Distributed Computing
https://doi.org/10.1007/s00446-023-00447-z

Near-optimal distributed dominating set in bounded arboricity graphs

Michal Dory[1] · Mohsen Ghaffari[2] · Saeed Ilchi[3]

Received: 16 November 2022 / Accepted: 28 March 2023
© The Author(s) 2023

Abstract

We describe a simple deterministic $O(\varepsilon^{-1}\log\Delta)$ round distributed algorithm for $(2\alpha+1)(1+\varepsilon)$ approximation of minimum weighted dominating set on graphs with arboricity at most α. Here Δ denotes the maximum degree. We also show a lower bound proving that this round complexity is nearly optimal even for the unweighted case, via a reduction from the celebrated KMW lower bound on distributed vertex cover approximation (Kuhn et al. in JACM 63:116, 2016). Our algorithm improves on all the previous results (that work only for unweighted graphs) including a randomized $O(\alpha^2)$ approximation in $O(\log n)$ rounds (Lenzen et al. in International symposium on distributed computing, Springer, 2010), a deterministic $O(\alpha\log\Delta)$ approximation in $O(\log\Delta)$ rounds (Lenzen et al. in international symposium on distributed computing, Springer, 2010), a deterministic $O(\alpha)$ approximation in $O(\log^2\Delta)$ rounds (implicit in Bansal et al. in Inform Process Lett 122:21–24, 2017; Proceeding 17th symposium on discrete algorithms (SODA), 2006), and a randomized $O(\alpha)$ approximation in $O(\alpha\log n)$ rounds (Morgan et al. in 35th International symposiumon distributed computing, 2021). We also provide a randomized $O(\alpha\log\Delta)$ round distributed algorithm that sharpens the approximation factor to $\alpha(1+o(1))$. If each node is restricted to do polynomial-time computations, our approximation factor is tight in the first order as it is NP-hard to achieve $\alpha-1-\varepsilon$ approximation (Bansal et al. in Inform Process Lett 122:21-24, 2017).

Keywords Distributed computing · Dominating set · Arboricity · Approximation algorithms

1 Introduction

The minimum dominating set (MDS) problem is a classic and central problem in graph algorithms. In this problem, the goal is to construct a minimum weight set of nodes S such that each node is either in S or has a neighbor in S. The MDS problem has been widely studied both in the classic centralized setting and in the distributed setting, and it has various applications, for example, clustering and routing in ad-hoc networks. It is well-known that a simple greedy algorithm gives $\ln(\Delta+1)$ approximation for the problem for graphs with maximum degree Δ [17], and that it is NP-hard to obtain a $c\ln\Delta$-approximation for a suitable constant c [10]. A similar approximation can be also obtained by efficient distributed algorithms [9, 16, 18, 19]. In particular, Kuhn, Moscibroda, and Wattenhofer showed a randomized $O(\log\Delta)$-approximation algorithm that takes $O(\log^2\Delta)$ rounds in the CONGESTmodel, where messages are restricted to $O(\log n)$ bits, or $O(\log n)$ rounds in the LOCALmodel, where the message size is unbounded [19]. There is also a deterministic poly-logarithmic $O(\log\Delta)$-approximation algorithm for the problem in CONGESTthat is obtained by combining the algorithm of Deurer, Kuhn, and Maus [9] with the recent deterministic network decomposition of Rozhoň and Ghaffari [25]. If one allows unbounded messages and exponential local computation, one can obtain even $(1+\varepsilon)$-approximation for the problem in poly-logarithmic time in the LOCALmodel using the algorithm of Ghaffari, Kuhn, and Maus [13]. On the lower bound side, Kuhn, Moscibroda, and Wattenhofer showed that one needs $\Omega(\log\Delta/\log\log\Delta)$ rounds or $\Omega(\sqrt{\log n/\log\log n})$ rounds (the minimum of

✉ Saeed Ilchi
saeed.ilchi@inf.ethz.ch

Michal Dory
mdory@ds.haifa.ac.il

Mohsen Ghaffari
ghaffari@mit.edu

[1] Department of Computer Science, University of Haifa, Haifa, Israel

[2] Department of Electrical Engineering and Computer Science, MIT, Cambridge, Massachusetts, USA

[3] Department of Computer Science, ETH Zurich, Zurich, Switzerland

Published online: 15 May 2023

these two lower bounds applies) to get a logarithmic approximation [20].

Since it is NP-hard to obtain better than logarithmic approximation in general graphs, a large body of research focused on finding algorithms with a better approximation in special graph families, such as planar graphs, graphs of bounded expansion, and more (see, e.g., [1, 3, 6, 8, 22, 23, 27]). One prominent example is the class of *bounded arboricity* graphs, which informally speaking are graphs that are sparse everywhere. The arboricity α of a graph is the minimum number of forests into which its edges can be partitioned. The class of bounded arboricity graphs includes many important graph classes such as planar graphs, graphs of bounded treewidth or genus, and graphs excluding a fixed minor. Many real-world graphs are sparse and believed to have low arboricity, for example, the World Wide Web graph and graphs representing social networks. This led to extensive study of graph problems in low-arboricity graphs (see, e.g., [4, 7, 12, 14, 15, 21, 24]).

1.1 MDS in bounded arboricity graphs

Lenzen and Wattenhofer showed the first algorithms for MDS in bounded arboricity graphs [21]. In particular, they showed a randomized $O(\alpha^2)$-approximation algorithm that takes $O(\log n)$ time, and a deterministic $O(\alpha \log \Delta)$-approximation algorithm that takes $O(\log \Delta)$ time. An $O(\alpha^2)$-approximation can be also obtained deterministically in $O(\log n)$ time as was shown recently by Amiri [2]. All these algorithms work in the CONGESTmodel. A recent line of work shows $O(\alpha)$-approximation algorithms for the problem. First, a centralized algorithm of Bansal and Umboh gives $(2\alpha + 1)$-approximation for the problem [4].[1] They also show that it is NP-hard to obtain an $(\alpha - 1 - \epsilon)$-approximation for the problem. The algorithm of Bansal and Umboh is based on LP-rounding, and can be implemented efficiently in the CONGESTmodel using an algorithm for approximating the LP. This leads to a deterministic $O(\log^2 \Delta/\varepsilon^4)$-round $(2\alpha + 1)(1 + \epsilon)$-approximation algorithm using the $(1 + \epsilon)$-approximation algorithm of Kuhn, Moscibroda, and Wattenhofer for approximating the LP [19]. A recent combinatorial algorithm for MDS in bounded arboricity graphs was shown by Morgan, Solomon and Wein [24]; they obtain a randomized $O(\alpha \log n)$-round $O(\alpha)$-approximation algorithm for the problem in CONGEST. All the above algorithms solve the unweighted version of the problem. Lastly, a very recent centralized algorithm of Sun [26] gives an $(\alpha+1)$-approximation for the weighted version of the problem. This algorithm however seems inherently sequential and does not seem to translate to an efficient distributed algorithm (see section 1.3 for a more detailed discussion).

1.2 Our contribution

Our first contribution is showing that $O(\alpha)$-approximation for weighted MDS can be obtained in just $O(\log \Delta)$ rounds in the CONGESTmodel. In particular, we show a simple deterministic algorithm that gives the following. For this and all the other algorithms in this paper, we assume that both Δ and α are known to all nodes. Please see remark 1 and remark 2 for a discussion on the setting where Δ and α are unknown. We also assume that $\alpha \geq 2$ throughout the whole paper. For $\alpha = 1$, there is a simple 3-approximation algorithm (refer to Appendix 1. for the proof).

