Theoretical Computer Science 906 (2022) 18–31

Contents lists available at ScienceDirect

Theoretical Computer Science

www.elsevier.com/locate/tcs

Distributed distance-*r* covering problems on sparse high-girth graphs

Saeed Akhoondian Amiri [a], Ben Wiederhake [b,c,*]

[a] University of Cologne, Köln, Nordrhein-Westfalen, 50931, Germany
[b] Max-Planck Institute for Informatics, Saarbrücken, 66123, Saarland, Germany
[c] Saarbrücken Graduate School, Saarland University, Saarbrücken, 66123, Saarland, Germany

ARTICLE INFO

Article history:
Received 29 June 2021
Received in revised form 21 December 2021
Accepted 4 January 2022
Available online 10 January 2022

Keywords:
Sparse graphs
Dominating set
Vertex cover
Upper bound
Lower bound

ABSTRACT

We prove that the distance-*r* dominating set, distance-*r* connected dominating set, distance-*r* vertex cover, and distance-*r* connected vertex cover problems admit constant factor approximations in the CONGEST model of distributed computing in a *constant* number of rounds on classes of sparse high-girth graphs. In this paper, sparse means bounded expansion, and high-girth means girth at least $4r + 2$. Our algorithm is quite simple; however, the proof of its approximation guarantee is non-trivial. To complement the algorithmic results, we show tightness of our approximation by providing a loosely matching lower bound on rings.

Our result is the first to show the existence of constant-factor approximations in a constant number of rounds in non-trivial classes of graphs for distance-*r* covering problems.

1. Introduction

In the sequential setting, many APX-hard covering problems can be well approximated if they are limited to the class of sparse graphs. Hence, it is interesting to understand how sparsity enables better distributed algorithms in distributed computing models, which could mean improving the approximation factor or reducing the number of communication rounds. In the distributed setting every node is considered as a processor that can communicate with its neighbors per synchronized rounds. The aim is to reduce the total number of such rounds while providing a *good* solution.

In this work, we continue the line of study on sparse graphs and explore the algorithmic properties of a wide range of sparse graphs, namely the class of graphs of bounded expansion, with an extra combinatorial property: sparse graphs of high girth. The girth of the graph is the length of its shortest cycle and for instance girth of a tree is infinity.

Girth plays a role in understanding structural properties of graphs. Sparse graphs of high girth appear in important constructions such as spanner graphs [1]. Similarly random graphs have only a few short cycles and at the same time, depending on their parameters, they could be quite sparse. In such a graph class (we will formally specify them later) we study several central covering problems in their most generic form: distance-*r* covering problems.

As a result, we answer some more cases of the famous question of Naor and Stockmeyer: "What can be computed locally?" [24] More precisely, we show that the problems 1. Distance-*r* Dominating Set 2. Distance-*r* Connected Dominating Set 3. Distance-*r* Vertex Cover and 4. Distance-*r* Connected Vertex Cover; on the considered graph class have constant

* Corresponding author.
E-mail addresses: amiri@cs.uni-koeln.de (S.A. Amiri), bwiederh@mpi-inf.mpg.de (B. Wiederhake).

https://doi.org/10.1016/j.tcs.2022.01.001

factor approximation in a constant number of rounds in the CONGEST model of computation, if r (the distance) is constant. Whenever feasible, we also give the precise relation to r.

The aforementioned problems are hard in general graphs when it comes to distributed settings. For instance there is no constant-factor approximation CONGEST algorithm for the distance-2 dominating set even if we let the algorithm to run for $o(n^2)$ rounds [11], where n is the number of nodes. Observe that in order to exchange information, two nodes require at most $O(n)$ rounds of communication, however, for such a restricted case of the problem (only distance-2) the amount of data to be transferred is too big to fit in one message that respects the bandwidth of the network. Hence it needs $\Omega(n^2)$ rounds of computation to merely approximate the optimum solution. Thus, a natural approach to progress on such problems is to consider them on restricted graph families.

Throughout the paper, we assume that a graph $G = (V, E)$ is given. In the (distance 1) dominating set problem, we seek a set $D \subseteq V$ such that every other vertex of G is a neighbor of a vertex in D. In the connected version of the problem, the induced graph of G on the vertices of $\hat{D}$ should be connected.

In the vertex cover problem, we seek a set C of vertices of the graph such that every edge in the graph is incident to at least one of the vertices in C. Similarly, the connected version of the problem asks for a vertex cover $\hat{C}$ such that the induced graph of C on G is connected. In all of the above problems we would like to minimize the size of the corresponding set.

The distance-r versions of covering problems are defined similarly to the classic versions: for the dominating set problem, we consider the distance-r neighborhood. In the distance-r vertex cover problem, we say that vertex v covers edge e if and only if vertex v is within distance r of both endpoints. Observe that for $r = 1$, these distance-r versions are equivalent to the classic versions.

We consider problems in the LOCAL and CONGEST distributed models of computation. Intuitively speaking, in both of these models, each vertex in the graph is a processor, has a unique identifier, and communicates only with its neighbors once per round. The CONGEST model restricts the bandwidth of communication links to a reasonable complexity. The aim is to solve the problem with the least number of communication rounds. A more rigorous definition follows in Section 2. We specifically look into the problem of finding a small distance-r dominating set, distance-r connected dominating set, distance-r vertex cover, or distance-r connected vertex cover where each vertex has to output its membership in the set.

Our main algorithmic contribution is that the distance-r dominating set problem admits a constant factor approximation in a constant number of rounds in sparse high-girth graphs (for constant r). We also extend this algorithm to the described related covering problems.

1.1. Related work

In distributed settings, for the dominating set problem on general graphs, Kuhn et al. [16] provided a $(1+\epsilon)(1+\log(\Delta+1))$-approximation in $f(n)$ rounds. In this bound, Δ is the maximum degree and $f : \mathbb{N} \to \mathbb{N}$ is the number of rounds that is needed to compute the network decomposition [7,8,18]. Given the recent breakthrough result of Rozhon and Ghaffari [28], the aforementioned algorithm runs in a polylogarithmic number of rounds.

For the vertex cover problem, Bar-Yehuda et al. [12] provided a constant factor approximation in a sublogarithmic number of rounds. This is complemented by the lower bound of Kuhn, Moscibroda and Wattenhofer (KMW) [20] shows that a logarithmic approximation in almost sublogarithmic time for the vertex cover problem and the minimum dominating set (and some other covering problems) is impossible in general graphs, even in the LOCAL model of computation. Their lower-bound graph for vertex cover has high girth, but it is also of unbounded arboricity (more generally unbounded average degree).

We investigate what happens in between graph classes. If we consider a graph class of very high girth and very low edge density, e.g. trees, then these problems are easy to approximate in zero rounds: take all non-leaf vertices. The above observations raise the questions: In which graph classes does the problem admit a constant approximation factor in a constant number of rounds? What about distance-r problems?

For the dominating set problem, Lenzen et al. [14,21] provided the first constant-factor approximation in a constant number of rounds in planar graphs, which was improved by Czygrinow et al. [14]. Later, Amiri et al. [4,5] provided a new analysis method to extend the result of Lenzen et al. to bounded genus graphs. This has been gradually generalized to excluded minor graphs [15] and bounded expansion [19].

A natural generalization of excluded minor graphs is the class of bounded expansion graphs. Simply put, bounded expansion graphs only exclude minors on a localized level; there may still be large clique minors in the graph.

