Lower Bounds for Local Approximation

Mika Göös
mika.goos@cs.helsinki.fi

Juho Hirvonen
juho.hirvonen@cs.helsinki.fi

Jukka Suomela
jukka.suomela@cs.helsinki.fi

Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki
P.O. Box 68, FI-00014 University of Helsinki, Finland

ABSTRACT

In the study of deterministic distributed algorithms it is commonly assumed that each node has a unique $O(\log n)$-bit identifier. We prove that for a general class of graph problems, local algorithms (constant-time distributed algorithms) do not need such identifiers: a port numbering and orientation is sufficient.

Our result holds for so-called *simple* PO*-checkable graph optimisation problems*; this includes many classical packing and covering problems such as vertex covers, edge covers, matchings, independent sets, dominating sets, and edge dominating sets. We focus on the case of bounded-degree graphs and show that if a local algorithm finds a constant-factor approximation of a simple PO-checkable graph problem with the help of unique identifiers, then the same approximation ratio can be achieved on anonymous networks.

As a corollary of our result and by prior work, we derive a tight lower bound on the local approximability of the *minimum edge dominating set problem*.

Our main technical tool is an algebraic construction of *homogeneously ordered graphs*: We say that a graph is (α, r)-homogeneous if its nodes are linearly ordered so that an α fraction of nodes have pairwise isomorphic radius-r neighbourhoods. We show that there exists a finite (α, r)-homogeneous $2k$-regular graph of girth at least g for any $\alpha < 1$ and any r, k, and g.

Categories and Subject Descriptors

C.2.4 [**Computer-Communication Networks**]: Distributed Systems; F.1.3 [**Computation by Abstract Devices**]: Complexity Measures and Classes; F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Nonnumerical Algorithms and Problems—*computations on discrete structures*

Keywords

approximation algorithms, deterministic distributed algorithms, edge dominating set, local algorithms, unique identifiers

© ACM, 2012. This is the authors' version of the work. It is posted here by permission of ACM for your personal use. Not for redistribution. The definitive version was published in Proc. 31st Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC'12, July 16–18, 2012, Madeira, Portugal).
http://doi.acm.org/10.1145/2332432.2332465

1. INTRODUCTION

In this work, we study deterministic distributed algorithms under three different assumptions; see Figure 1a for illustrations.

(ID) *Networks with unique identifiers.* Each node is given a unique $O(\log n)$-bit label.

(OI) *Order-invariant algorithms.* There is a linear order on nodes.

Equivalently, the nodes have unique labels, but the output of an algorithm is not allowed to change if we relabel the nodes while preserving the relative order of the labels.

(PO) *Anonymous networks with a port numbering and orientation.* For each node, there is a linear order on the incident edges, and for each edge, there is a linear order on the incident nodes.

Equivalently, a node of degree d can refer to its neighbours by integers $1, 2, \ldots, d$, and each edge is oriented so that the endpoints know which of them is the head and which is the tail.

While unique identifiers are often useful, we will show that they are seldom needed in local algorithms (constant-time distributed algorithms): there is a general class of graph problems such that local algorithms in PO are able to produce as good approximations as local algorithms in OI or ID.

1.1 Graph Problems

We study graph problems that are related to the structure of an unknown communication network. Each node in the network is a computer; each computer receives a *local input*, it can exchange messages with adjacent nodes, and eventually it has to produce a *local output*. The local outputs constitute a solution of a graph problem—for example, if we study the dominating set problem, each node produces one bit of local output, indicating whether it is part of the dominating set. The *running time* of an algorithm is the number of synchronous communication rounds.

From this perspective, the models ID, OI, and PO are easy to separate. Consider, for example, the problem of finding a maximal independent set in an n-cycle. In ID model the problem can be solved in $\Theta(\log^* n)$ rounds [6, 18], while in OI model we need $\Theta(n)$ rounds, and the problem is not solvable at all in PO, as we cannot break symmetry—see Figure 1b. Hence ID is strictly stronger than OI, which is strictly stronger than PO.