Theorem 1 *For any $0 < \varepsilon < 1$, there is a deterministic $(2\alpha + 1)(1 + \varepsilon)$-approximation algorithm for the minimum weighted dominating set problem in graphs with arboricity at most α. The algorithm runs in $O\left(\frac{\log(\Delta/\alpha)}{\varepsilon}\right)$ rounds in the* CONGEST*model.*

Our algorithm is faster compared to the two previous $O(\alpha)$-approximation algorithms that take $O(\log^2 \Delta/\varepsilon^4)$ rounds [4, 19] and $O(\alpha \log n)$ rounds [24], and its $O(\log \Delta)$ complexity is nearly optimal, as we will discuss later when describing lower bounds. Its approximation ratio of $(2\alpha + 1)(1+\varepsilon)$ matches the best approximation that was previously obtained by a distributed algorithm. It is a deterministic algorithm, where the previous $O(\alpha \log n)$ round algorithm was randomized. Moreover, to the best of our knowledge, our algorithm is the first distributed algorithm that solves the *weighted* version of the problem.

Improved approximation

We also show a randomized algorithm with an improved approximation of $\alpha(1 + o(1))$, giving the following.

Theorem 2 *For any $1 \leq t \leq \frac{\alpha}{\log \alpha}$, there is a randomized algorithm with expected $\left(\alpha + O\left(\frac{\alpha}{t}\right)\right)$-approximation factor for the minimum weighted dominating set problem in graphs with arboricity at most α. The algorithm runs in $O(t \log \Delta)$ rounds in the* CONGEST*model.*

In particular, by setting $t = \frac{\alpha}{\log \alpha}$, we can get $(\alpha + O(\log \alpha))$-approximation in $O(\alpha \log \Delta)$ time. For algorithms that are only using polynomial time computations, this approximation factor is tight in the first order as it is NP-hard to achieve $\alpha - 1 - \varepsilon$ approximation [4].

As a byproduct of our randomized algorithm, we improve on the approximation factor for the minimum dominating set problem on general graphs with maximum degree Δ. Previously, the best approximation factor for the problem in $O(k^2)$

[1] The paper claimed a 3α-approximation, but optimizing the parameters of the algorithm gives a $(2\alpha + 1)$-approximation, as was observed by Dvořák [11].

(for a parameter k) rounds was due to the work of Kuhn, Moscribroda, and Wattenhofer [19] where they provided a randomized algorithm with expected approximation factor $O(k\Delta^{\frac{2}{k}}\log\Delta)$. We drop the $O(\log\Delta)$ term in their result.

Theorem 3 *For any k, there is randomized algorithm that computes a weighted dominating set with expected approximation factor at most $\Delta^{\frac{1}{k}}(\Delta^{\frac{1}{k}}+1)(k+1)\cdot = O(k\Delta^{\frac{2}{k}})$ in $O(k^2)$ rounds in the* CONGEST*model.*

Lower Bound

Our algorithms provide $O(\alpha)$-approximation in $O(\log\Delta)$ rounds. A natural question is whether a logarithmic dependence on Δ is needed in the time complexity. While in general graphs it is known that $\Omega(\log\Delta/\log\log\Delta)$ rounds are required for obtaining a constant or logarithmic approximation [20], in special graph families faster algorithms are known. For example, in planar graphs one can obtain $O(1)$-approximation in $O(1)$ rounds in the LOCALmodel [22, 27]. Also, in the special case of trees that have arboricity 1, a trivial algorithm that takes all non-leaf nodes gives a 3-approximation for unweighted MDS (see Appendix 1 for the proof). Interestingly, we show that as soon as the arboricity is increased from 1 to 2, the locality of the minimum dominating set approximation problem changes radically, and any constant or poly-logarithmic approximation needs $\Omega\left(\frac{\log\Delta}{\log\log\Delta}\right)$ time, even in the LOCALmodel where the size of messages is unbounded.

Theorem 4 *Any distributed algorithm that computes any constant or poly-logarithmic approximation of the minimum dominating set on graphs of arboricity 2 requires $\Omega\left(\frac{\log\Delta}{\log\log\Delta}\right)$ rounds in the* LOCAL*model.*

Hence, the $O(\log\Delta)$ round complexity of our algorithms is nearly optimal.

1.3 Our techniques

At a high-level, our algorithms construct a dominating set in two steps. In the first step, we construct a partial dominating set S that has the following nice properties. First, the weight of the set S is a good approximation for the optimal dominating set. Second, the nodes T that are undominated by S have a nice structure that allows us to find efficiently a dominating set for T with a good approximation guarantee. In the second step, we construct a dominating set for T. In fact, in our first algorithm, we show that if for each node in T we add one node to the dominating set, we already obtain a $(2\alpha+1)(1+\varepsilon)$-approximation. In our second algorithm, we show how to exploit the structure of T to obtain a better approximation. Our algorithm for constructing the partial dominating set is inspired by the primal-dual method. To explain the idea, we first describe in section 3 a simpler variant of our algorithm that works for unweighted graphs. Next, in section 4, we generalize the algorithm for weighted graphs, and also show our randomized algorithm with improved approximation. Our lower bound appears in section 5.

Comparison to [26] We remark that in a very recent independent work [26], the author uses the primal-dual method to obtain approximation algorithm for MDS in bounded arboricity graphs in the centralized setting. This algorithm however does not seem to translate to an efficient distributed algorithm, as it has a reverse-delete step that makes it inherently sequential. In this step, the algorithm goes over all the nodes that were added to the dominating set S in reverse order and removes them from S if it is still a valid dominating set. This part is crucial for obtaining a good approximation ratio, and the analysis crucially relies on this part. While in our algorithm we also use the primal-dual method, we use it in a different way, and only to construct a partial solution for the problem.

2 Preliminaries

The input graph is $G=(V,E)$ with n nodes, m edges, maximum degree Δ, and arboricity α. For each node $v\in V$, let N_v be the set of neighbors of v and $N_v^+=\{v\}\cup N_v$. For a set of nodes $S\subseteq V$, let $N_S^+=\bigcup_{v\in S}N_v^+$ be set of nodes that are dominated by S. Let w_v be the weight of v and for set $S\subseteq V$, let $w_S=\sum_{v\in S}w_v$ be the total weight of the set S. We assume all the weights are positive integers and are bounded by n^c for some constant c.

Our algorithms are inspired by the primal-dual method. We associate a packing value $0\le x_v$ to each node such that for any node u, the value $X_u=\sum_{v\in N_u^+}x_v\le w_u$. From weak duality, we have the following.

Lemma 5 *For any feasible packing, $\sum_{v\in V}x_v\le \mathrm{OPT}$ where* OPT *is the weight of the minimum dominating set of G.*

Proof Let S^* be a dominating set of G with weight OPT. We have:

$$\mathrm{OPT}=\sum_{v\in S^*}w_v\ge\sum_{v\in S^*}X_v\ge\sum_{u\in V}x_u$$

The last inequality comes from the fact that S^* is a dominating set. Hence, each node u contributes to at least one of X_v for $v\in S^*$. □

Model

Our algorithms work in the standard CONGESTmodel. We have a communication network with n nodes that is identical to the input graph. Nodes communicate with each other by sending $O(\log n)$ bit messages in synchronous rounds. In the beginning, each node only knows its own weight and set of neighbors. At the end of the algorithm, each node should know if it is part of the constructed dominating set. Our lower bound works even for the more powerful LOCALmodel where the size of messages is unbounded.

3 Algorithm for unweighted MDS

As a warm-up, we start by describing a simpler variant of our algorithm that works for unweighted graphs, showing the following.

Theorem 6 *For any* $0 < \varepsilon < 1$*, there is a deterministic* $(2\alpha + 1)(1 + \varepsilon)$*-approximation algorithm for the minimum dominating set problem in unweighted graphs with arboricity at most* α *that runs in* $O\left(\frac{\log(\Delta/\alpha)}{\varepsilon}\right)$ *rounds in the* CONGEST*model.*

Our approach is inspired by the primal-dual method. Each node v has a packing value x_v, such that during the algorithm for each node, the value $X_v = \sum_{u \in N_v^+} x_u \leq 1$. From lemma 5 we have that $\sum_{v \in V} x_v \leq \text{OPT}$, where OPT is an optimal solution. Our algorithm first builds a partial dominating set S with the following properties.

