Online Dominating Set

Boyar, Joan; Eidenbenz, Stephen J.; Favrholdt, Lene Monrad; Kotrbcik, Michal; Larsen, Kim Skak

Published in:
15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016

DOI:
10.4230/LIPIcs.SWAT.2016.21

Publication date:
2016

Document version:
Final published version

Citation for pulished version (APA):
Boyar, J., Eidenbenz, S. J., Favrholdt, L. M., Kotrbcik, M., & Larsen, K. S. (2016). Online Dominating Set. In R. Pagh (Ed.), *15th Scandinavian Symposium and Workshops on Algorithm Theory, SWAT 2016* (pp. 1-15). Article 21 Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik. https://doi.org/10.4230/LIPIcs.SWAT.2016.21

Online Dominating Set*

Joan Boyar[1], Stephan J. Eidenbenz[2], Lene M. Favrholdt[3], Michal Kotrbčík[4], and Kim S. Larsen[5]

1 **University of Southern Denmark, Odense, Denmark**
joan@imada.sdu.dk
2 **Los Alamos National Laboratory, Los Alamos, USA**
eidenben@lanl.gov
3 **University of Southern Denmark, Odense, Denmark**
lenem@imada.sdu.dk
4 **University of Southern Denmark, Odense, Denmark**
kotrbcik@imada.sdu.dk
5 **University of Southern Denmark, Odense, Denmark**
kslarsen@imada.sdu.dk

Abstract

This paper is devoted to the online dominating set problem and its variants on trees, bipartite, bounded-degree, planar, and general graphs, distinguishing between connected and not necessarily connected graphs. We believe this 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 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. In most cases, we obtain tight bounds on the competitive ratios. Our results show that requiring the graphs to be presented in a connected fashion allows the online algorithms to obtain provably better solutions. Furthermore, 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.

1998 ACM Subject Classification F.1.2 [Modes of Computation] Online Computation, G.2.2 [Graph Theory] Graph Algorithms, I.1.2 [Algorithms] Analysis of Algorithms

Keywords and phrases online algorithms, dominating set, competitive analysis, graph classes, connected graphs

Digital Object Identifier 10.4230/LIPIcs.SWAT.2016.21

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

* Supported in part by the Danish Council for Independent Research and the Villum Foundation.

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 [19], Berge [3], and Ore [21]. The concept of domination readily lends itself to modelling 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 [20] 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 [14], total domination [15], and connected domination [12], 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. The online algorithm must maintain a dominating set, i.e., after each vertex has arrived, D must be a dominating set for the subgraph given so far. In particular, this means that the first vertex must always be included in the solution, except for the case of total dominating set. Since the graph consisting of a single vertex does not have a total dominating set at all, we allow an online algorithm for TDS to *not* include isolated vertices in the solution, unlike the other variants of DS. 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.

Defining the nature of the irrevocable decisions is a modelling 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 [10] 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 nonconnected dominating set. A similar observation is made in [18] for this model; their focus is 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. 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* [22, 16]. 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 Opt. Therefore, Opt has to solve the same problem and hence the restriction on Opt^{INC}. While such an attention to comparing algorithms to an appropriate 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 (1) [7] which considers *fair* algorithms that have to accept a request whenever possible, and thus require Opt to be fair as well, (2) [8] which studies *k-bounded-space* algorithms for bin packing that have at any time at most k open bins and requires Opt to also adhere to this restriction, and (3) [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 Alg, we let $\text{Alg}(I)$ denote the size of the dominating set computed by Alg on I, and we define Alg to be *c-competitive* if there exists a constant α such that for all input sequences I, $\text{Alg}(I) \leq c\,\text{Opt}(I) + \alpha$, where Opt may be Opt^{INC} or Opt^{OFF}, depending on the context. The (asymptotic) *competitive ratio* of Alg is the infimum over all such c and we denote this $\mathbb{CR}^{\text{INC}}(\text{Alg})$ and $\mathbb{CR}^{\text{OFF}}(\text{Alg})$, respectively. In some results, we use the strict competitive ratio, i.e., the inequality above holds without an additive constant. For these results, when the strict result is linear in n, we write the asymptotic competitive ratio in Table 2 without any additive constant.

