

Setting Ports in an Anonymous Network: How to Reduce the Level of Symmetry?

Ralf Klasing^{*†}

Adrian Kosowski^{‡§}

Dominik Pajak^{§¶}

Abstract

A fundamental question in the setting of anonymous graphs concerns the ability of nodes to spontaneously break symmetries, based on their local perception of the network. In contrast to previous work, which focuses on symmetry breaking under arbitrary port labelings, in this paper, we study the following design question: Given an anonymous graph G without port labels, how to assign labels to the ports of G , in interval form at each vertex, so that symmetry breaking can be achieved using a message-passing protocol requiring as few rounds of synchronous communication as possible?

More formally, for an integer $l > 0$, the *truncated view* $\mathcal{V}_l(v)$ of a node v of a port-labeled graph is defined as a tree encoding labels encountered along all walks in the network which originate from node v and have length at most l , and we ask about an assignment of labels to the ports of G so that the views $\mathcal{V}_l(v)$ are distinct for all nodes $v \in V$, with the goal being to minimize l .

We present such efficient port labelings for any graph G , and we exhibit examples of graphs showing that the derived bounds are asymptotically optimal in general. More precisely, our results imply the following statements.

1. For any graph G with n nodes and diameter D , a uniformly random port labeling achieves $l = O(\min(D, \log n))$, w.h.p.
2. For any graph G with n nodes and diameter D , it is possible to construct in polynomial time a labeling that satisfies $l = O(\min(D, \log n))$.
3. For any integers $n \geq 2$ and $D \leq \log_2 n - \log_2 \log_2 n$, there exists a graph G with n nodes and diameter D which satisfies $l \geq \frac{1}{2}D - \frac{5}{2}$.

Keywords: Anonymous network, Port-labeled network, View, Level of symmetry.

1 Introduction

Consider a network G in which nodes have no identifiers but ports at each node of degree d are uniquely labeled with the integers from 1 to d (in any order). A *view* from a node v in such a network is a rooted infinite tree defined recursively. This tree is composed of the views from all the neighbors of v connected to the root (corresponding to node v) using the edges with the same port labels as the edges connecting v to their children in graph G .

The concept of a view is useful for many applications as it allows for identification of the network topology and for breaking of symmetries between nodes. For example, the topological

^{*}LaBRI, CNRS — Université de Bordeaux, 33400 Talence, France. E-mail: ralf.klasing@labri.fr

[†]Research partially supported by the ANR project DISPLEXITY (ANR-11-BS02-014). This study has been carried out in the frame of the Investments for the future Programme IdEx Bordeaux - CPU (ANR-10-IDEX-03-02).

[‡]Inria and LIAFA, CNRS — Université Paris Diderot, 75013 Paris, France. E-mail: adrian.kosowski@inria.fr

[§]Research partially supported by the National Science Centre, Poland - grant number 2015/17/B/ST6/01897.

[¶]Institute of Informatics, Faculty of Fundamental Problems of Technology, Wrocław University of Technology, 50-370 Wrocław, E-mail: dominik.pajak@pwr.edu.pl

knowledge which can be gathered by a node of an anonymous graph running a deterministic algorithm, in a setup defined identically (uniformly) over nodes, corresponds to its view. More precisely, in an anonymous communication network, in k rounds, each node can learn up to k levels of its view. An important question for such applications is to determine the smallest possible integer l (also called the level of symmetry of G) such that for all the different views, their subtrees truncated at depth l are also different. As observed in [16] in the context of leader election in the \mathcal{LOCAL} model, learning l levels of all the views is sufficient and necessary to determine whether leader election in the \mathcal{LOCAL} model is feasible, hence l determines the number of communication rounds to elect a leader.

Another example of applications concerns walker-based models of computation. The information gathered by a walker traversing an anonymous graph, which does not have the ability to write to its environment, is simply a subtree of the view from its starting vertex. Hence, for example, rendezvous of deterministic walkers is only possible if they start from positions with different views.

A lot of the related work so far has considered the question of bounding the largest possible level of symmetry (taken over all labelings). By contrast, in this work, we would like to focus on the best case labelings, i.e., those having the smallest possible level of symmetry. Moreover, we also look at the case of uniformly random labelings to verify whether the average case is closer to the best or the worst case.

1.1 Overview of results

For a graph G with labeling λ , we define the *level of symmetry* $l(G, \lambda)$ as the smallest integer l such that views from any two nodes, truncated at depth l , are different. If there exists a pair of nodes whose infinite views are equal, we say that the level of symmetry is infinite. For an unlabeled graph G , we refer to its level of symmetry as $l(G) = \min_{\lambda} l(G, \lambda)$, where the minimum is taken over all port labelings λ of G . We show the following results:

We first show that, for any graph G with n nodes and diameter D , the level of symmetry in the best case is $O(\min(D, \log n))$ by proposing an algorithm that labels any graph in such a way that the resulting labeled graph has level of symmetry $O(\min(D, \log n))$. Secondly, we show that a uniformly random labeling λ achieves level of symmetry $l(G, \lambda) = O(\min(D, \log n))$, w.h.p. Thirdly, we exhibit examples of graphs showing that these bounds are asymptotically tight in general.