On graphs of bounded expansion, only a logarithmic time constant factor approximation is known for the dominating set problem; however, it seems that one can extend the algorithm of [15] to bounded expansion graphs, as they only consider "local" minors. If we go slightly beyond these graphs, to graphs of bounded arboricity (where every subgraph has a constant edge density), the situation is worse: only an $O(\log \Delta)$-approximation in $O(\log n)$ rounds was known. There was a $O(\log n)$ round $O(1)$-approximation in such graphs; however, this algorithm is randomized [22], only recently it has been proven that it can be performed deterministically [2].

All these results are about the distance-1 dominating set problem. Significantly less work exists on the topic of the distance-r dominating set problem. We are only aware of the algorithm of Amiri et al. [3] for bounded expansion graphs that provides a constant factor approximation in a logarithmic number of rounds.

We are not aware of any paper tackling the distance-r vertex cover problem, except the general techniques known for bounded degree graphs and the generic algorithmic techniques that one can apply to bounded arboricity graphs.

1.2. Existing approaches for (distance-r) dominating set

There are several existing approaches one might employ to tackle the problem: 1) Take the r-th power of the graph and go back to the distance-1 dominating set, 2) Decentralize existing decomposition methods in the sequential setting and employ them, 3) Use existing fast distributed graph decomposition methods for sparse graphs. In the following, we explain how all of the above approaches, without introducing new ideas, are impractical in providing sublogarithmic round algorithms for distance-r covering problems.

For the first approach, we lose the sparsity of the graph already on stars. Hence, we cannot rely on existing algorithms for solving the domination problem in sparse graphs.

Decentralizing the existing sequential decomposition methods does not seem promising if one hopes to achieve anything better than logarithmic rounds: To the best of our knowledge, every such sequential decomposition already consumes polylogarithmically many rounds. Even assuming the decomposition is given in advance, such methods handle the clusters sequentially; however the number of clusters is usually at least logarithmic, requiring at least logarithmically many rounds.

For the third approach, these existing fast distributed graph decomposition methods are mostly inspired by existing methods in classical settings, like Baker's method [10]; this includes, e.g., the $O(\log^* n)$ round algorithm of [13]. The idea is to partition a sparse graph into connected clusters such that each cluster has a small diameter and the number of in-between cluster edges is small. Then, find the optimal solution inside each cluster efficiently, and because the number of edges between a pair of clusters is small, we can just ignore conflicts.

However, this fails already for distance-2 domination, since the number of edges between clusters in the power graph is high. As mentioned earlier, recent research has shown that there is a lowerbound of $\tilde{\Omega}(n^2)$ for distance-2 dominating set, both for solving it exactly [9], and even for constant-factor approximations [11].

Also, we cannot rely on global properties (such as tree decomposition) like in the sequential setting, since this increases the number of rounds to the diameter of the graph, which can easily be superlogarithmic.

Therefore, any distributed algorithm that solves distance-r covering problems has to exploit special properties of the underlying graph class or problem, motivating our choice of sparse high-girth graphs.

1.3. Our approach and results

We consider a generalization of the dominating set and vertex cover problem, and fill a gap between the lower bounds and upper bounds by analyzing the complexity of the problems on graphs of high girth, similar to the lower bound graph by Kuhn et al. Given that the lower bound graph in that work is relatively dense, we restrict the graph class further to sparse graphs, in particular, to bounded expansion graphs (similar to the work of [3]).

Let us present the algorithm for the distance-r dominating set problem: Each vertex chooses its dominator to be the neighbor that has a maximum degree in the r-th power graph. To implement such a simple algorithm in the CONGEST model without actually constructing the r-th power graph (which is basically impossible in our desired running time even for $r = 2$), we exploit the fact that the neighborhood of a vertex looks like a tree. The output of the algorithm is the set D of dominators, which solves the problem.

To prove the correctness of the algorithm, we partition the vertices of the graph into Voronoi cells where the center of each cell is one of the vertices of the optimum distance-r dominating set. Then we construct a candidate set, based on the Voronoi boundaries, and show that the algorithm effectively only ever selects vertices from this candidate set, which in turn is bounded in terms of the optimal distance-r dominating set.

This is used to prove Theorem 1. We generalize the algorithm to handle the Distance-r Connected Dominating Set (proven by Proposition 29), Distance-r Vertex Cover (Proposition 31), and Distance-r Connected Vertex Cover (Corollary 33) problem.

Theorem 1. *Let $\mathcal{C}$ be a graph class of bounded expansion $f(r)$ and girth at least $4r + 3$. There is a CONGEST algorithm that runs in $O(r)$ rounds and provides an $O(r \cdot f(r))$-approximation of minimum distance-r dominating set on $\mathcal{C}$, even if $r \geq 2$ is a non-constant function of n.*

Given that the distance-r dominating set problem is equivalent to the dominating set problem of the r-th power of the input graph, the algorithm can also be used to provide a constant factor approximation in a non-trivial class of dense graphs for covering problems. There are very few known algorithms with a constant factor guarantee in a constant number of rounds on non-trivial dense graphs, e.g., the algorithm of Schneider et al. [29] on graphs of bounded independence number (for the independent set and the connected dominating set problem) partially falls into this category.

To show that our upper bound is reasonably tight, we provide a lower bound as well. This we obtain by a reduction from a lower bound for independent set on the ring [14,21] to the distance-r dominating set on the ring (naturally, a ring with high girth). More formally we prove Theorem 2.

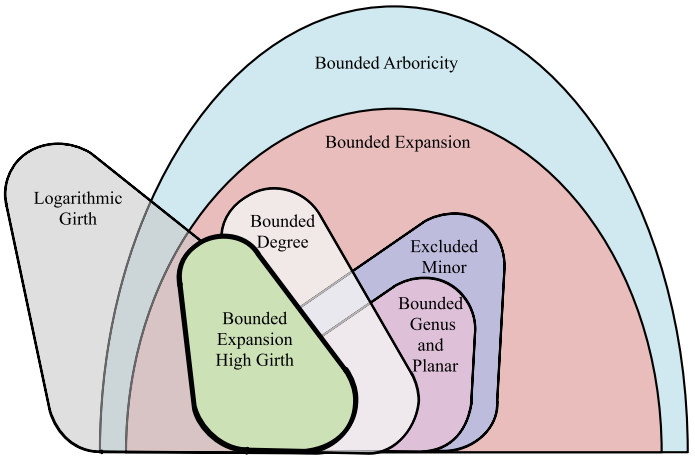

Fig. 1. Diagram of the relation of sparse graph classes. The graph class in the lower bound construction of Kuhn et al. [20] is a subclass of logarithmic girth class of unbounded arboricity. The bounded expansion class is a subclass of bounded arboricity class. Bounded expansion is also a superclass of many common sparse graph classes: planar, bounded genus, excluded minors, and bounded degree. The class of bounded expansion with high girth intersects each of the other four classes, but neither contains nor is fully contained in any of them.

Theorem 2. *Assume an arbitrary but fixed $\delta > 0$ and $r > 1$, with $r \in o(\log^* n)$. Then, there is no deterministic LOCAL algorithm that finds in $O(r)$ rounds a $(2r + 1 - \delta)$-approximation of distance-r dominating set for all $G \in \mathcal{C}$, where $\mathcal{C}$ is the class of cycles of length $\gg 4r + 3$.*

We will formally introduce the notion of bounded expansion in Section 2. The relation between the sparse graph classes is shown in Fig. 1.

Some of these results previously appear in CIAC 2021 [6]. This paper contains as new contributions a simplified analysis that also improves the result by factor 2.

2. Preliminaries and notation

We assume basic familiarity with graph theory. In the following, we introduce basic graph notation to avoid ambiguities. We refer to the book by Diestel [17] for further reading.