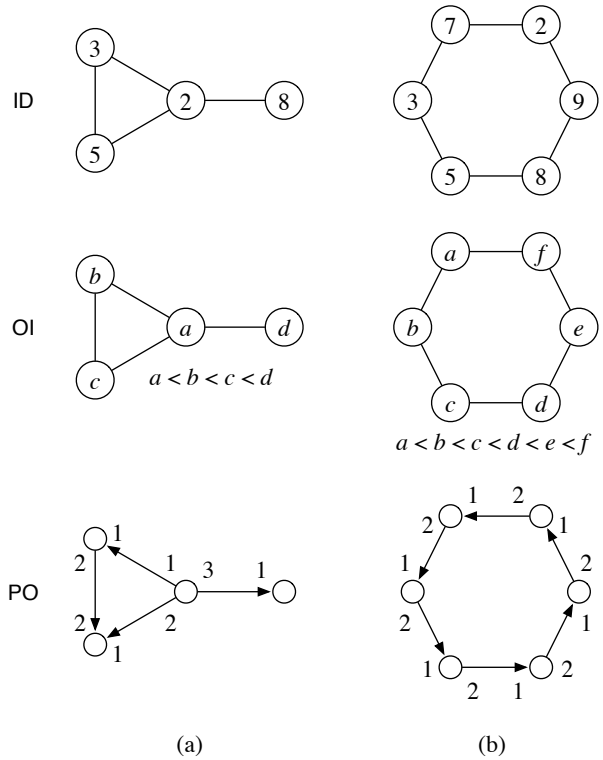

Figure 1: (a) Three models of distributed computing. (b) In model ID, the numerical identifiers break symmetry everywhere—for example, in a cycle, a maximal independent set can be found in $O(\log^* n)$ rounds. In model OI, we can have a cycle with only one "seam", and in model PO we can have a completely symmetric cycle.

1.2 Local Algorithms

In this work we focus on *local algorithms*, i.e., distributed algorithms that run in a constant number of synchronous communication rounds, independently of the number of nodes in the network [20, 23]. The above example separating ID, OI, and PO no longer applies, and there has been a conspicuous lack of *natural* graph problems that would separate ID, OI, and PO from the perspective of local algorithms.

Indeed, there are results that show that many problems that can be solved with a local algorithm in ID also admit a local algorithm in OI or PO. For example, the seminal paper by Naor and Stockmeyer [20] studies so-called *locally checkable labellings*, or LCL problems for short—these include problems such as graph colouring and maximal matchings on bounded-degree graphs. The authors show that ID and OI are indeed equally expressive among LCL problems. The followup work by Mayer, Naor, and Stockmeyer [19] hints of a stronger property:

(i) *Weak* 2*-colouring* is an LCL problem that can be solved with a local algorithm in ID model [20]. It turns out that the same problem can be solved in PO model as well [19].

Granted, contrived counterexamples do exist: there are LCL problems that are solvable in OI but not in PO. However, most of the classical graph problems that are studied in the field of distributed computing are *optimisation problems*, not LCL problems.

1.3 Local Approximation

In what follows, we will focus on graph problems in the case of *bounded-degree graphs*; that is, there is a known constant Δ such that the degree of any node in any graph that we may encounter is at most Δ. Parity often matters; hence we also define $\Delta' = 2\lfloor \Delta/2 \rfloor$.

In this setting, the best possible approximation ratios are surprisingly similar in ID, OI, and PO. The following hold for any given $\Delta \geq 2$ and $\epsilon > 0$:

(ii) *Minimum vertex cover* can be approximated to within factor 2 in each of these models [3, 5]. This is tight: $(2-\epsilon)$-approximation is not possible in any of these models [9, 17, 23].

(iii) *Minimum edge cover* can be approximated to within factor 2 in each of these models [23]. This is tight: $(2-\epsilon)$-approximation is not possible in any of these models [9, 17, 23].

(iv) *Minimum dominating set* can be approximated to within factor $\Delta' + 1$ in each of these models [4]. This is tight: $(\Delta' + 1 - \epsilon)$-approximation is not possible in any of these models [9, 17, 23].

(v) *Maximum independent set* cannot be approximated to within any constant factor in any of these models [9, 17].

(vi) *Maximum matching* cannot be approximated to within any constant factor in any of these models [9, 17].

This phenomenon has not been fully understood: while there are many problems with identical approximability results for ID, OI, and PO, it has not been known whether these are examples of a more general principle or merely isolated coincidences. In fact, for some problems, tight approximability results have been lacking for ID and OI, even though tight results are known for PO:

(vii) *Minimum edge dominating set* can be approximated to within factor $4 - 2/\Delta'$ in each of these models [22]. This is tight for PO but only near-tight for ID and OI: $(4 - 2/\Delta' - \epsilon)$-approximation is not possible in PO [22], and $(3-\epsilon)$-approximation is not possible in ID and OI [9, 17, 23].