1.2 Related work

The notion of *view* was introduced and first studied by Yamashita and Kameda in [29] in the context of distributed message passing algorithms. Yamashita and Kameda proved that if views of two nodes truncated to depth n^2 are identical, then their infinite views are identical [29], where n is the number of nodes of the network. The bound has been improved to $n - 1$ by Norris [24]. Although this bound is asymptotically tight [1, 24], it is far from being accurate for many networks. Hence, one may ask for bounds expressed as a function of different graph invariants. Fraigniaud and Pelc proved in [15] that if two nodes have the same views to depth $\hat{n} - 1$ then their views are the same, where \hat{n} is the number of nodes having different views (or equivalently, \hat{n} is the size of the quotient graph [29]). For some works on view computation see, e.g., [2, 26]. Recently, Hendrickx [20] proved (for simple graphs with symmetric port labeling) an upper bound of $O(D \log(n/D))$ on the depth to which views need to be checked in order to be distinguished, where D is the diameter of the network, leaving the tightness of this bound as an open problem. A complementary bound of $\Omega(D \log(n/D))$ was shown in [9], and independently in [16] for the special case when $D = O(1)$.

View-based approaches have been successfully used when designing algorithms for various network problems, for example in leader election [4, 7, 11, 12, 16, 27, 30]. In anonymous, port-labeled networks, the time to elect a leader or to declare the election infeasible is equal to $\Theta(D + l)$ [16] hence is $\Theta(D \log(n/D))$ [9, 20] for the worst-case port labeling. Our results show that for the best- and random-case labeling the time of leader election reduces to $O(\min(D, \log n))$.

View-based approaches have also been used for map construction [3, 10, 12], rendezvous [6, 8, 19], and other tasks [14, 28].

The result of Norris [24] is indeed more general as it works also for directed graphs and shows that in directed graphs the level of symmetry is at most $n - 1$. We may also consider the problem of distinguishing the views of two nodes from different graphs. We are interested in whether the isomorphism of two views truncated to a certain depth implies the isomorphism of the infinite views. Krebs and Verbitsky [23] showed that it is possible to construct two directed graphs, both with at most n nodes, such that the truncated views of two nodes from different graphs are isomorphic up to depth $(2 - o(1))n$.

In port-labeled graphs, the problem of setting the port labels to allow for quick exploration by a simple agent is well-studied. In most of the existing results, the considered agent is performing a so-called *basic walk* which consists in taking the port number $x + 1$ after entering a node via port number x (where $x + 1$ is taken modulo the degree of the node). It turns out that it is possible to find a port labeling such that an agent following the basic walk will explore the given graph in a number of rounds linear in n [5, 13, 17, 18, 21, 22]. The question whether a port labeling allowing the basic walk to visit all the vertices can be assigned by a mobile agent making only local changes was posed in [13]. Steinová [25] answered affirmatively by exhibiting an agent with $O(\log n)$ bits of memory and one droppable pebble that assigns such a labeling.

1.3 Preliminaries and notation

In this work, we consider anonymous port labeled (simple and undirected) networks $G = (V, E)$ (the terms graph and network are used interchangeably throughout) in which the nodes do not have identifiers and each edge $\{u, v\}$ has two integers assigned to its endpoints, called the *port numbers* at u and v , respectively. The port numbers are assigned in such a way that for each node v they are pairwise different and they form a consecutive set of integers $\{1, \dots, d\}$, where d is the number of neighbors of v in G .

The number of neighbors of v in G is called the *degree* of v and is denoted by $\deg_G(v)$. The *maximum degree* of G (taken over all nodes v) is denoted by $\Delta(G)$. The *diameter* of G , denoted by D , is the maximum (taken over all pairs of nodes u and v) length of a shortest path between u and v in G . For a node $v \in V$, we write (G, v) to denote the rooted graph G with root v . For any $V' \subseteq V$, we denote by $G[V']$ the subgraph of G induced by V' and by $G[U] \setminus G[U']$ graph $G[U]$ without the edges that belong to $G[U']$. The set of nodes at distance at most l from a node $v \in V$ is denoted by $N_l(v)$, with $N(v) = N_1(v)$. A rooted subgraph $\mathcal{B}_l(v)$ of all nodes and edges reachable from v in a walk of at most l steps, $\mathcal{B}_l(v) = (G[N_l(v)] \setminus G[N_l(v) \setminus N_{l-1}(v)], v)$, will be called the *radius- l ball* around v . We recall the definition of a view [29]. Let G be a graph, v be a node of G and let λ be a port labeling for G . Given any $l \geq 0$, the (*truncated*) *view up to level l* , $\mathcal{V}_l(v)$, is defined as follows. $\mathcal{V}_0(v)$ is a tree consisting of a single node x_0 . Then, $\mathcal{V}_{l+1}(v)$ is the port-labeled tree rooted at x_0 and constructed as follows. For every node v_i , $i \in \{1, \dots, \deg_G(v)\}$, adjacent to v in G there is a child x_i of x_0 in $\mathcal{V}_{l+1}(v)$ such that the port number at x_0 corresponding to edge $\{x_0, x_i\}$ equals $\lambda(v, v_i)$, and the port number at x_i corresponding to edge $\{x_0, x_i\}$ equals $\lambda(v_i, v)$. For each $i \in \{1, \dots, \deg_G(v)\}$ the node x_i is the root of the truncated view $\mathcal{V}_l(v_i)$. The *view* from v in G is the infinite port-labeled rooted tree $\mathcal{V}(v)$ such that $\mathcal{V}_l(v)$ is its truncation to level l , for each $l \geq 0$.

We will denote by $\mathcal{C}(G)$ the number of distinct labelings of a graph G . Notice that for each node v , its labeling can be defined independently of the other nodes in $\deg_G(v)!$ different ways, hence if the nodes of the graph were distinguishable then the number of distinct labelings would be equal to $\prod_{v \in V} \deg_G(v)!$. In our setting, nodes have no identifiers hence $\mathcal{C}(G) \leq \prod_{v \in V} \deg_G(v)!$.