We consider the four dominating set problem variants on various graph types, including trees, bipartite, bounded-degree (letting Δ denote the maximum degree), and to some extent planar graphs. 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 connectivity assumption allows us to obtain provably better bounds on the performance of online algorithms, at least compared to Opt^{OFF}, and these bounds are of course more meaningful for the relevant applications.

The results for online algorithms are summarized in Tables 1 and 2. The results for Opt^{INC} against Opt^{OFF} are identical to the results of Table 2, except that for DS on trees, $\mathbb{CR}^{\text{OFF}}(\text{Opt}^{\text{INC}}) = 2$ and for DS on always-connected planar graphs, $\mathbb{CR}^{\text{OFF}}(\text{Opt}^{\text{INC}}) = \lceil n/2 \rceil$. The results are discussed in the conclusion.

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

Graph class	DS	CDS	TDS	IDS
Trees	2	1		1
Bipartite	$[n/4, n/2]$			
Always-connected bipartite	$n/4$			
Bounded degree	$[\frac{\Delta}{2}; \Delta + 1]$	$[\frac{\Delta}{2}; \Delta]$	$[\frac{\Delta}{2}; \Delta]$	
Always-connected bounded degree		$[\frac{\Delta}{2}; \Delta - 1]$		

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

Graph class	DS	CDS	TDS	IDS
Trees	$[2; 3]$	1	2	n
Bipartite	n		$n/2$	
Always-connected bipartite	$n/2$			
Bounded degree	$[\Delta; \Delta + 1]$	$\Delta + 1$	$[\Delta - 1; \Delta]$	Δ
Always-connected bounded degree	$[\frac{\Delta}{2}; \Delta + 1]$	$[\Delta - 2; \Delta - 1]$		$[\Delta - 1; \Delta]$
Planar	n		$n/2$	n
Always-connected planar				

2 Preliminaries

Since we are studying online problems, the order in which vertices are given is important. Throughout the paper, we will assume 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 Alg using this ordering. When no confusion can occur, we implicitly assume that the dominating set being constructed by an online algorithm 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 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. 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 $\textsc{Opt}^{\text{INC}}$, 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 $K_{1,n-1}$. A *leaf* is a vertex of degree 1, and an *internal* vertex is a vertex of degree at least 2. 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 $\textsc{Opt}^{\text{OFF}}$. We also use these indices on $\textsc{Opt}^{\text{INC}}$ to indicate which variant is under consideration. We use Δ to denote the maximum degree of the graph under consideration. Similarly, n denotes 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 a leaf, 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.

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 focus on the comparison of algorithms bound to the same irrevocable decisions. We do so by comparing any online algorithm with $\textsc{Opt}^{\text{INC}}$ and $\textsc{Opt}^{\text{OFF}}$, investigating the role played by the (absence of) knowledge of the future. We start by using the size of a given dominating set to bound the sizes of some connected or incremental equivalents.

▶ **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:*

1. *There is a connected dominating set S' of G such that $|S'| \leq |S| + 2(\mathrm{c}(S) - 1)$.*
2. *There is an incremental connected dominating set R' of G such that $|R'| \leq |R| + \mathrm{c}(R) - 1$.*
3. *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. The proof of 1. can be found in the full version of the paper [6].

To prove 2., 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. 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, 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 3., consider an algorithm $\textsc{Alg}$ 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

not in 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 20 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, 1. and 2. in Theorem 1 cannot be combined even on bipartite planar graphs. The situation is different for trees: Corollary 10 1. 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.

▶ **Proposition 2.** *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. ◀

Since a correct incremental algorithm is uniquely defined by this proposition by a forced move in every step, OPT^{INC} must behave exactly the same. This fills the column for independent dominating set in Table 1.