In this work we prove a theorem unifying all of the above observations—they are indeed examples of a general principle. As a simple application of our result, we settle the local approximability of the minimum edge dominating set problem by proving a tight lower bound in ID and OI.

1.4 Main Result

A *simple graph problem* Π is an optimisation problem in which a feasible solution is a subset of nodes or a subset of edges, and the goal is to either minimise or maximise the size of a feasible solution. We say that Π is a PO-*checkable graph problem* if there is a local PO-algorithm A that recognises a feasible solution. That is, $\mathsf{A}(\mathcal{G}, X, v) = 1$ for all nodes $v \in V(\mathcal{G})$ if X is a feasible solution of problem Π in graph $\mathcal{G}$, and $\mathsf{A}(\mathcal{G}, X, v) = 0$ for some node $v \in V(\mathcal{G})$ otherwise—here

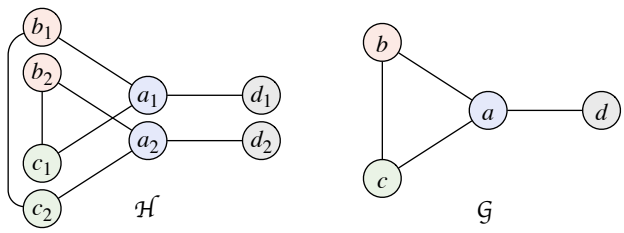

Figure 2: Graph $\mathcal{H}$ is a lift of $\mathcal{G}$. The covering map $\varphi\colon V(\mathcal{H}) \to V(\mathcal{G})$ maps $a_i \mapsto a$, $b_i \mapsto b$, $c_i \mapsto c$, and $d_i \mapsto d$ for each $i = 1, 2$. The fibre of $a \in V(\mathcal{G})$ is $\{a_1, a_2\} \subseteq V(\mathcal{H})$; all fibres have the same size.

$\mathsf{A}(\mathcal{G}, X, v)$ is the output of a node v if we run algorithm A on graph $\mathcal{G}$ and the local inputs form an encoding of X.

Let $\varphi\colon V(\mathcal{H}) \to V(\mathcal{G})$ be a surjective graph homomorphism from graph $\mathcal{H}$ to graph $\mathcal{G}$. If φ preserves vertex degrees, i.e., $\deg_{\mathcal{H}}(u) = \deg_{\mathcal{G}}(\varphi(u))$, then φ is called a *covering map*, and $\mathcal{H}$ is said to be a *lift* of $\mathcal{G}$. The *fibre* of $u \in V(\mathcal{G})$ is the set $\varphi^{-1}(u)$ of pre-images of u. We usually consider n-lifts that have fibres of the same cardinality n. It is a basic fact that a connected lift $\mathcal{H}$ of $\mathcal{G}$ is an n-lift for some n. See Figure 2 for an illustration.

Let $\mathcal{F}$ be a family of graphs. We say that $\mathcal{F}$ is *closed under lifts* if $\mathcal{G} \in \mathcal{F}$ implies $\mathcal{H} \in \mathcal{F}$ for all lifts $\mathcal{H}$ of $\mathcal{G}$. A family is *closed under connected lifts* if $\mathcal{G} \in \mathcal{F}$ implies $\mathcal{H} \in \mathcal{F}$ whenever $\mathcal{H}$ and $\mathcal{G}$ are connected graphs and $\mathcal{H}$ is a lift of $\mathcal{G}$.

Now we are ready to state our main theorem.

THEOREM 1 (MAIN THEOREM). *Let Π be a simple* PO*-checkable graph problem. Assume one of the following:*

- *General version: $\mathcal{F}$ is a family of bounded degree graphs, and it is closed under lifts.*
- *Connected version: $\mathcal{F}$ is a family of connected bounded degree graphs, it does not contain any trees, and it is closed under connected lifts.*

If there is a local ID*-algorithm* A *that finds an α-approximation of Π in $\mathcal{F}$, then there is a local* PO*-algorithm* B *that finds an α-approximation of Π in $\mathcal{F}$.*

While the definitions are somewhat technical, it is easy to verify that the result is widely applicable:

(a) Vertex covers, edge covers, matchings, independent sets, dominating sets, and edge dominating sets are simple PO-checkable graph problems.

(b) Bounded-degree graphs, regular graphs, and cyclic graphs are closed under lifts.

(c) Connected bounded-degree graphs, connected regular graphs, and connected cyclic graphs are closed under connected lifts.

