating set of T for any $p \leq 1$ and $q \geq 1$.*

Proof. Since s has no incoming arcs, any (p,q)-edge dominating set must select at least one arc from $\{(s,v)\} \cup \delta^+(v)$ for every $v \in V \setminus \{s\}$ in order to (p,q)-dominate (s,v). Because the arc sets $\{(s,v)\} \cup \delta^+(v)$ are mutually disjoint over all $v \in V \setminus \{s\}$, any (p,q)-edge dominating set has size at least $|\delta^+(s)|$. Now, observe that $\delta^+(s)$ $(0,1)$-dominates every arc of T. ◀

▶ **Lemma 22.** *On tournaments on n vertices, for any $p \geq 2$ we have:* $OPT_{(p,2)dEDS} \leq OPT_{(2,2)dEDS} \leq 2\log n + 3$.

Proof. The first inequality trivially holds, so we prove the second inequality. Let $T = (V,E)$ be a tournament on n vertices. If T has no source, then $OPT_{(2,2)dEDS} \leq OPT_{(0,2)dEDS} \leq OPT_{DS} \leq \log n + 1$, where the second and the last inequality follow from Lemma 20 and Lemma 1, respectively. If T^{rev} contains no source, observe that a $(0,2)$-edge dominating set of T^{rev} is a $(2,0)$-edge dominating set of T and the statement holds.

Therefore, we may assume that T has a source s and a sink t. Let $S_1 \subseteq V \setminus \{s\}$ be a dominating set of $T - s$ of size at most $\log n + 1$. Clearly, every arc (u,v) of $T - s$ lies on a directed path of length at most two from some vertex of S_1. Let $D_1 \subseteq E$ be a minimal arc set such that $D_1 \cap \delta^-(v) \neq \emptyset$ for every $v \in S_1$. Since every $v \in S_1$ has positive in-degree, such a set D_1 exists and we have $|D_1| \leq |S_1|$. Observe that D_1 (0,2)-dominates every arc of $T - s$. Applying a symmetric argument to $T^{rev} - t$, we know that there exists an arc set D_2 of size at most $\log n + 1$ which $(2,0)$-dominates every arc of $T - t$. Now $D_1 \cup D_2$ (2,2)-dominates every arc incident with $V \setminus \{s,t\}$. Therefore, $D_1 \cup D_2 \cup \{(s,t)\}$ is a $(2,2)$-dEDS. ◀

▶ **Lemma 23.** *There is an FPT reduction from* Dominating Set *on tournaments parameterized by solution size to* (p, q)-EDS *parameterized by solution size, when* $p = 2$ *or* $q = 2$.

Proof. Without loss of generality we assume that $q = 2$. Let $T = (V, E)$ be an input tournament to Dominating Set, and let k be the solution size. It can be assumed that T has no source. We construct a tournament T' on vertex set $V \cup \{t\}$, in which t is a sink. Given a dominating set D of T, we select an arbitrary arc set K of T' so that $\delta^-_K(v) = 1$ for each $v \in D$. It is easy to see that K $(0, 2)$-dominates every arc of T': any arc (u, v) with $u \in D$ is clearly dominated by K. For any arc (u, v) with $u \notin D$, there is $w \in D$ such that $(w, u) \in E$ and thus K $(0, 2)$-dominates (u, v).

Conversely, suppose that K is a $(p, 2)$-edge dominating set of size at most k and let K^+ be the set of heads of K found in V. Let K^- be the set of vertices $u \in V$ such that $(u, t) \in K$. We have $|K^+ \cup K^-| \leq k$, because each arc of K either contributes an element in K^+ or in K^-. We claim that $K^+ \cup K^-$ is a dominating set of T. Suppose the contrary, therefore there exists $u \in V \setminus (K^+ \cup K^-)$ that is not dominated by $K^+ \cup K^-$. However, the arc (u, t) is dominated by K. We have $(u, t) \notin K$, as $u \notin K^-$. Therefore, since t is a sink, (u, t) is $(0, 2)$-dominated by an arc $(v, w) \in K$. This means that either $w = u$, or the arc (w, u) exists. However, $w \in K^+$, which means that u is dominated. ◀

▶ **Theorem 24.** *On tournaments, the problems* $(p, 2)$-dEDS *are W[2]-hard for each fixed* p.

Proof. For all problems, we use the reduction from Set Cover to Dominating set on Tournaments given in Theorem 13.14 of [10] and our results follow from the W[2]-hardness of that problem (see also Theorem 13.28 therein) and Lemma 23. ◀

▶ **Theorem 25.** *On tournaments, the problems* $(0, 2)$-dEDS, $(1, 2)$-dEDS *and* $(2, 2)$-dEDS *can be solved in time* $n^{O(\log n)}$.