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. For connected dominating set on trees, PARENT is 1-competitive, even against OPT^{OFF}:

▶ **Proposition 3.** *For any tree T,* Parent(T) *outputs a connected dominating set of T and*

$$\textsc{Parent}(T) = \begin{cases} \gamma_C(T)+1 & \text{if } v_1 \text{ is a leaf of } T \\ \gamma_C(T) & \text{otherwise.} \end{cases}$$

Proof. For trees with at least two vertices, Parent selects the internal vertices plus at most one leaf. Clearly, the size of the minimal connected dominating set of any tree T equals the number of its internal vertices. ◀

To show that for TDS on trees, Parent is 1-competitive against $\textsc{Opt}^{\text{INC}}$, we prove:

▶ **Lemma 4.** *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, the set D_i induces a subgraph of G with at least two components, and let i be the smallest index with this property. It follows that the vertex v_i constitutes a singleton component of the subgraph induced by D_i. Thus, v_i cannot be dominated by any other vertex of D_i, contradicting that the solution was incremental. ◀

▶ **Corollary 5.** *For any tree T on n vertices,*

$$\textsc{Opt}_T^{\text{INC}}(T) = \textsc{Opt}_C^{\text{INC}}(T) = \begin{cases} \text{int}(T)+1 & \text{if } v_1 \text{ is a leaf of } T \\ \text{int}(T) & \text{otherwise,} \end{cases}$$

where $\text{int}(T)$ *is the number of internal vertices of T. Consequently, when given in the standard order* $\textsc{Opt}_C^{\text{INC}}(P_n) = \textsc{Opt}_T^{\text{INC}}(P_n) = n-1$ *for every* $n \geq 3$.

▶ **Proposition 6.** *For any $n \in \mathbb{Z}^+$ and P_n given in the standard order,* $\textsc{Opt}^{\text{INC}}(P_n) = \lceil n/2 \rceil$.

▶ **Proposition 7.** *For any online algorithm* Alg *for DS and $n > 0$, there is a tree T with n vertices such that the dominating set constructed by* Alg *for T has at least $n-1$ vertices.*

Proof. We prove that the adversary can maintain the invariant that at most one vertex is not included in the solution of 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 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 Alg must select an additional vertex. ◀

▶ **Proposition 8.** *For any always-connected bipartite graph G, the smaller partite set of G (plus, possibly, the vertex v_1) forms an incremental dominating set.*

As a corollary of Proposition 7 and Proposition 8, we get the following result.

▶ **Corollary 9.** *For any online algorithm* Alg *for DS on trees,* $\mathbb{CR}^{\text{INC}}(\textsc{Alg}) \geq 2$.

▶ **Corollary 10.** *For trees, the following hold.*

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

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.

▶ **Proposition 11.** *For any always-connected graph G, the set computed by* Parent *on G is an incremental connected dominating set of G.*

Proof. We prove the claim by induction on n. Since Parent always selects v_1, the statement holds for $n = 1$. Consider the graph G_i, for some $i > 1$, and assume that D_{i-1} is an incremental connected dominating set of G_{i-1}. If v_i is already dominated by a vertex in D_{i-1}, then Parent keeps D unchanged (that is, $D_i = D_{i-1}$) and thus D_i is an incremental connected dominating set of G_i. If v_i is not dominated by D_{i-1}, then Parent chooses a neighbor v of v_i in G_{i-1}. Clearly, this implies that D_i is an incremental dominating set of G_i. Since D_{i-1} is an incremental connected dominating set of G_{i-1} and the vertex v is adjacent to the only component of D_{i-1}, D_i is connected, which concludes the proof. ◀

▶ **Proposition 12.** *For DS and CDS on always-connected bipartite graphs, $\mathbb{CR}^{\text{OFF}}(\text{Parent}) \leq n/2$.*

▶ **Proposition 13.** *Let G be a graph with n vertices and maximum degree Δ. For any graph G, $\gamma_C(G) \geq \gamma(G) \geq n/(\Delta+1)$ and $\gamma_T(G) \geq n/\Delta$.*

Proposition 13 implies that any algorithm computing an incremental dominating set is no worse than $(\Delta+1)$-competitive.