1.5 An Application

The above result provides us with a powerful tool for proving lower-bound results: we can easily transfer negative results from PO to OI and ID. We demonstrate this strength by deriving a new lower bound result for the minimum edge dominating set problem.

THEOREM 2. *Let $\Delta \geq 2$, and let* A *be a local* ID*-algorithm that finds an α-approximation of a minimum edge dominating set on connected graphs of maximum degree Δ. Then $\alpha \geq \alpha_0$, where $\alpha_0 = 4 - 2/\Delta'$ and $\Delta' = 2\lfloor \Delta/2 \rfloor$. This is tight: there is a local* ID*-algorithm that finds an α_0-approximation.*

PROOF. By prior work [22], it is known that there is a connected Δ'-regular graph $\mathcal{G}_0$ such that the approximation factor of any local PO-algorithm on $\mathcal{G}_0$ is at least α_0. Let $\mathcal{F}_0$ consist of all connected lifts of $\mathcal{G}_0$, and let $\mathcal{F}$ consist of all connected graphs of degree at most Δ. We make the following observations.

(a) We have $\mathcal{F}_0 \subseteq \mathcal{F}$; by assumption, A finds an α-approximation in $\mathcal{F}_0$.

(b) Family $\mathcal{F}_0$ consists of connected graphs of degree at most Δ. As $\mathcal{G}_0$ is not a tree, family $\mathcal{F}_0$ does not contain any trees. Moreover, $\mathcal{F}_0$ is by construction closed under connected lifts. Hence we can apply the connected version of the main theorem: there is a local PO-algorithm B that finds an α-approximation in $\mathcal{F}_0$.

(c) However, $\mathcal{G}_0 \in \mathcal{F}_0$, and hence $\alpha \geq \alpha_0$.

The matching upper bound is presented in prior work [22]. □

1.6 Overview

Informally, our proof of the main theorem is structured as follows.

(a) Fix a graph problem Π, a graph family $\mathcal{F}$, and an ID-algorithm A as in the statement of Theorem 1. Let r be the running time of ID-algorithm A.

(b) Let $\mathcal{G} \in \mathcal{F}$ be a graph with a port numbering and orientation.

(c) Section 3.2: We construct a certain lift $\mathcal{G}_\epsilon \in \mathcal{F}$ of $\mathcal{G}$. Graph $\mathcal{G}_\epsilon$ inherits the port numbering and the orientation from $\mathcal{G}$.

(d) Section 4.1: We show that there exists a linear order $<_\epsilon$ on the nodes of $\mathcal{G}_\epsilon$ that gives virtually no new information in comparison with the port numbering and orientation. If we have an OI-algorithm A' with running time r, then we can simulate A' with a PO-algorithm B' almost perfectly on $\mathcal{G}_\epsilon$: the outputs of A' and B' agree for a $(1 - \epsilon)$ fraction of nodes. We deduce that the approximation ratio of A' on $\mathcal{F}$ cannot be better than the approximation ratio of B' on $\mathcal{F}$.

(e) Section 4.2: We apply Ramsey's theorem to show that the unique identifiers do not help, either. We can construct a PO-algorithm B that simulates A in the following sense: there exists an assignment of unique identifiers on a lift $\mathcal{H} \in \mathcal{F}$ of $\mathcal{G}_\epsilon$ such that the outputs of A and B agree for a $(1 - \epsilon)$ fraction of nodes. We deduce that the approximation ratio of A on $\mathcal{F}$ cannot be better than the approximation ratio of B on $\mathcal{F}$.

Now if graph $\mathcal{G}$ was a directed cycle, the construction would be standard; see, e.g., Czygrinow et al. [9]. In particular, $\mathcal{G}_\epsilon$ and $\mathcal{H}$ would simply be long cycles, and $<_\epsilon$ would order the nodes along the cycle—there would be only one "seam" in $(\mathcal{G}_\epsilon, <_\epsilon)$ that could potentially help A' in comparison with B', and only an ϵ fraction of nodes are near the seam.

However, the case of a general $\mathcal{G}$ is more challenging. Our main technical tool is the construction of so-called homogeneous graphs; see Section 3.1. Homogeneous graphs are regular graphs with a linear order that is useless from the perspective of OI-algorithms: for a $(1-\epsilon)$ fraction of nodes, the local neighbourhoods are isomorphic. Homogeneous graphs trivially exist; however, our proof calls for homogeneous graph of an arbitrarily high degree and an arbitrarily large girth (i.e., there are no short cycles—the graph is locally tree-like). In Section 5 we use an algebraic construction to prove that such graphs exist.

