Algorithmica manuscript No.
(will be inserted by the editor)

Online Dominating Set

Joan Boyar · Stephan J. Eidenbenz ·
Lene M. Favrholdt · Michal Kotrbčík ·
Kim S. Larsen

Received: date / Accepted: date

Abstract This paper is devoted to the online dominating set problem and its variants. We believe the paper represents the first systematic study of the effect of two limitations of online algorithms: making irrevocable decisions while not knowing the future, and being incremental, i.e., having to maintain solutions to all prefixes of the input. This is quantified through competitive analyses of online algorithms against two optimal algorithms, both knowing the entire

A preliminary version of this paper appeared in the *15th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT)*, LIPIcs, vol. 53, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016, pp. 21:1–21:15. Most of the work was done while the fourth author was at the University of Southern Denmark. The first, third, fourth, and fifth authors were supported in part by the Danish Council for Independent Research, Natural Sciences, grants DFF-1323-00247 and DFF-7014-00041, and the Villum Foundation, grant VKR023219.

Joan Boyar
Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark
E-mail: joan@imada.sdu.dk

Stephan J. Eidenbenz
Los Alamos National Laboratory, Los Alamos, NM, USA
E-mail: eidenben@lanl.gov

Lene M. Favrholdt
Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark
E-mail: lenem@imada.sdu.dk

Michal Kotrbčík
School of Mathematics and Physics, University of Queensland, Brisbane, Australia
E-mail: m.kotrbcik@gmail.com

Kim S. Larsen
Department of Mathematics and Computer Science, University of Southern Denmark, Odense, Denmark
E-mail: kslarsen@imada.sdu.dk

input, but only one having to be incremental. We also consider the competitive ratio of the weaker of the two optimal algorithms against the other.

We consider important graph classes, distinguishing between connected and not necessarily connected graphs. For the classic graph classes of trees, bipartite, planar, and general graphs, we obtain tight results in almost all cases. We also derive upper and lower bounds for the class of bounded-degree graphs. From these analyses, we get detailed information regarding the significance of the necessary requirement that online algorithms be incremental. In some cases, having to be incremental fully accounts for the online algorithm's disadvantage.

1 Introduction

We consider online versions of a number of NP-complete graph problems, *Dominating Set* (DS), and variants hereof. Given an undirected graph $G = (V, E)$ with vertex set V and edge set E, a set $D \subseteq V$ is a *dominating set* for G if for all vertices $u \in V$, either $u \in D$ (containment) or there exists an edge $\{u, v\} \in E$, where $v \in D$ (dominance). The objective is to find a dominating set of minimum cardinality.

In the variant *Connected* Dominating Set (CDS), we add the requirement that D be connected (if G is not connected, D should be connected for each connected component of G). In the variant *Total* Dominating Set (TDS), every vertex must be dominated by another, corresponding to the definition above with the "containment" option removed. We also consider *Independent* Dominating Set (IDS), where we add the requirement that D be independent, i.e., if $\{u, v\} \in E$, then $\{u, v\} \not\subseteq D$. In both this introduction and the preliminaries section, when we refer to Dominating Set, the statements are relevant to all the variants unless explicitly specified otherwise.

The study of Dominating Set and its variants dates back at least to seminal books by König [18], Berge [3], and Ore [20]. The concept of domination readily lends itself to modeling many conceivable practical problems. Indeed, at the onset of the field, Berge [3] mentions a possible application of keeping all points in a network under surveillance by a set of radar stations, and Liu [19] notes that the vertices in a dominating set can be thought of as transmitting stations that can transmit messages to all stations in the network. Several monographs are devoted to domination [13], total domination [14], and connected domination [11], and we refer the reader to these for further details.

We consider *online* [5] versions of these problems. More specifically, we consider the vertex-arrival model where the vertices of the graph arrive one at a time and with each vertex, the edges connecting it to previous vertices are also given. If the online algorithm decides to include a vertex in the set D, this decision is irrevocable. Note, however, that not just a new vertex but also vertices given previously may be added to D at any time. An online algorithm must make this decision without any knowledge about possible future vertices.

Note that, since an online algorithm does not know the size of the input graph, it has to maintain a feasible solution at any time. Since the graph consisting of a single vertex does not have a total dominating set at all and isolated vertices do not dominate any vertices, we allow an online algorithm for TDS to *not* include isolated vertices in the solution, unlike the other variants of DS.

Defining the nature of the irrevocable decisions is a modeling issue, and one could alternatively have made the decision that also the act of *not* including the new vertex in D should be irrevocable, i.e., not allowing algorithms to include already given vertices in D at a later time. The main reason for our choice of model is that it is much better suited for applications such as routing in wireless networks for which domination is intensively studied; see for instance [9] and the citations thereof. Indeed, when domination models a (costly) establishment of some service, there is no reason why *not* establishing a service at a given time should have any inherent costs or consequences, such as preventing one from doing so later. Furthermore, the stricter variant of irrevocability results in a problem for which it becomes next to impossible for an online algorithm to obtain a non-trivial result in comparison with an optimal offline algorithm. Consider, for example, an instance where the adversary starts by giving a vertex followed by a number of neighbors of that vertex. If the algorithm ever rejects one of these neighbors, the remaining part of the sequence will consist of neighbors of the rejected vertex and the neighbors must all be selected. This shows that, using this model of irrevocability, online algorithms for DS or TDS would have to select at least $n-1$ vertices, while the optimal offline algorithm selects at most two. For CDS it is even worse, since rejecting any vertex could result in a disconnected dominating set. A similar observation is made in [17] for this model, though they focus more on a different model, where the vertices are known in advance, and all edges incident to a particular vertex are presented when that vertex arrives.

An online algorithm can be seen as having two characteristics: it maintains a feasible solution at any time, and it has no knowledge about future requests. The first is a consequence of the algorithm not knowing the length of the sequence. We also define a larger class of algorithms: An *incremental* algorithm is an algorithm that maintains a feasible solution at any time. It may or may not know the whole input from the beginning.

We analyze the quality of online algorithms for the dominating set problems using *competitive analysis* [21,15]. Thus, we consider the size of the dominating set an online algorithm computes up against the result obtained by an optimal offline algorithm, OPT.

As something a little unusual in competitive analysis, we are working with two different optimal algorithms. This is with the aim of investigating whether it is predominantly the requirement to maintain feasible solutions or the lack of knowledge of the future which makes the problem hard. Thus, we define OPT^{INC} to be an *optimal incremental* algorithm and OPT^{OFF} to be an *optimal offline* algorithm, i.e., it is given the entire input, and then produces a dominating set for the whole graph. The reason for this distinction is that in order to properly measure the impact of the knowledge of the future, it is necessary that it is the sole difference between the algorithm and $\textsc{Opt}$. Therefore, $\textsc{Opt}$ has to solve the same problem and hence the restriction on $\textsc{Opt}^{\textsc{inc}}$. While such an attention to comparing algorithms to an appropriate $\textsc{Opt}$ already exists in the literature, to the best of our knowledge the focus also on the comparison of different optimum algorithms is a novel aspect of our work. Previous results requiring the optimal offline algorithm to solve the same problem as the online algorithm include [6] which considers *fair* algorithms that have to accept a request whenever possible, and thus require $\textsc{Opt}$ to be fair as well, [7] which studies *k-bounded-space* algorithms for bin packing that have at most k open bins at any time and requires $\textsc{Opt}$ to also adhere to this restriction, and [4] which analyzes the performance of online algorithms for a variant of bin packing against a *restricted offline optimum* algorithm that knows the future, but has to process the requests in the same order as the algorithm under consideration.

Given an input sequence I and an algorithm $\textsc{Alg}$, we let $\textsc{Alg}(I)$ denote the size of the dominating set computed by $\textsc{Alg}$ on I. Then $\textsc{Alg}$ is *c-competitive* if there exists a constant α such that for all input sequences I, $\textsc{Alg}(I) \leq c\,\textsc{Opt}(I)+\alpha$, where $\textsc{Opt}$ may be $\textsc{Opt}^{\textsc{inc}}$ or $\textsc{Opt}^{\textsc{off}}$, depending on the context. The (asymptotic) *competitive ratio* of $\textsc{Alg}$ is the infimum over all such c and we denote this $\mathbb{CR}^{\textsc{inc}}(\textsc{Alg})$ and $\mathbb{CR}^{\textsc{off}}(\textsc{Alg})$, respectively. If the inequality above holds without the additive constant α, the algorithm is said to be *strictly c-competitive*, and the *strict competitive ratio* is the infimum over all such c. When considering competitive ratios that are linear in the input size, n, we will use the strict competitive ratio. This is mainly to avoid technicalities arising from the fact that if an algorithm is n/a-competitive for some constant a, then it is also $(n/a-b)$-competitive for any constant b.

We consider the four dominating set problem variants on various graph types, including trees, bipartite, and general graphs and to some extent planar graphs, obtaining tight results in almost all cases. We also consider graphs of bounded degree, giving upper and lower bounds as a function of the maximum degree, Δ. In all cases, we also consider the online variant where the adversary is restricted to giving the vertices in such a manner that the graph given at any point in time is connected. In this case, the graph is called *always-connected*. One motivation is that graphs in applications such as routing in networks are most often connected.

The results for online algorithms are summarized in Tables 1 and 2. The strict upper bound on the competitive ratio against $\textsc{Opt}^{\textsc{inc}}$ for general graphs is $\frac{n+3}{4}$. Note that for this, and other strict competitive ratios containing n, we ignore the additive constant (in the table), writing $n/4$ in this case. The results for $\textsc{Opt}^{\textsc{inc}}$ against $\textsc{Opt}^{\textsc{off}}$ are identical to the results of Table 2, except that for DS on trees, $\mathbb{CR}^{\textsc{off}}(\textsc{Opt}^{\textsc{inc}})=2$, for DS on always-connected planar graphs, $\mathbb{CR}^{\textsc{off}}(\textsc{Opt}^{\textsc{inc}})=n/2$, and for always-connected bounded-degree graphs, the lower bound that we prove is $\mathbb{CR}^{\textsc{off}}(\textsc{Opt}^{\textsc{inc}}) \geq (\Delta-1)/2$. The results are discussed in the conclusion.

<table>
<tr><th>Graph class</th><th>DS</th><th>CDS</th><th>TDS</th><th>IDS</th></tr>
<tr><td>Trees</td><td>2</td><td colspan="2">1</td><td rowspan="6">1</td></tr>
<tr><td>Bipartite</td><td colspan="2" rowspan="2">$n/4$</td><td>$n/4$</td></tr>
<tr><td>Always-connected bipartite</td><td>$[n/6; 2n/9]$</td></tr>
<tr><td>Bounded-degree</td><td rowspan="2">$[\Delta/2 - 1/4; \Delta]$</td><td>$[\Delta/2; \Delta + 1]$</td><td>$[\Delta/2; \Delta]$</td></tr>
<tr><td>Always-conn. bounded-degree</td><td>$[\Delta/3; \Delta - 1]$</td><td>$[\Delta/3; \Delta]$</td></tr>
<tr><td>General graphs</td><td colspan="3">$n/4$</td></tr>
</table>

Table 1 Bounds on the competitive ratio of any online algorithm with respect to $\textsc{Opt}^{\textsc{inc}}$.