▶ **Corollary 14.** *For any algorithm* Alg *for DS, $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \leq \Delta + 1$. Furthermore, for any algorithm* Alg *for TDS, $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \leq \Delta$.*

▶ **Proposition 15.** *For any algorithm* Alg *for CDS, $\mathbb{CR}^{\text{OFF}}(\text{Alg}) \leq \Delta - 1$.*

In the next result and in Proposition 19 in Section 4 we use *layers* in an always-connected graph G defined by letting L assign layer numbers to vertices in the following manner. Let $L(v_1) = 0$ and for $i > 1$, $L(v_i) = 1 + \min\{L(v_j) \mid v_j \text{ is a neighbor of } v_i \text{ in } G_i\}$.

Our next aim is to show that for always-connected bipartite graphs, there is an $n/4$-competitive algorithm against Opt^{INC}. This is achieved by considering the following *first parent* algorithm, denoted FirstParent, which generalizes Parent. For DS and CDS, the algorithm FirstParent always selects v_1 and 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 TDS, we add the following to FirstParent, so that the dominating set produced is total: If, when v_i arrives, v_i and v_j $(j < i)$ are the only vertices of a component of size 2, then besides v_j, FirstParent also selects v_i.

▶ **Theorem 16.** *For DS, CDS, and TDS on always-connected bipartite graphs, we have $\mathbb{CR}^{\text{INC}}(\text{FirstParent}) \leq n/4$ for $n \geq 4$.*

Proof. We consider DS and CDS first. Since FirstParent is an instantiation of Parent, Proposition 11 implies that the incremental dominating set constructed by FirstParent is connected. Therefore, the fact that for any graph G with at least three vertices $\text{Opt}^{\text{INC}}(G) \leq \text{Opt}_T^{\text{INC}}(G) \leq \text{Opt}_C^{\text{INC}}(G) + 1$ implies that it is sufficient to prove that FirstParent is $n/4$-competitive against Opt^{INC}. Furthermore, we only need to consider the case $\text{Opt}^{\text{INC}}(G) < 4$, since otherwise FirstParent is trivially $n/4$-competitive. Since G is bipartite, there are no edges between vertices of a single layer. Our first aim is to bound the number of layers.
Claim: If $\text{Opt}^{\text{INC}}(G) < 4$, then G has at most 6 layers.

To establish the claim, we prove that if an always-connected graph G has $2k+1$ layers, then $\text{Opt}^{\text{INC}}(G) > k$. For the sake of contradiction, suppose that there exist graphs G that are always-connected with $2k+1$ layers such that $\text{Opt}^{\text{INC}}(G) \leq k$, and among all such graphs

choose one, G, with the smallest number of vertices. Since any dominating set contains at least one vertex, we have $k \geq 1$. Let D be an incremental dominating set of G with $|D| \leq k$ and let l be the largest integer such that G_l has $2k-1$ layers. Since G is the smallest counterexample, we have $\textsc{Opt}^{\text{INC}}(G_l) \geq k$. Recall that D_l is defined as $D \cap G_l$. The fact that D is an incremental dominating set implies that D_l is a dominating set of G_l. We claim that $|D_l| = k$, since otherwise D_l would be an incremental dominating set of G_l with $|D_l| < k$, contradicting the fact that $\textsc{Opt}^{\text{INC}}(G_l) \geq k$. The fact that $|D_l| = k$ is equivalent to $D \subseteq V(G_l)$ and, in particular, $L(v) \leq 2k-1$ for each vertex v from D. Let w be a vertex of G such that $L(w) = 2k+1$, such a vertex exists since G has $2k+1$ layers. By the definition of layers the vertex w does not have a neighbor in any of the first $2k-1$ layers and thus is not adjacent to any vertex of D, contradicting the fact that D is a dominating set of G. This concludes the proof of the claim.