Lemma 7 *For any* $0 < \varepsilon < 1$*, there is a deterministic algorithm that takes* $O\left(\frac{\log(\Delta/\alpha)}{\varepsilon}\right)$ *rounds and outputs a partial dominating set* $S \subseteq V$ *along with packing values* $\{x_v\}_{v \in V}$ *such that*

1. $|S| \leq (2\alpha + 1)(1 + \varepsilon) \sum_{v \in N_S^+} x_v$.
2. *For each node* $v \notin N_S^+$*, we have* $x_v \geq \frac{1}{(2\alpha+1)(1+\varepsilon)}$.

Before proving lemma 7, let us first describe how S can be extended to a dominating set. Let $T = V \backslash N_S^+$ be the set of undominated nodes. We next show that adding the nodes of T to the dominating set S results in the desired approximation.

Claim 8 The set $S \cup T$ is a dominating set of size at most $(2\alpha + 1)(1 + \varepsilon)\text{OPT}$.

Proof The set $S \cup T$ is clearly a dominating set, as we added to S all the undominated nodes. From lemma 7, we have that

$$|S| \leq (2\alpha + 1)(1 + \varepsilon) \sum_{v \in N_S^+} x_v,$$

$$|T| = \sum_{v \in T} 1 \leq (2\alpha + 1)(1 + \varepsilon) \sum_{v \in T} x_v.$$

Hence, we get that $|S \cup T| \leq (2\alpha + 1)(1 + \varepsilon) \sum_{v \in V} x_v \leq (2\alpha + 1)(1 + \varepsilon)\text{OPT}$, where the last inequality follows from lemma 5. □

To complete the proof, our goal is to prove lemma 7.

Proof of lemma 7 Let us start with the description of the algorithm. Let r be the integer such that

$$(1 + \varepsilon)^r \frac{1}{\Delta + 1} \leq \frac{1}{(2\alpha + 1)(1 + \varepsilon)} < (1 + \varepsilon)^{r+1} \frac{1}{\Delta + 1}.$$

Our algorithm consists of $r + 1$ iterations. At the beginning, all the nodes are unmarked and all packing values x_v are set to $\frac{1}{\Delta+1}$. Then, per iteration, we run the following on each node v. All nodes run each line simultaneously.

1. Compute $X_v = \sum_{u \in N_v^+} x_u$.
2. If $X_v \geq \frac{1}{1+\varepsilon}$, add v to S and mark all the nodes in N_v^+.
3. If v is not marked, set $x_v \leftarrow x_v(1 + \varepsilon)$.

A straightforward but important observation of this algorithm is the following.

Observation 9 For all nodes, X_v is always at most 1.

Proof At the start of the first iteration, $X_v \leq \frac{|N_v^+|}{\Delta+1} \leq 1$. At the beginning of iteration $i \geq 2$, if v is not in S, it means $X_v < \frac{1}{1+\varepsilon}$ in iteration $i - 1$ and so $X_v < 1$ in this iteration. If v is in S, then X_v is not changed after that and so $X_v \leq 1$. □

To provide the upper bound on the size of S in lemma 7, we use the following property of bounded arboricity graphs.

Observation 10 Let G be a graph with arboricity at most α, then the edges of G can be oriented such that the out-degree of each node is at most α.[2]

Observation 10 follows from the fact that the edges of the graph can be partitioned into α forests, and in each one of them we can orient the edges with out-degree one, by fixing a root in each tree and orienting the edges towards the root. We fix one of those orientations. We emphasize that this orientation is used only in the analysis, and we do not construct it in the algorithm. With this in mind, for each node v, let N_v^{in} be the set of incoming neighbors and N_v^{out} be the set of outgoing neighbors of v with respect to this fixed orientation. Note that for all nodes in N_S^+, the packing value is increased at most r times. Hence, from the choice of r, we have that $x_v \leq \frac{1}{(2\alpha+1)(1+\varepsilon)}$ for $v \in N_S^+$. For each $v \in S$,

[2] All of our algorithms work as long the input graph is orientable with the maximum out-degree of at most α. Hence, our results can be applied to the slightly larger class of graphs that are decomposable to at most α-pseudoforests. A psuedoforest is a graph in which every connected component has at most one cycle.

we have (all the values x_v are considered at the end of the algorithm):

$$X_v = \sum_{u \in N_v^{\text{in}}} x_u + x_v + \sum_{u' \in N_v^{\text{out}}} x_{u'} \geq \frac{1}{1+\varepsilon}$$

$$\Rightarrow \sum_{u \in N_v^{\text{in}}} x_u \geq \frac{1}{1+\varepsilon} - \frac{\alpha+1}{(2\alpha+1)(1+\varepsilon)}$$

$$\geq \frac{\alpha}{(2\alpha+1)(1+\varepsilon)}$$

Let $\lambda = \frac{\alpha}{(2\alpha+1)(1+\varepsilon)}$. From the above, we have that $\frac{1}{\lambda}\sum_{u\in N_v^{\text{in}}} x_u \geq 1$. We can bound the size of S as follows:

$$|S| = \sum_{v\in S} 1 \leq \frac{1}{\lambda}\sum_{v \in S}\sum_{u \in N_v^{\text{in}}} x_u$$

$$\leq \frac{1}{\lambda}\sum_{u\in N_S^+} x_u \sum_{v \in S} \mathbb{I}[u \in N_v^{\text{in}}]$$

$$\leq \frac{\alpha}{\lambda}\sum_{u \in N_S^+} x_u \leq (2\alpha+1)(1+\varepsilon)\text{OPT}.$$

Note that we bound $\sum_{v\in S}\mathbb{I}[u \in N_v^{in}]$ with α. This is because the out-degree of u is at most α, so u can be an incoming neighbor of at most α nodes.

So far we have proved the first guarantee of lemma 7. To conclude the proof of lemma, we should show that for each node $v \notin N_S^+$, we have $x_v \geq \frac{1}{(2\alpha+1)(1+\varepsilon)}$. This follows from the choice of r, as for nodes $v \notin N_S^+$ we increase the packing value $r+1$ times.

Each iteration can be implemented in $O(1)$ rounds in the CONGESTmodel. In total, $O(r) = O\left(\frac{\log(\Delta/\alpha)}{\varepsilon}\right)$ rounds. □

4 Algorithm for weighted MDS

We next show a generalized version of our algorithm from section 3. First, the algorithm works for weighted graphs. Second, we allow for different trade-offs between the two dominating sets we compute in the algorithm. Recall that in section 3 we start by computing a partial dominating set S, and then we add to it additional set S' to dominate the rest of the nodes. Previously we just constructed S' as the set of all undominated nodes, but we will see in section 4.2 an algorithm that exploits the structure of undominated nodes to get an improved approximation for this part. To get a better approximation for the whole algorithm, we can stop the algorithm for computing a partial dominating set earlier and get an improved approximation for the first part as well. We start by presenting our general scheme, which allows us to get a deterministic algorithm for weighted graphs with the same guarantees obtained in section 3, and then we show a randomized algorithm that can obtain an improved approximation.

4.1 Deterministic algorithm

We again follow the primal-dual method. We associate a packing value $0 \leq x_v$ to each node such that for any node u, the value $X_u = \sum_{v\in N_u^+} x_v \leq w_u$. For each node $v \in V$, let $\tau_v = \min_{u \in N_v^+} w_u$ be the minimum weight of a node that can dominate v.

Lemma 11 *For any $0 < \varepsilon < 1$ and $0 < \lambda < \frac{1}{(\alpha+1)(1+\varepsilon)}$, there is a deterministic algorithm that outputs a partial dominating set $S \subseteq V$ along with packing values $\{x_v\}_{v\in V}$ with the following properties:*

(a) $w_S \leq \alpha\left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \sum_{v\in N_S^+} x_v$.