Lemma 1.1. *The following bound holds for all $l > 1$ and all connected graphs G with more than 2 nodes:*

$$\mathcal{C}(\mathcal{B}_l(v)) \leq \mathcal{C}(G[N_l(v)]) \leq (\Delta(G[N_{l-1}(v)]))^{2^{|E(G[N_{l-1}(v))|}} \leq 2^{2^{|E(\mathcal{B}_l(v))|} \log_2 \Delta(\mathcal{B}_l(v))}. \quad (1)$$

Proof. From the definitions of $N_l(v)$ and $\mathcal{B}_l(v)$ we have, for any l , that $\mathcal{B}_{l-1}(v)$ is a subgraph of $G[N_{l-1}(v)]$, and $G[N_{l-1}(v)]$ is a subgraph of $\mathcal{B}_l(v)$. This shows the first and the last inequalities of (1).

Denote $H = G[N_{l-1}(v)]$. We have $\mathcal{C}(H) \leq \prod_{v \in H} \deg_H(v)!$. The middle inequality of (1) can be shown using the fact that $k! \leq k^k$, for all $k > 0$. \square

2 Lower bounds

We start by providing a lower bound based on the following observation. Consider a graph G with labeling λ . If, for some integer $l > 0$ and for a pair of distinct nodes $u, v \in V$, their views truncated to depth l are distinct, but their radius- l balls are isomorphic (i.e., $\mathcal{V}_l(u) \neq \mathcal{V}_l(v)$ and $\mathcal{B}_l(u) \equiv \mathcal{B}_l(v)$), then the labeling λ around nodes u and v , restricted to their radius- l balls, must be distinct, i.e., $\lambda[\mathcal{B}_l(u)] \neq \lambda[\mathcal{B}_l(v)]$. It follows that if the number of isomorphic radius- l balls around nodes of G exceeds the number of possible labelings of such balls, we must have $l(G) > l$. (Here, isomorphism of rooted graphs (H_1, v_1) and (H_2, v_2) is understood as an isomorphism between H_1 and H_2 which maps v_1 into v_2 .)

Lemma 2.1. *Given a graph G and $l > 0$, let $S \subseteq V$ be a subset of vertices such that all radius- l balls around nodes from S are isomorphic to some rooted graph B , $\mathcal{B}_l(v) \equiv B$. Then, $\mathcal{C}(B) < |S| \implies l(G) > l$. \square*

The above lemma immediately implies the following two corollaries, the first of which applies inequality (1).

Corollary 2.1. *Given a connected graph G and $l > 0$, let $S \subseteq V$ be a subset of vertices such that all radius- l balls around nodes from S are isomorphic to some rooted graph B , $\mathcal{B}_l(v) \equiv B$. Then, $|E(B)| \log_2 \Delta(B) < \frac{1}{2} \log_2 |S| \implies l(G) \geq l$. \square*

Corollary 2.2. *When G is a line or a ring, we have $l(G) \geq \frac{1}{2} \log_2 n - 2$. \square*

Proposition 2.1. *For any integers $n \geq 2$ and $D \leq \log_2 n - \log_2 \log_2 n$, there exists a graph G with n nodes and diameter D which satisfies $l(G) \geq \frac{1}{2}D - \frac{5}{2}$.*

Proof. Assume that D is even. If $D < 6$ then the claim is trivial hence we assume that $D \geq 6$. It can be easily verified that for $D \leq \log_2 n - \log_2 \log_2 n$ and $n \geq 2$ we have $\lfloor \frac{n-1}{D-2} \rfloor \geq 4$. We construct a wheel graph G from a cycle with $n-1$ nodes and a node v connected to $x = \lfloor \frac{n-1}{D-2} \rfloor$ almost-equidistant nodes s_1, s_2, \dots, s_x on the cycle (see Figure 1 for an example). We choose nodes s_1, s_2, \dots, s_x along the cycle in such a way that the distance between two consecutive nodes s_i and s_{i+1} (and between s_x and s_1) on the cycle is either $D-1$ or $D-2$. Let c_i be a middle node of each such path and let c_x be a middle node on the path between s_x and s_1 (if the path has $D-1$ edges then there are two middle nodes). Distance between any two (not

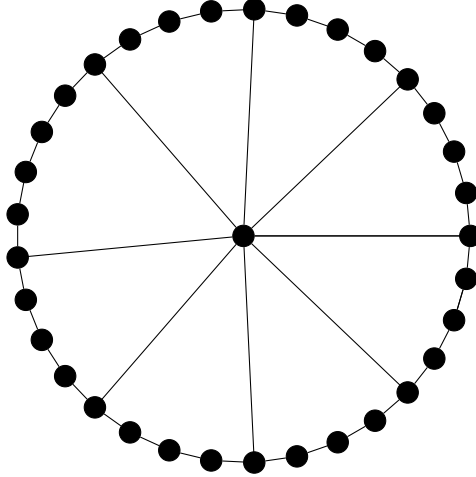


Figure 1: Illustration of wheel graph for $n = 34$ and $D = 6$.

consecutive) c_i and c_j is exactly D , because the shortest path goes via v and such not consecutive middle nodes exist because $x \geq 4$. Observe that the diameter of the graph is exactly D , because distance from v to any node is at most $D/2$. View from each c_i up to depth $D/2 - 2$ is a path with $D - 3$ nodes. There are 2^{D-4} possible labellings of such paths because at each vertex, different from c_i , we can either put port 1 towards c_i or away from c_i . Observe that

$$2^{D-4} \leq \frac{1}{16} 2^{\log_2 n - \log_2 \log_2 n} = \frac{n}{16 \log_2 n} < \left\lfloor \frac{n-1}{D-2} \right\rfloor.$$

Hence, by the pigeonhole principle, for at least two different nodes c_i, c_j their respective views up to depth $D/2 - 2$ are equal. Now observe that the view from the central node v is unique already at depth 1 hence by [20, Lemma 4], views from any pair of nodes is unique up to depth $D + 1$. This shows that $l(G) \geq D/2 - 2$.