In the rest of the proof, we distinguish several cases according to the number of layers of G. If there are at most two layers, then FirstParent selects only the root v_1 and the result easily follows. Let l_i denote the size of the i-th layer and s_i the number of vertices selected by FirstParent from the i-th layer. For convenience, we will ignore the terms s_0 and l_0, both of which are one, which is viable since we are dealing with the asymptotic competitive ratio. Because FirstParent can add a vertex from the i-th layer to the dominating set only when a (non-dominated) vertex from the $(i+1)$-st layer arrives, we have

$$s_i \leq l_{i+1}. \tag{Ai}$$

Clearly,

$$s_i \leq l_i. \tag{Bi}$$

The letter i in equations (A) and (B) indicates the layer for which the equation is applied. If there are precisely three layers, then $\textsc{Opt}^{\text{INC}}(G) \geq 2$ and we must prove that $s_1 + s_2 \leq n/2$. However, $s_2 = 0$, and $s_1/2 \leq l_1/2$ by $(B1)$ and $s_1/2 \leq l_2/2$ by $(A1)$. Adding the last two inequalities yields $s_1 \leq l_1/2 + l_2/2 = n/2$, as required.

We use the same idea as for three layers also in the cases of four and five layers, albeit the counting is slightly more complicated. First we deal separately with the case where $\textsc{Opt}^{\text{INC}}(G) = 2$, and, consequently, there are four layers. Note that the two vertices in the optimal solution are necessarily in layers 0 and 2, and it follows that $l_2 = 1$. Furthermore, $(A1)$ implies that $s_1 \leq 1$ and $(B2)$ implies that $s_2 \leq 1$. Since $s_3 = 0$, FirstParent always selects at most 3 vertices, which yields the desired result. Assume now that $\textsc{Opt}^{\text{INC}}(G) \geq 3$ and therefore, our aim is to prove that $\textsc{FirstParent}(G) \leq 3n/4$. Adding 1/4 times $(A1)$, 3/4 times $(B1)$, 1/2 times $(A2)$, and 1/2 times $(B2)$ yields

$$s_1 + s_2 \leq 3l_1/4 + 3l_2/4 + l_3/2. \tag{1}$$

If there are four layers, then $s_3 = 0$ and the right-hand side of (1) satisfies $3l_1/4+3l_2/4+l_3/2 \leq 3(l_1+l_2+l_3)/4 = 3n/4$, which yields the desired result. If there are five layers, we add 3/4 times $(A3)$ and 1/4 times $(B3)$ to (1), which gives $s_1+s_2+s_3 \leq 3(l_1+l_2+l_3+l_4)/4 = 3n/4$, as required. The last remaining case is that of six layers and $\textsc{Opt}^{\text{INC}}(G) = 3$, which is dealt with similarly to that of four layers and $\textsc{Opt}^{\text{INC}}(G) = 2$. In particular, the vertices selected by $\textsc{Opt}^{\text{INC}}$ necessarily lie in layers 0, 2, and 4, and thus $l_0 = l_2 = l_4 = 1$. Now observing that $s_5 = 0$ and adding (Bi) for all even i to (Ai) for $i = 1$ and $i = 3$ yields that $\textsc{FirstParent}(G) \leq 5$, which implies the result in the always-connected case.

For TDS, the additional vertices accepted by FirstParent must by accepted by any incremental online algorithm, so the result also holds for TDS. ◀

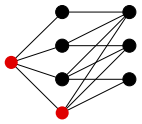

Figure 1 A two-layer construction; the minimum connected dominating set is depicted in red (Proposition 18).

▶ **Proposition 17.** *For DS, CDS, and TDS, we have* $\mathbb{CR}^{\text{INC}}(\textsc{FirstParent}) \leq n/2$ *for* $n \geq 2$.

Proof. Since for any graph, FirstParent constructs an incremental dominating set, we need to consider only the cases where $\textsc{Opt}^{\text{INC}}(G) \leq 1$, $\textsc{Opt}_C^{\text{INC}}(G) \leq 1$, and $\textsc{Opt}_T^{\text{INC}}(G) \leq 1$. For TDS, either G has no edges, in which case the empty set of vertices is a feasible solution constructed both by $\textsc{Opt}_T^{\text{INC}}$ and FirstParent, or G contains an edge, in which case $\textsc{Opt}_T^{\text{INC}}(G) \geq 2$ and the bound follows. Since $\textsc{Opt}^{\text{INC}}(G) \leq \textsc{Opt}_C^{\text{INC}}(G)$, it is sufficient to consider the case where $\textsc{Opt}^{\text{INC}}(G) = 1$. If, at any point, G_i has more than one component, then $\textsc{Opt}^{\text{INC}}(G_i) \geq 2$. Thus, if $\textsc{Opt}^{\text{INC}}(G_i) = 1$, G is a star and is always-connected. Thus, the center vertex must arrive as either the first or second request, so $\textsc{FirstParent}(G) \leq 2 \leq n$. ◀