(b) *For each undominated node $v \notin N_S^+$, its packing value x_v is at least $\lambda\tau_v$.*

The algorithm runs in $O\left(\frac{\log(\Delta\lambda)}{\varepsilon}\right)$ rounds in the CONGEST*model.*

Before proving this lemma, we illustrate it can be used to derive one of our main results which is presented in the following theorem.

Theorem 1 *For any $0 < \varepsilon < 1$, there is a deterministic $(2\alpha+1)(1+\varepsilon)$-approximation algorithm for the minimum weighted dominating set problem in graphs with arboricity at most α. The algorithm runs in $O\left(\frac{\log(\Delta/\alpha)}{\varepsilon}\right)$ rounds in the* CONGEST*model.*

Proof We run the algorithm of lemma 11 with the same ε and with λ equals to $\frac{1}{(2\alpha+1)(1+\varepsilon)}$. We want to find a set S' such that $S \cup S'$ is a dominating set. For this, we go over all the undominated nodes $v \notin N_S^+$, and add a node from N_v^+ with weight τ_v to S'. Clearly, $S \cup S'$ is a dominating set and its weight can be bounded as follows:

$$w_{S\cup S'} = w_S + \sum_{v \in V\setminus N_S^+} \tau_v$$

$$\leq \alpha\left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \sum_{v' \in N_S^+} x_{v'} + \sum_{v\in V\setminus N_S^+} \frac{x_v}{\lambda}$$

$$\leq (2\alpha+1)(1+\varepsilon)\sum_{v\in V} x_v$$

$$\leq (2\alpha+1)(1+\varepsilon)\cdot\text{OPT}$$

In the first inequality, we use property (a) of lemma 11 to bound the first term, and we invoked property (b) of this lemma, i.e. $\lambda\tau_v \leq x_v$, to bound the second term. □

Proof of lemma 11 For each node v, initialize x_v to $\frac{\tau_v}{\Delta+1}$. This gives us a feasible packing since for each node u:

$$X_u = \sum_{v\in N_u^+} x_v = \sum_{v\in N_u^+} \frac{\tau_v}{\Delta+1} \leq \sum_{v\in N_u^+} \frac{w_u}{\Delta+1} \leq w_u$$

If $\lambda < \frac{1}{\Delta+1}$, we can satisfy the two required properties by setting S to the empty set. Assume $\lambda \geq \frac{1}{\Delta+1}$ and let $r \geq 1$ be the integer that

$$(1+\varepsilon)^{r-1}\frac{1}{\Delta+1} \leq \lambda < (1+\varepsilon)^r\frac{1}{\Delta+1}$$

We run the following procedure for r iterations.

1. For each node u, compute $X_u = \sum_{v\in N_u^+} x_v$.
2. If $X_u \geq \frac{w_u}{1+\varepsilon}$, add u to S.
3. For each undominated node $v \notin N_S^+$, set $x_v \leftarrow x_v(1+\varepsilon)$.

Observation 12 Through the algorithm, $\{x_v\}_{v\in V}$ is always a feasible packing.

Observation 13 In the end, the packing value of each undominated node, i.e. nodes in $V \setminus N_S^+$, is strictly greater than $\lambda\tau_v$ and the packing value of each dominated node is at most $\lambda\tau_v$.

Proof of observation 13 Let $v \in V \setminus N_S^+$. Its packing value is multiplied by $1+\varepsilon$ in all the r iterations. So its final value is $(1+\varepsilon)^r\frac{\tau_v}{\Delta+1} > \lambda\tau_v$. If $v \in N_S^+$, it is multiplied by $1+\varepsilon$ at most $r-1$ times. So its final value is at most $(1+\varepsilon)^{r-1}\frac{\tau_v}{\Delta+1} \leq \lambda\tau_v$. □

Back to the proof of lemma 11, to bound w_S, we use observation 10 and orient the edges of G such that the out-degree of each node is at most α. The orientation is used only for the analysis. For each node v, let N_v^{in} be the set of incoming neighbors and N_v^{out} be the set of outgoing neighbors of v in this fixed orientation.

Consider the packing values at the end of the algorithm. Note that through the algorithm, we freeze the packing value of a node as soon as it gets dominated. With this, we can write the following for each node $u \in S$:

$$\begin{aligned} X_u &= \sum_{v\in N_u^{\text{in}}} x_v + x_u + \sum_{v'\in N_u^{\text{out}}} x_{v'} \geq \frac{w_u}{1+\varepsilon} \\ &\Rightarrow \sum_{v\in N_u^{\text{in}}} x_v \geq \frac{w_u}{1+\varepsilon} - \lambda\tau_u - \sum_{v'\in N_u^{\text{out}}} \lambda\tau_{v'} \\ &\geq w_u\left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right). \end{aligned}$$

This implies:

$$\begin{aligned} w_S = \sum_{u\in S} w_u &\leq \sum_{u\in S}\left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \sum_{v\in N_u^{\text{in}}} x_v \\ &\leq \left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \sum_{v\in N_S^+} x_v \sum_{u\in S} \mathbb{I}[v \in N_u^{\text{in}}] \qquad (1) \\ &\leq \alpha\left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \sum_{v\in N_S^+} x_v. \end{aligned}$$

In the last inequality, we upper bound $\sum_{u\in S}\mathbb{I}[v \in N_u^{in}]$ with α. This is because out-degree of v is at most α, so v can be an incoming neighbor of at most α nodes.

The bound of eq. 1 along with observation 13 guarantee property (a) and property (b). The only remaining component is the round complexity. Note that each iteration of the procedure runs in $O(1)$ rounds in the CONGESTmodel. So in total, there are $O(r) = O(\log_{1+\varepsilon}\Delta\lambda) = O\left(\frac{\log(\Delta\lambda)}{\varepsilon}\right)$ many rounds. □

Remark 1 (Unknown Δ) We can transform the algorithm of theorem 1 into one that works in the setting where Δ is unknown. Recall that the algorithm has two phases. In the first phase, it computes a partial dominating set S by applying the algorithm of lemma 11 with λ being $\frac{1}{(2\alpha+1)(1+\varepsilon)}$. Then, in the second phase, for each node v that is not dominated by S, we add a node with weight τ_v from N_v^+ to the final dominating set. To convert our algorithm, first we initialize the packing value x_v of each node v with

$$\frac{\tau_v}{\max_{u\in N_v^+}|N_u^+|}$$

rather than with $\frac{\tau_v}{\Delta+1}$. Next, we run the iterations of the algorithm of lemma 11 similarly with one extra step at the beginning of each iteration. In this extra step, each undominated node v with a packing value strictly larger than $\lambda\tau_v$ adds a node with weight τ_v in its neighborhood to the final dominating set. Observe that after $O\left(\frac{\log\Delta}{\varepsilon}\right)$ rounds, all nodes are dominated and the approximation analysis goes through similarly. Intuitively, since a node cannot decide locally when the first phase terminates, we add this extra step for each iteration to simulate its effect.

Remark 2 (Unknown α) When α is unknown, we are not aware of a way to keep the algorithm's round complexity independent of n, while preserving the approximation factor. However, we can find a dominating set with approximation factor $(2\alpha+1)(2+\varepsilon)$ in $O(\frac{\log n}{\varepsilon})$ rounds (note that here we assume that all nodes know n). For that, first we apply the orientation algorithm of Barenboim and Elkin [5] to find an orientation of edges where the out-degree of each node is at most $(2+\varepsilon)\alpha$. This algorithm runs in $O(\frac{\log n}{\varepsilon})$ rounds. Next, each node v computes a local approximation of arboricity, denoted by $\hat{\alpha}_v$, for itself which is the maximum out-degree of the nodes in N_v^+. To find the partial dominating set S, each node initializes its packing value with $\frac{1}{n+1}$. We run the algorithm of lemma 11 where each node has its own λ, denoted by λ_v, and which equals $\lambda_v = \frac{1}{(2\hat{\alpha}+1)(1+\varepsilon)}$. Similar to remark 1, we add an extra step in the beginning of each iteration on which any undominated node v with a packing value more than threshold $\lambda_v \tau_v$, adds a node with weight τ_v in N_v^+ to the final dominating set. After $O\left(\frac{\log n}{\varepsilon}\right)$ iterations, all nodes are dominated and the approximation factor is $(2\alpha + 1)(2 + O(\varepsilon))$.