<table>
<tr><th>Graph class</th><th>DS</th><th>CDS</th><th>TDS</th><th>IDS</th></tr>
<tr><td>Trees</td><td>$[2; 3]$</td><td>1</td><td>2</td><td rowspan="3">n</td></tr>
<tr><td>Bipartite</td><td colspan="2">n</td><td rowspan="2">$n/2$</td></tr>
<tr><td>Always-connected bipartite</td><td colspan="2">$n/2$</td></tr>
<tr><td>Bounded-degree</td><td>Δ</td><td>$\Delta + 1$</td><td rowspan="2">$[\Delta - 1; \Delta]$</td><td>Δ</td></tr>
<tr><td>Always-conn. bounded-degree</td><td>$[\Delta - 2; \Delta]$</td><td>$[\Delta - 2; \Delta - 1]$</td><td>$[\Delta - 1; \Delta]$</td></tr>
<tr><td>Planar</td><td colspan="2" rowspan="2">n</td><td rowspan="2">$n/2$</td><td rowspan="2">n</td></tr>
<tr><td>Always-connected planar</td></tr>
</table>

Table 2 Bounds on the competitive ratio of any online algorithm with respect to $\textsc{Opt}^{\textsc{off}}$.

2 Preliminaries

Since we are studying online problems, the order in which vertices are given is important. We assume throughout the paper that the indices of the vertices of G, $v_1, \ldots, v_n$, indicate the order in which they are given to the online algorithm, and we use $\textsc{Alg}(G)$ to denote the size of the dominating set computed by $\textsc{Alg}$ using this ordering. When no confusion can occur, we implicitly assume that the dominating set being constructed by an online algorithm $\textsc{Alg}$ is denoted by D. We use the phrase *select a vertex* to mean that the vertex in question is added to the dominating set in question. We use G_i to denote the subgraph of G induced by $\{v_1, \ldots, v_i\}$. We let D_i denote the dominating set constructed by $\textsc{Alg}$ after processing the first i vertices of the input. When no confusion can occur, we sometimes implicitly identify a dominating set D and the subgraph it induces. For example, we may say that D *has k components* or *is connected*, meaning that the subgraph of G induced by D has k components or is connected, respectively.

Online algorithms must compute a solution for all prefixes of the input seen by the algorithm, since the input could terminate at any point. Given the irrevocable decisions, this can of course affect the possible final sizes of a dominating set. When we want to emphasize that a bound is derived under this restriction, we use the word *incremental* to indicate this, i.e., if we discuss the size of an incremental dominating set D of G, this means that $D_1 \subseteq D_2 \subseteq \cdots \subseteq D_n = D$ and that D_i is a dominating set of G_i for each i. Note in particular that any incremental algorithm, including $\mathrm{OPT}^{\mathrm{INC}}$, for DS, CDS, or IDS must select the first vertex.

Throughout the text, we use standard graph-theoretic notation. In particular, the *path on n vertices* is denoted P_n. A *star* with n vertices is the complete bipartite graph $K_{1,n-1}$. In a rooted tree, an *internal* vertex is a vertex that has at least one child vertex. For a vertex v, $N(v)$ denotes the set of neighbors of v. We use $\mathrm{c}(G)$ to denote the number of components of a graph G. The size of a minimum dominating set of a graph G is denoted by $\gamma(G)$. We use indices to indicate variants, using $\gamma_C(G)$, $\gamma_T(G)$, and $\gamma_I(G)$ for Connected, Total, and Independent Dominating Set, respectively. This is an alternative notation for the size computed by $\mathrm{OPT}^{\mathrm{OFF}}$. We also use these indices on $\mathrm{OPT}^{\mathrm{INC}}$ to indicate which variant is under consideration. Sometimes, when the problem considered is clear from the context or we consider more problems at the same time, we may omit the index. We use Δ to denote the maximum degree of the graph under consideration. Similarly, we always let n denote the number of vertices in the graph.

In many of the proofs of lower bounds on the competitive ratio, when the path, P_n, is considered, either as the entire input or as a subgraph of the input, we assume that it is given in the *standard order*, the order where the first vertex given is one of the two endpoints, and each subsequent vertex is a neighbor of the vertex given in the previous step. When the path is a subgraph of the input graph, we often extend this standard order of the path to an *adversarial order* of the input graph – a fixed ordering of the vertices that yields an input attaining the bound. Typically, the adversarial order consists of a path in the standard order, followed by one or more high-degree vertices off the path.

In some online settings, we are interested in connected graphs, where the vertices are given in an order such that the subgraph induced at any point in time is connected. In this case, we use the term *always-connected*, indicating that we are considering a connected graph G, and all the partial graphs G_i are connected. We implicitly assume that trees are always-connected and we drop the adjective. Since all the classes we consider are hereditary (that is, any induced subgraph also belongs to the class), no further restriction of partial inputs G_i is necessary. In particular, these conventions imply that for trees, the vertex arriving at any step (except the first) is connected to exactly one of the vertices given previously, and since we consider unrooted trees, we can think of that vertex as the *parent* of the new vertex.

3 The Cost of Being Online

In this section, we analyze the the performance of online algorithms for the four variants of Dominating Set. We compare the algorithms to $\mathrm{OPT}^{\mathrm{INC}}$, thus comparing algorithms restricted to making the same irrevocable decisions, and thereby investigating the role played by the (absence of) knowledge of the future. We also compare the online algorithms to $\mathrm{OPT}^{\mathrm{OFF}}$.

We start with Independent Dominating Set.

Proposition 1 *For any graph G, there is a unique incremental independent dominating set.*

Proof We fix G and proceed inductively. The first vertex has to be selected due to the online requirement. When the next vertex, v_{i+1}, is given, if it is dominated by a vertex in D_i, it cannot be selected, since then D_{i+1} would not be independent. If v_{i+1} is not dominated by a vertex in D_i, then v_{i+1} or one of its neighbors must be selected. However, none of v_{i+1}'s neighbors can be selected, since if they were not selected already, then they are dominated, and selecting one of them would violate the independence criteria. Thus, v_{i+1} must be selected. In either case, D_{i+1} is uniquely defined. $\square$

Since a correct incremental algorithm is uniquely defined by this proposition by a forced move in every step, $\textsc{Opt}^{\textsc{inc}}$ must behave exactly the same. This fills the column for Independent Dominating Set in Table 1.

For Dominating Set, Connected Dominating Set, and Total Dominating Set, we start by using the size of a given dominating set to bound the sizes of some connected or incremental equivalents. The following theorem does not address TDS directly, but in many cases, it can be applied to this problem as well, since any connected dominating set including more than one vertex in each connected component is a total dominating set.

Theorem 1 *Let G be always-connected, let S be a dominating set of G, and let R be an incremental dominating set of G. Then the following hold:*

(i) *There is a connected dominating set S' of G such that $|S'| \leq |S| + 2(\mathrm{c}(S) - 1)$.*
(ii) *There is an incremental connected dominating set R' of G such that $|R'| \leq |R| + \mathrm{c}(R) - 1$.*
(iii) *If G is a tree, there is an incremental dominating set R'' of G such that $|R''| \leq |S| + \mathrm{c}(S)$.*

Moreover, all three bounds are tight for infinitely many graphs.

Proof To obtain the upper bound of (i), we argue that by selecting additionally at most $2(\mathrm{c}(S) - 1)$ vertices, we can connect all the components in S. We do this inductively. If there are two components that can be connected by a path of at most two unselected vertices, we select all the vertices on this path and continue inductively. Otherwise, assume to the contrary that all pairs of components require the selection of at least three vertices to become connected. We choose a shortest such path of length k consisting of vertices $u_1, \ldots, u_k$, where u_i is dominated by a component C_i for all i. If $C_1 \neq C_2$, we can connect them by selecting u_1 and u_2, which would be a contradiction. If $C_1 = C_2$, then we have found a shorter path between C_1 and C_k; also a contradiction. We conclude that $|S'| \leq |S| + 2(\mathrm{c}(S) - 1)$, which proves (i).

To see that the bound is tight, consider a path P_n in the standard order, where $n \equiv 0 \pmod 3$. Clearly, the size of a minimum dominating set S of P_n is $n/3$ and $\mathrm{c}(S) = n/3$. On the other hand, the size of any minimum connected dominating set of P_n is $n - 2$ and $n - 2 = |S| + 2(\mathrm{c}(S) - 1)$.

To prove (ii), we label the components of R in the order in which their first vertices arrive. Thus, let $C_1, \ldots, C_k$ be the components of R, and, for $1 \leq i \leq k$, let v_{j_i} be the first vertex of C_i that arrives. Note that we assume that v_{j_i} arrives before $v_{j_{i+1}}$ for each $i = 1, \ldots, k-1$. We prove that for each component C_i of R, there is a path of length 2 joining v_{j_i} with C_h in G_{j_i} for some $h < i$, i.e., a path with only one vertex not belonging to either component. Let $P = v_{l_1}, \ldots, v_{l_m}, v_{j_i}$ be a shortest path in G_{j_i} connecting v_{j_i} and some component C_h, $h < i$, and assume for the sake of contradiction that $m \geq 3$. In G_{j_i}, the vertex v_{l_3} is not adjacent to a vertex in any component $C_{h'}$, where $h' < i$, since in that case a shorter path would exist. However, since vertices cannot be unselected as the online algorithm proceeds, it follows that in G_{l_3}, v_{l_3} is not dominated by any vertex, which is a contradiction. Thus, $m \leq 2$ and selecting just one additional vertex at the arrival of v_{i_j} connects C_i to an earlier component, and the result follows inductively.

To see that the bound is tight, observe that the optimal incremental connected dominating set of P_n has $n-1$ vertices, while for even n, there is an incremental dominating set of size $n/2$ with $n/2$ components.

To obtain (iii), consider an algorithm ALG processing vertices greedily, while always selecting all vertices from S. That is, v_1 and all vertices of S are always selected, and when a vertex $v \notin S$ arrives, it is selected if and only if it is not dominated by already selected vertices, in which case it is called a *bad* vertex. Clearly, ALG produces an incremental dominating set, R'', of G.

To prove the upper bound on $|R''|$, we gradually mark components of S. For a bad vertex v_i, let v be a vertex from S dominating v_i, and let C be the component of S containing v. Mark C. To prove the claim it suffices to show that each component of S can be marked at most once, since each bad vertex leads to some component of S being marked.