▶ **Proposition 18.** *For any online algorithm* Alg *for DS, CDS, or TDS on always-connected bipartite graphs,* $\mathbb{CR}^{\text{INC}}(\textsc{Alg}) \geq n/4$ *and* $\mathbb{CR}^{\text{INC}}(\textsc{Alg}) \geq \Delta/2$.

Proof. We prove that for any online algorithm Alg for DS, CDS, or TDS and for any integer $\Delta \geq 2$, there is a bipartite graph G with maximum degree Δ such that $\textsc{Alg}(G) = \Delta \geq n/2$ and $\textsc{Opt}^{\text{INC}}(G) = \textsc{Opt}_C^{\text{INC}}(G) = \textsc{Opt}_T^{\text{INC}}(G) = 2$. Consider the graph consisting of a root v, Δ vertices $u_1, \ldots, u_\Delta$ adjacent to the root and constituting the first layer, and an additional $\Delta - 1$ vertices $w_1, \ldots, w_{\Delta-1}$, which will be given in that order, constituting the second layer, with adjacencies as follows: For $i = 1, \ldots, \Delta - 1$, the i-th vertex w_i of the second layer is adjacent to $\Delta - i + 1$ vertices of the first layer in such a way that we obtain the following strict set containment of sets of neighbors of these vertices: $N(w_i) \supset N(w_{i+1})$ for all $i = 1, \ldots, \Delta - 2$. An example of this construction for $\Delta = 4$ is depicted in Figure 1. After the entire first layer is presented to the algorithm, the vertices of the first layer are indistinguishable to the algorithm and $D_{\Delta+1}$ does not necessarily contain more than one vertex. For each $i = 1, \ldots, \Delta - 1$, the neighbors of w_i are chosen from the first layer in such a way that $N(w_{i-1}) \supset N(w_i)$, the degree of w_i is $\Delta - i + 1$, and $N(w_i)$ contains as many vertices not contained in the dominating set constructed by Alg so far as possible. Consider the situation when the vertex w_i arrives. It is easy to see that if the set $N(w_i)$ does not contain a vertex from the dominating set constructed so far, then Alg must select at least one additional vertex at this time. The last observation implies that Alg selects at least $\Delta - 1$ vertices from the first and second layer, plus the root.

Since there is a vertex u in the first layer that is adjacent to all vertices in the second layer, $\{u, v\}$ is an incremental connected dominating set of G, which concludes the proof. ◀

4 The Cost of Being Incremental

This section is devoted to comparing the performance of incremental algorithms and $\textsc{Opt}^{\text{OFF}}$. Since $\textsc{Opt}^{\text{OFF}}$ performs at least as well as $\textsc{Opt}^{\text{INC}}$ and $\textsc{Opt}^{\text{INC}}$ performs at least as well as any online algorithm, each lower bound in Table 2 is at least the maximum of the corresponding

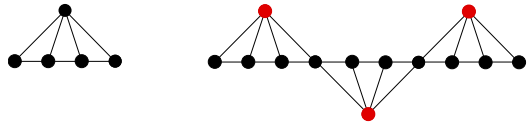

Figure 2 A fan with $\Delta = 4$ (left; Proposition 24) and an alternating fan with $k = 3$ and $\Delta = 4$ (right; Proposition 25).

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 and corresponding upper bound for $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}})$ is at least 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.

The following result generalizes the idea of Proposition 8.

▶ **Proposition 19.** *For DS on always-connected graphs,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \leq n/2$.