If D is odd we construct the wheel graph on $n - 1$ nodes and diameter $D - 1$ and attach an extra node w to one of the middle nodes c . The construction of the wheel graph is possible because $\lfloor \frac{n-2}{D-3} \rfloor \geq 4$ holds true under our assumptions. We disregard the subpath to which we attach node w hence we obtain $\lfloor \frac{n-2}{D-3} \rfloor - 1$ paths of length $D - 4$. Similarly as in the previous case we have 2^{D-5} possible labelings of the paths and

$$2^{D-5} \leq \frac{1}{32} 2^{\log_2 n - \log_2 \log_2 n} = \frac{n}{32 \lg n} < \left\lfloor \frac{n-2}{D-3} \right\rfloor - 1.$$

Hence for odd D we obtain that $l(G) \geq (D - 1)/2 - 2$. □

3 Upper bounds

In this section we want to propose algorithms for labeling of general graphs that will guarantee that all views up to certain depths will be distinct.

We will start by describing a simple procedure **Greedy** for labeling the ports of the graph. We start with a non-empty set X of nodes such that for any pair of nodes $u, v \in X$, if there is an edge between u and v then it has already been labeled (i.e., both ends of the edge have been labeled). Procedure **Greedy** labels all the remaining edges.

Procedure Greedy: We pick any node $v \notin X$ and adjacent to some node from X . We label all the edges between v and the nodes from X (in any order) by choosing the smallest available label at both endpoints of such edges and we add v to X . Repeat until all edges are labeled.

Let us also define two invariants. We will show that procedure Greedy maintains those invariants.

Invariants ():* If at some step vertex $x \in X$ has α neighbors in X then it has port labels from 1 to α used to label the edges to these neighbors. The second invariant states that each $x \in X$ has at least one neighbor in X .

Lemma 3.1. *If set X satisfies invariants (*) then procedure Greedy used with X as the initial set, labels all the remaining edges without using label pair (1, 1) on any newly labeled edge.*

Proof. Since graph G is connected then by repeating procedure Greedy we label all the edges of the graph.

In a single step we pick a vertex y not in X , but adjacent to some vertices from X , we label all the edges connecting y to vertices from X using the first available port numbers hence the invariant is maintained. From the invariant, port pair (1, 1) will not be used on any edge labeled by the procedure Greedy. \square

Theorem 3.1. *For any graph G of diameter D , we have $l(G) \leq D + 1$. Moreover, a labeling λ such that $l(G, \lambda) \leq D + 1$ can be constructed in polynomial time.*

Proof. We want to first show a procedure assigning labels in any graph in such a way that there always exists a node with a unique view up to depth 1. The rest follows from [20, Lemma 4] because then each view up to depth $D + 1$ is unique.

Take any graph G with at least 2 edges. If G has a leaf (i.e., a node with degree 1) we assign label 1 to both endpoints of its unique incident edge. In this case we define set X to initially contain both endpoints of the edge.

On the other hand if G has no leaf then it has a cycle. We take this cycle (call it \mathcal{C}) and label both endpoints of its edges with port numbers 1 and 2 as in Figure 2. We fix an arbitrary node v from cycle \mathcal{C} and label its outgoing edges with label pairs (1, 1) and (2, 2). All other edges of cycle \mathcal{C} get labeled with label pairs (1, 2). If there are other edges in G between the nodes of the cycle their labels are chosen by simply taking the first available label. In this case set X is defined to initially contain all nodes of the cycle. Observe that in both cases all edges with both endpoints in X have already been labeled.

The remaining labels are chosen by repeating procedure Greedy.

Since invariants (*) are satisfied at the beginning of the procedure then procedure Greedy will not put port numbers (1, 1) on any newly labeled edge.

Now observe that in the first case the view from the leaf is unique and in the second case the view from v is unique already at depth 1. \square

For a given orientation \vec{G} of the edges of G , we denote by $\text{indeg}_{\vec{G}}(v)$ ($\text{outdeg}_{\vec{G}}(v)$) the number of oriented edges entering (leaving) a vertex v . We also denote by $\vec{N}_l(v) \subseteq N_l(v)$ the subset of nodes reachable from v by following an outward-oriented path of at most l edges, starting from node v . Likewise, the radius- l out-ball of v is defined as $\vec{B}_l(v) = (\vec{G}[\vec{N}_l(v)] \setminus \vec{G}[\vec{N}_l(v) \setminus \vec{N}_{l-1}(v)], v)$.