Assume for the sake of contradiction that some component, C, of S is marked twice. This happens because a vertex v of C is adjacent to a bad vertex b, and a vertex v' (not necessarily different from v) of C is adjacent to some later bad vertex b'. Since G is always-connected and b' was bad, b and b' are connected by a path not including v'. Furthermore, v and v' are connected by a path in C. Thus, the edges $\{b, v\}$ and $\{b', v'\}$ imply the existence of a cycle in G, contradicting the fact that it is a tree.

To see that the bound is tight, let $v_1, \ldots, v_m$, $m \equiv 2 \pmod 6$, be a path in the standard order. Let G be obtained from P_m by attaching m pendant vertices (new vertices of degree 1) to each of the vertices $v_2, v_5, v_8, \ldots, v_m$, where the pendant vertices arrive in arbitrary order, though respecting that G should be always-connected. Each minimum incremental dominating set of G contains each of the vertices $v_2, v_5, v_8, \ldots, v_m$, the vertex v_1, and one of the vertices v_{3i} and v_{3i+1} for each i, and thus it has size $2(m+1)/3$. On the other hand, the vertices $v_2, v_5, v_8, \ldots, v_m$ form a dominating set S of G with $\mathrm{c}(S) = (m+1)/3$. □

Theorem 1 is best possible in the sense that none of the assumptions can be omitted. Indeed, Proposition 11 implies that it is not even possible to bound the size of an incremental (connected) dominating set in terms of the size of a (connected) dominating set, much less to bound the size of an incremental connected dominating set in terms of the size of a dominating set. Therefore, (i) and (ii) in Theorem 1 cannot be combined even on bipartite planar graphs. The situation is different for trees: Proposition 3 (i) essentially leverages the fact that any connected dominating set D on a tree can be produced by an incremental algorithm without increasing the size of D.

3.1 Trees

For DS and CDS, we let Parent denote the following algorithm for trees. The algorithm selects the first vertex. When a new vertex v arrives, if v is not already dominated by a previously arrived vertex, then the parent vertex that v is adjacent to is added to the dominating set. Note that Parent accepts all internal nodes of the tree rooted at the first vertex, creating an incremental connected dominating set. For CDS on trees, Parent is 1-competitive, even against Opt^{OFF}:

Lemma 1 *For CDS on any tree T,*

$$\text{Parent}(T) \leq \begin{cases} \gamma_C(T) + 1, & \text{if } v_1 \text{ has degree } 1 \text{ in } T \\ \gamma_C(T), & \text{otherwise.} \end{cases}$$

Proof For CDS, Parent selects no vertices of degree 1, except possibly v_1. Thus, the algorithm selects all vertices of degree at least 2 plus at most one vertex of degree 0 or 1.

For trees with at most two vertices, the minimum size of a connected dominating set is 1. For trees with more than two vertices, the minimum size of a connected dominating set of any tree T equals the number of vertices with degree at least 2. □

For TDS, Parent is the same as for DS and CDS, except that it selects v_1 only if v_2 arrives, in which case it selects both v_1 and v_2. Thus, Parent for TDS selects at most one more vertex than Parent for DS and CDS. To show that for TDS on trees, Parent is 1-competitive against Opt^{INC}, we prove the following:

Lemma 2 *For any incremental total dominating set D for an always-connected graph G, all D_i are connected.*

Proof For the sake of a contradiction, suppose that for some i, D_i is not connected, and let i be the smallest index with this property. It follows that the vertex v_i constitutes a singleton component of D_i. Thus, v_i cannot be dominated by any other vertex of D_i, contradicting that the solution is incremental. □

Lemma 3 *For TDS on any tree T,* $\text{Parent}(T) = \text{Opt}_T^{\text{INC}}(T)$.

Proof If T consists of only one vertex, $\textsc{Parent}(T) = \textsc{Opt}_T^{\text{INC}}(T) = 0$. Otherwise, $\textsc{Parent}$ selects v_1, v_2, and all later internal vertices. $\textsc{Opt}^{\text{INC}}$ also selects v_1 and v_2, and by Lemma 2, it has to select all internal vertices. Thus, the two algorithms select exactly the same set of vertices. □

Lemma 4 *For any online algorithm* $\textsc{Alg}$ *for DS or CDS, there exist arbitrarily large trees* T*, such that* $\textsc{Alg}(T) \geq n - 1$.

Proof We prove that the adversary can construct an arbitrarily large tree, maintaining the invariant that at most one vertex is not included in the solution of $\textsc{Alg}$. The algorithm has to select the first vertex, so the invariant holds initially. When presenting a new vertex v_i, the adversary checks whether all vertices given so far are included in $\textsc{Alg}$'s solution. If this is the case, v_i is connected to an arbitrary vertex, and the invariant still holds. Otherwise, v_i is connected to the unique vertex not included in D_{i-1}. Now v_i is not dominated, so $\textsc{Alg}$ must select an additional vertex. □

Proposition 2 *For any online algorithm* $\textsc{Alg}$ *for DS on trees,* $\mathbb{CR}^{\text{INC}}(\textsc{Alg}) \geq 2$.

Proof We argue that, for any always-connected bipartite graph, G, we have that $\textsc{Opt}^{\text{INC}}(G) \leq \frac{n+1}{2}$. Since trees are bipartite, the result then follows from Lemma 4. The smaller partite set S of any connected bipartite graph G is a dominating set of G. If the first presented vertex v_1 belongs to S, then S is an incremental dominating set of G. Otherwise, $S \cup \{v_1\}$ is an incremental dominating set of G. □

The adversary strategy used in the proof of Lemma 4 cannot give a lower bound larger than 2 against $\textsc{Opt}^{\text{OFF}}$, since the resulting tree may not have any dominating set with fewer than $n/2$ vertices. Consider, for example, a caterpillar graph where each vertex of the central path has exactly one neighbor not belonging to the central path.

The following proposition concludes on the results for DS, CDS, and TDS on trees.

Proposition 3 *For trees, the following hold.*

(i) For DS, $\mathbb{CR}^{\text{INC}}(\textsc{Parent}) = 2$ *and* $\mathbb{CR}^{\text{OFF}}(\textsc{Parent}) = 3$.
(ii) For CDS, $\mathbb{CR}^{\text{INC}}(\textsc{Parent}) = \mathbb{CR}^{\text{OFF}}(\textsc{Parent}) = 1$.
(iii) For TDS, $\mathbb{CR}^{\text{INC}}(\textsc{Parent}) = 1$ *and* $\mathbb{CR}^{\text{OFF}}(\textsc{Parent}) = 2$.

Proof We prove (i) first. The lower bound on $\mathbb{CR}^{\text{INC}}(\textsc{Parent})$ follows directly from Proposition 2. For the corresponding upper bound, note that Lemma 1 in combination with Theorem 1(ii) imply that $\textsc{Parent}(T) \leq \gamma_C(T) + 1 \leq 2 \cdot \textsc{Opt}^{\text{INC}}(T)$, for any tree T. The result on $\mathbb{CR}^{\text{OFF}}(\textsc{Parent})$ follows from Theorem 1(i) and the proof that Theorem 1(i) is tight.

Item (ii) follows directly from Lemma 1.

In item (iii), the result on $\mathbb{CR}^{\text{INC}}(\textsc{Parent})$ follows directly from Lemma 3.

For the upper bound on $\mathbb{CR}^{\text{OFF}}(\textsc{Parent})$, let S be an optimal total dominating set for a tree T. Assume that $|S| \geq 3$ and consider the following calculations which we argue for below.

$$\begin{aligned}\textsc{Parent}(T) &= \textsc{Opt}_T^{\text{INC}}(T), \text{ by Lemma 3}\\ &\leq \textsc{Opt}_C^{\text{INC}}(T)\\ &\leq \textsc{Opt}_C^{\text{OFF}}(T)+1\\ &\leq |S|+2(c(S)-1)+1, \text{by Theorem 1(i)}\\ &\leq 2|S|-1\end{aligned}$$

The first inequality in the calculations above follows from the fact that any connected dominating set of size at least 2 is a total dominating set, and since we assumed that an optimal total dominating set for T has at least three vertices, any connected dominating set for T must have at least two vertices.

The second inequality follows from Lemma 1, since Parent is an incremental algorithm.

The last inequality follows from the fact that any connected component in a total dominating set has at least two vertices.

For the lower bound on $\mathbb{CR}^{\text{OFF}}(\textsc{Parent})$, consider a path $v_1, v_2, \ldots, v_{4n}$ for a positive integer n. When given in the standard order, Parent will select the first $4n-1$ vertices, whereas an optimal total dominating set is the set $\{v_{4i+2}, v_{4i+3} \mid 0 \leq i \leq n-1\}$ of size $2n$. □

3.2 Bipartite, bounded-degree, and general graphs

We extend the Parent algorithm for graphs that are not trees as follows. When a vertex v_i, $i > 1$, arrives, which is not already dominated by one of the previously presented vertices, Parent selects *any* of the neighbors of v_i in G_i. Again, it is easily seen that Parent creates an incremental connected dominating set. We start with a few positive results for Parent.

Proposition 4 *The following hold.*

(i) For DS and CDS on always-connected bipartite graphs, for $n \geq 4$,

$$\mathbb{CR}^{\text{OFF}}(\textsc{Parent}) \leq n/2.$$

(ii) For DS and CDS on always-connected graphs, for $n \geq 4$,

$$\mathbb{CR}^{\text{OFF}}(\textsc{Parent}) \leq n-2.$$

(iii) For TDS, $\mathbb{CR}^{\text{OFF}}(\textsc{Parent}) \leq n/2$.

Proof For item (i), if $\gamma(G) \geq 2$, there is nothing to prove. Therefore, we assume that there is a single vertex v adjacent to every other vertex. Since G is bipartite, there is no edge between any of the vertices adjacent to v, so G is a star. Since G_i is connected for each i, the vertex v arrives either as the first or the second vertex. Furthermore, if another vertex arrives after v, then v is selected by Parent. Once v is selected, all future vertices are already dominated by v, so no more vertices are selected, implying that $\text{Parent}(G) \leq 2$, which concludes the proof.

For item (ii), we only need to consider the case of $\gamma(G) = 1$, since otherwise there is nothing to prove, and thus there is a vertex v adjacent to every other vertex of G. Since after the arrival of any vertex, Parent increases the size of the dominating set by at most one, it suffices to prove that, immediately after some vertex has been processed, there are two vertices not selected by Parent. First note that once v is selected, Parent does not select any other vertex and thus we can assume that v is not the first vertex. Suppose that v arrives after v_i, $i \geq 2$. The vertex v_i has not yet been selected when v arrives, and v is dominated by v_1, so there are two vertices not selected. The last remaining case is when v arrives as the second vertex. In this case we distinguish whether v_3 is adjacent to v_1, or not. If v_3 is adjacent to v_1, then v is not selected, there are two vertices not selected (v and v_3), and we are done. If v_3 is not adjacent to v_1, then Parent selects v when v_3 arrives. No further vertex will be added to the dominating set, concluding the proof.