1.7 Discussion

In the field of distributed algorithms, the running time of an algorithm is typically analysed in terms of two parameters: n, the number of nodes in the graph, and Δ, the maximum degree of the graph. In our work, we assumed that Δ is a constant—put otherwise, our work applies to algorithms that have a running time independent of n but arbitrarily high as a function of Δ. The work by Kuhn et al. [13–15] studies the dependence on Δ more closely: their lower bounds on approximation ratios apply to algorithms that have, for example, a running time sublogarithmic in Δ.

While our result is very widely applicable, certain extensions have been left for future work. One example is the case of planar graphs [9], [16, §13]. The family of planar graphs is not closed under lifts, and hence Theorem 1 does not apply. Another direction that we do not discuss at all is the case of randomised algorithms; if each node has access to a stream of random bits, the distinction between ID, OI, and PO essentially vanishes, as the random bits can be used to generate unique identifiers w.h.p.

2. THREE MODELS OF DISTRIBUTED COMPUTING

In this section we make precise the notion of a *local algorithm* in each of the models ID, OI and PO. First, we discuss the properties common to all the models.

We start by fixing a graph family $\mathcal{F}$ where every $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G})) \in \mathcal{F}$ has maximum degree at most $\Delta \in \mathbb{N}$. We consider algorithms A that operate on graphs in $\mathcal{F}$; the properties of A (e.g., its running time) are allowed to depend on the family $\mathcal{F}$ (and, hence, on Δ). We denote by $\mathsf{A}(\mathcal{G}, u) \in \Omega$ the output of A on a node $u \in V(\mathcal{G})$. Here, Ω is a finite set of possible outputs of A in $\mathcal{F}$. If the solutions to Π are sets of vertices, we shall have $\Omega = \{0, 1\}$ so that the solution produced by A on $\mathcal{G}$, denoted $\mathsf{A}(\mathcal{G})$, is the set of nodes u with $\mathsf{A}(\mathcal{G}, u) = 1$. Similarly, if the solutions to Π are sets of edges, we shall have $\Omega = \{0, 1\}^{\Delta}$ so that the ith component of the vector $\mathsf{A}(\mathcal{G}, u)$ indicates whether the ith edge incident to u is included in the solution $\mathsf{A}(\mathcal{G})$—in each of the models a node will have a natural ordering of its incident edges.

Let $r \in \mathbb{N}$ denote the constant running time of A in $\mathcal{F}$. This means that a node u can only receive messages from nodes within distance r in $\mathcal{G}$, i.e., from nodes in the radius-r neighbourhood

$$B_{\mathcal{G}}(u, r) = \big\{v \in V(\mathcal{G}) : \mathrm{dist}_{\mathcal{G}}(u, v) \leq r\big\}.$$

Let $\tau(\mathcal{G}, u)$ denote the structure $(\mathcal{G}, u)$ restricted to the vertices $B_{\mathcal{G}}(u, r)$, i.e., in symbols, $\tau(\mathcal{G}, u) = (\mathcal{G}, u) \restriction B_{\mathcal{G}}(u, r)$. Then $\mathsf{A}(\mathcal{G}, u)$ is a function of the data $\tau(\mathcal{G}, u)$ in that

$$\mathsf{A}(\mathcal{G}, u) = \mathsf{A}(\tau(\mathcal{G}, u)).$$

The models ID, OI and PO impose further restrictions on this function.

2.1 Model ID

Local ID-algorithms are not restricted in any additional way. We follow the convention that the vertices have unique $O(\log n)$-bit labels, i.e., an instance $\mathcal{G} \in \mathcal{F}$ of order $n = |V(\mathcal{G})|$ has $V(\mathcal{G}) \subseteq \{1, 2, \ldots, s(n)\}$ where $s(n)$ is some fixed polynomial function of n. Our presentation assumes $s(n) = \omega(n)$, even though this assumption can often be relaxed as we discuss in Remark 1.