Proof. For a fixed ordering of G, consider the layers $L(v)$ assigned to vertices of G. It is easy to see that the set of vertices in the even layers is an incremental solution for DS and similarly for the set of vertices in odd layers plus the vertex v_1. Therefore, OPT^{INC} can select the smaller of these two sets, which necessarily has at most $n/2$ vertices. ◀

▶ **Proposition 20.** *The following hold for the strict competitive ratio:*

- *For DS on bipartite planar graphs,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq n - 1$ *and* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta$.
- *For CDS on bipartite planar graphs,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq n$.

▶ **Proposition 21.** *For IDS and for the strict competitive ratio,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta$ *and* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq n - 1$.

▶ **Proposition 22.** *For IDS on always-connected graphs,* $\Delta - 1 \leq \mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \leq \Delta$.

Theorem 1 3. implies the following bound on the performance of OPT^{INC} on trees.

▶ **Corollary 23.** *For DS on trees,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \leq 2$.

All of the following results are lower bounds. Specific examples of the families of graphs used to obtain these lower bounds are depicted in the following figures; the details of the proofs appear in the full paper [6].

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.

▶ **Proposition 24.** *For always-connected planar graphs (and, thus, also on general planar graphs), the following strict competitive ratio results hold.*

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

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 2. 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.

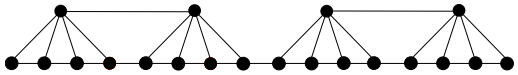

Figure 3 A modular bridge with $k = 4$ and $\Delta = 5$ (Proposition 26).

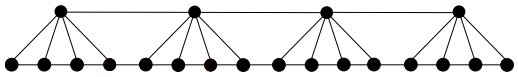

Figure 4 A bridge with $k = 4$ and $\Delta = 6$ (Proposition 27).

▶ **Proposition 25.** *For DS on always-connected graphs,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq (\Delta - 1)/2$.

A *modular bridge of degree* Δ *with* k *sections*, where k is even, is the graph obtained from a path on $k(\Delta - 1)$ vertices, with an additional k *chord* vertices. There is a perfect matching on the chord vertices $u_1, \ldots, u_k$ with u_{2i} is adjacent to u_{2i-1} for all $i = 1, \ldots, k/2$. Furthermore, the i-th chord vertex is adjacent to the vertices of the i-th *section*; see Figure 3 for an example. The adversarial order of a modular bridge is defined by the standard order of the path, followed by the chord vertices in any order.

▶ **Proposition 26.** *For TDS on always-connected graphs,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta - 1$.

A *bridge of degree* Δ *with* k *sections* is obtained from a modular bridge of degree $\Delta - 1$ with k sections by joining vertices u_{2i} and u_{2i+1} by an edge for each $i = 1, \ldots, k/2 - 1$; see Figure 4 for an example. The adversarial order of a bridge is identical with the adversarial order of the underlying modular bridge.

▶ **Proposition 27.** *For CDS on always-connected graphs,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta - 2$.

A *rotor of degree* Δ, where $\Delta \geq 2$ is even, is a graph obtained from a star, $K_{1,\Delta}$, on $\Delta + 1$ vertices by adding the edges of a perfect matching on the pendant vertices, as in Figure 5. The adversarial order of a rotor G of degree Δ is any fixed order such that G_{2i} is a graph with a perfect matching for each $i = 1, \ldots, \Delta/2$ and the central vertex of the original star is the last vertex to arrive.

▶ **Proposition 28.** *For CDS,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta + 1$*, and for TDS,* $\mathbb{CR}^{\text{OFF}}(\text{OPT}^{\text{INC}}) \geq \Delta/2$.

For any $n \geq 2$, the *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 adversarial order of a two-sided fan is defined by the standard order of the path, followed by the two additional vertices. See Figure 5 for an illustration of a two-sided fan of size 10.

▶ **Proposition 29.** *For any incremental algorithm* ALG *for CDS or TDS on always-connected bipartite graphs,* $\mathbb{CR}^{\text{OFF}}(\text{ALG}) \geq (n - 3)/2$ *holds for the strict competitive ratio.*

5 Conclusion and Open Problems

Online algorithms for four variants of the dominating set problem are compared using competitive analysis to OPT^{INC} and OPT^{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.