Graph, Neighborhood, Distance-r: We only consider simple, connected, undirected graphs $G = (V, E)$. For $u, v \in V$, define $d(u, v)$ as the distance (in number of edges) between the two vertices. For a set $S \subseteq V$, we define $d(u, S)$ as the maximum distance between vertex u and any vertex in S. Two vertices $u, v \in V$ are neighbors in G if there is an edge $e = \{u, v\} \in E$. We extend this definition to the distance-r neighborhood $N^r[v]$ and open distance-r neighborhood $N^r(v)$ of a vertex v in the following way:

$$N^r[v] := \{u \in V \mid d(u, v) \leq r\} \qquad N^r(v) := N^r[v] \setminus \{v\}$$

Similarly for a set $S \subseteq V$ we define:

$$N^r[S] := \bigcup_{v \in S} N^r[v] \qquad N^r(S) := N^r[S] \setminus S$$

Girth, Radius: The girth g of a graph G is the length of its shortest cycle, or ∞ if acyclic. The radius R of G is the minimum integer R for which $\exists v \in V : N^R[v] = V$.

Distance-r Dominating Set: A set $M \subseteq V$ is a distance-r dominating set if $V = N^r[M]$. If there is no smaller such set, then M is a minimum distance-r dominating set of G.

Distance-r Connected Dominating Set: A set of vertices $D \subseteq V$ is a distance-r connected dominating set of G if D is a distance-r dominating set of G and the subgraph of G induced on vertices in D is connected.

Distance-r Vertex Cover: A set $C \subseteq V$ is a distance-r vertex cover of G if for every edge $e = \{u, v\}$, there exists a vertex $w \in C$ such that the distance of both u and v from w is at most r. The special case of $r = 1$ is the standard vertex cover problem. Note that in contrast to dominating set, there is no equivalence for vertex cover between the distance-r and power graph version.

Distance-r Connected Vertex Cover: Similarly, a set $\hat{C}$ is a distance-r connected vertex cover of G if it is a distance-r vertex cover of G and the induced subgraph of G on vertices of $\hat{C}$ is connected.

Edge Density, r-Shallow Minor, Expansion: Let $G = (V, E)$ be a graph; its edge density is $|E|/|V|$. A graph H is an r-shallow minor of G if H can be obtained from G by the following operations. First, we take a subgraph S of G and then

partition the vertices of S into vertex disjoint connected subgraphs $S_1, \dots, S_t$ of S, each of radius at most r and, at the end, contract each S_i ($i \in [t]$) to a single vertex to obtain H. We denote by $\nabla_r(G)$ the maximum edge density among all r-shallow minors of the graph G.

A graph class $\mathcal{C}$ is bounded expansion if there is a function $f : \mathbb{N} \to \mathbb{N}$ such that for every graph $G \in \mathcal{C}$ and integer $r \in \mathbb{N}$ we have $\nabla_r(G) \le f(r)$; here f is the *expansion function*. A class of graphs $\mathcal{C}$ has constant expansion if we also have $f \in O(1)$.

Every planar, bounded genus, and excluded minor graph is a constant expansion graph. Every class of bounded degree graphs is also bounded expansion, but not of constant expansion. For more information on bounded expansion graphs, we refer the reader to the book by Nešetřil and Ossona de Mendez [25].

LOCAL and CONGEST Model of Computation: The LOCAL model of computation assumes that the problem is being solved in a distributed manner: Each vertex in the graph is also a computational node, the input graph is also the communication graph, and initially, each vertex only knows its own unique identifier and its neighbors. An algorithm proceeds in synchronous rounds on each vertex in parallel. In each round, the algorithm can run an arbitrary amount of local computation, send a message of arbitrary size to its neighboring vertices, and then receive all messages from its neighbors. Each vertex can decide locally whether it halts with an output or continues. The most common metric is the number of communication rounds.

This model was first introduced by Linial [23]; later Peleg [26] named it LOCAL model.

The CONGEST model is very similar to the LOCAL model, except that identifiers can be represented in $O(\log n)$ bits, and each message can only hold $O(\log n)$ bits, where n is the number of vertices in the network.

3. Distributed approximation algorithm for dominating set

In this section we prove the following theorem.

Theorem 1. *Let $\mathcal{C}$ be a graph class of bounded expansion $f(r)$ and girth at least $4r + 3$. There is a CONGEST algorithm that runs in $O(r)$ rounds and provides an $O(r \cdot f(r))$-approximation of a minimum distance-r dominating set on $\mathcal{C}$, even if $r \ge 2$ is a non-constant function of n.*

We prove this by providing Algorithm 1, satisfying all bounds. At its core, the algorithm is simple: Each vertex computes the size of its distance-r neighborhood, i.e., the distance-r degree. This degree is propagated so that each vertex selects in its distance-r neighborhood the vertex with the highest such degree. The output is the set of all selected vertices. We expect this to yield a good approximation because only few candidates can be selected.

Algorithm 1 defines this formally. The main technical contribution is Lemma 9, which concludes that Algorithm 1 is correct and thus proves Theorem 1.

Algorithm 1: CONGEST computation of r-MDS, on each vertex v in parallel.

```
1: Compute |N^r(v)|, e.g. using Algorithm 2
2: // Phase 1: Select the vertex with the highest degree:
3: (prio^v, sel^v) := (|N^r(v)|, v)
4: for r rounds do
5:     Send (prio^v, sel^v) to all neighbors
6:     Receive (prio^u, sel^u) from each neighbor u
7:     (prio^v, sel^v) := max_{u∈N^1[v]}{(prio^u, sel^u)}
8:     Remember all received messages that contained (prio^v, sel^v)
9: end for
10: // Phase 2: Propagate back to the selected vertex:
11: D^v := {sel^v}
12: for r rounds do
13:     for each neighbor u ∈ N^1(v) do
14:         Determine which vertices sent by u in Phase 1 are in D^v
15:         Send these to u, encoded as a bitset of size r
16:     end for
17:     Receive bitsets, extend D^v accordingly
18: end for
19: Join the dominating set if and only if v ∈ D^v, in other words:
20: return v ∈ D^v
```

We say that Algorithm 1 computes a set D by returning $\top$ for all vertices in the set, and $\bot$ for all others. Naturally, messages and comparisons only consider the ID of vertices, and not the vertices themselves. This abuse of notation simplifies the algorithm and analysis. In line 7, we order tuples lexicographically: Tuples are ordered by the first element (the size of the distance-r neighborhood); ties are broken by the second element (the ID of the vertex).

In lines 15 and 17, we suggest to use "bitsets" to communicate a potentially large set of nodes efficiently. Here we exploit that vertices v and its neighbor u know which candidate-vertices were announced by vertex v in Phase 1, and in which

order. This means that vertex v can encode a set of vertices simply sending e.g. 0 to indicate absence and 1 to indicate presence in the set. Note that this obeys the message size restriction of CONGEST if and only if $r \in O(\log n)$. Observe that this is satisfied for constant r and $r \in O(\log^* n)$. For larger $r \in \omega(\log n) \cap O(n)$, this scheme of communication is not viable. A different scheme could be used: e.g. counting a new variable i backwards from r to 1 in each round, and only sending a single bit each round, indicating whether the node announced in Phase 1 round i was selected by anyone. We propose the bitset-approach because it seems easier to explain and understand, and because $r \in O(\log n)$ holds for all interesting cases.

Algorithm 2: CONGEST computation of $\left|N^r(v)\right|$, on each vertex v in parallel.