Lemma 3.2. *Let λ be a port labeling of G chosen uniformly at random. Then, for any pair of distinct nodes $u, v \in V$ and any orientation \vec{G} of G , we have under labeling λ :*

$$\Pr[\mathcal{V}_l(u) = \mathcal{V}_l(v)] \leq \left(\prod_{w \in \vec{N}_l(v)} \left((\text{indeg}_{\vec{B}_l(v)}(w) - 1)! \deg_G(w) \right) \right)^{-1/2}.$$

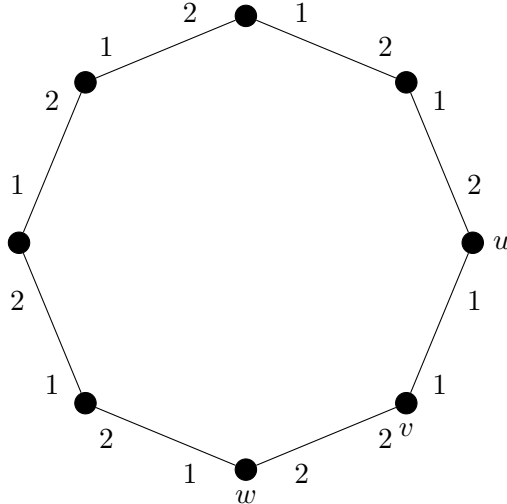


Figure 2: Labeling of cycle \mathcal{C} .

Proof. We consider a process in which the labels of all ports at all nodes in λ are treated as initially covered cards with numbers written on their backs. Each node of degree d hands out a perfectly shuffled set of d cards with numbers $\{1, \dots, d\}$ written on their backs to the d ports adjacent to it. We consider a process in which we sequentially consider edges of G (in some, possibly adaptive order), and in each time step uncover the cards at both ports of both endpoints of the edge. By $p_t(w)$ we denote the indegree of w with respect to covered incoming edges of \vec{G} , only, at the start of the t -th time step (initially, $p_1(w) = \text{indeg}_{\vec{\mathcal{B}}_i(v)}(w)$). By $q_t(w)$ we denote a variable defined as $q_t(w) = \frac{\text{deg}_G(w)}{\text{indeg}_{\vec{\mathcal{B}}_i(v)}(w)}$ if all edges adjacent to w are covered at time t , and $q_t(w) = 1$ otherwise. For any pair of nodes u, v and integer i we define a subgraph $\text{VPM}_u(\vec{\mathcal{B}}_i(v))$ as a “view preserving mapping” of $\vec{\mathcal{B}}_i(v)$. We build $\text{VPM}_u(\vec{\mathcal{B}}_i(v))$ in the following way. We take all directed paths of length i starting at v . For each such path we search for isomorphic with respect to λ path starting from u . Whenever such a path exists, we add its edges to $\text{VPM}_u(\vec{\mathcal{B}}_i(v))$. If for some path P from v there is no isomorphic path from u then we will say that the path corresponding to P is empty (in this case we immediately distinguish views of u and v at depth i). We denote by S_t the event that after t steps of the process the views from u and v are equal.

The process proceeds in l stages, and we assume that at the beginning of the i -th stage, $i = (1, 2, \dots, l)$, all edges from $\vec{\mathcal{B}}_i(v) \cup \text{VPM}_u(\vec{\mathcal{B}}_i(v))$ have been uncovered. In the i -th stage of the process, we sequentially consider edges connecting node pairs $\{w_{i-1}, w_i\}$, where $w_{i-1} \in \vec{N}_{i-1}(v)$, $w_i \in \vec{N}_i(v)$, such that the edge $\{w_{i-1}, w_i\} \in E(G)$ is oriented from w_{i-1} towards w_i in $E(G)$.

By the definition of our process, some shortest oriented path $\vec{P} = (v = w_0, w_1, \dots, w_{i-1}, w_i)$, with $w_j \in \vec{N}_j(v) \setminus \vec{N}_{j-1}(v)$, has already been uncovered save for its last edge $\{w_{i-1}, w_i\}$, and so has the unoriented path P' originating at u and isomorphic to the unoriented version of \vec{P} . If edge $\{w_{i-1}, w_i\}$ is still uncovered, we uncover it together with its (possibly uncovered) counterpart $\{w'_{i-1}, w'_i\}$ in path P . (Remark that if $w_i = w'_i$, then we have an immediate distinction of the two considered views.) We observe that if two edges are uncovered in the current step t , then we have:

$$\Pr[S_t] \leq \frac{1}{\max\{p_t(w_i)q_t(w_i), p_t(w'_i)q_t(w'_i)\}} \Pr[S_{t-1}]$$

with $p_{t+1}(w_i) = p_t(w_i) - 1$, $p_{t+1}(w'_i) = p_t(w'_i) - 1$, and $p_{t+1}(x) = p_t(x)$ for all other nodes x ; we also set $q_{t+1}(w_i) = q_{t+1}(w'_i) = 1$ and $q_{t+1}(x) = q_t(x)$ for all other nodes x . However, if edge $\{w'_{i-1}, w'_i\}$ was already previously uncovered, then we have:

$$\Pr[S_t] \leq \frac{1}{p_t(w_i)q_t(w_i)} \Pr[S_{t-1}]$$

with $p_{t+1}(w_i) = p_t(w_i) - 1$, and $p_{t+1}(x) = p_t(x)$ for all other nodes x ; we also set $q_{t+1}(w_i) = 1$ and $q_{t+1}(x) = q_t(x)$ for all other nodes x . Denoting $\Pi_t = \prod_{w \in \vec{\mathcal{B}}_l(v)} (p_t(w)!q_t(w))$, we notice that in both cases the following inequality holds:

$$\Pr[S_t] \leq \sqrt{\frac{\Pi_{t+1}}{\Pi_t}} \Pr[S_{t-1}]$$

By combining the above inequalities over all l phases of the process, in the final step m of the process we have $\Pi_m = 1$ and we eventually obtain:

$$\Pr[\mathcal{V}_l(u) = \mathcal{V}_l(v)] = \Pr[S_l] \leq \Pi_1^{-1/2},$$

which, taking into account the definition of Π_1 , p_1 , and q_1 , is exactly the claim of the lemma. \square

Choosing as \vec{G} a BFS-out-orientation from vertex v (with edges within BFS levels arbitrarily oriented), we have $\vec{N}_l(v) = N_l(v)$, and we obtain the following corollary that follows directly from Lemma 3.2.