4.2 Randomized algorithm

In the previous section, to extend our partial dominating set S to a dominating set, we simply added one node to S for each undominated node and this introduced a factor 2 in the approximation factor. Here, we show how we can get $\alpha + O(\log \alpha)$ approximation, but we need $O(\frac{\alpha}{\log \alpha} \log \Delta)$ rounds rather than $O(\log \Delta)$ rounds. The algorithm also becomes randomized.

To reduce the approximation factor, we can leverage property (b) of lemma 11. To explain the intuition, we focus first on the unweighted case. If the problem is unweighted (so τ_v is 1 for all the nodes), then property (b) implies that each node has at most λ^{-1} undominated neighbors. The reason is that $X_u = \sum_{v \in N_u^+} x_v \leq 1$ for all u. Now since any node $v \notin N_S^+$ has $x_v \geq \lambda$, there can only be at most λ^{-1} undominated nodes in N_u^+. So, dominating the set of nodes that are not in N_S^+ is a set cover problem with maximum set size λ^{-1} which can be approximated with a factor of $O(\log \lambda^{-1})$ in $O(\log \lambda^{-1} \log \Delta)$ rounds according to [19] (Note that an undominated node can be a neighbor of many dominated nodes, so the frequency in the set cover instance can be as large as $\Delta + 1$). Recall that the size of S is bounded by $\alpha \left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \cdot \mathrm{OPT}$. On the other hand, for extending S, we add $O(\log \lambda^{-1} \cdot \mathrm{OPT})$ many nodes. If we set $\lambda = \Theta\left(\frac{\log \alpha}{\alpha^2}\right)$ and $\varepsilon = \Theta\left(\frac{\log \alpha}{\alpha}\right)$, with straightforward calculations we can show that the expected size of the final dominating set is $(\alpha + O(\log \alpha)) \cdot \mathrm{OPT}$ and the algorithm takes $O\left(\frac{\alpha}{\log \alpha} \log \Delta\right)$ many rounds. The bottleneck for the round complexity is the first phase when we construct S as there the number of rounds depends linearly on $\frac{1}{\varepsilon}$.

The weighted case is more subtle as there is no bound on the set size. Now lemma 11 implies that $x_v \geq \lambda \tau_v$, which is a different value for every v, hence we cannot argue anymore that each node has at most λ^{-1} undominated neighbors. We are not aware of a result in the literature that we can use as a black box or with a clean reduction for this case, but we still want to exploit property (b) of lemma 11 to get an improved approximation. To do so, we devise a simple iterative randomized algorithm for this case. Towards resolving this, we also improve on the results of [18, 19] for solving the dominating set problem on general graphs. There, they presented an $O(k^2)$ rounds randomized algorithm with expected $O(k \Delta^{\frac{2}{k}} \log \Delta)$ approximation factor for the dominating set problem. We shave off the factor $\log \Delta$ in their bound as it is stated in theorem 3.

Our algorithm of extending S in full generality is stated in the following lemma.

Lemma 14 *For* $0 < \lambda$ *and* $1 < \gamma$*, there is a randomized algorithm that given the output of lemma 11, it finds* $S' \subseteq V$ *such that* $S \cup S'$ *is a dominating set and* $\mathbb{E}[w_{S'}] = \gamma(\gamma+1)\lceil \log_\gamma \lambda^{-1} \rceil \cdot \mathrm{OPT}$. *The algorithm runs in* $O(\log_\gamma \lambda^{-1} \log_\gamma \Delta)$ *rounds in the* CONGEST*model.*

Before proving lemma 14, we first show how we can prove the claim of theorem 2 via optimizing the parameters in lemma 11 and lemma 14. The theorem is restated below.

Theorem 2 *For any* $0 < \varepsilon < 1$*, there is a deterministic* $(2\alpha+1)(1+\varepsilon)$*-approximation algorithm for the minimum weighted dominating set problem in graphs with arboricity at most* α. *The algorithm runs in* $O\left(\frac{\log(\Delta/\alpha)}{\varepsilon}\right)$ *rounds in the* CONGEST*model.*

Proof We first execute the algorithm of lemma 11 and then lemma 14 with a suitable set of parameters such that $w_S \leq \alpha\left(1 + \frac{1}{t}\right)$ and $w_{S'} = O\left(\frac{\alpha}{t}\right)$. Since $S \cup S'$ is a dominating set, this gives us the desired approximation factor. For the parameters, we set $\varepsilon = \frac{1}{4t}$, $\lambda = \frac{\varepsilon}{\alpha+1}$, and $\gamma = \max(2, \alpha^{\frac{1}{2t}})$. To bound w_S, note that:

$$\begin{aligned} w_S &\leq \alpha \left(\frac{1}{1+\varepsilon} - \lambda(\alpha+1)\right)^{-1} \cdot \mathrm{OPT} \\ &\leq \alpha(1-2\varepsilon)^{-1} \cdot \mathrm{OPT} \\ &\leq \alpha(1+4\varepsilon) \cdot \mathrm{OPT} \\ &\leq \left(\alpha + \frac{\alpha}{t}\right) \cdot \mathrm{OPT} \end{aligned}$$

In the second last inequality, we use the fact that $\varepsilon = \frac{1}{4t} \leq \frac{1}{4}$. To bound $w_{S'}$, first note that $\lambda^{-1} = 4t(\alpha+1) = O(\alpha^2)$ because $t \leq \frac{\alpha}{\log \alpha}$. If γ is 2, we have:

$$\begin{aligned} \mathbb{E}[w_{S'}] &= O(\gamma^2 \log_\gamma \lambda^{-1} \cdot \mathrm{OPT}) \\ &= O(\log \alpha) \cdot \mathrm{OPT} \\ &= O\left(\frac{\alpha}{t}\right) \cdot \mathrm{OPT}. \end{aligned}$$

If γ is $\alpha^{\frac{1}{2t}}$, then $t \le \frac{\log \alpha}{2}$, and we have:

$$\begin{aligned}\mathbb{E}[w_{S'}] &= O(\gamma^2 \log_\gamma \lambda^{-1}) \cdot \mathrm{OPT}\\ &= O(\alpha^{\frac{1}{t}} t) \cdot \mathrm{OPT}\\ &= O\left(\frac{\alpha}{t}\right) \cdot \mathrm{OPT}.\end{aligned}$$

To derive the last equality, we can compare the logarithm of the two bounds. Note that $\log(\alpha/t) = \log\alpha - \log t$ and since $t \le (\log\alpha)/2$, we have $\log(\alpha/t) = \Omega(\log\alpha)$. On the other hand, $\log(\alpha^{1/t}t) = (\log\alpha)/t + \log t$ and using $t \le (\log\alpha)/2$, we get $\log(\alpha^{1/t}t) = O(\log\alpha)$.