1: $n_u := 1$ for all $u \in N^1(v)$
2: **for** r rounds **do**
3: To each vertex $u \in N^1(v)$, send $1 + \sum_{w \in N^1(v) \setminus \{u\}} n_w$
4: $n_u :=$ the number received from u, for each $u \in N^1(v)$
5: **end for**
6: **return** $\sum_{w \in N^1(v)} n_w$

Algorithm 2 implements the computation of the distance-r neighborhood. The intuition is to compute the size of a *rooted* tree, for all possible roots at once. The high girth of G and line 3 mean that each vertex is counted only once (if at all).

The remainder of this section proves Algorithm 1's correctness and approximation factor.

3.1. Correctness

First, we will show basic correctness properties. One can trivially check that all messages contain only $O(\log n)$ many bits. Specifically, the bitsets have only size $r \in o(\log^* n)$.

Lemma 3. *Algorithm 2 computes the size of $N^r(v)$ for each vertex v in parallel.*

Proof. First, observe that in only $r-1$ rounds of communication, no cycle can be detected, as the girth is at least $4r+3$. This means that $N^i(v)$ is a tree for every $i \le r-1$ and $v \in V$. We define the tree $T^{\neg v}_{u,i}$ as the (set of vertices in the) tree of edge-depth i, rooted at vertex u, excluding the branch to vertex v, where v is a neighbor of vertex u.

We prove by induction: At vertex v, after the i-th round[1] (where $0 \le i \le r-1$), n_u stores the size of the tree $T^{\neg v}_{u,i}$.

For the induction basis $i = 0$, we know $\forall u, v : n_u = 1 = \left|T^{\neg v}_{u,0}\right| = |\{u\}|$.

This leaves the induction step: At the beginning of the i-th round (for $1 \le i \le r-1$), we know that $n_u = \left|T^{\neg v}_{u,i-1}\right|$ by the induction hypothesis, for every u, v. Consider vertex v. By construction, its distance-i open neighborhood is the union of all edge-depth $i-1$ trees of v's neighbors, so: $N^i(v) = \bigcup_{u \in N^1(v)} T^{\neg v}_{u,i-1}$. Due to the high girth requirement, we know that all sets in this union are disjoint. Vertex v can therefore compute $\left|N^i(v)\right|$ by summing up all its n_us, and can even compute $\left|T^{\neg u}_{v,i}\right|$ for an arbitrary vertex u by subtracting the corresponding n_u again. This is exactly what happens in line 3. Then v sends $\left|T^{\neg u}_{v,i}\right|$ to each neighbor u, which stores it in the corresponding variable n_v. By symmetry, this also means that vertex v now has stored $\left|T^{\neg v}_{u,i}\right|$ in n_u, thus proving the induction step.

With the meaning of n_u established, line 6 of Algorithm 2 must compute $\left|N^r(v)\right|$. □

Next, we show that Algorithm 1 selects the maximum degree neighbor:

Lemma 4. *In Algorithm 1, at the start of Phase 2 (line 10 et seq.), each vertex v has selected a vertex sel^v. This is the unique vertex* $\arg\max_{u \in N^r[v]}\{(\left|N^r(u)\right|, u)\}$.

Proof. By construction, only tuples of the form $(\left|N^r(w)\right|, w)$ with $w \in V$ are ever stored. The max operator is commutative and associative, so it is sufficient to prove that each vertex v considers precisely the tuples for $w \in N^r[v]$. This can be shown by straightforward induction: After round i, vertex v considers precisely the tuples for $w \in N^i[v]$. The base case is $i = 0$, and the induction step is straight-forward. □

Hence, each vertex selects the maximum neighbor. Next, we show that this is back-propagated:

Lemma 5. *If there is a vertex u that selects v ($sel^u = v$), then $v \in D^v$.*

[1] We interpret "after the zeroth round" as "before the first round".

Proof. Consider the path along which v was forwarded during the selection phase. By straight-forward induction, one can see that after i rounds of propagation, for all vertices w on the path with $d(w,u) \leq i$, have $v \in D^w$. The path has length at most r edges, so $v \in D^v$ after r rounds. □

And because no further vertices are added into any D^v, we get:

Corollary 6. *The selected vertices are precisely the computed set: $D = \{sel^v \mid v \in V\}$.*

Together with Lemma 4, this shows that Algorithm 1 computes a dominating set.

Lemma 7. *The computed set D is a distance-r dominating set.*

Proof. Assume towards contradiction that a vertex v is not dominated. Lemma 4 shows that v selected a vertex sel^v in its distance-r neighborhood. Corollary 6 shows that $sel^v \in D$, which distance-r-dominates v, which is a contradiction. □

The time complexity analysis is trivial.

Lemma 8. *Algorithm 1 runs in $O(r)$ rounds.*

3.2. Approximation analysis

So far we have seen the correctness of the algorithm and the running time bound. In this subsection, we prove the approximation bound of Lemma 9. Specifically, we prove that the size of D, the set of selected vertices, is within factor $1 + 2r \cdot f(r)$ of $|M|$, a minimum distance-r dominating set.

Lemma 9. *If the graph class $\mathcal{C}$ has expansion $f(r)$ and girth at least $4r + 3$, then the set of vertices D selected by Algorithm 1 is small: $|D|/|M| \in O(r \cdot f(r))$.*

In the remainder of this subsection, we prove Lemma 9. Note that this means that if r is constant, then the approximation factor is constant, too.

We now analyze the behavior of Algorithm 1 on a particular graph $G \in \mathcal{C}$. First, we want to get rid of an edge-case:

Lemma 10. *If $|M| = 1$, then $|D| = 1$.*

Proof. Let $M = \{v\}$. Then $\left|N^r[v]\right| = |V|$. Let u be the vertex with $\left|N^r[u]\right| = |V|$ with maximum ID; this may or may not be different from vertex v. By construction and Lemma 4, all vertices must select u, therefore $D = \{u\}$. □

We begin by showing that the optimal solution implies a partition into Voronoi cells [27], which we will use throughout the analysis. First, we define what a *covering* vertex is. Note that this can be (and often is) different from the vertex selected by the algorithm.

Definition 11. Let $c : V \to M$ be the mapping from vertices in V to corresponding dominating vertices in M, breaking ties first by distance and then by ID:

$$c(v) := \operatorname*{arg\,min}_{u \in N^r[v] \cap M} \{(d(u,v), u)\} \tag{1}$$

We order tuples lexicographically, again. The equivalent term $\operatorname{arg\,min}_{u \in M}\{(d(u,v),u)\}$ provides shorter notation: By construction, each vertex v has a vertex in M in its r-neighborhood, so arg min will select from $N^r[v]$ anyway. Next, we partition V into Voronoi cells $H_m := \{v \in V \mid c(v) = m\}$ for each $m \in M$.

Corollary 12. *Each H_m is connected and has radius at most r.*

Proof. As vertex m dominates all vertices in H_m, we know that there is a path of length at most r from vertex m to each vertex in H_m. □

We use the high-girth property to show that the Voronoi cells behave nicely:

Lemma 13. *The subgraph induced by H_m in G is a tree.*

Proof. Assume towards contradiction that there is a cycle C' in H_m. We construct a cycle that has length at most $2r+1$, a contradiction.

Specifically, construct a BFS-tree of H_m rooted in m. Then, the cycle C' must contain an edge e between $u, v \in H_m$. Consider the cycle that consists of the path from u to v along the BFS-tree and the edge $\{u, v\}$. This cycle must have length at most $r+r+1$, because the BFS-tree has depth at most r. This contradicts G having a girth of at least $4r+3$. □

Lemma 14. *For any two Voronoi cells $H_m \neq H_n$, there is at most one edge between them.*

Proof. Let $\{u, v\} \in E$ and $\{s, t\} \in E$ be two different edges between H_m and H_n. W.l.o.g. assume $c(u) = c(s) = m$ and $c(v) = c(t) = n$, and assume $v \neq t$ (but $u = s$ is possible).