2.2 Model OI

A local OI-algorithm A does not directly use unique vertex identifiers but only their relative *order*. To make this notion explicit, let the vertices of $\mathcal{G} \in \mathcal{F}$ be linearly ordered by $<$, and call $(\mathcal{G}, <)$ an *ordered graph*. Denote by $\tau(\mathcal{G}, <, u)$ the restriction of the structure $(\mathcal{G}, <, u)$ to the r-neighbourhood $B_{\mathcal{G}}(u, r)$, i.e., in symbols,

$$\tau(\mathcal{G}, <, u) = (\mathcal{G}, <, u) \restriction B_{\mathcal{G}}(u, r).$$

Then, the output $\mathsf{A}(\mathcal{G}, <, u)$ depends only on the *isomorphism type* of $\tau(\mathcal{G}, <, u)$, so that if $\tau(\mathcal{G}, <, u) \simeq \tau(\mathcal{G}', <', u')$ then $\mathsf{A}(\mathcal{G}, <, u) = \mathsf{A}(\mathcal{G}', <', u')$.

2.3 Model PO

In the PO model the nodes are considered anonymous and only the following node specific structure is available: a node can communicate with its neighbours through ports numbered $1, 2, \ldots, \deg(u)$, and each communication link has an orientation.

2.3.1 Edge-Labelled Digraphs

To model the above, we consider *L-edge-labelled directed graphs* (or *L-digraphs*, for short) $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G}), \ell_{\mathcal{G}})$, where the edges $E(\mathcal{G}) \subseteq V(\mathcal{G}) \times V(\mathcal{G})$ are directed and each edge $e \in E(\mathcal{G})$ carries a label $\ell_{\mathcal{G}}(e) \in L$. We restrict our considerations to *proper* labellings $\ell_{\mathcal{G}} \colon E(\mathcal{G}) \to L$ that for each $u \in V(\mathcal{G})$ assign the incoming edges $(v, u) \in E(\mathcal{G})$ distinct labels and the outgoing edges $(u, w) \in E(\mathcal{G})$ distinct labels; we allow $\ell_{\mathcal{G}}(v, u) = \ell_{\mathcal{G}}(u, w)$. We refer to the outgoing edges of a node by the labels L and to the incoming edges by the formal letters $L^{-1} = \{\ell^{-1} : \ell \in L\}$. In the context of L-digraphs, covering maps $\varphi \colon V(\mathcal{H}) \to V(\mathcal{G})$ are required to preserve edge labels so that $\ell_{\mathcal{H}}(u, v) = \ell_{\mathcal{G}}(\varphi(u), \varphi(v))$ for all $(u, v) \in E(\mathcal{H})$.

A port numbering on $\mathcal{G}$ gives rise to a proper labelling $\ell_{\mathcal{G}}(v, u) = (i, j)$, where u is the ith neighbour of v, and v is the jth neighbour of u; see Figure 3. We now fix L to contain every possible edge label that appears when a graph $\mathcal{G} \in \mathcal{F}$ is assigned a port numbering and an orientation. Note that $|L| \leq \Delta^2$.

2.3.2 Views

The information available to a PO-algorithm computing on a node $u \in V(\mathcal{G})$ in an L-digraph $\mathcal{G}$ is usually modelled as follows [1, 23, 24]. The *view* of $\mathcal{G}$ from u is an L-edge-labelled rooted (possibly infinite) directed tree $\mathcal{T} = \mathcal{T}(\mathcal{G}, u)$, where the vertices $V(\mathcal{T})$ correspond to all non-backtracking walks on $\mathcal{G}$ starting at u; see Figure 3c. Formally, a k-step walk can be identified with a word of length k in the letters $L \cup L^{-1}$. A non-backtracking walk is a *reduced* word where neither $\ell\ell^{-1}$ nor $\ell^{-1}\ell$ appear. If $w \in V(\mathcal{T})$ is a walk on $\mathcal{G}$ from u to v, we define $\varphi(w) = v$. In particular, the root of $\mathcal{T}$ is the