For any graph with at least one edge, any total dominating set contains at least two vertices. Thus, if Parent selects more vertices than $\text{Opt}_T^{\text{OFF}}$, $\text{Opt}_T^{\text{OFF}}$ selects at least two vertices. This proves (iii). □

The following result shows that Proposition 4(ii) is tight.

Proposition 5 *For any online algorithm,* Alg*, for DS or CDS on always-connected planar graphs,* $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \geq n - 2$.

Proof By Lemma 4, the adversary can construct a tree on $n-1$ vertices, such that any online algorithm selects at least $n-2$ vertices. If the adversary then gives one vertex connected to all $n-1$ vertices in the tree, this last vertex constitutes a connected dominating set. It is not difficult to see that any such graph is indeed planar. □

For DS, let Greedy be the algorithm that selects an arriving vertex, if and only it is not dominated by a previously selected vertex.

Proposition 6 *For graphs of maximum degree Δ, the following hold.*

(i) For any algorithm Alg *for DS or CDS,* $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \leq \Delta + 1$.
(ii) For DS, $\mathbb{CR}^{\text{OFF}}(\text{Greedy}) \leq \Delta$.
(iii) For any algorithm Alg *for TDS,* $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \leq \Delta$.
(iv) For any algorithm Alg *for CDS on connected graphs,* $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \leq \Delta - 1$.

Proof For DS and CDS, each vertex can only dominate itself and its at most Δ neighbors. Thus, $\gamma_C(G) \geq \gamma(G) \geq n/(\Delta + 1)$, proving item (i).

For item (ii), consider a dominating set $S = \{s_1, s_2, \ldots, s_k\}$ of size $k = \gamma(G)$. Partition the vertices of G into k sets $V_1, V_2, \ldots, V_k$ such that $s_i \in V_i$

and all vertices in $V_i \setminus \{s_i\}$ are dominated by s_i. Clearly, $|V_i| \leq \Delta + 1$ and if V_i has $\Delta + 1$ vertices, it is called a *critical* set. If there are exactly d critical sets, then $n \leq d(\Delta + 1) + (k - d)\Delta$. Thus, $\gamma(G) = k \geq (n - d)/\Delta$.

For each critical set V_i, each vertex in the set is connected to at least one other vertex. Thus, if GREEDY selects the Δ first vertices of V_i, it will not select the last vertex of V_i. This shows that, from each critical set, GREEDY will select at most Δ vertices. Hence, $\text{GREEDY}(G) \leq n - d$, concluding the proof of item (ii).

For TDS, a vertex can only dominate its at most Δ neighbors. Thus, $\gamma_T(G) \geq n/\Delta$, proving item (iii).

For item (iv), let D be a minimum connected dominating set of a connected graph G with $|D| = k$. The sum of the degrees of vertices in D is bounded by $k\Delta$ which is then also an upper bound on how many vertices D can dominate outside D. Since D is connected, any spanning tree of D contains $k - 1$ edges and each endpoint is adjacent to the other endpoint in the spanning tree. Thus, no vertices outside D are dominated via these edges. Thus, at most $k\Delta - (2k - 2)$ vertices not in D can be dominated by D, giving $n \leq k\Delta - k + 2 = k(\Delta - 1) + 2$ vertices in G. It follows that $\gamma_C(G) \geq (n - 2)/(\Delta - 1)$ and thus, for any algorithm ALG for CDS, $\mathbb{CR}^{\text{OFF}}(\text{ALG}) \leq \Delta - 1$. □

The upper bound of Proposition 6(ii) is almost tight, even for always-connected bounded-degree graphs:

Proposition 7 *For any online algorithm* ALG *for DS on always-connected bounded-degree graphs,* $\mathbb{CR}^{\text{OFF}}(\text{ALG}) \geq \Delta - 2$.

Proof We adapt the construction in the proof of Lemma 4 to work for bounded-degree graphs. The adversary first gives n_1 vertices inducing a tree. For convenience, we let n_1 be a multiple of Δ.

For the first n_1 vertices, the adversary uses the following strategy. If there is a vertex $v \notin D_{i-1}$ with degree less than $\Delta - 1$, v_i is connected to v. Otherwise, v_i is connected to any vertex $v \in D_{i-1}$ with degree less than $\Delta - 1$. Thus, the following invariant is maintained. At most one vertex $v \notin D_i$ has degree less than $\Delta - 1$. After the first n_1 vertices, $n_2 = n_1/\Delta$ vertices are given such that each of the first n_1 vertices is adjacent to exactly one of the last n_2 vertices.

Let V_1 be the set consisting of the first n_1 vertices, and let V_2 contain the last n_2 vertices. By construction, each vertex in $V_1 \setminus D$, except at most one, has at least $\Delta - 1$ neighbors in V_1, and for any pair of neighbors, $u, v \in V_1$, at least one of u and v is included in D. Thus, there are more than $(|V_1 \setminus D| - 1)(\Delta - 1)$ edges between $V_1 \setminus D$ and $V_1 \cap D$. Together with the fact that the number of edges in the subgraph induced by V_1 is $n_1 - 1$, this means that $(|V_1 \setminus D| - 1)(\Delta - 1) \leq n_1 - 1$, implying $|V_1 \setminus D| \leq (n_1 - 1)/(\Delta - 1) + 1$. Thus, $|D| \geq |D \cap V_1| = n_1 - |V_1 \setminus D| \geq n_1 - (n_1 - 1)/(\Delta - 1) - 1 > (\Delta - 2)n_1/(\Delta - 1) - 1$. Since V_2 constitutes a dominating set of size n_1/Δ, this proves that the asymptotic competitive ratio satisfies $\mathbb{CR}^{\text{OFF}}(\text{ALG}) \geq \Delta(\Delta - 2)/(\Delta - 1) > \Delta - 2$. □

Our next aim is to show that there exists an algorithm which is $n/4$-competitive against OPT^{INC} on every graph. Later, in Propositions 8 and 9, we prove that this is optimal. For the algorithm, we use *layers* in a graph G. The function L assigns layer numbers to vertices as follows: If v_i has no neighbors when it arrives, let $L(v_i) = 1$; otherwise, let

$$L(v_i) = 1 + \min\{L(v_j) \mid v_j \text{ is a neighbor of } v_i \text{ in } G_i\}.$$

The algorithm, denoted LowParent, is a specialization of Parent. For each vertex v_i, $i > 1$, if v_i is not dominated by one of the already selected vertices, it selects a neighbor of v_i with the smallest layer number. For CDS, if the vertex v_i connects two or more connected components, the algorithm also adds a minimum-sized set of vertices to D to make it connected. This will include the current vertex and at most one neighbor in each component being connected. Furthermore, for DS and CDS, the algorithm also adds the first vertex to arrive in each of layers 3 and 5.

The pseudocode for LowParent for DS and CDS is given in Algorithm 1.

Algorithm 1: Algorithm LowParent for DS and CDS.

1 $D \leftarrow \emptyset$
2 **while** a vertex v_i is presented **do**
3   **if** v_i has no neighbors in G_i **then**
4         $L(v_i) \leftarrow 1$
5         $D \leftarrow D \cup \{v_i\}$
6   **else**
7         $L(v_i) \leftarrow 1 + \min\{L(v_j) \mid v_j \text{ is a neighbor of } v_i \text{ in } G_i\}$
8   **if** there is no $v_j \in D$ such that v_j dominates v_i **then**
9   Choose a neighbor v_j of v_i with $L(v_j) = L(v_i) - 1$
10             $D \leftarrow D \cup \{v_j\}$
11   **if** the problem is CDS **then**
12   **if** v_i connects vertices belonging to different connected components in G_{i-1} **then**
13   Add a minimum-sized set of vertices to D connecting the corresponding components of D
14   **if** $L(v_i) \in \{3, 5\}$ **then**
15   **if** $|\{v_j \in G_i \mid L(v_j) = L(v_i)\}| = 1$ **then**
16             $D \leftarrow D \cup \{v_i\}$

The pseudocode for LowParent for TDS is given in Algorithm 2. Algorithm 2 is obtained from Algorithm 1 by omitting lines 5 and 11–16 and adding the following (lines 10–11): For each vertex in layer 1, its first neighbor v to arrive is added to D.

We prove that LowParent is asymptotically optimal in most cases. We consider DS and CDS first.

Algorithm 2: Algorithm LowParent for TDS.

1 $D \leftarrow \emptyset$
2 **while** a vertex v_i is presented **do**
3   **if** v_i has no neighbors in G_i **then**
4         $L(v_i) \leftarrow 1$
5   **else**
6         $L(v_i) \leftarrow 1 + \min\{L(v_j) \mid v_j \text{ is a neighbor of } v_i \text{ in } G_i\}$
7   **if** there is no $v_j \in D$ such that v_j dominates v_i **then**
8   Choose a neighbor v_j of v_i with $L(v_j) = L(v_i) - 1$
9             $D \leftarrow D \cup \{v_j\}$
10   **if** v_i has an undominated neighbor **then**
11             $D \leftarrow D \cup \{v_i\}$

Lemma 5 *Consider a graph G and an incremental algorithm* Alg *for DS or CDS. For each connected component, H, of the subgraph G_i of G, the following hold.*

(i) Alg *selects all vertices of the first layer of H.*
(ii) For any two consecutive layers, j and $j+1$ of H, if no vertices in layer j are included in the final solution, the first vertex of layer $j+1$ is selected by Alg.
(iii) If H has at least $2k+1$ layers, $k \in \mathbb{Z}$, Alg *accepts at least $k+1$ vertices in H.*

Proof Item (i) follows immediately from the fact that each vertex in layer 1 is isolated when it arrives.

For item (ii), note that when the first vertex v of layer $j+1$ arrives, it is only connected to vertices in layer j, and hence it is not dominated. Since Alg does not select any vertices from layer j, v must be selected.

Item (iii) follows directly from items (i) and (ii). □

Theorem 2 *For DS and CDS, $\mathbb{CR}^{\text{INC}}(\text{LowParent}) \leq (n+3)/4$.*

Proof First, if $\text{Opt}^{\text{INC}}(G) \geq 4$, then $\text{LowParent}(G) \leq n \leq \frac{n}{4}\text{Opt}^{\text{INC}}(G)$. Furthermore, if $\text{Opt}^{\text{INC}}(G) = 1$, then $\text{LowParent}(G) = \text{Opt}^{\text{INC}}(G)$. Thus, we need only consider graphs, G, with $2 \leq \text{Opt}^{\text{INC}}(G) \leq 3$.