By Corollary 12, we know that the subgraphs induced by H_m and H_n are each connected, so there must be a path p_m entirely in H_m between vertices u and s, possibly of length 0. Likewise, a path p_n must exist entirely in H_n between vertices v and t. The union of the paths and the assumed edges forms a cycle $C_{u,v,s,t}$, as no vertex can be repeated. We will now prove that $C_{u,v,s,t}$ is too small.

The paths p_m and p_n have length at most $2r$ each. Therefore, we have found the cycle $C_{u,v,s,t}$ to have length at most $4r+2$, in contradiction to the minimum girth $4r+3$. □

Let $G' = (V', E')$ be the result of contracting H_m to a single vertex, for each $m \in M$.

Lemma 15. *The edge set E' is small: $|E'| \leq f(r) \cdot |M|$.*

Proof. Using Corollary 12, we can apply the definition of the function $f(r)$:

$$|E'| = \frac{|E'|}{|V'|} \cdot |V'| \leq f(r) \cdot |M| \quad \square$$

Next, we construct a set of candidates, based on each Voronoi cell.

Definition 16. For each Voronoi cell H_m, we define the set of vertices $\mathcal{T}_m$ as the union of all shortest paths $P_{m,u}$ between vertex m and each vertex u on the Voronoi boundary, and we define $\mathcal{T}$ as the union of all $\mathcal{T}_m$:

$$\mathcal{T}_m := \bigcup_{\substack{\{u,v\}\in E \\ c(u)=m, c(v)\neq m}} P_{m,u} \qquad \mathcal{T} := \bigcup_{m\in M} \mathcal{T}_m$$

This is well-defined due to Lemma 13. Observe that $\mathcal{T}_m$ is not necessarily equal to H_m: All leaves in $\mathcal{T}_m$ have edges in G that lead outside the Voronoi cell. A leaf-vertex in H_m that has no such edges will not be in $\mathcal{T}_m$. This reduces the number of candidates sufficiently:

Lemma 17. *The set $\mathcal{T}$ is small: $|\mathcal{T}| \leq (1 + 2r \cdot f(r))\,|M|$.*

Proof. Consider an arbitrary but fixed $\{v, u\} \in E$ with $c(v) \neq c(u)$. Each path $P_{c(v),v}$ has at most r vertices not in M, because it is a shortest path, and by construction all vertices are dominated by $c(v)$. Each edge in E' corresponds to at most two such paths, due to Lemma 14. With Lemma 15, this bounds the number of paths to at most $2f(r)\,|M|$.

Therefore, $\mathcal{T}$ contains at most $2r \cdot f(r)\,|M| + |M|$ vertices. □

Now we can prove in two steps that the algorithm only selects vertices from the candidate set $\mathcal{T}$:

Lemma 18. *If $|M| > 1$, then there is always a vertex just out of reach (i.e. in distance $r+1$):*

$$\forall v \in V\ \exists u \in V : d(u, v) = r + 1 \tag{2}$$

Proof. Assume towards contradiction that there is a vertex v for which no such vertex u exists. Then, there is also no vertex u' with $d(u', v) > r+1$, because one could pick a shortest path and construct such a u. Therefore, $D' = \{v\}$ would be a dominating set, a contradiction to $|M| > 1$. □

Lemma 19. *Let v be a vertex selected by u. Then $v \in \mathcal{T}_{c(v)}$.*

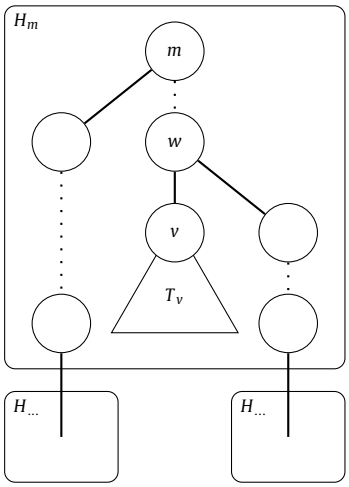

Fig. 2. Typical vertex layout in proof of Lemma 19. The identity of vertex u does not matter; hence, it is not shown.

Proof. Assume towards contradiction that $v \notin \mathcal{T}_{c(v)}$. For brevity, let $m := c(v)$. Observe that $m \neq v$, because $m \in \mathcal{T}_m$. Let w be the next vertex on a shortest path from v towards m; possibly m itself. We now analyze the properties of vertex w and conclude that vertex u should not have selected v. Refer to Fig. 2 for an overview.

By Lemma 13, the subgraph induced by H_m is a tree. If we root this H_m-tree at vertex m, we can denote the subtree rooted at v as T_v. This subtree has depth at most $r-1$ (because $v \neq m$), so w covers the entire subtree: $T_v \subseteq N^r(w)$. All vertices $x \in V \setminus T_v$ are closer to w than to v, as all paths from x to v must go through w. So, the neighborhood of v is included in the neighborhood of w: $N^r[v] \subseteq N^r[w]$.

Now we can use Lemma 18: There must be a vertex t that has distance $r+1$ to vertex v, so $t \notin N^r[v]$. This means that $t \notin N^r[v]$ and $t \in N^r[w]$. Therefore, the degree of w is strictly larger: $\left|N^r[w]\right| > \left|N^r[v]\right|$. Thus, vertex u would prefer selecting w over v, and also was able to do so (because $u \in N^r[v] \subseteq N^r[w]$).

This leads to a contradiction: Vertex u selected v, although vertex w should be preferred by the algorithm. □

We can combine the previous lemmas to achieve an exact bound:

Corollary 20. *The set D is small: $|D| \leq (1 + 2r \cdot f(r))\,|M|$.*

Proof. Follows from Lemmas 17 and 19. □

This proves Lemma 9, and thus Theorem 1. More specifically, we have proved the upper bound $(1 + 2r \cdot f(r)) \cdot |M|$ on $|D|$.

3.3. Tightness of the analysis

We have seen that the algorithm is an $O(r \cdot f(r))$ approximation. Is it possible that the algorithm actually performs significantly better than what the analysis guarantees? This subsection proves that there are graphs for which the algorithm yields an $\Omega(r \cdot f(r))$ approximation, meaning that the above analysis of the algorithm is asymptotically tight.

We will focus on the tightness of Corollary 20, by constructing a worst-case input graph. See also Lemma 9.

Lemma 21. *For arbitrary but fixed values $r \geq 2$, $v \geq 2$, there is a graph $G_{r,v}$ such that:*

- *the singleton graph class $\mathcal{C}_{r,v} := \{G_{r,v}\}$ has expansion $f(r) = v$ and girth at least $4r + 3$, and*
- *executing Algorithm 1 on $G_{r,v}$ computes a $(r \cdot f(r))$-approximation (or worse) of the minimum distance-r dominating set of $G_{r,v}$.*

Proof. The remainder of this subsection constructs the graph $G_{r,v}$ and proves its properties. □

The construction is a modified version of the subdivided biclique. Let X and Y be two disjoint sets of vertices, each of size $2v$. For each pair $(x, y) \in X \times Y$, create a path $P_{x,y}$ starting at vertex x and ending in vertex y, with $2r$ new vertices, such that $d(x, y) = 2r + 1$. This means that no vertex can simultaneously cover x and y, i.e., no vertex is within distance r of both x and y. In path $P_{x,y}$, let $b_{x,y}$ be the second vertex (the one after x). Create a set $B_{x,y}$ of $k = 2rv$ new vertices, and

connect each vertex in $B_{x,y}$ by a single edge to the vertex b. Let V be the union of all the sets X, Y, $P_{x,y}$, $B_{x,y}$; and let E be the set of edges as described. Then $G_{r,v} = (V, E)$ is the constructed graph.