Corollary 3.1. *Let λ be a port labeling of G chosen uniformly at random. Then, for any pair of distinct nodes $u, v \in V$, we have under labeling λ :*

$$\Pr[\mathcal{V}_l(u) = \mathcal{V}_l(v)] \leq \prod_{w \in N_l(v)} (\deg_G(w))^{-1/2}.$$

We use the above to show the following theorem.

Theorem 3.2. *There exists an absolute constant c such that, for any graph G with n nodes and diameter D , a uniformly random labeling λ satisfies $l(G, \lambda) \leq c \min\{D, \log_2 n\}$, w.h.p.¹*

Proof. Let us call any node with degree at least 2 a *non-leaf*. In our proof we will consider two cases. First assume that in G there are at least $6 \log_2 n$ non-leaves. Let $l = \min\{6 \log_2 n + 1, D\}$. For any v , in $\mathcal{B}_l(v)$ there are at least $6 \log_2 n$ non-leaves. Hence by Corollary 3.1, for any u, v we have $\Pr[\mathcal{V}_l(u) = \mathcal{V}_l(v)] \leq n^{-3}$. By taking a union bound over all pairs of vertices we get that with probability at least n^{-1} all pairs of vertices are distinguished within radius l .

Now, consider the case where in G there are less than $6 \log_2 n$ non-leaves. Then the diameter of G is at most $6 \log_2 n + 1$. We will first compute the probability that all pairs of non-leaves can be distinguished within radius D . Let us fix any pair of different non-leaves u, v and compute the probability that, under random labeling, their views are equal up to depth D . We will, similarly as in the proof of Lemma 3.2 expose the port labels sequentially. If $\deg_G u \neq \deg_G v$ then u and v are distinguished within radius 1. Among non-leaves there exists a node u' (possibly equal to u , but different from v) with degree at least $n/(6 \log_2 n) - 1$. Take the shortest path P from u to u' . At every edge of P expose the port label of the port closer to u (the other port remains covered). Consider a path P' that has the same port numbers on the exposed ports but it starts at v . If the views of u and v are to be equal, P' must lead to another vertex with degree at least

¹With high probability means here with probability at least $1 - O(\text{polylog} n/n)$.

$n/(6 \log_2 n) - 1$, call it v' . If $u' = v'$, the views are already distinguished by the properties of the port labeling. Otherwise observe that the last ports on paths P and P' have to be equal, and since there are at least $n/(6 \log_2 n) - 1$ possibilities, then it is equal with probability at most $6 \log_2 n / (n - 6 \log_2 n)$. Hence $\Pr[\mathcal{V}_l(u) = \mathcal{V}_l(v)] \leq 6 \log_2 n / (n - 6 \log_2 n)$. By taking the union bound over all non-leaves we get that with probability at least $1 - 108 \log_2^3 n / (n - 6 \log_2 n)$ all pairs of non-leaves are distinguished within radius D . This finishes the proof because the leaves can be distinguished using their unique neighbor (which is a non-leaf). \square

We remark that for a graph constructed from two equal stars connected with an edge, the probability that the centers are indistinguishable (to any depth) is $4/(n - 2)$. Hence the probability in our bound can be improved by at most a polylogarithmic factor.

The labelings given by Theorem 3.2 are non-constructive. By Theorem 3.1, we know how to constructively obtain a labeling λ satisfying $l(G, \lambda) = O(D)$. We now show how to constructively obtain $l(G, \lambda) = O(\log n)$. As a warmup exercise, we first perform the construction for the case when G is a path.

Proposition 3.1. *When G is a path, it is possible to construct in polynomial time a labeling λ such that $l(G, \lambda) = O(\log n)$.*

Proof. Encode the sequence of integers $1 * 2 * \dots *$ on the ports of the path in binary, using a bit coding convention: $0 = aa$, $1 = ab$, $* = abbabba$. Here, a means an edge with identical port labels at its endpoints, and b an edge with different ones.

We want to show that under such a labeling, within radius $l = 2 \log_2 n + 12$, each node has a unique view. We can see the view from a node as a string of a and b . Observe that if the view has depth l , then it contains at least two complete sequences $*$. Moreover, this sequence $*$ is the only place in the string that contains at least 2 consecutive b -s hence it can be identified. Since, sequence $*$ is not a palindrome, then it allows to determine the orientation of the string. The substring between the two sequences $*$ can then be correctly decoded as a number. From the value of the number and the position of the node with respect to the first node encoding this number, we can uniquely determine the position of the node on the path. Hence the view up to depth l distinguishes all the nodes of the path. \square

Using the result for paths we can construct for any graph a labeling that yields logarithmic level of symmetry.

Proposition 3.2. *For any graph G , it is possible to construct in polynomial time a labeling λ such that $l(G, \lambda) = O(\log n)$.*

Proof. If $D = O(\log n)$, then the statement follows from Theorem 3.1. We assume here that $D \geq 4(\log_2 n + 1)$.