We distinguish several cases according to the number, ℓ, of layers of G. If $\ell \leq 2$, then $\text{LowParent}(G) = \text{Opt}^{\text{INC}}(G)$. If $\ell \geq 7$, then by Lemma 5, $\text{Opt}^{\text{INC}}(G) \geq 4$. Hence, we only need to consider the range $3 \leq \ell \leq 6$.

We consider DS first. For $i \geq 1$, let n_i denote the size of the ith layer and s_i the number of vertices in the ith layer selected by LowParent in Line 5 or 10 (thus, *not* including the selections in Line 16). Note that $s_\ell = n_{\ell+1} = 0$.

Since each vertex in layer $i+1$ causes at most one vertex in layer i to be selected,

$$s_i \leq n_{i+1}, \text{ for } i \geq 2, \quad \text{and} \quad s_i \leq n_i, \text{ for } i \geq 1.$$

From these two inequalities independently, we get

$$\sum_{i=2}^{\ell-1}\frac{i-1}{\ell-1}s_i \leq \sum_{i=2}^{\ell-1}\frac{i-1}{\ell-1}n_{i+1} = \sum_{i=3}^{\ell}\frac{i-2}{\ell-1}n_i \quad \text{and} \quad \sum_{i=2}^{\ell-1}\frac{\ell-i}{\ell-1}s_i \leq \sum_{i=2}^{\ell-1}\frac{\ell-i}{\ell-1}n_i\,.$$

Adding these two inequalities, we obtain

$$\sum_{i=2}^{\ell-1}s_i \leq \sum_{i=2}^{\ell}\frac{\ell-2}{\ell-1}n_i\,.$$

Let n' be the total number of vertices selected in lines 5 and 16. If $3 \leq \ell \leq 4$, then $n' = n_1 + 1$. Finally, if $\ell \geq 5$, then $n' = n_1 + 2$.

Since $\textsc{LowParent}(G) = n' + \sum_{i=2}^{\ell-1} s_i$, we get

$$\textsc{LowParent}(G) - n' = \sum_{i=2}^{\ell-1}s_i \leq \frac{\ell-2}{\ell-1}\sum_{i=2}^{\ell}n_i = \frac{\ell-2}{\ell-1}(n-n_1)\,. \tag{1}$$

We consider always-connected graphs first, for which $n_1 = 1$.

For $\ell = 3$, Inequality (1) yields $\textsc{LowParent}(G) \leq (n-1)/2+2 = (n+3)/2$. Since $\textsc{Opt}^{\text{INC}}(G) \geq 2$, $\textsc{LowParent}(G)/\textsc{Opt}^{\text{INC}}(G) \leq (n+3)/4$.

For $\ell = 4$, Inequality (1) gives $\textsc{LowParent}(G) \leq 2(n-1)/3+2 = (2n+4)/3$. If $\textsc{Opt}^{\text{INC}}(G) = 3$, then $\textsc{LowParent}(G)/\textsc{Opt}^{\text{INC}}(G) \leq (2n+4)/9 < (n+3)/4$. If $\textsc{Opt}^{\text{INC}}(G) = 2$, it follows from Lemma 5 that the vertices selected by $\textsc{Opt}^{\text{INC}}$ are the first vertices in layers 1 and 3. Since these vertices are selected on arrival by $\textsc{LowParent}$ as well, $\textsc{LowParent}$ selects the same vertices as $\textsc{Opt}^{\text{INC}}$, plus a parent of the first vertex in layer 3. Thus, it selects $3/2\,\textsc{Opt}^{\text{INC}}$ vertices. This ratio is smaller than $(n+3)/4$, since $n \geq 4$.

For $\ell = 5$, Inequality (1) yields $\textsc{LowParent}(G) \leq 3(n-1)/4+3 = (3n+9)/4$. By Lemma 5, $\textsc{Opt}^{\text{INC}}(G) \geq 3$ and hence, $\textsc{LowParent}(G)/\textsc{Opt}^{\text{INC}}(G) \leq (3n+9)/12 = (n+3)/4$.

For $\ell = 6$, it follows from Lemma 5 that $\textsc{Opt}^{\text{INC}}(G) = 3$ and the vertices selected by $\textsc{Opt}^{\text{INC}}$ are the first vertices in layers 1, 3, and 5. Since these vertices are selected on arrival by $\textsc{LowParent}$ as well, $\textsc{LowParent}$ selects the same vertices as $\textsc{Opt}^{\text{INC}}$, plus a parent of the first vertex in layer 3 and a parent of the first vertex in layer 5. Thus, it selects $5/3\,\textsc{Opt}^{\text{INC}}$ vertices. This ratio is smaller than $(n+3)/4$, since $n \geq 6$.

We now consider graphs which are not always-connected. Note that we still assume that $\textsc{Opt}^{\text{INC}}(G) \leq 3$, and Inequality (1) still holds. If G is given in a disconnected order, layer 1 contains at least two vertices and by Lemma 5, $\textsc{Opt}^{\text{INC}}$, just as $\textsc{LowParent}$, accepts all vertices of layer 1. Therefore, if $\textsc{Opt}^{\text{INC}}(G) = 2$, then $\ell \leq 2$, and $\textsc{LowParent}(G) = \textsc{Opt}^{\text{INC}}(G)$. Moreover, if layer 1 contains three vertices, then $\ell \leq 2$, and $\textsc{Opt}^{\text{INC}}(G) = 3 = \textsc{LowParent}(G)$. Hence, we only need to consider the case where $\textsc{Opt}^{\text{INC}}(G) = 3$ and the first layer contains exactly two vertices. Note that, in this case, $3 \leq \ell \leq 4$.

If $\ell = 3$, then by Inequality (1), $\textsc{LowParent}(G) \leq (n-2)/2+3 = (n+4)/2$. It follows that $\textsc{LowParent}(G)/\textsc{Opt}^{\text{INC}}(G) \leq (n+4)/6 < (n+3)/4$.

If $\ell = 4$, then by Inequality (1), $\textsc{LowParent}(G) \leq 2(n-2)/3+3 = (2n+5)/3$. It follows that $\textsc{LowParent}(G)/\,\textsc{Opt}^{\text{INC}}(G) \leq (2n+5)/9 < (n+3)/4$.

We now consider CDS. If the graph is always-connected, LowParent for CDS selects the same vertices as for DS, so the calculations for DS also hold for CDS. Thus, we only need to consider graphs that are not always-connected.

If layer 1 contains three vertices, then the graph cannot have an incremental CDS with fewer than four vertices, contradicting $\textsc{Opt}^{\text{INC}}(G) \leq 3$. Thus, we can assume that the graph never has more than two connected components and that the two components arrive in an always-connected manner. If the two components remain unconnected, the above analysis for DS holds for each component. Otherwise, $\textsc{Opt}^{\text{INC}}$ connects the two components by selecting exactly one vertex, v, and since $\textsc{Opt}^{\text{INC}}$ is incremental, the two components must be unconnected until the arrival of v. Thus, the vertices selected by LowParent are exactly the three vertices selected by $\textsc{Opt}^{\text{INC}}$ and the first vertex of layer 3, if it arrives. Hence, $\textsc{LowParent}(G)/\,\textsc{Opt}^{\text{INC}}(G) \leq 4/3 < \frac{n+3}{4}$, since $n \geq 3$. □

We now consider TDS. For general graphs, we obtain an upper bound of approximately $n/4$, as we did for DS and CDS. For always-connected graphs, the upper bound is improved to approximately $2n/9$.

Theorem 3 *For TDS,* $\mathbb{CR}^{\text{INC}}(\textsc{LowParent}) \leq (n+2)/4$*, and for TDS on always-connected graphs,* $(2n+1)/9 \leq \mathbb{CR}^{\text{INC}}(\textsc{LowParent}) \leq (2n+2)/9$.

Proof We use the same notation as in the proof of Theorem 2. Thus, n_i denotes the size of the ith layer, s_i denotes the number of vertices in the ith layer selected by LowParent, and ℓ is the total number of layers.

Since the first vertex in each layer $i > 1$ is only connected to vertices in layer $i-1$, and choosing that first vertex does not dominate it, any incremental algorithm must choose at least one vertex in each layer, except the last. Hence, $\textsc{Opt}^{\text{INC}}(G) \geq \ell - 1$. Since $\textsc{LowParent}(G) \leq n$, this means that for general graphs, we can assume $\ell \leq 4$. For always-connected graphs, we can assume $\ell \leq 5$, since $\ell \geq 6$ implies a ratio of at most $n/5 < 2n/9$.

If $\ell = 1$, $\textsc{LowParent}(G) = \textsc{Opt}^{\text{INC}}(G) = 0$. Hence, it suffices to consider $\ell \geq 2$.

We use inequalities similar to those for DS:

$$\begin{aligned}
s_1 &\leq n_1 \;\text{ and }\; s_1 \leq n_2 \\
s_2 &\leq n_2 \;\text{ and }\; s_2 \leq n_1 + n_3 \\
s_i &\leq n_i \;\text{ and }\; s_i \leq n_{i+1}, \text{ for } i \geq 3
\end{aligned}$$

Before using the inequalities, we strengthen the inequality $s_2 \leq n_1 + n_3$. When the first vertex of layer 2 arrives, $\textsc{Opt}^{\text{INC}}$ as well as LowParent will select this vertex and a vertex from layer 1. If LowParent selects one more vertex from layer 2, $\textsc{Opt}^{\text{INC}}$ will also have to select an additional vertex, and hence, $\textsc{Opt}^{\text{INC}}(G) \geq 3$. Thus,

$$s_2 = 1, \text{ if } \textsc{Opt}^{\text{INC}}(G) = 2\,.$$

Note that no vertex in layer 1 has a neighbor outside of layer 2. Consider a vertex u in layer 1. When the first neighbor, v, of u arrives, any incremental algorithm has to select v. Thus, the vertices in layer 2 that LowParent selects in order to dominate vertices in layer 1 are also selected by Opt^{INC}. Hence, since Opt^{INC} selects at least one vertex in layer 1,

$$s_2 \leq \text{Opt}^{\text{INC}}(G) - 1 + n_3\,.$$

We consider general graphs first. Recall that for general graphs, we only need to consider $2 \leq \ell \leq 4$.

For $\ell = 2$,

$$\begin{aligned}\text{LowParent}(G) &= s_1 + s_2 \\ &\leq \left(\frac{1}{2}n_1 + \frac{1}{2}n_2\right) + (\text{Opt}^{\text{INC}}(G) - 1 + n_3) \\ &= \frac{1}{2}(n-2) + \text{Opt}^{\text{INC}}(G),\ \text{since } n_3 = 0.\end{aligned}$$

Hence, $\text{LowParent}(G)/\,\text{Opt}^{\text{INC}}(G) \leq \frac{1}{4}(n-2) + 1 = \frac{1}{4}(n+2)$.