First, we prove that the graph class satisfies all requirements.

Lemma 22. *The graph class $\mathcal{C}_{r,v} = \{G_{r,v}\}$ has expansion $f(r) = v$ and girth at least $4r + 3$.*

Proof. The girth of $G_{r,v}$ is at least $4 \cdot (2r + 1) > 4r + 3$, as a cycle needs to pass through at least two vertices from X and two vertices from Y.

To prove the low expansion of $\mathcal{C}_{r,v}$, it suffices to show $\nabla_r(G) = v$. This can be shown by contracting as much as possible around all vertices in $X \cup Y$, which results in the biclique $K_{2v,2v}$, with $4v$ vertices and $4v^2$ edges. Therefore, the constructed graph has $\nabla_r(G) \geq v$. This is the contraction choice that maximizes the average degree, as it eliminates all degree-2 vertices. Therefore $\nabla_r(G) = v$, and thus $f(r) = v$. □

Next, we show that the algorithm computes a comparatively large dominating set:

Lemma 23. *Algorithm 1 computes a $(r \cdot f(r))$-approximation (or worse) of minimum distance-r dominating set on $G_{r,v}$.*

Proof. By construction, $X \cup Y$ is a dominating set, meaning $|M| \leq 4f(r)$. Therefore, it suffices to show that $|D| \geq 4r \cdot f(r)^2$.

We do so by simulating the algorithm on $G_{r,v}$. We only need to consider the vertices selected by vertices on the paths do. Specifically, pick a specific path $P_{x,y}$ between $x \in X$ and $y \in Y$. Vertices v_x closer to x than to y cover the attached vertices $B_{x,y}$, so $\left|N^r(v_x)\right| \geq 2r + k = 2r + 2r \cdot f(r)$. The vertices closer to vertex x cover more of the other paths ending in x, each step increases $\left|N^r(v_x)\right|$ by at least $2f(r) - 1$, and loses at most 1 vertex out of sight in the y direction. Note that we ignore the vertices in $B_{x,y'}$ with $y' \neq y$, which would only make this argument stronger. The important property is that $\left|N^r(v_x)\right|$ strictly increases towards x, among vertices v_x with $d(v_x, x) < d(v_x, y)$.

Each vertex v_y closer to y than to x does not cover the attached vertices $B_{x,y}$ close to vertex x, as distance r from them would imply distance r to x. We can compute $\left|N^r(v_y)\right| \leq r + N^r(v_r) + 1 - 1 = r + r \cdot 2f(r) < 2r + 2r \cdot f(r) \leq \left|N^r(v_x)\right|$, so vertex v_y will choose some vertex v_x. As we already established, $\left|N^r(v_x)\right|$ increases with decreasing distance to x. Therefore, each v_y will select the vertex closest to x, meaning at least half of each path will be selected, specifically the one on the v_l side.

This means the algorithm selects at least r vertices per path, and there is one such path for each $X \times Y$ combination. Hence $|D| \geq r \cdot 4 \cdot f(r)^2$. Recall that $|M| \leq 4f(r)$, so the algorithm achieves an approximation factor of at least $r \cdot f(r)$ for the constructed graph. Compared with the upper bound of $1 + (2r \cdot f(r))$ this is asymptotically tight. □

This concludes the proof of Lemma 21 (tightness of approximation). This does not mean that the problem is hard. It only shows that in the worst case, the presented algorithm may use up the approximation slack.

4. Lower bound

In this section, we prove that a significantly better approximation of the problem is hard. Intuitively speaking, this is because symmetry cannot be broken in $o(\log^* n)$ rounds, and without that, it is hard to construct any non-trivial distance-r dominating set.

We show the hardness by a reduction from the "large" independent set problem to the distance-r dominating set problem, on the graph class of cycles. Intuitively speaking we find a distance-r dominating set D on cycle C; two consecutive vertices of D on C are of distance at most $2r + 1$ from each other, and hence, these vertices help us to break the symmetry, and as $r \in o(\log^* n)$ it yields an independent set of size $O(n)$ in $o(\log^* n)$ rounds.

Theorem 2. *Assume an arbitrary but fixed $\delta > 0$ and $r > 1$, with $r \in o(\log^* n)$. Then, there is no deterministic LOCAL algorithm that finds in $O(r)$ rounds a $(2r + 1 - \delta)$-approximation of distance-r dominating set for all $G \in \mathcal{C}$, where $\mathcal{C}$ is the class of cycles of length $\gg 4r + 3$.*

As we will see later, the trivial distance-r dominating set V (i.e., the set of all vertices), is a $(2r + 1)$-approximation in the case of cycles.

This has been proved implicitly in the work of [21]. However, we find it simpler to provide a new proof tailored for our setting, but only for n being a multiple of $2r + 1$. In essence, we show a reduction from the "large" independent set problem to the distance-r dominating set problem, on the graph class of cycles. Intuitively speaking, any algorithm that does significantly better than the trivial dominating set *anywhere* on the cycle leads to a linear sized independent set; and the bound is constructed such that the algorithm needs to do better than trivial somewhere indeed.

The idea is simple: Find a distance-r dominating set D on cycle C; we know two consecutive vertices of D on C are of distance at most $2r + 1$ from each other, and hence, these vertices help us to break the symmetry, and as $r \in o(\log^* n)$ it yields an independent set of size $O(n)$ in $o(\log^* n)$ rounds. In the remainder, we formalize this argument.

Assume towards contradiction that $\mathcal{ALG}$ is such a deterministic distributed algorithm, which finds a distance-r dominating set in $G \in \mathcal{C}$ of size at most $(2r+1-\delta)|M|$, where M is a minimum distance-r dominating set and r and δ as in Theorem 2.

We show that $\mathcal{ALG}$ can be used to construct an algorithm violating known lowerbounds on "large" independent set [14, 21]:

Lemma 24 *(Lemma 4 of [14]). There is no deterministic distributed algorithm that finds an independent set of size $\Omega(n/\log^* n)$ in a cycle on n vertices in $o(\log^* n)$ rounds.*

We present the reduction algorithm in Algorithm 3.