Let $x = 2\lceil \log_2 n \rceil + 15$ and let S be a set of vertices in G that form a maximal independent set in G^{2x} . Such a set can be constructed in polynomial time. For any vertex v we can then find a vertex $s \in S$ that is at distance at most $2x$ to v . Moreover any two vertices $s, s' \in S$ are at distance more than $2x$. Each $s \in S$ can be seen a cluster center and the ball of radius $2x$ around s is its cluster. Note, that our clusters are not necessarily disjoint but each cluster center belongs to exactly one cluster and balls of radius x from each cluster center are disjoint. Hence, for each $s \in S$ we can assign a simple path P_s (starting in s) of length x in such a way that for any $s, s' \in S$, their paths P_s and $P_{s'}$ are disjoint. For each s , we will label ports along path P_s using only port labels 1 and 2. We pick an arbitrary order s_1, s_2, \dots, s_k of the cluster centers. For each s_i , we write its identifier i on the path P_{s_i} using port labels 1 and 2 using the same coding as in the proof of Proposition 3.1. On P_{s_i} we write $*i*$, where $* = abbabba$

and i is encoded in binary where $0 = aa$ and $1 = bb$. Here a means an edge with identical port endpoints, and b an edge with different ones. We label the path in such a way that the first a of each encoding is always a pair $(1, 1)$ (the encoding defines the remaining labels). If the encoding ends with pair $(2, 2)$, we label one more edge with port pair $(1, 1)$. The length of the whole encoding is then at most $2\lceil \log_2 n \rceil + 15$ hence the length of the path is sufficient. Let \bar{P}_{s_i} be the subpath of P_{s_i} containing exactly the edges used to encode the identifier $*i*$ (possibly together with the additional edge).

The remaining edges are labeled as follows. We start with a set X of all the nodes whose at least one adjacent edge has been labeled. First we label all edges connecting two vertices from X , by picking the smallest available port number at both endpoints. We will not use port 1 because it has been used along the paths. Now all the remaining edges are labeled using procedure **Greedy**. Observe that now set X satisfies invariants $(*)$ hence by using procedure **Greedy**, we can label the remaining edges without putting port pair $(1, 1)$ on any newly labeled edge.

Now we want to show that under such labeling, all nodes have distinct views up to depth $3x$. To show that all the views are distinct, for any node v we will find a *tag* i.e. a labeled path in the view of v that will not appear in the view of any other node. In the view up to depth $3x$, vertex v has at least one cluster center $s \in S$ and its whole path \bar{P}_s . In labeled ball $\mathcal{B}_{3x}(v)$ we identify the set \mathcal{P} of all paths induced by edges for which both endpoints were labeled using only ports 1 and 2. We keep in \mathcal{P} only those paths which start and end with an edge labeled with $(1, 1)$ and remove the other paths. By the construction of our labeling, edge labeled with $(1, 1)$ can only appear on a path \bar{P}_s of some cluster center s .

Now we remove from \mathcal{P} all the paths that have more than x edges. Finally we remove from \mathcal{P} the paths that are subpaths of other paths from \mathcal{P} . Each of the remaining paths $A \in \mathcal{P}$ is exactly \bar{P}_s for some cluster center s . Path A cannot contain more than one \bar{P} path because the length of A is at most x and the distance between the cluster centers is more than $2x$. Path A contains at least one \bar{P} because A starts and ends with an edge labeled with $(1, 1)$ and A is not a subpath of any \bar{P} path.

From each $A \in \mathcal{P}$ we can decode the identifier of the corresponding cluster center since $*$ is not a palindrome and hence we can deduce the direction of the path. If we obtained multiple identifiers of cluster centers for some starting vertex v , we simply pick the smallest one, call it s_i . The tag of v will then be the shortest path from v to s_i concatenated with \bar{P}_{s_i} (if there are multiple shortest paths, we choose the one lexicographically smallest under the chosen labeling). We conclude the proof by observing that if for v, u their tags contain the same \bar{P}_{s_i} then, by the properties of the port labeling, their paths to s_i have to be different. \square

4 Conclusion

In this paper, we have considered the problem of assigning labels to the ports of a graph G so that the views $\mathcal{V}_l(v)$ are distinct for all nodes $v \in V$, and such that l is minimized. We have shown that, for any graph G with n nodes and diameter D , a uniformly random labeling achieves $l = O(\min(D, \log n))$, w.h.p., and that it is possible to construct in polynomial time a labeling that satisfies $l = O(\min(D, \log n))$. In addition, we exhibited examples of graphs showing that these bounds are asymptotically tight in general.

An interesting direction of future work would be an analogue of the result of Steinová [25] in our setting. The goal is to propose a mechanism for the assignment (or reassignment) of the port labels by a mobile walker which starts in a graph with initially empty (or arbitrary) port labels, with the goal of obtaining a port labeling with small (best-case) level of symmetry. E.g., one could consider the implementation of the algorithm from Theorem 3.1 using a deterministic

walker.

Acknowledgements. The authors would like to thank Philippe Duchon and David Ilcinkas for proposing the problem, and for some initial discussions.