The algorithm of lemma 11 runs in $O(\frac{\log\Delta}{\varepsilon}) = O(t\log\Delta)$ rounds and the algorithm of lemma 14 runs in

$$\begin{aligned}O(\log_\gamma \lambda^{-1} \log_\gamma \Delta) &= O(\log_{\alpha^{\frac{1}{2t}}} \alpha^2 \log_\gamma \Delta)\\ &= O(t \log \Delta).\end{aligned}$$

rounds. □

Proof of lemma 14 We construct S' in several steps. For a moment, assume that the problem is unweighted (all nodes have weight one). From what we have discussed before, each node can dominate at most $q = O(\alpha)$ nodes in $V \setminus N_S^+$. In the first step of constructing S', we try to reduce q to $\frac{q}{2}$. We call a node heavy if it has at least $\frac{q}{2}$ undominated neighbors. To get rid of heavy nodes, first, we sample each of them with probability $\frac{1}{\Delta+1}$. We add all the sampled nodes to S' and update the set of undominated nodes to $V \setminus N^+_{S\cup S'}$. The set of heavy nodes is updated accordingly. Then, we sample each heavy node with probability $\frac{2}{\Delta+1}$. We repeat this for $O(\log\Delta)$ iterations until the sampling probability becomes 1. This ensures that in the end, there is no heavy node. To show that there are not too many sampled nodes, let n' be the number of undominated nodes before the first iteration and observe that we need at least $\frac{n'}{q}$ nodes to dominate them. It can be shown that after all the iterations, the expected number of sampled nodes is $O(\frac{n'}{q})$. So in $O(\log\Delta)$ rounds and with an additive loss of $O(1)$ in the approximation factor, we can reduce q to $\frac{q}{2}$. Repeating this for $O(\log q) = O(\log\alpha)$ times, we get a set S' in $O(\log\alpha\log\Delta)$ rounds with expected size $O(\log\alpha \cdot \mathrm{OPT})$ such that $S \cup S'$ is a dominating set. A detailed discussion on the parameterized version of this algorithm that works also for the weighted case is given in the following.

We start S' as an empty set. Unlike the previous parts, for each node u, we set $X_u = \sum_{v \in N_u^+ \cap (V \setminus N^+_{S\cup S'})} x_v$ to be the summation of packing values of only undominated nodes in N_u^+. Let $\Gamma_1 = \{u \notin S \cup S' : X_u \ge \gamma^{-1} w_u\}$. We sample nodes of Γ_1 and update it for $r = \lceil \log_\gamma(\Delta+1)\rceil + 1$ iterations. Before the first iteration, we initialize the sampling probability p with $\frac{1}{\Delta+1}$. Then, we run the following in each iteration:

1. Sample each node in the current Γ_1 with probability p.
2. Add the sampled nodes to S'.
3. Update X_u for each node u. That is, compute

$$X_u = \sum_{v \in N_u^+ \cap (V \setminus N^+_{S\cup S'})} x_v$$

.

4. Remove all the nodes from Γ_1 with $X_u < \gamma^{-1} w_u$.
5. Set $p \leftarrow \min(\gamma p, 1)$.

For each element v, we define a random variable c_v. If v is already dominated by S or if it is not dominated by S' after all the iterations, we set c_v to zero. Otherwise, let i be the first iteration where v is dominated. We set c_v to be the number of sampled nodes in iteration i that dominates v.

Observation 15 For each node v, we have $\mathbb{E}[c_v] \le \gamma + 1$.

Proof If v is already dominated by S, then c_v is always zero. So suppose this is not the case and assume that it is in d sets of Γ_1 before the first iteration. Let $d = d_1 \ge d_2 \ge \cdots \ge d_r$ be the sequence that maximize the expected value of $\mathbb{E}[c_v]$ where d_i is the number of sets in Γ_1 that contains v in the beginning of iteration i. Let $p_i = \frac{\gamma^{i-1}}{\Delta+1}$ be the sampling probablity at iteration i. We have:

$$\mathbb{E}[c_v] \le \sum_{i=1}^{r} p_i d_i \prod_{j=1}^{i-1} (1-p_j)^{d_j} \le \sum_{i=1}^{r} p_i d_i \prod_{j=1}^{i-1} e^{-p_j d_j}$$

To simplify the notation, we define $\beta_i = p_i d_i$. Denote the prefix sum of sequence β_i as $\bar{\beta}_i = \sum_{j=1}^{i-1} \beta_i$. Let us emphasize that $\bar{\beta}_i$ does not include β_i. Rewriting the above inequality using β and $\bar{\beta}$, we have:

$$\mathbb{E}[c_v] \le \sum_{i=1}^{r} \beta_i \prod_{j=1}^{i-1} e^{-\beta_j} = \sum_{i=1}^{r} \beta_i e^{-\bar{\beta}_i}$$

Since the sequence $d_1, \ldots, d_r$ is non-increasing and $p_i = \gamma p_{i-1}$, we have $\beta_i \le \gamma\beta_{i-1}$ for any $2 \le i$. Using this, we can deduce the following:

$$\begin{aligned}\mathbb{E}[c_v] \le \sum_{i=1}^{r} \beta_i e^{-\bar{\beta}_i} &= \beta_1 + \sum_{i=2}^{r} \beta_i e^{-\bar{\beta}_i}\\ &\le \frac{d_1}{\Delta+1} + \sum_{i=2}^{r} \gamma\beta_{i-1} e^{-\bar{\beta}_i}\\ &\le 1 + \gamma \sum_{i=2}^{r} \int_{\bar{\beta}_{i-1}}^{\bar{\beta}_i} e^{-x} dx\end{aligned}$$

In the last inequality, note that $\bar{\beta}_i - \bar{\beta}_{i-1} = \beta_{i-1}$ by definition. Since all the integral ranges are disjoint, we have:

$$\mathbb{E}[c_e] \leq 1 + \gamma \int_0^\infty e^{-x} dx = \gamma + 1$$

□

Observation 16 The expected total weight of sampled nodes in all iterations is at most $\gamma(\gamma + 1) \cdot \mathrm{OPT}$.

Proof Suppose a node u that is sampled in iteration i. Let T_u be the set of nodes in N_u^+ that are undominated in the beginning of iteration i. At that point, u is in Γ_1 implying $\gamma^{-1} w_u \leq \sum_{v \in T_u} x_v$. So we can upper bound the weight of any sampled node u by $w_u \leq \gamma \sum_{v \in T_u} x_v$. From the definition of c_v, each node v appears in c_v many T_us for a sampled node T_u. So the total weight of sampled nodes is upper bounded by $\gamma \sum_{v \in V} c_v x_v$. Finally, we know that $\mathbb{E}[c_v] \leq \gamma + 1$ from lemma 15 which concludes the proof. □

Back to the proof of lemma 14, we now try to use these two observations to show the claims of this lemma. First, note that the set $S \cup S'$ is not necessarily a dominating set. Consider the subproblem of dominating $V \backslash N^+_{S \cup S'}$. Clearly, we can dominate these nodes with a set of weight at most OPT. Moreover, $\{x_v\}_{v \notin N^+_{S \cup S'}}$ is a feasible packing for this subproblem. Multiply the packing value of each node $v \notin S \cup S'$ by a factor γ and update X_u for each $u \notin S \cup S'$. The packing for this subproblem remains feasible. This is because, in the last iteration, we sample all nodes of Γ_1. So at the end of this iteration, a node u that is not in $S \cup S'$ has $X_u \leq \gamma^{-1} w_u$ and as a result, multiplying the packing values by a factor γ is safe.

Now, we define Γ_2 as $\{u \notin S \cup S' : X_u \geq \gamma^{-1} w_u\}$ and run the procedure with Γ_2. We repeat this for $t = \lceil \log_\gamma \lambda^{-1} \rceil$ times and claim that after that, $S \cup S'$ is a dominating set.

Suppose it is not and let v be an undominated node. At the very beginning, $x_v \geq \lambda \tau_v$ due to property (b) of lemma 11. Since v is undominated, its packing value is $\gamma^i x_v$ after we finish the process of Γ_i. Since $t = \lceil \log_\gamma \lambda^{-1} \rceil$, there should be an i such that the packing value of v is at least $\gamma^{-1} \tau_v$ when we finish working with Γ_{i-1}. On the other hand and from the definition of τ_v, there is a neighbor u of v with weight τ_v. This means that u is in Γ_i and it remains in Γ_i until it is sampled. This contradicts that v is not dominated.