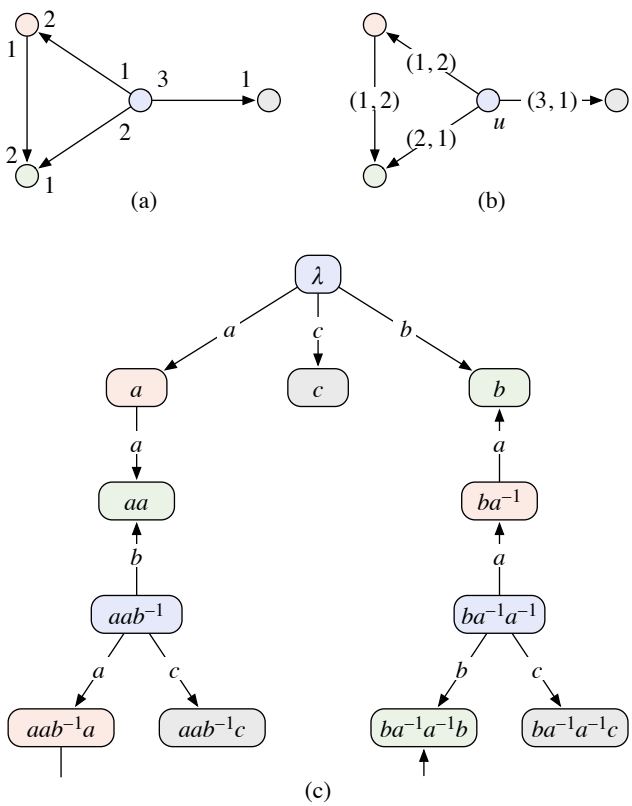

Figure 3: (a) A graph $\mathcal{G}$ with a port numbering and an orientation. (b) A proper labelling $\ell_{\mathcal{G}}$ that is derived from the port numbering. We have an L-digraph with $L = \{a, b, c\}$, $a = (1, 2)$, $b = (2, 1)$, and $c = (3, 1)$. (c) The view of $\mathcal{G}$ from u is an infinite directed tree $\mathcal{T} = \mathcal{T}(\mathcal{G}, u)$; there is a covering map φ from $\mathcal{T}$ to $\mathcal{G}$ that preserves adjacencies, orientations, and edge labels. For example, $\varphi(\lambda) = \varphi(aab^{-1}) = u$.

empty word λ with $\varphi(\lambda) = u$. The directed edges of $\mathcal{T}$ (and their labels) are defined in such a way that $\varphi\colon V(\mathcal{T}) \to V(\mathcal{G})$ becomes a covering map. Namely, $w \in V(\mathcal{T})$ has an out-neighbour $w\ell$ for every $\ell \in L$ such that $\varphi(w)$ has a outgoing edge labelled ℓ.

2.3.3 *Local* PO*-Algorithms*

The inability of a PO-algorithm B to detect cycles in a graph is characterised by the fact that $\mathsf{B}(\mathcal{G}, u) = \mathsf{B}(\mathcal{T}(\mathcal{G}, u))$. In fact, we *define* a local PO-algorithm as a function B satisfying

$$\mathsf{B}(\mathcal{G}, u) = \mathsf{B}(\tau(\mathcal{T}(\mathcal{G}, u))).$$

An important consequence of this definition is that the output of a PO-algorithm is invariant under lifts, i.e., if $\varphi\colon V(\mathcal{H}) \to V(\mathcal{G})$ is a covering map of L-digraphs, then $\mathsf{B}(\mathcal{H}, u) = \mathsf{B}(\mathcal{G}, \varphi(u))$. The intuition is that nodes in a common fibre are always in the same state during computation as they see the same view.

The following formalism will become useful. Denote by $(\mathcal{T}^*, \lambda)$ the complete L-labelled rooted directed tree of radius r with $V(\mathcal{T}^*)$ consisting of reduced words in the letters $L \cup L^{-1}$, i.e., every non-leaf vertex in $\mathcal{T}^*$ has an outgoing edge and an incoming edge for each $\ell \in L$; see Figure 4. The

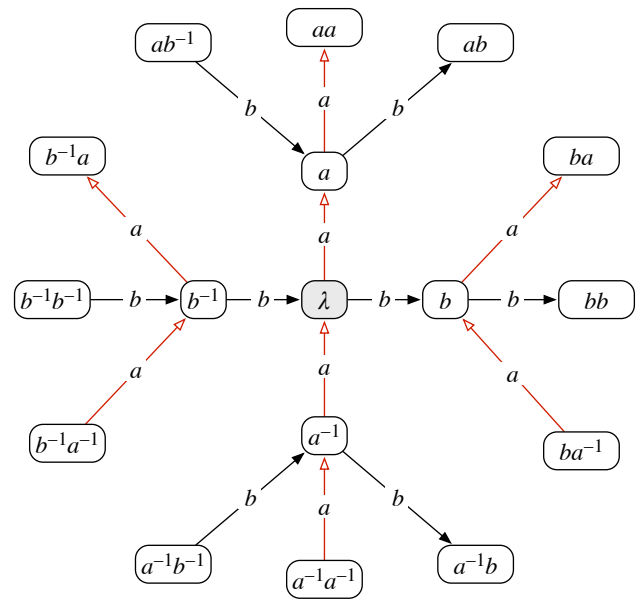

Figure 4: The complete L-labelled rooted directed tree $(\mathcal{T}^*, \lambda)$ of radius $r = 2$, for $L = \{a, b\}$.