References

- [1] P. Boldi and S. Vigna. Fibrations of graphs. *Discrete Mathematics*, 243(1-3):21–66, 2002.
- [2] P. Boldi and S. Vigna. Universal dynamic synchronous self-stabilization. *Distributed Computing*, 15(3):137–153, 2002.
- [3] J. Chalopin, S. Das, and A. Kosowski. Constructing a map of an anonymous graph: Applications of universal sequences. In *Proc. 14th Int. Conf. on Principles of Distributed Systems (OPODIS 2010)*, volume 6490 of *LNCS*, pages 119–134. Springer, 2010.
- [4] J. Chalopin and Y. Métivier. An efficient message passing election algorithm based on Mazurkiewicz’s algorithm. *Fundamenta Informaticae*, 80(1-3):221–246, 2007.
- [5] J. Czyzowicz, S. Dobrev, L. Gasieniec, D. Ilcinkas, J. Jansson, R. Klasing, I. Lignos, R. Martin, K. Sadakane, and W. Sung. More efficient periodic traversal in anonymous undirected graphs. *Theor. Comput. Sci.*, 444:60–76, 2012.
- [6] J. Czyzowicz, A. Kosowski, and A. Pelc. How to meet when you forget: log-space rendezvous in arbitrary graphs. *Distributed Computing*, 25(2):165–178, 2012.
- [7] S. Das, P. Flocchini, A. Nayak, and N. Santoro. Effective elections for anonymous mobile agents. In *Proc. 17th Int. Symp. on Algorithms and Computation (ISAAC 2006)*, volume 4288 of *LNCS*, pages 732–743. Springer, 2006.
- [8] S. Das, M. Mihalák, R. Srámek, E. Vicari, and P. Widmayer. Rendezvous of mobile agents when tokens fail anytime. In *Proc. 12th Int. Conf. on Principles of Distributed Systems (OPODIS 2008)*, volume 5401 of *LNCS*, pages 463–480. Springer, 2008.
- [9] D. Dereniowski, A. Kosowski, and D. Pajak. Distinguishing views in symmetric networks: A tight lower bound. *Theor. Comput. Sci.*, 582:27–34, 2015.
- [10] D. Dereniowski and A. Pelc. Drawing maps with advice. *J. Parallel Distrib. Comput.*, 72(2):132–143, 2012.
- [11] D. Dereniowski and A. Pelc. Leader election for anonymous asynchronous agents in arbitrary networks. *Distributed Computing*, 27(1):21–38, 2014.
- [12] D. Dereniowski and A. Pelc. Topology recognition and leader election in colored networks. *Theor. Comput. Sci.*, 621:92–102, 2016.
- [13] S. Dobrev, J. Jansson, K. Sadakane, and W. Sung. Finding short right-hand-on-the-wall walks in graphs. In *Proc. 12th Int. Coll. on Structural Information and Communication Complexity (SIROCCO 2005)*, volume 3499 of *LNCS*, pages 127–139. Springer, 2005.
- [14] P. Flocchini, A. Roncato, and N. Santoro. Computing on anonymous networks with sense of direction. *Theor. Comput. Sci.*, 1-3(301):355–379, 2003.

- [15] P. Fraigniaud and A. Pelc. Decidability classes for mobile agents computing. In *Proc. 10th Latin American Symposium on Theoretical Informatics (LATIN 2012)*, volume 7256 of *LNCS*, pages 362–374. Springer, 2012.
- [16] E. G. Fusco and A. Pelc. Knowledge, level of symmetry, and time of leader election. *Distributed Computing*, 28(4):221–232, 2015.
- [17] L. Gasieniec, R. Klasing, R. A. Martin, A. Navarra, and X. Zhang. Fast periodic graph exploration with constant memory. *J. Comput. Syst. Sci.*, 74(5):808–822, 2008.
- [18] L. Gasieniec and T. Radzik. Memory efficient anonymous graph exploration. In *Proc. 34th Int. Workshop on Graph-Theoretic Concepts in Computer Science (WG 2008)*, volume 5344 of *LNCS*, pages 14–29, 2008.
- [19] S. Guilbault and A. Pelc. Asynchronous rendezvous of anonymous agents in arbitrary graphs. In *Proc. 15th Int. Conf. on Principles of Distributed Systems (OPODIS 2011)*, volume 7109 of *LNCS*, pages 421–434. Springer, 2011.
- [20] J. M. Hendrickx. Views in a graph: To which depth must equality be checked? *IEEE Trans. Parallel Distrib. Syst.*, 25(7):1907–1912, 2014.
- [21] D. Ilcinkas. Setting port numbers for fast graph exploration. *Theor. Comput. Sci.*, 401(1-3):236–242, 2008.
- [22] A. Kosowski and A. Navarra. Graph decomposition for memoryless periodic exploration. *Algorithmica*, 63(1-2):26–38, 2012.
- [23] A. Krebs and O. Verbitsky. Universal covers, color refinement, and two-variable counting logic: Lower bounds for the depth. In *Proc. 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS 2015)*, pages 689–700. IEEE, 2015.
- [24] N. Norris. Universal covers of graphs: Isomorphism to depth $N - 1$ implies isomorphism to all depths. *Discrete Applied Mathematics*, 56(1):61–74, 1995.
- [25] M. Steinová. On the power of local orientations. In *Proc. 15th Int. Coll. on Structural Information and Communication Complexity (SIROCCO 2008)*, volume 5058 of *LNCS*, pages 156–169. Springer, 2008.
- [26] S. Tani. Compression of view on anonymous networks — folded view. *IEEE Trans. Parallel Distrib. Syst.*, 23(2):255–262, 2012.
- [27] S. Tani, H. Kobayashi, and K. Matsumoto. Exact quantum algorithms for the leader election problem. *ACM Transactions on Computation Theory*, 4(1):1, 2012.
- [28] M. Yamashita and T. Kameda. Computing functions on asynchronous anonymous networks. *Mathematical Systems Theory*, 29(4):331–356, 1996.
- [29] M. Yamashita and T. Kameda. Computing on anonymous networks: Part I-characterizing the solvable cases. *IEEE Transactions on Parallel and Distributed Systems*, 7(1):69–89, 1996.
- [30] M. Yamashita and T. Kameda. Leader election problem on networks in which processor identity numbers are not distinct. *IEEE Trans. Parallel Distrib. Syst.*, 10(9):878–887, 1999.