From lemma 16, the expected weight of the sampled nodes in one phase is $\gamma(\gamma+1) \cdot \mathrm{OPT}$. So in total, the expected weight of S' is $\gamma(\gamma + 1) \lceil \log_\gamma \lambda^{-1} \rceil \cdot \mathrm{OPT}$.

Each iteration of each phase can be run in $O(1)$ rounds in the CONGESTmodel. So in total, the algorithm needs

$$O(t \cdot r) = O(\log_\gamma \lambda^{-1} \log_\gamma \Delta)$$

many rounds. □

Theorem 3 *For any k, there is randomized algorithm that computes a weighted dominating set with expected approximation factor at most $\Delta^{\frac{1}{k}}(\Delta^{\frac{1}{k}} + 1)(k + 1) \cdot = O(k\Delta^{\frac{2}{k}})$ in $O(k^2)$ rounds in the* CONGEST*model.*

Proof In lemma 14, we assume S is empty and set λ to $\frac{1}{\Delta+1}$. This does not violate any condition in the lemma. By setting γ to $\Delta^{\frac{1}{k}}$, the output S' of the lemma is a dominating set with the claimed size. □

5 Lower Bound

Theorem 4 *ThmLB Any distributed algorithm that computes any constant or poly-logarithmic approximation of the minimum dominating set on graphs of arboricity 2 requires $\Omega\left(\frac{\log \Delta}{\log \log \Delta}\right)$ rounds in the* LOCAL*model.*

Proof Kuhn, Moscibroda and Wattenhofer proved that obtaining a constant or poly-logarithmic approximation for the Minimum Vertex Cover (MVC) problem requires $\Omega\left(\frac{\log \Delta}{\log \log \Delta}\right)$ rounds in the LOCALmodel [20]. In fact, their lower bound holds even for the *fractional* version of the problem, in which the goal is to assign a value x_v for each node such that for each edge $\{u, v\} \in E$, we have that $x_v + x_u \geq 1$, and $\sum_v x_v$ is minimized. Let G be the Kuhn-Moscibroda-Wattenhofer (KMW) lower bound graph for approximating the Minimum Fractional Vertex Cover (MFVC), with maximum degree Δ, n nodes, and m edges. We build a new graph H as follows: Take Δ^2 copies of G, let us call them $G_1, G_2, \ldots, G_{\Delta^2}$. Add a set T of n additional nodes, one for each node of G, and connect each new node to all copies of that original G-node. So the degree of each node in T is Δ^2. Next, for each G_i, add a node in the middle of each of its edges. This completes the construction of H. See fig 1 for an illustration. Some observations on H is in the following:

- Each G_i has $n + m$ nodes (one copy for each node of G and m middle nodes) and $2m$ edges. Taking T into consideration, H has $\Delta^2(n+m)+n$ nodes and $\Delta^2(2m+n)$ edges.
- The maximum degree of H is Δ^2.
- The arboricity of H is 2. For each middle node, orient its two incident edges outward. For each node in T, orient its Δ^2 incident edges inward. This gives us an orientation of all the edges of H. There is no directed cycle in this orientation and its maximum out-degree is 2, so the arboricity of H is 2.
- Let $\mathrm{OPT}^H_{\mathrm{MDS}}$ be the size of the minimum dominating set of H and $\mathrm{OPT}^G_{\mathrm{MVC}}$ ($\mathrm{OPT}^G_{\mathrm{MFVC}}$) be the size of the

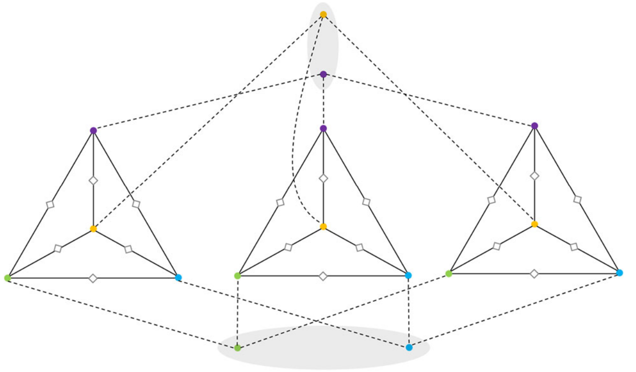

Fig. 1 The lower bound graph H assumes that the KMW graph G is K_4. Only three copies (rather than nine copies) of G are drawn. The middle nodes are indicated by hollow diamonds and T is the set of four nodes in the gray area

minimum (fractional) vertex cover of G, then:

$$\begin{aligned}\mathrm{OPT}^H_{\mathrm{MDS}} &\leq \Delta^2 \cdot \mathrm{OPT}^G_{\mathrm{MVC}} + n \\ &= \Delta^2 \cdot \mathrm{OPT}^G_{\mathrm{MFVC}} + n.\end{aligned} \tag{2}$$

The first inequality is because T along with copies of a minimum vertex cover of G in each G_i is a dominating set of H. This is because every node in G_i that is not a middle-node has a neighbor in T. On the other hand, all the middle-nodes are dominated as we add a vertex cover for each G_i. For the equality $\mathrm{OPT}^G_{\mathrm{MVC}} = \mathrm{OPT}^G_{\mathrm{MFVC}}$ in (2), we leverage the fact that the KMW graph G is bipartite and as a result the integrality gap of vertex cover on G is 1.[3] In addition, $\mathrm{OPT}^G_{\mathrm{MFVC}} \geq \frac{m}{\Delta}$. This holds if $\{x_v\}_v$ is the optimal fractional solution. So we have that

$$\begin{aligned} m &\leq \sum_{\{u,v\}\in E} (x_u + x_v) \leq \Delta \sum_v x_v \\ &= \Delta \cdot \mathrm{OPT}^G_{\mathrm{MFVC}} \end{aligned}$$

On the other hand, since $\mathrm{OPT}^G_{\mathrm{MFVC}} \geq \frac{m}{\Delta}$ and also because for the KMW graph we have $m \geq n$, we can write:

$$\mathrm{OPT}^H_{\mathrm{MDS}} \leq (\Delta^2 + \Delta) \cdot \mathrm{OPT}^G_{\mathrm{MFVC}}.$$

To prove the theorem, suppose there is an algorithm $\mathcal{A}$ with round complexity $o\left(\frac{\log \Delta}{\log\log \Delta}\right)$ that computes a c-approximation of Minimum Dominating Set (MDS) on H for some constant or poly-logarithmic c. Using this algorithm, we will show that we get an algorithm that computes a $c(1 + \frac{1}{\Delta})$ approximation of MFVC of G in the same number of rounds, hence putting us in contradiction with the lower bound of [20], thus completing the proof.

Notice that the graph G can simulate the graph H in the LOCALmodel, where each node simulates all its copies in H, and each node in H that corresponds to an edge in G is simulated by one of its endpoints. Note that for each edge in H, its two endpoints are simulated either by the same node in G or by two neighboring nodes in G, hence we can simulate a round of an algorithm in H in one round in G. Let us run the algorithm on the graph H, by actually working on the graph G. Let S be the computed dominated set of H. We want to turn S into a fractional vertex cover of G. First, replace each middle node in S with one of its endpoints. This can only decrease the size of S and after it, all the middle nodes are still dominated. Let S_i be the set of nodes of G_i in S. Since S dominates all the middle nodes, S_i is a vertex cover of the original graph G. As $\mathcal{A}$ is a c-approximation algorithm for MDS, we have:

$$\sum_{i\in[\Delta^2]} |S_i| \leq |S| \leq c(\Delta^2 + \Delta) \cdot \mathrm{OPT}^G_{\mathrm{MFVC}}$$

At the end, each node v of G, computes $y_v = \frac{|\{i: v\in S_i\}|}{\Delta^2}$. Observe that $\sum_v y_v \leq c(1 + \frac{1}{\Delta}) \cdot \mathrm{OPT}^G_{\mathrm{MFVC}}$, and we show that $\{y_v\}_v$ is a fractional vertex cover of G. This follows from the fact that each S_i is a vertex cover of G. Hence, for each edge $\{u, v\} \in G$, if we denote by $\{u_i, v_i\}$ the corresponding edge in G_i, we have that at least one of u_i, v_i is in the vertex cover. Let y^i_v indicate if $v_i \in S_i$. We have

$$y_u + y_v = \frac{1}{\Delta^2} \sum_{1\leq i\leq \Delta^2} (y^i_u + y^i_v) \geq 1,$$

as needed. So if $\mathcal{A}$ exists, then we can compute a $c(1 + \frac{1}{\Delta})$ approximation of minimum fractional vertex cover of G in $o\left(\frac{\log \Delta}{\log\log \Delta}\right)$ rounds contradicting the lower bound of [20]. □

Appendix A missing proofs

Observation 17 There is a single-round algorithm that computes a 3-approximation of minimum unweighted dominating set on graphs of arboricity 1.