Proof. For $(0, 2)$-dEDS and $(1, 2)$-dEDS, the case when a given tournament contains a source can be solved in polynomial time by Lemma 21. If the input tournament contains no source, then by Lemma 20 we have $OPT_{(1,2)dEDS} \leq OPT_{(0,2)dEDS} \leq OPT_{DS}$, which is bounded by $\log n + 1$ by Lemma 1. Lemma 22 states that $OPT_{(p,2)dEDS} \leq 2 \log n + 3$. Exhaustive search over vertex subsets of size $O(\log n)$ performs in the claimed runtime. ◀

5.3 P-time solvable: $p + q \leq 1$ or, $2 \notin \{p, q\}$ and $\max\{p, q\} \geq 3$

▶ **Theorem 26.** $(0, 1)$-dEDS *can be solved in polynomial time on tournaments.*

Proof. We will show that $OPT_{(0,1)dEDS} = n - 1$ and give a polynomial-time algorithm for finding such an optimal solution. First, given a tournament $T = (V, E)$, to see why $OPT_{(0,1)dEDS} \geq n - 1$ consider any optimal solution $K \subseteq E$: if there exists a pair of vertices $u, v \in V$ with $d^-_K(u) = d^-_K(v) = 0$, i.e. a pair of vertices, neither of which has an arc of K as an incoming arc, then the arc between them (without loss of generality let its direction be (v, u)) is not dominated: as $d^-_K(u) = 0$, the arc itself does not belong in K and as $d^-_K(v) = 0$, there is no arc preceding it that is in K. This leaves (v, u) undominated. Therefore, there cannot be two vertices with no incoming arcs in any optimal solution, implying any solution must include at least $n - 1$ arcs.

To see $OPT_{(0,1)dEDS} \leq n-1$, consider a partition of T into strongly connected components $C_1, \ldots, C_l$, where we can assume these are given according to their topological ordering, i.e. for $1 \leq i < j \leq l$, all arcs between C_i and C_j are directed towards C_j. Let S be the set of arcs traversed in breadth-first-search (BFS) from some vertex $s \in C_1$ until all vertices of C_1 are spanned. Also let S' be the set of arcs $(s, u), \forall u \in C_i, \forall i \in [2, l]$, i.e. all outgoing arcs from s to every vertex of $C_2, \dots, C_l$. Note that set S' must contain an arc from s to *every* vertex that is not in C_1: T being a tournament means every pair of vertices has an arc between them and C_1 being the first component in the topological ordering means all arcs between its vertices and those of subsequent components are oriented away from C_1. Then $K := S \cup S'$ is a directed $(0, 1)$-edge dominating set of size $n - 1$ in T: observe that $d^-_K(u) = 1, \forall u \neq s \in T$, i.e. every vertex in T has positive in-degree within K except s. Thus all outgoing arcs from all such vertices u are $(0, 1)$-dominated by K, while all outgoing arcs from s are in K, due to the BFS selection for S and the definition of S'.

Since such an optimal solution K can be computed in polynomial time (partition into strongly connected components, BFS), the claim follows. ◀

▶ **Theorem 27.** *For any p, q with $\max\{p, q\} \geq 3$, $p \neq 2$ and $q \neq 2$, (p, q)-dEDS can be solved in polynomial time on tournaments.*

Proof. Suppose without loss of generality that $q \geq 3$, as otherwise we can solve (q, p)-dEDS on T^{rev}, the tournament obtained by reversing the orientation of every arc. In any tournament T, there always exists a *king* vertex, that is, a vertex with a path of length at most 2 to any other vertex in the graph. One such vertex is the vertex of maximum out-degree v. If v is not a source, it suffices to select one of its incoming arcs: since there is a path of length at most 2 from v to any other vertex u in the graph, any outgoing arc from any such u will be $(0, 3)$-dominated by this selection. This is clearly optimal.

Suppose now that s is a source. We consider two cases: if $p \leq 1$, then Lemma 21 implies that $\delta^+(s)$ is optimal. Finally, suppose s is a source and $p \geq 3$. If T does not have a sink, then a king of T^{rev} has an incoming arc, which $(0, 3)$-dominates T^{rev} as observed above, and thus T has a $(0, 3)$-edge dominating set of size 1.

Therefore, we may assume that T has both a source s and a sink t. Let s' and t' be vertices of $V \setminus \{s, t\}$ with maximum out- and in-degree, respectively. Now $\{(s, t), (s, s'), (t', t)\}$ is a $(3, 3)$-edge dominating set. This is because s' is a king of $T - s$ and thus every arc (u, v) with $u \neq s$ is $(0, 3)$-dominated by (s, s'). Similarly, every arc (u, v) with $v \neq t$ is $(3, 0)$-dominated by (t', t). The only arc not $(3, 3)$-dominated by these two arcs is (s, t), which is dominated by itself. Examining all vertex subsets of size up to 3, we can compute an optimal $(3, 3)$-edge dominating set in polynomial time. ◀

References

1 Paola Alimonti and Viggo Kann. Some apx-completeness results for cubic graphs. *Theor. Comput. Sci.*, 237(1-2):123–134, 2000.