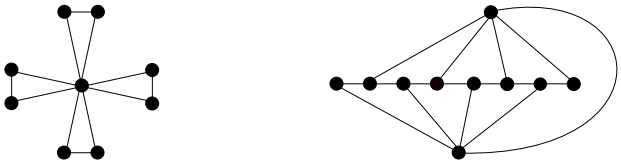

Figure 5 The rotor of degree 8 (left, Proposition 28) and two-sided fan of size 10 (right, Proposition 29).

The difference between OPT^{INC} and OPT^{OFF} is that OPT^{INC} is required to maintain an incremental solution (as any online algorithm), while OPT^{OFF} is only required to produce an offline solution for the final graph. The algorithms are compared to both OPT^{INC} and OPT^{OFF}, and OPT^{INC} is compared to OPT^{OFF}, in order to investigate why all algorithms tend to perform poorly against OPT^{OFF}. Is this due to the requirement to be incremental, or is it 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 OPT^{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 4. 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 [17], $W[2]$-hard [11], and not approximable within $c \log n$ for any constant c on general graphs [13]. On the positive side, on planar graphs, the problem is FPT [1], admits a PTAS [2], and is approximable within $\log \Delta$ on bounded degree graphs [9]. 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 could 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 the anonymous referees for constructive comments.

References

1 J. Alber, H. L. Bodlaender, H. Fernau, T. Kloks, and R. Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002.

2 B. S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, 1994.

3 C. Berge. *Theory of Graphs and its Applications*. Meuthen, London, 1962.

4 M. Böhm, J. Sgall, and P. Veselý. Online colored bin packing. In E. Bampis and O. Svensson, editors, *12th International Workshop on Approximation and Online Algorithms (WAOA)*, volume 8952 of *Lecture Notes in Computer Science*, pages 35–46. Springer, 2015.

5 A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

6 J. Boyar, S. J. Eidenbenz, L. M. Favrholdt, M. Kotrbčík, and K. S. Larsen. Online dominating set. Technical Report arXiv:1604.05172 [cs.DS], arXiv, 2016.

7 J. Boyar and K. S. Larsen. The seat reservation problem. *Algorithmica*, 25(4):403–417, 1999.

8 M. Chrobak, J. Sgall, and G. J. Woeginger. Two-bounded-space bin packing revisited. In C. Demetrescu and M. M. Halldórsson, editors, *19th Annual European Symposium (ESA)*, volume 6942 of *Lecture Notes in Computer Science*, pages 263–274. Springer, 2011.

9 V. Chvátal. A greedy heuristic for the set-covering problem. *Mathematics of Operations Research*, 4(3):233–235, 1979.

10 B. Das and V. Bharghavan. Routing in ad-hoc networks using minimum connected dominating sets. In *IEEE International Conference on Communications (ICC)*, volume 1, pages 376–380, 1997.

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

12 D.-Z. Du and P.-J. Wan. *Connected Dominating Set: Theory and Applications*. Springer, New York, 2013.

13 U. Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM*, 45(4):634–652, 1998.

14 T. W. Haynes, S. Hedetniemi, and P. Slater. *Fundamentals of Domination in Graphs*. Marcel Dekker, New York, 1998.

15 M. Henning and A. Yao. *Total Domination in Graphs*. Springer, New York, 2013.

16 A. R. Karlin, M. S. Manasse, L. Rudolph, and D. D. Sleator. Competitive snoopy caching. *Algorithmica*, 3:79–119, 1988.

17 R. M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972.

18 G.-H. King and W.-G. Tzeng. On-line algorithms for the dominating set problem. *Information Processing Letters*, 61(1):11–14, 1997.

19 D. König. *Theorie der Endlichen und Unendlichen Graphen*. Chelsea, New York, 1950.

20 C. L. Liu. *Introduction to Combinatorial Mathematics*. McGraw-Hill, New York, 1968.

21 O. Ore. *Theory of Graphs*, volume 38 of *Colloquium Publications*. American Mathematical Society, Providence, 1962.

22 D. D. Sleator and R. E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of the ACM*, 28(2):202–208, 1985.