For $\ell = 3$, we first consider the case $\text{Opt}^{\text{INC}}(G) = 2$. In this case, $s_2 = 1$. Hence,

$$\text{LowParent}(G) = s_1 + s_2 \leq \left(\frac{1}{2}n_1 + \frac{1}{2}n_2\right) + 1 < \frac{1}{2}n + 1\,.$$

and $\text{LowParent}(G)/\,\text{Opt}^{\text{INC}}(G) < \frac{1}{4}(n+2)$. For $\text{Opt}^{\text{INC}}(G) = 3$, we note that

$$\begin{aligned}\text{LowParent}(G) &= s_1 + s_2 \\ &\leq \left(\frac{3}{4}n_1 + \frac{1}{4}n_2\right) + \left(\frac{1}{2}n_2 + \frac{1}{2}(\text{Opt}^{\text{INC}}(G) - 1 + n_3)\right) \\ &< \frac{3}{4}n + \frac{1}{2}\text{Opt}^{\text{INC}}(G)\,.\end{aligned}$$

Thus, $\text{LowParent}(G)/\,\text{Opt}^{\text{INC}}(G) < \frac{1}{4}n + \frac{1}{2} = \frac{1}{4}(n+2)$.

For $\ell = 4$, $\text{Opt}^{\text{INC}} \geq 3$ and

$$\begin{aligned}&\text{LowParent}(G) \\ &= s_1 + s_2 + s_3 \\ &\leq \left(\frac{3}{4}n_1 + \frac{1}{4}n_2\right) + \left(\frac{1}{2}n_2 + \frac{1}{2}(\text{Opt}^{\text{INC}}(G) - 1 + n_3)\right) + \left(\frac{1}{4}n_3 + \frac{3}{4}n_4\right) \\ &< \frac{3}{4}n + \frac{1}{2}\,\text{Opt}^{\text{INC}}(G)\,.\end{aligned}$$

Thus, $\text{LowParent}(G)/\,\text{Opt}^{\text{INC}}(G) < \frac{1}{4}n + \frac{1}{2} = \frac{1}{4}(n+2)$.

We now consider always-connected graphs, for which $s_1 = n_1 = 1$. We have argued that for the upper bound, it is sufficient to consider $2 \leq \ell \leq 5$.

If $\ell = 2$, $\text{LowParent}(G) = \text{Opt}^{\text{INC}}(G) = 2$.

For $\ell = 3$, we consider $\textsc{Opt}^{\textsc{inc}}(G) = 2$ first. In this case, $\textsc{LowParent}(G) = s_1 + s_2 = 1 + 1 = \textsc{Opt}^{\textsc{inc}}(G)$. For $\textsc{Opt}^{\textsc{inc}}(G) \geq 3$, note that

$$\begin{aligned}\textsc{LowParent}(G) &= s_1 + s_2 \\ &\leq 1 + \left(\frac{1}{2}n_2 + \frac{1}{2}(n_3 + 1)\right) \\ &= 1 + \frac{1}{2}n \\ &= \frac{n+2}{2}\,.\end{aligned}$$

Thus, $\textsc{LowParent}(G)/\,\textsc{Opt}^{\textsc{inc}}(G) \leq (n+2)/6 < (2n+2)/9$, since $n > 2$.

If $\ell = 4$, then $\textsc{Opt}^{\textsc{inc}}(G) \geq 3$. Moreover,

$$\begin{aligned}\textsc{LowParent}(G) &= s_1 + s_2 + s_3 \\ &\leq 1 + \left(\frac{2}{3}n_2 + \frac{1}{3}(n_3 + 1)\right) + \left(\frac{1}{3}n_3 + \frac{2}{3}n_4\right) \\ &= \frac{4}{3} + \frac{2}{3}(n-1) \\ &= \frac{2n+2}{3}\,.\end{aligned}$$

Thus, $\textsc{LowParent}(G)/\,\textsc{Opt}^{\textsc{inc}}(G) \leq \frac{2n+2}{9}$.

If $\ell = 5$, then $\textsc{Opt}^{\textsc{inc}}(G) \geq 4$. Moreover,

$$\begin{aligned}&\textsc{LowParent}(G) \\ &= s_1 + s_2 + s_3 + s_4 \\ &\leq 1 + \left(\frac{3}{4}n_2 + \frac{1}{4}(n_3 + 1)\right) + \left(\frac{1}{2}n_3 + \frac{1}{2}n_4\right) + \left(\frac{1}{4}n_4 + \frac{3}{4}n_5\right) \\ &= \frac{5}{4} + \frac{3}{4}(n-1) \\ &= \frac{3n+2}{4}\,.\end{aligned}$$

Thus, $\textsc{LowParent}(G)/\,\textsc{Opt}^{\textsc{inc}}(G) \leq \frac{3n+2}{16} < \frac{2n}{9}$, since $n \geq 5$.

Finally, we prove the lower bound for always-connected graphs, using the following adversarial input sequence defining a graph G with four layers. The first layer consists of the vertex u. The following three layers each have m vertices, for some large integer m. The vertices of the second, third, and fourth layers are called $v_1, \ldots, v_m$, $w_1, \ldots, w_m$, and $x_1, \ldots, x_m$, respectively. The vertices are given layer by layer, in the order according to their numbering.

No vertex in the second layer is connected to any other vertex in the same layer. In the third layer, w_1 is connected to v_1, and for $2 \leq i \leq m$, w_i is connected to v_i and w_1. In the fourth layer, for $1 \leq i \leq m-1$, x_i is connected to w_1 and w_{i+1}, and x_m is connected w_1.

$\textsc{Opt}^{\textsc{inc}}$ selects the three vertices u, v_1, and w_1.

LowParent selects u and v_1 on arrival. For $2 \le i \le m$, it selects v_i when w_i arrives. Hence, when the first three layers have arrived, all vertices of layers 1 and 2 have been selected. Each vertex $x_1, \ldots, x_{m-1}$ can be dominated by either w_1 or w_{i+1}. If LowParent always chooses the latter, it will select all vertices $w_2, \ldots, w_m$, and when x_m arrives, it must select w_1. In total, LowParent selects $1 + 2m = 1 + 2(n-1)/3 = (2n+1)/3$, yielding a ratio of $\text{LowParent}(G)/\,\text{Opt}^{\text{INC}}(G) = (2n+1)/9$. □

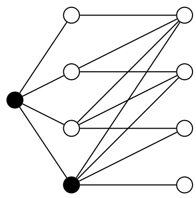

Figure 1 A three-layer construction; the minimum connected dominating set is indicated by the black vertices (Proposition 8).

Proposition 8 *On bipartite graphs, the following hold for any online algorithm* Alg *for DS or CDS.*

(i) For DS and CDS on always-connected graphs, $\mathbb{CR}^{\text{INC}}(\text{Alg}) \ge n/4$.

(ii) For DS on always-connected bounded-degree graphs,

$$\mathbb{CR}^{\text{INC}}(\text{Alg}) \ge \Delta/2 - 1/4.$$

(iii) For CDS on always-connected bounded-degree graphs,

$$\mathbb{CR}^{\text{INC}}(\text{Alg}) \ge \Delta/3.$$

(iv) For CDS on bounded-degree graphs, $\mathbb{CR}^{\text{INC}}(\text{Alg}) \ge \Delta/2$.

Proof For items (i) and (iv), we prove that for any integer $\Delta \ge 2$, there is an always-connected bipartite graph, G, with maximum degree Δ such that $\text{Alg}(G) \ge \Delta = n/2$ and $\text{Opt}^{\text{INC}}(G) = \text{Opt}_C^{\text{INC}}(G) = 2$.

The graph G consists of three layers. The first layer contains only one vertex u, and the second layer contains $\Delta - 1$ vertices $v_1, \ldots, v_{\Delta-1}$ adjacent to u. After the entire second layer is presented to the algorithm, the vertices of the second layer are indistinguishable to the algorithm. The last layer consists of $\Delta-1$ vertices $w_1, \ldots, w_{\Delta-1}$, which will be given in that order, with adjacencies as follows: For $i = 1, \ldots, \Delta - 1$, w_i is connected to $\Delta - i$ vertices of the second layer in such a way that $N(w_{i+1}) \subset N(w_i)$ and $N(w_i)$ contains as few vertices from D_{i-1} as possible. An example of this construction for $\Delta = 4$ is depicted in Figure 1.

Consider the situation when the vertex w_i arrives. If the set $N(w_i)$ does not contain a vertex from D_{i-1}, then Alg must select at least one additional vertex at this time. Thus, ALG selects at least $\Delta - 1 = (n-1)/2$ vertices from the second and third layer, plus the root. Since there is a vertex v in the second layer that is adjacent to all vertices in the third layer, $\{u, v\}$ is an incremental connected and total dominating set of G, concluding the proof of (i). Since the adversary can use any number of copies of G, this also finishes the proof of (iv).

For items (ii) and (iii), note that the adversary can use any number of copies of G, with one vertex in the third layer of copy k connected to the vertex in the first layer of copy $k+1$.

For DS, note that for any given algorithm ALG, G is constructed in such a way that ALG must select at least one vertex from layers 1 and 2 and at least $\Delta - 1$ vertices from layers 2 and 3 in each copy of G. Thus, if ALG selects all $\Delta - 1$ vertices of layer 3 in some copy of G, it selects at least Δ vertices from this copy. Otherwise, the adversary can connect the next copy of G to a vertex w not selected by ALG. In this case, the algorithm will have to select w or the first vertex of the next copy of G. Hence, from two consecutive copies of G, ALG selects at least $2\Delta - 1$ vertices. On the other hand, choosing two vertices from each copy as described above will result in an incremental dominating set. This proves (ii).

For CDS, the adversary will connect adjacent copies of G in the following way. The vertex in layer 3 connected to all vertices in layer 2 will be connected to the first vertex of the following copy of G. Thus, an incremental connected dominating set can be created by selecting one vertex from each of layers 1 and 2 as described above plus the vertex in layer 3 connected to the next copy of G. Again, ALG will select at least $\Delta - 1$ vertices from layers 2 and 3 in each copy of G, and to make the dominating set connected, it will also select the vertex in layer 1. This proves (iii). $\square$

The above adversary strategy does not work for TDS, since OPT^{INC} needs to accept the two first vertices of the graph G. Thus, we use a slightly different graph to prove the following proposition.

Proposition 9 *On bipartite graphs, the following hold for any online algorithm* ALG.