Proof Take all non-leaf nodes as the dominating set. This is a dominating set, we next show the approximation ratio. Let S^* be an optimal dominating set. A graph of arboricity 1 is a forest, we can fix a root in each one of the trees. We create a dominating set S' as follows. For each node $v \in S^*$, we add to S' the node v along with its parent and grandparent in the tree (if they exist). Clearly $|S'| \leq 3|S^*|$. We show that S' contains all non-leaf nodes, completing our proof. Assume to the contrary that there is a node $v \notin S'$ that is an internal

[3] Alternatively, we can use the fact that $\mathrm{OPT}^G_{\mathrm{MVC}} \leq 2 \cdot \mathrm{OPT}^G_{\mathrm{MFVC}}$ in any graph, which would only change the constant in our analysis.

node in the tree, and let u be a child of v (that exists, as v is non-leaf). It follows that $u \notin S^*$, as otherwise $v \in S'$. Since v and u are not in S^*, and S^* is a dominating set, there is a child w of u such that $w \in S^*$, but then $v \in S'$, a contradiction. □

Acknowledgements This work was supported in part by funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 853109), and the Swiss National Foundation (project grant 200021-184735).
Michal Dory and Mohsen Ghaffari were affiliated to ETH Zurich when the project is done.
Michal Dory has several joint with Merav Parter (a member of the journal editorial) in the last three years.
This work is purely theoretical and no data is used to achieve the provided results.

Author Contributions All authors have contributed equally

Funding Open access funding provided by Swiss Federal Institute of Technology Zurich.

Declaration

Conflict of interest The authors declare no competing interests.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.

References

1. Akhoondian Amiri S., Ossona de Mendez P., Rabinovich R., et al.: Distributed domination on graph classes of bounded expansion. In: Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures (SPAA), pp 143–151 (2018)
2. Amiri, S.A.: Deterministic CONGEST algorithm for MDS on bounded arboricity graphs. CoRR abs/2102.08076 (2021)
3. Amiri, S.A., Schmid, S., Siebertz, S.: Distributed dominating set approximations beyond planar graphs. ACM Transact Algorithm (TALG) **15**(3), 1–18 (2019)
4. Bansal, N., Umboh, S.W.: Tight approximation bounds for dominating set on graphs of bounded arboricity. Inform. Process. Lett. **122**, 21–24 (2017)
5. Barenboim, L., Elkin, M.: Sublogarithmic distributed mis algorithm for sparse graphs using nash-williams decomposition. Distrib. Comput. **22**(5), 363–379 (2010)
6. Bonamy, M., Cook, L., Groenland, C., et al.: A tight local algorithm for the minimum dominating set problem in outerplanar graphs. In: 35th International Symposium on Distributed Computing, DISC 2021, LIPIcs, vol 209. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp 13:1–13:18 (2021)
7. Cambus, M., Choo, D., Miikonen, H., et al.: Massively parallel correlation clustering in bounded arboricity graphs. In: 35th International Symposium on Distributed Computing, DISC 2021, LIPIcs, vol 209. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp 15:1–15:18 (2021)
8. Czygrinow, A., Hańćkowiak, M., Wawrzyniak, W.: Fast distributed approximations in planar graphs. In: International Symposium on Distributed Computing, Springer, pp 78–92 (2008)
9. Deurer, J., Kuhn, F., Maus, Y.: Deterministic distributed dominating set approximation in the congest model. In: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC), pp 94–103 (2019)
10. Dinur, I., Steurer, D.: Analytical approach to parallel repetition. In: Proceedings of the forty-sixth annual ACM symposium on Theory of computing, pp 624–633 (2014)
11. Dvořák, Z.: On distance-dominating and-independent sets in sparse graphs. J. Graph Theory **91**(2), 162–173 (2019)
12. Eden, T., Levi, R., Ron, D.: Testing bounded arboricity. ACM Transact. Algorithms (TALG) **16**(2), 1–22 (2020)
13. Ghaffari, M., Kuhn, F., Maus, Y.: On the complexity of local distributed graph problems. In: Proc. 49th ACM Symp. on Theory of Computing (STOC), pp 784–797 (2017)
14. Gianinazzi, L., Besta, M., Schaffner, Y., et al.: Parallel algorithms for finding large cliques in sparse graphs. In: Proceedings of the 33rd ACM Symposium on Parallelism in Algorithms and Architectures (SPAA), pp 243–253 (2021)
15. Goel, G., Gustedt, J.: Bounded arboricity to determine the local structure of sparse graphs. In: International Workshop on Graph-Theoretic Concepts in Computer Science, Springer, pp 159–167 (2006)
16. Jia, L., Rajaraman, R., Suel, T.: An efficient distributed algorithm for constructing small dominating sets. Distrib. Comput. **15**(4), 193–205 (2002)
17. Johnson, D.S.: Approximation algorithms for combinatorial problems. J. Comput. Syst. Sci. **9**(3), 256–278 (1974)
18. Kuhn, F., Wattenhofer, R.: Constant-time distributed dominating set approximation. Distrib. Comput. **17**(4), 303–310 (2005)
19. Kuhn, F., Moscibroda, T., Wattenhofer, R.: The price of being near-sighted. In: Proceeding 17th Symposium on Discrete Algorithms (SODA), pp 980–989 (2006)
20. Kuhn, F., Moscibroda, T., Wattenhofer, R.: Local computation: Lower and upper bounds. JACM 63(2) (2016)
21. Lenzen, C., Wattenhofer, R.: Minimum dominating set approximation in graphs of bounded arboricity. In: International symposium on distributed computing (DISC), Springer, pp 510–524 (2010)
22. Lenzen, C., Oswald, YA., Wattenhofer, R.: What can be approximated locally? case study: Dominating sets in planar graphs. In: Proceedings of the twentieth annual symposium on Parallelism in algorithms and architectures (SPAA), pp 46–54 (2008)
23. Lenzen, C., Pignolet, Y.A., Wattenhofer, R.: Distributed minimum dominating set approximations in restricted families of graphs. Distrib. Comput. **26**(2), 119–137 (2013)
24. Morgan, A., Solomon, S., Wein, N.: Algorithms for the minimum dominating set problem in bounded arboricity graphs: Simpler, faster, and combinatorial. In: 35th International Symposium on Distributed Computing, DISC 2021, LIPIcs, vol 209. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp 33:1–33:19 (2021)

25. Rozhoň, V., Ghaffari, M.: Polylogarithmic-time deterministic network decomposition and distributed derandomization. In: Proc. 52nd ACM Symp. on Theory of Computing (STOC), pp 350–363 (2020)
26. Sun, H.: An improved approximation bound for minimum weight dominating set on graphs of bounded arboricity. In: International Workshop on Approximation and Online Algorithms, Springer, pp 39–47 (2021)
27. Wawrzyniak, W.: A strengthened analysis of a local algorithm for the minimum dominating set problem in planar graphs. Inform. Process. Lett. **114**(3), 94–98 (2014)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.