Algorithm 3: CONGEST computation of an IS on a cycle $G \in \mathcal{C}$, for each v in parallel.

```
1: Compute a distance-r dominating set D by simulating ALG.
2: Determine the connected components V \ D.
3: for each component C_i do
4:     Determine the two adjacent vertices to C_i, i.e. u, v ∈ N(C_i).
5:     Let u be the vertex with the lower ID, name it representor of C_i.
6:     All vertices of odd distance to u in C_i join I.
7: end for
8: return I
```

We begin by showing basic correctness:

Lemma 25. *Algorithm 3 runs in $o(\log^* n)$ rounds.*

Proof. By assumption, $\mathcal{ALG}$ executes in $O(r)$ rounds. On the other hand, observe that each vertex in D only covers up to a distance of r. Because D is a dominating set, all components must have length at most $2r$. Hence, discovering the adjacent vertex of lowest ID can be done in $O(r)$ as well as propagating the distance information. By construction $r \in o(\log^* n)$, so Algorithm 3 takes $o(\log^* n)$ rounds. □

Lemma 26. *Algorithm 3 computes set I, which is an independent set.*

Proof. For two distinct vertices $u, v \in I$, if they belong to different components, then there is no edge between them; otherwise, if they are in the same component, their distance is at least 2, as they are distinct vertices of odd distance from their representor. □

Now we can show that this yields a large independent set:

Lemma 27. *The dominating set is not too large: $|D| \leq (1-\delta')n$ for some $\delta' > 0$.*

Proof. By assumption, we know $|D| \leq (2r+1-\delta)|M|$, where M is the minimum distance-r dominating set. Construct M' by picking every $2r+1$-th vertex so that $|M'| = n/(2r+1)$. Note that M' is a distance-r dominating set, so we have $|M| \leq |M'|$. Together we get $|D| \leq (2r+1-\delta)n/(2r+1) = (1-\delta')n$, for $\delta' := 1/(2r+1) > 0$. □

Lemma 28. *The set I is large: $|I| \in \Omega(n/\log^* n)$.*

Proof. Many vertices must be part of some component: $|V \setminus D| \geq \delta' n$ for some $\delta' > 0$ by Lemma 27. At least half of those vertices are taken into I, thus $|I| \geq \delta' n/2 \in \Omega(n/\log^* n)$. □

Proof of Theorem 2. Lemmas 24 and 28 imply that algorithm $\mathcal{ALG}$ cannot exist. □

Note that this does not preclude randomized algorithms. This is because randomized algorithms can indeed achieve a better approximation quality, at least on cycles, by randomly joining the dominating set with sufficiently small probability if necessary, for several rounds, and finally all uncovered vertices join.

5. Vertex cover, connected dominating set, and connected vertex cover

In this section, we extend Algorithm 1 to solve several other covering problems, namely Distance-r Vertex Cover, Distance-r Connected Vertex Cover, and Distance-r Connected Dominating Set; the precise definitions can be found in Section 2. Again, we assume the input graph is $G = (V, E)$ so that we can directly refer to its edge and vertex set.

The rough idea for each of these extensions is simple: For connected variations, we take the original set and connect two vertices that are within small distance via short paths. It is possible to show that due to the structure of bounded expansion graphs, this approach does not add much extra overhead and it is a constant factor approximation to the problem. For vertex cover, observe that every vertex cover is already a dominating set, to go the other way around we define boundary vertices of each Voronoi cell of the dominating set, then include them into the solution to vertex cover. It is possible to show that this approach provides a constant factor approximation for the problem (distance-r variation).

Proposition 29. *There is a CONGEST algorithm that computes an $O((r \cdot f(r))^2)$ approximation of Distance-r Connected Dominating Set in graphs of bounded expansion with high girth in $O(r)$ rounds.*

Proof. We prove this by constructing Algorithm 4 as a simple extension of Algorithm 1, or any other appropriate CONGEST distance-r dominating set algorithm.

Algorithm 4: CONGEST computation of connected r-MDS, on each vertex v in parallel.

1: Compute a Distance-r Dominating Set D of the graph
2: Determine the closest dominating vertex sel^v
3: Determine the path P^v from v to sel^v
4: If any neighbor u has a $sel^u \neq sel^v$, call vertex v a border
5: **return** $\hat{D}$ as the union of D, all border vertices, and their paths P^v

Algorithm 4 is a CONGEST algorithm, as the distance-r dominating set algorithm is CONGEST, and all other messages only contain a constant amount of identifiers.

Algorithm 4 takes $O(r)$ rounds, because the distance-r dominating set algorithm does so, too, and all other steps also only take $O(r)$ rounds.

$\hat{D}$ is a dominating set because $D \subseteq \hat{D}$ is a dominating set.

Define Voronoi cells H_d for each $d \in D$. Note that Corollary 12 and Lemma 13 apply analogously.

We show that $\hat{D}$ is connected by construction, if G is connected: Vertices v within a Voronoi cell $H_d \cap \hat{D}$ are connected by construction, as they are all connected to $d \in \hat{D}$. Furthermore, for every path P_G in the input graph G, one can construct a corresponding walk W_H in the Voronoi graph by mapping each vertex to its Voronoi cell (i.e. sel^v). Thus, the Voronoi cells are connected.

Finally, we show the approximation quality: Consider the minimal Distance-r Connected Dominating Set $\hat{M}$. One can easily see that the minimum distance-r dominating set M is not larger: $|M| \leq \left|\hat{M}\right|$. An argument similar to Lemma 15 shows that the number of border vertices is bounded in $|D|$; and by construction of D the Voronoi cells have radius at most r (and therefore so do the paths). By Theorem 1, we can now deduce:

$$\begin{aligned}\left|\hat{D}\right| &\in O(r \cdot f(r) \cdot |D|)\\ &\subseteq O((r \cdot f(r))^2 \cdot |M|)\\ &\subseteq O((r \cdot f(r))^2 \cdot \left|\hat{M}\right|) \qquad \square\end{aligned}$$

For constant r, the terms simplify to:

Corollary 30. *There is a CONGEST algorithm that computes a constant factor approximation of Distance-r Connected Dominating Set for constant r in graphs of bounded expansion with high girth in constant number of rounds.*

Likewise, we can solve the related vertex cover problem. Intuitively speaking, we can define Voronoi cells according to the computed dominating set, determine borders between cells, and include all borders into the vertex cover. More formally:

Proposition 31. *There is a CONGEST algorithm that computes an $O(r \cdot f(r) \cdot f(r))$ approximation of Distance-r Vertex Cover in graphs of bounded expansion with high girth in $O(r)$ rounds.*

Proof. We prove this by constructing Algorithm 5 as a simple extension of Algorithm 1, or any other appropriate CONGEST distance-r dominating set algorithm.

Algorithm 5: CONGEST computation of distance-r vertex cover, on each vertex v in parallel.

1: Compute a Distance-r Dominating Set D of the graph
2: Determine the closest dominating vertex sel^v
3: If any neighbor u of vertex v has a $sel^u \neq sel^v$, call v a border
4: **return** C as the union of D and all border vertices

Algorithm 5 is a CONGEST algorithm as the Distance-r Dominating Set algorithm is CONGEST, and all other messages only contain a constant amount of identifiers. Algorithm 5 takes $O(r)$ rounds, because the Distance-r Dominating Set algorithm does so, too, and all other steps also only take $O(r)$ rounds.

Define Voronoi cells according to sel^v for $v \in V$. The set C is a distance-r vertex cover by simple case distinction: All edges $e = u, v$ that are fully inside a Voronoi cell, i.e., there is a vertex $w \in C$ with $w = sel^u = sel^v$, is covered by vertex w. All edges $e = u, v$ with $sel^u \neq sel^v$ are covered by vertices u and v, as both vertices were detected as borders.

Finally, we show the approximation quality: Consider the minimal Distance-r Vertex Cover M^{VC}. One can easily see that the minimum distance-r dominating set M is smaller: $|M| \leq \left|M^{VC}\right|$. An argument similar to Lemma 15 shows that the number of border vertices is bounded in $|D|$. By Theorem 1, we can now deduce:

$$\begin{aligned} |C| &\in O(f(r) \cdot |D|) \\ &\subseteq O(r \cdot f(r) \cdot f(r) \cdot |M|) \\ &\subseteq O(r \cdot f(r) \cdot f(r) \cdot \left|M^{VC}\right|) \end{aligned}$$ □

Again, for constant r, the terms simplify to:

Corollary 32. *There is a CONGEST algorithm that computes a constant factor approximation of the minimum Distance-r Vertex Cover in a constant number of rounds.*

From Proposition 29 and Corollary 32, the following is straight-forward to see:

Corollary 33. *There is a CONGEST algorithm that computes a constant factor approximation of the minimum Distance-r Connected Vertex Cover in a constant number of rounds.*

This proves that Algorithm 1 can be extended to compute a Distance-r Vertex Cover instead, as mentioned in Section 5.