(i) For TDS, on always-connected graphs,

$$\mathbb{CR}^{\text{INC}}(\text{ALG}) \geq n/6 \quad \textit{and} \quad \mathbb{CR}^{\text{INC}}(\text{ALG}) \geq \Delta/3.$$

(ii) For TDS, $\mathbb{CR}^{\text{INC}}(\text{ALG}) \geq n/4$ *and* $\mathbb{CR}^{\text{INC}}(\text{ALG}) \geq (\Delta+1)/2$.

Proof For item (i), we use a graph, G', identical to the graph G used in the proof of Proposition 8, except that the second layer has Δ vertices, and no vertex in layer 3 is connected to the first vertex of layer 2. For bounded-degree graphs, the adversary gives many copies of G', and for each copy except the last, the first vertex of layer 2 is connected to the first vertex of layer 2 in the following copy. In all copies of G', except the first, the first vertex of layer 2 is given before the vertex of layer 1.

For each copy of G', any incremental algorithm for TDS will select the vertex of layer 1 and the first vertex of layer 2, and ALG will also select the remaining $\Delta - 1$ vertices of layer 2. Among the last $\Delta - 1$ vertices selected by ALG, OPT^{INC} will only select the last one to be selected by ALG. This proves item (i).

For item (ii), we use a graph consisting of only two layers. The vertices of layer 1 are given first. Then, the following is repeated. As long as there is a vertex in layer 1 not selected by ALG, a vertex is given which is adjacent to exactly the vertices in layer 1 not yet selected by ALG. For each of these vertices, ALG has to select a vertex in layer 1. It follows that layer 1 contains $\Delta \geq n/2$ vertices, and ALG selects at least $\Delta+1$ vertices, all of those in layer 1 and the first in layer 2. On the other hand, OPT^{INC} chooses only the first vertex of layer 2 and the last vertex of layer 1 to be included in ALG's dominating set. This proves (ii). □

4 The Cost of Being Incremental

This section is devoted to comparing the performance of incremental algorithms and OPT^{OFF}. Since OPT^{OFF} performs at least as well as OPT^{INC} and OPT^{INC} performs at least as well as any online algorithm, each lower bound in Table 2 is at least the maximum of the corresponding lower bound in Table 1 and the corresponding lower bound for $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}})$. Similarly, each upper bound in Table 1 is at most the corresponding upper bound in Table 2. In both cases, we mention only bounds that cannot be obtained in this way from cases considered already. We first give two positive results.

Proposition 10 *For DS, the following hold.*

(i) On trees, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \leq 2$.
(ii) On always-connected graphs, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \leq \lceil n/2 \rceil$.

Proof Item (i) follows directly from Theorem 1(iii).

We now consider item (ii). For a fixed ordering of the vertices of G, consider the layers $L(v)$ assigned to vertices of G. It is easy to see that the set of vertices in the odd layers is an incremental solution for DS and similarly for the set of vertices in even layers plus the vertex v_1. Therefore, OPT^{INC} can select the smaller of these two sets, which necessarily has at most $\lfloor (n-1)/2 \rfloor + 1 = \lceil n/2 \rceil$ vertices. □

The remaining results are negative results.

Proposition 11 *On bipartite planar graphs, the following hold.*

(i) For DS, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta$ *and* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq n - 1$.
(ii) For CDS, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta + 1$ *and* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq n$.

Proof We prove that for each $\Delta \geq 3$, $i > 0$, and $n = i(\Delta+1)$, there is a bipartite planar graph G with n vertices and maximum degree Δ such that

$$\begin{aligned}
&\text{OPT}^{\text{INC}}(G) = \frac{\Delta}{\Delta+1}n\,,\ \text{OPT}_C^{\text{INC}}(G) = n\,,\ \text{and} \\
&\gamma(G) = \gamma_C(G) = \frac{n}{\Delta+1}\,,
\end{aligned}$$

implying the first lower bound of both (i) and (ii). Letting $i = 1$, and hence $n = \Delta + 1$, gives the second lower bound of both (i) and (ii).

Let G consist of i disjoint copies of the star on $\Delta+1$ vertices, with the center of each star arriving as the last vertex among the vertices of that particular star. Clearly, $\gamma(G) = \gamma_C(G) = n/(\Delta+1)$. On the other hand, any incremental dominating set has to contain every vertex, except the last vertex of each star, since all these vertices are pairwise non-adjacent. In addition, any incremental connected dominating set has to contain the centers of the stars to preserve connectedness of the solution in each component. It follows that for Dominating Set, OPT^{INC} selects $n\Delta/(\Delta+1)$ vertices, and for Connected Dominating Set, it selects all n vertices. □

Proposition 12 *For IDS on bipartite planar graphs, the following hold.*

(i) On always-connected graphs, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq n-1$.
(ii) On bounded-degree graphs, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta$.
(iii) On always-connected bounded-degree graphs, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta-1$.

Proof For (i), let G be a star, where the second vertex to arrive is the center vertex. Clearly, $\gamma_I(G) = 1$. Since the first vertex is always selected by any incremental algorithm, the center vertex cannot be selected. Consequently, all $n-1$ vertices of degree 1 have to be selected in the dominating set, which proves the lower bound of the first part.

For (ii), note that the adversary can give any number of copies of G.

For (iii), note that the adversary can make arbitrarily many copies of G and connect two consecutive copies by identifying two vertices of degree 1, one from each copy. □

Proposition 13 *For IDS,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \leq \Delta \leq n-1$

Proof To prove the upper bound of Δ, consider any graph, G, with maximum degree Δ, and let $S = \{s_1, \ldots, s_k\}$ be an independent dominating set of G of size $k = \gamma_I(G)$.

Let $R_1, \ldots, R_k$ be a partition of V such that all vertices in R_i are dominated by s_i. Let $R'_i = R_i \setminus \{s_i\}$ and note that $R'_1, \ldots, R'_k$ is a partition of $V \setminus \{s_1, \ldots, s_k\}$. For each i, the vertex s_i can be in an independent dominating set D only if $R'_i \cap D = \emptyset$. Thus, $|D| \leq \sum_{i=1}^k \max\{|\{s_i\}|, |R'_i|\} = \sum_{i=1}^k \max\{1, |R'_i|\}$, and $|D|/|S|$ is bounded by the maximum possible size of R'_i, which is Δ. Since $\Delta \leq n-1$ for all simple graphs, this concludes the proof. □

Lemma 6 *For any positive integer $n \geq 3$ and P_n given in the standard order,*

$$\textsc{Opt}^{\text{INC}}(P_n) = \lceil n/2 \rceil \quad \textit{and} \quad \textsc{Opt}_C^{\text{INC}}(P_n) = \textsc{Opt}_T^{\text{INC}}(P_n) = n - 1\,.$$

Proof The result for $\textsc{Opt}^{\text{INC}}$ follows from Lemma 5(iii) and the fact that selecting the vertices with odd index results in an incremental dominating set.

For $\textsc{Opt}_C^{\text{INC}}$, note that v_1 must be selected and hence, each v_i, $2 \leq i \leq n-1$, must be selected no later than when v_{i+1} arrives.

The result on $\textsc{Opt}_T^{\text{INC}}$ follows from Lemma 2 and the result on $\textsc{Opt}_C^{\text{INC}}$. $\square$

A *fan* of degree Δ is the graph obtained from a path P_Δ by addition of a vertex v that is adjacent to all vertices of the path, as in Figure 2. The adversarial order of a fan is defined by the standard order of the underlying path, followed by the vertex v.

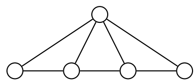

Figure 2 A fan of degree 4 (Proposition 14).

Proposition 14 *For always-connected planar graphs, the following hold.*

(i) For DS, $\mathbb{CR}^{\text{OFF}}(\textsc{Opt}^{\text{INC}}) \geq n/2$.
(ii) For CDS, $\mathbb{CR}^{\text{OFF}}(\textsc{Opt}^{\text{INC}}) \geq n - 2$.
(iii) For TDS, $\mathbb{CR}^{\text{OFF}}(\textsc{Opt}^{\text{INC}}) \geq n/2 - 1$.

Proof Let G be a fan of degree $\Delta = n-1$, where n is even, given in the adversarial order. By Lemma 6, $\textsc{Opt}^{\text{INC}}(G) = n/2$ and $\textsc{Opt}_C^{\text{INC}}(G) = \textsc{Opt}_T^{\text{INC}}(G) = n-2$. Furthermore, $\gamma(G) = \gamma_C(G) = 1$, and $\gamma_T(G) = 2$, since v_n forms a connected dominating set of size 1, and $\{v_1, v_n\}$ is a total dominating set of size 2. This proves (i)–(iii). $\square$

An *alternating fan* with k fans of degree Δ consists of k copies of the fan of degree Δ, where the individual copies are joined in a path-like manner by identifying some of the vertices of degree 2, as in Figure 3. Thus, $n = k(\Delta + 1) - (k - 1)$ and $k = (n-1)/\Delta$. The adversarial order of an alternating fan is defined by the concatenation of the adversarial orders of the underlying fans.

A *bridge of degree Δ with k sections* is obtained from a path of $k(\Delta-2)$ vertices $v_1, v_2, \ldots, v_{k(\Delta-2)}$, in that order, together with k vertices $u_1, u_2, \ldots, u_k$. For $1 \leq i \leq k$, u_i is connected to the $\Delta - 2$ vertices

$$v_{(i-1)(\Delta-2)+1}, v_{(i-1)(\Delta-2)+2}, \ldots, v_{i(\Delta-2)},$$

and for $1 \leq i \leq k-1$, u_i is connected to u_{i+1}. See Figure 4 for an example. The adversarial order of a bridge of degree Δ with k sections is v_1, v_2, $\ldots$, $v_{k(\Delta-2)}$, u_1, u_2, $\ldots$, u_k.

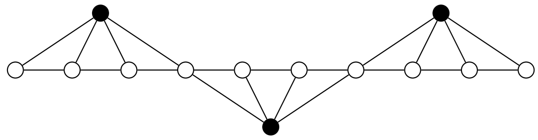

Figure 3 An alternating fan with 3 fans of degree 4 (Proposition 15(i)).

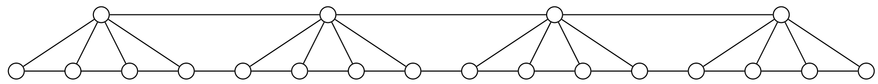

Figure 4 A bridge of degree 6 with 4 sections (Proposition 15(ii)).

For even k, a *modular bridge of degree Δ with k sections* is the same as a bridge of degree $\Delta - 1$ with k sections, except that for even i, the edge between u_i and u_{i+1} is not present.

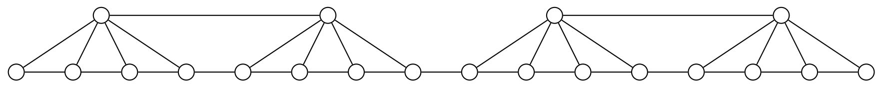

Figure 5 A modular bridge of degree 5 with 4 sections (Proposition 15(iii)).