2 Brenda S. Baker. Approximation Algorithms for NP-Complete Problems on Planar Graphs. *J. ACM*, 41(1):153–180, 1994.

3 Daniel Binkele-Raible and Henning Fernau. Enumerate and measure: Improving parameter budget management. In *IPEC*, volume 6478 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2010.

4 Arindam Biswas, Varunkumar Jayapaul, Venkatesh Raman, and Srinivasa Rao Satti. The Complexity of Finding (Approximate Sized) Distance-d Dominating Set in Tournaments. In *Frontiers in Algorithmics*, pages 22–33, 2017.

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

6 Rodica Boliac, Kathie Cameron, and Vadim V. Lozin. On computing the dissociation number and the induced matching number of bipartite graphs. *Ars Comb.*, 72, 2004.

7 Glencora Borradaile and Hung Le. Optimal dynamic program for r-domination problems over tree decompositions. In *IPEC*, volume 63 of *LIPIcs*, pages 8:1–8:23. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2016.

8 Jean Cardinal, Stefan Langerman, and Eythan Levy. Improved approximation bounds for edge dominating set in dense graphs. *Theor. Comput. Sci.*, 410(8-10):949–957, 2009.

9 Miroslav Chlebík and Janka Chlebíková. Approximation hardness of edge dominating set problems. *Journal of Combinatorial Optimization*, 11(3):279–290, 2006.

10 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

11 Erik D. Demaine, Fedor V. Fomin, Mohammad Taghi Hajiaghayi, and Dimitrios M. Thilikos. Fixed-parameter algorithms for (k, r)-center in planar graphs and map graphs. *ACM Trans. Algorithms*, 1(1):33–47, 2005.

12 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *STOC*, pages 624–633. ACM, 2014.

13 Rodney G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness i: Basic results. *SIAM J. Comput.*, 24(4):873–921, 1995.

14 Rodney G. Downey and Michael R. Fellows. Parameterized Computational Feasibility. In *Feasible Mathematics II*, pages 219–244, 1995.

15 David Eisenstat, Philip N. Klein, and Claire Mathieu. Approximating k-center in planar graphs. In *SODA*, pages 617–627. SIAM, 2014.

16 Michael R. Fellows, Danny Hermelin, Frances A. Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theor. Comput. Sci.*, 410(1):53–61, 2009.

17 Henning Fernau. edge dominating set: Efficient Enumeration-Based Exact Algorithms. In *Parameterized and Exact Computation*, pages 142–153. Springer Berlin Heidelberg, 2006.

18 Fedor V. Fomin, Serge Gaspers, Saket Saurabh, and Alexey A. Stepanov. On two techniques of combining branching and treewidth. *Algorithmica*, 54(2):181–207, 2009.

19 Toshihiro Fujito and Hiroshi Nagamochi. A 2-approximation algorithm for the minimum weight edge dominating set problem. *Discrete Applied Mathematics*, 118(3):199–207, 2002.

20 Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Co., 1979.

21 Torben Hagerup. Kernels for edge dominating set: Simpler or smaller. In *MFCS*, volume 7464 of *Lecture Notes in Computer Science*, pages 491–502. Springer, 2012.

22 Tesshu Hanaka, Naomi Nishimura, and Hirotaka Ono. On directed covering and domination problems. In *ISAAC*, volume 92 of *LIPIcs*, pages 45:1–45:12. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017.

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

24 Joseph D. Horton and Kyriakos Kilakos. Minimum edge dominating sets. *SIAM J. Discret. Math.*, 6(3):375–387, 1993.

25 Ken Iwaide and Hiroshi Nagamochi. An improved algorithm for parameterized edge dominating set problem. *J. Graph Algorithms Appl.*, 20(1):23–58, 2016.

26 Stephan Kreutzer and Siamak Tazari. Directed nowhere dense classes of graphs. In *SODA '12*, pages 1552–1562, 2012.

27 Dana Moshkovitz. The projection games conjecture and the np-hardness of ln n-approximating set-cover. *Theory of Computing*, 11:221–235, 2015.

28 Yury L. Orlovich, Alexandre Dolgui, Gerd Finke, Valery S. Gordon, and Frank Werner. The complexity of dissociation set problems in graphs. *Discrete Applied Mathematics*, 159(13):1352–1366, 2011.

29 Richard Schmied and Claus Viehmann. Approximating edge dominating set in dense graphs. *Theor. Comput. Sci.*, 414(1):92–99, 2012.

30 Mingyu Xiao, Ton Kloks, and Sheung-Hung Poon. New parameterized algorithms for the edge dominating set problem. *Theor. Comput. Sci.*, 511:147–158, 2013.

31 Mingyu Xiao and Shaowei Kou. Exact algorithms for the maximum dissociation set and minimum 3-path vertex cover problems. *Theor. Comput. Sci.*, 657:86–97, 2017.

32 Mingyu Xiao and Hiroshi Nagamochi. A refined exact algorithm for edge dominating set. *Theor. Comput. Sci.*, 560:207–216, 2014.

33 Mihalis Yannakakis. Node-deletion problems on bipartite graphs. *SIAM J. Comput.*, 10(2):310–327, 1981.

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