6. Conclusion

We have analyzed important distance-r covering problems in a deterministic setting on graphs of bounded expansion $f(r)$ (e.g. planar graphs) and high girth. We have provided CONGEST algorithms and proved that they achieve a constant factor approximation in constant rounds (for a constant r). We have shown that the standard $\Omega(\log^* n)$ lower bound on bounded degree graphs also holds here, even if r is super-constant. This means that Algorithm 1 can only be improved up to a factor of $f(r)$, if at all.

We believe that with our algorithmic analysis tools it is possible to extend the lower bound of Kuhn et al. [20]. This result relies on line graphs that typically result in graphs with very low girth. We believe that this can be done without taking the line graph, thus extending the result to high-girth graphs for dominating set. On the other hand, the necessity of some kind of sparsity requirement seems clear. To what extent can this requirement be reduced?

We have covered the dominating set problem and related covering problems. The next step is to weaken either the sparsity condition or the girth requirement, with the goal of finding the most generic class of graphs with a reasonable approximation in constant rounds, and eventually discovering their connection to dense graphs.

For distance-r computation, one might also consider other interesting problems such as independent set and coloring problems. These relate to network decomposition, and it might help to find a faster network decomposition in the CONGEST model.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] I. Althöfer, G. Das, D.P. Dobkin, D. Joseph, J. Soares, On sparse spanners of weighted graphs, Discrete Comput. Geom. 9 (1993) 81–100.

[2] S.A. Amiri, Deterministic CONGEST algorithm for MDS on bounded arboricity graphs, CoRR, arXiv:2102.08076 [abs], 2021.
[3] S.A. Amiri, P. Ossona de Mendez, R. Rabinovich, S. Siebertz, Distributed domination on graph classes of bounded expansion, in: Proceedings of the 30th on Symposium on Parallelism in Algorithms and Architectures, ACM, 2018, pp. 143–151.
[4] S.A. Amiri, S. Schmid, Brief announcement: a log-time local MDS approximation scheme for bounded genus graphs, in: Proc. 30th International Symposium on Distributed Computing (DISC), Springer, 2016.
[5] S.A. Amiri, S. Schmid, S. Siebertz, A local constant factor MDS approximation for bounded genus graphs, in: Proc. ACM Symposium on Principles of Distributed Computing (PODC), 2016.
[6] S.A. Amiri, B. Wiederhake, Distributed distance-r covering problems on sparse high-girth graphs, in: T. Calamoneri, F. Corò (Eds.), Algorithms and Complexity - 12th International Conference, CIAC 2021, Proceedings, in: Lecture Notes in Computer Science, vol. 12701, Springer, 2021, pp. 37–60.
[7] B. Awerbuch, B. Berger, L. Cowen, D. Peleg, Fast network decomposition, in: Proceedings of the Eleventh Annual ACM Symposium on Principles of Distributed Computing, PODC '92, ACM, 1992, pp. 169–177.
[8] B. Awerbuch, A.V. Goldberg, M. Luby, S.A. Plotkin, Network decomposition and locality in distributed computation, in: 30th Annual Symposium on Foundations of Computer Science, 1989, 1989, pp. 364–369.
[9] N. Bachrach, K. Censor-Hillel, M. Dory, Y. Efron, D. Leitersdorf, A. Paz, Hardness of distributed optimization, in: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, ACM, 2019, pp. 238–247.
[10] B.S. Baker, Approximation algorithms for NP-complete problems on planar graphs, J. ACM 41 (1) (1994) 153–180.
[11] R. Bar-Yehuda, K. Censor-Hillel, Y. Maus, S. Pai, S.V. Pemmaraju, Distributed approximation on power graphs, in: Proceedings of the 2020 ACM Symposium on Principles of Distributed Computing, PODC 2020, ACM, 2020, pp. 501–510.
[12] R. Bar-Yehuda, K. Censor-Hillel, G. Schwartzman, A distributed $(2+\epsilon)$-approximation for vertex cover in $o(\log\delta/\epsilon \log\log \delta)$ rounds, in: Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, 2016, pp. 3–8.
[13] A. Czygrinow, M. Hańćkowiak, E. Szymańska, Distributed approximation algorithms for planar graphs, in: Italian Conference on Algorithms and Complexity, Springer, 2006, pp. 296–307.
[14] A. Czygrinow, M. Hańćkowiak, W. Wawrzyniak, Fast distributed approximations in planar graphs, in: Distributed Computing, Springer, Berlin, Heidelberg, 2008, pp. 78–92.
[15] A. Czygrinow, M. Hańćkowiak, W. Wawrzyniak, M. Witkowski, Distributed approximation algorithms for the minimum dominating set in K_h-minor-free graphs, in: 29th International Symposium on Algorithms and Computation, ISAAC, 2018, pp. 22:1–22:12.
[16] J. Deurer, F. Kuhn, Y. Maus, Deterministic distributed dominating set approximation in the CONGEST model, in: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, 2019, pp. 94–103.
[17] R. Diestel, Graph Theory, 4th edition, Graduate Texts in Mathematics, vol. 173, Springer, 2012.
[18] M. Ghaffari, Distributed maximal independent set using small messages, in: Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2019, 2019, pp. 805–820.
[19] S. Kublenz, S. Siebertz, A. Vigny, Constant round distributed domination on graph classes with bounded expansion, in: Structural Information and Communication Complexity - 28th International Colloquium, SIROCCO 2021, Proceedings, in: Lecture Notes in Computer Science, vol. 12810, Springer, 2021, pp. 334–351.
[20] F. Kuhn, T. Moscibroda, R. Wattenhofer, Local computation: lower and upper bounds, J. ACM 63 (2) (Mar. 2016) 17.
[21] C. Lenzen, Y.A. Pignolet, R. Wattenhofer, Distributed minimum dominating set approximations in restricted families of graphs, Distrib. Comput. 26 (2) (2013) 119–137.
[22] C. Lenzen, R. Wattenhofer, Minimum dominating set approximation in graphs of bounded arboricity, in: Distributed Computing, vol. 6343, 2010, pp. 510–524.
[23] N. Linial, Locality in distributed graph algorithms, SIAM J. Comput. 21 (1) (1992) 193–201.
[24] M. Naor, L. Stockmeyer, What can be computed locally?, in: Proc. ACM 25th Annual ACM Symposium on Theory of Computing (STOC), 1993, pp. 184–193.
[25] J. Nešetřil, P.O. De Mendez, Sparsity: Graphs, Structures, and Algorithms, vol. 28, Springer Science & Business Media, 2012.
[26] D. Peleg, Distributed Computing: A Locality-Sensitive Approach, Society for Industrial and Applied Mathematics, 2000.
[27] F.P. Preparata, M.I. Shamos, Computational Geometry - An Introduction, Texts and Monographs in Computer Science., Springer, 1985.
[28] V. Rozhon, M. Ghaffari, Polylogarithmic-time deterministic network decomposition and distributed derandomization, in: Proc. ACM 52nd Annual ACM SIGACT Symposium on Theory of Computing (STOC), ACM, 2020, pp. 350–363.
[29] J. Schneider, R. Wattenhofer, A log-star distributed maximal independent set algorithm for growth-bounded graphs, in: Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, 2008, pp. 35–44.