Proposition 15 *For always-connected bounded-degree planar graphs, the following hold.*

(i) For DS, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq (\Delta - 1)/2$.
(ii) For CDS, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta - 2$.
(iii) For TDS, $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta - 1$.

Proof For (i), let G be an alternating fan with k fans of degree Δ, for any $\Delta \geq 4$, given in the adversarial order. We prove that $\text{OPT}^{\text{INC}}(G) > n(\Delta - 1)/(2\Delta)$ and $\gamma(G) \leq (n-1)/\Delta$. Starting with the latter, a fan consists of $\Delta + 1$ vertices, but the fans share one vertex, so a new one starts every Δ vertices, except for the final vertex which accounts for the -1. For the former claim, in Figure 3, the vertices belonging to a dominating set of size $k = (n-1)/\Delta$ are filled in (black). Since, by Lemma 6, any incremental dominating set on a path P in the standard order has at least $\lceil |V(P)|/2 \rceil$ vertices, OPT^{INC} must select at least $\lceil (n-k)/2 \rceil$ vertices of G. Inserting $k = (n-1)/\Delta$ into $(n-k)/2$ gives $(n(\Delta - 1) + 1)/(2\Delta)$, resulting in a ratio larger than $(\Delta - 1)/2$.

For (ii), let G be a bridge of degree Δ with k sections, given in the adversarial order, and let $m = k(\Delta - 2)$. By Lemma 6, we have $\text{OPT}^{\text{INC}}(G) \geq \text{OPT}^{\text{INC}}(P_m) = k(\Delta - 2) - 1$. The last k vertices form a connected dominating set of G and, thus, $\gamma_C(G) \leq k$.

For (iii), let G be a modular bridge of degree Δ with k sections given in the adversarial order. Let $m = k(\Delta - 1)$. By Lemma 6, we have $\textsc{Opt}^{\text{INC}}(G) \geq \textsc{Opt}^{\text{INC}}(P_m) = k(\Delta - 1) - 1$. Clearly, $\gamma_T(G) \leq k$, and the result follows. □

For any $n \geq 2$, a *two-sided fan of size n* is the graph obtained from a path on $n-2$ vertices by attaching two additional vertices, one to the even-numbered vertices of the path and the other to the odd-numbered vertices of the path. The two additional vertices are connected by an edge. An adversarial order of a two-sided fan is defined by the standard order of the path, followed by the two additional vertices in any order. See Figure 6 for an illustration of a two-sided fan of size 10.

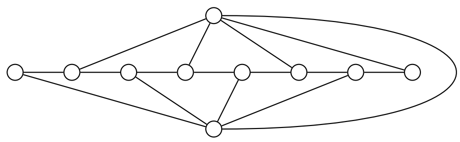

Figure 6 A two-sided fan of size 10 (Proposition 16).

Proposition 16 *For both CDS and TDS on always-connected bipartite planar graphs, we have* $\mathbb{CR}^{\text{OFF}}(\textsc{Opt}^{\text{INC}}) \geq (n-3)/2$.

Proof Let G_n be a two-sided fan of size n, given in an adversarial order. It suffices to prove that $\textsc{Opt}_C^{\text{INC}}(G_n) = \textsc{Opt}_T^{\text{INC}}(G_n) = n - 3$ and $\gamma(G_n) = \gamma_C(G_n) = \gamma_T(G_n) = 2$. This is straightforward from the facts that the first $n-2$ vertices of G induce a path and any incremental connected or total dominating set on P_{n-2} given in the standard order has size at least $n-3$. □

5 Conclusion and Open Problems

Online algorithms for four variants of the dominating set problem are analyzed using competitive analysis comparing to $\textsc{Opt}^{\text{INC}}$ and $\textsc{Opt}^{\text{OFF}}$, two reasonable alternatives for the optimal algorithm having knowledge of the entire input. Several graph classes are considered, and tight results are obtained in most cases.

The difference between $\textsc{Opt}^{\text{INC}}$ and $\textsc{Opt}^{\text{OFF}}$ is that $\textsc{Opt}^{\text{INC}}$ is required to maintain an incremental solution (as any online algorithm), while $\textsc{Opt}^{\text{OFF}}$ is only required to produce a solution for the final graph. The online algorithms are compared to both $\textsc{Opt}^{\text{INC}}$ and $\textsc{Opt}^{\text{OFF}}$, and $\textsc{Opt}^{\text{INC}}$ is compared to $\textsc{Opt}^{\text{OFF}}$, in order to investigate why all online algorithms tend to perform poorly against $\textsc{Opt}^{\text{OFF}}$. Is this due only to the requirement to be incremental, or is it more generally because of the lack of knowledge of the future?

Inspecting the results in the tables, perhaps the most striking conclusion is that the competitive ratios of any online algorithm and $\textsc{Opt}^{\text{INC}}$, respectively, against OPT^{OFF}, are almost identical. This indicates that the requirement to maintain an incremental dominating set is a severe restriction, which can be offset by the full knowledge of the input only to a very small extent. On the other hand, when we restrict our attention to online algorithms against OPT^{INC}, it turns out that the handicap of not knowing the future still presents a barrier, leading to competitive ratios of the order of n or Δ in most cases.

One could reconsider the nature of the irrevocable decisions, which originally stemmed from practical applications. Which assumptions on irrevocability are relevant for practical applications, and which irrevocability components make the problem hard from an online perspective? We expect that these considerations will apply to many other online problems as well.

There is relatively little difference observed between three of the variants of Dominating Set considered: Dominating Set, Connected Dominating Set, and Total Dominating Set. In fact, the results for Total Dominating Set generally followed directly from those for Connected Dominating Set as a consequence of Lemma 2. The results for Independent Dominating Set were significantly different from the others. It can be viewed as the minimum maximal independent set problem since any maximal independent set is a dominating set. This problem has been studied in the context of investigating the performance of the greedy algorithm for the independent set problem. In fact, the unique incremental independent dominating set is the set produced by the greedy algorithm for independent set.

In yet another orthogonal dimension, we compare the results for various graph classes. Dominating Set is a special case of Set Cover and is notoriously difficult in classical complexity, being NP-hard [16], $W[2]$-hard [10], and not approximable within $c \log n$ for any constant c on general graphs [12]. On the positive side, on planar graphs, the problem is FPT [1] and admits a PTAS [2], and it is approximable within $\log \Delta$ on bounded-degree graphs [8]. On the other hand, the relationship between the performance of online algorithms and structural properties of graphs is not particularly well understood. In particular, there are problems where the absence of knowledge of the future is irrelevant; examples of such problems in this work are CDS and TDS on trees, and IDS on any graph class. As expected, for bounded-degree graphs, the competitive ratios are of the order of Δ, but closing the gap between $\Delta/2$ and Δ seems to require additional ideas. On the other hand, for planar graphs, the problem, rather surprisingly, seems to be as difficult as the general case when compared to OPT^{OFF}. When online algorithms for planar graphs are compared to OPT^{INC}, we suspect there might be an algorithm with constant competitive ratio. At the same time, this case is the most notable open problem directly related to our results. Drawing inspiration from classical complexity, one may want to eventually consider more specific graph classes in the quest for understanding exactly what structural properties make the problem solvable. From this perspective, our consideration of planar, bipartite, and bounded-degree graphs is a natural first step.

Acknowledgment

The authors would like to thank an anonymous referee for constructive suggestions.

References

1. Alber, J., Bodlaender, H.L., Fernau, H., Kloks, T., Niedermeier, R.: Fixed parameter algorithms for dominating set and related problems on planar graphs. Algorithmica **33**(4), 461–493 (2002)
2. Baker, B.S.: Approximation algorithms for NP-complete problems on planar graphs. Journal of the ACM **41**(1), 153–180 (1994)
3. Berge, C.: Theory of Graphs and its Applications. Meuthen, London (1962)
4. Böhm, M., Sgall, J., Veselý, P.: Online colored bin packing. In: E. Bampis, O. Svensson (eds.) 12th International Workshop on Approximation and Online Algorithms (WAOA), *Lecture Notes in Computer Science*, vol. 8952, pp. 35–46. Springer (2015)
5. Borodin, A., El-Yaniv, R.: Online Computation and Competitive Analysis. Cambridge University Press (1998)
6. Boyar, J., Larsen, K.S.: The seat reservation problem. Algorithmica **25**(4), 403–417 (1999)
7. Chrobak, M., Sgall, J., Woeginger, G.J.: Two-bounded-space bin packing revisited. In: C. Demetrescu, M.M. Halldórsson (eds.) 19th Annual European Symposium (ESA), *Lecture Notes in Computer Science*, vol. 6942, pp. 263–274. Springer (2011)
8. Chvátal, V.: A greedy heuristic for the set-covering problem. Mathematics of Operations Research **4**(3), 233–235 (1979)
9. Das, B., Bharghavan, V.: Routing in ad-hoc networks using minimum connected dominating sets. In: IEEE International Conference on Communications (ICC), vol. 1, pp. 376–380 (1997)
10. Downey, R.G., Fellows, M.R.: Fixed-parameter tractability and completeness I: Basic results. SIAM Journal on Computing **24**(4), 873–921 (1995)
11. Du, D.Z., Wan, P.J.: Connected Dominating Set: Theory and Applications. Springer, New York (2013)
12. Feige, U.: A threshold of $\ln n$ for approximating set cover. Journal of the ACM **45**(4), 634–652 (1998)
13. Haynes, T.W., Hedetniemi, S., Slater, P.: Fundamentals of Domination in Graphs. Marcel Dekker, New York (1998)
14. Henning, M., Yao, A.: Total Domination in Graphs. Springer, New York (2013)
15. Karlin, A.R., Manasse, M.S., Rudolph, L., Sleator, D.D.: Competitive snoopy caching. Algorithmica **3**, 79–119 (1988)
16. Karp, R.M.: Reducibility among combinatorial problems. In: R.E. Miller, J.W. Thatcher (eds.) Complexity of Computer Computations, The IBM Research Symposia Series, pp. 85–103. Plenum Press, New York (1972)
17. King, G.H., Tzeng, W.G.: On-line algorithms for the dominating set problem. Information Processing Letters **61**(1), 11–14 (1997)
18. König, D.: Theorie der Endlichen und Unendlichen Graphen. Chelsea, New York (1950)
19. Liu, C.L.: Introduction to Combinatorial Mathematics. McGraw-Hill, New York (1968)
20. Ore, O.: Theory of Graphs, *Colloquium Publications*, vol. 38. American Mathematical Society, Providence (1962)
21. Sleator, D.D., Tarjan, R.E.: Amortized efficiency of list update and paging rules. Communications of the ACM **28**(2), 202–208 (1985)