output of B on every graph $\mathcal{G} \in \mathcal{F}$ is completely determined after specifying its output on the subtrees of $(\mathcal{T}^*, \lambda)$. More precisely, let $\mathfrak{W}$ consist of vertex sets $W \subseteq V(\mathcal{T}^*)$ such that $(\mathcal{T}^*, \lambda) \restriction W = \tau(\mathcal{T}(\mathcal{G}, u))$ for some $\mathcal{G} \in \mathcal{F}$ and $u \in V(\mathcal{G})$. Then a function $\mathsf{B}\colon \mathfrak{W} \to \Omega$ defines a PO-algorithm by identifying

$$\mathsf{B}((\mathcal{T}^*, \lambda) \restriction W) = \mathsf{B}(W).$$

3. ORDER HOMOGENEITY

In this section we introduce some key concepts that are used in controlling the local symmetry breaking information that is available to a local OI-algorithm.

3.1 Homogeneous Graphs

In the following, we take the *isomorphism type* of an r-neighbourhood $\tau = \tau(\mathcal{G}, <, u)$ to be some canonical representative of the isomorphism class of τ.

Definition 1. Let $(\mathcal{H}, <)$ be an ordered graph. If there is a set $U \subseteq V(\mathcal{H})$ of size $|U| \geq \alpha|\mathcal{H}|$ such that the vertices in U have a common r-neighbourhood isomorphism type τ^*, then we call $(\mathcal{H}, <)$ an (α, r)-*homogeneous graph* and τ^* the associated *homogeneity type* of $\mathcal{H}$.

Homogeneous graphs are useful in fooling OI-algorithms: an (α, r)-homogeneous graph forces any local OI-algorithm to produce the same output in at least an α fraction of the nodes in the input graph. However, there are some limitations to how large α can be: Let $(\mathcal{G}, <)$ be a connected ordered graph on at least two vertices. If u and v are the smallest and the largest vertices of $\mathcal{G}$, their r-neighbourhoods $\tau(\mathcal{G}, <, u)$ and $\tau(\mathcal{G}, <, v)$ cannot be isomorphic even for $r = 1$. Thus, non-trivial finite graphs are not $(1, 1)$-homogeneous. Moreover, an ordered $(2k-1)$-regular graph cannot be $(\alpha, 1)$-homogeneous for any $\alpha > 1/2$; this is the essence of the weak 2-colouring algorithm of Naor and Stockmeyer [20].

Our main technical tool will be a construction of graphs that satisfy the following properties:

(1) $(1-\epsilon, r)$-homogeneous for any $\epsilon > 0$ and r,
(2) $2k$-regular for any k,
(3) large girth,
(4) finite order.

Note that it is relatively easy to satisfy any three of these properties:

(1), (2), (3) Infinite $2k$-regular trees admit a $(1, r)$-homogeneous linear order; see Figure 5 for an example.

(1), (2), (4) We can construct a sufficiently large k-dimensional toroidal grid graph (cartesian product of k directed cycles) and order the nodes lexicographically coordinate-wise; see Figure 6 for an example. However, these graphs have girth 4 when $k \geq 2$.

(1), (3), (4) A sufficiently large directed cycle is $(1-\epsilon, r)$-homogeneous and has large girth. However, all the nodes have degree 2.

(2), (3), (4) It is well known that regular graphs of arbitrarily high girth exist.

Our construction satisfies all four properties simultaneously.

THEOREM 3. *Let $k, r \in \mathbb{N}$. For every $\epsilon > 0$ there exists a finite $2k$-regular $(1-\epsilon, r)$-homogeneous connected graph $(\mathcal{H}_\epsilon, <_\epsilon)$ of girth larger than $2r+1$. Furthermore, the following properties hold:*

(a) The homogeneity type τ^* of $(\mathcal{H}_\epsilon, <_\epsilon)$ does not depend on ϵ.

(b) The graph $\mathcal{H}_\epsilon$ and the type τ^* are k-edge-labelled digraphs.

We defer the proof of Theorem 3 to Section 5. There, it turns out that Cayley graphs of *soluble* groups suit our needs: The homogeneous toroidal graphs mentioned above are Cayley graphs of the abelian groups $\mathbb{Z}_n^k$. Analogously, we use the decomposition of a soluble group into abelian factors to guarantee the presence of a suitable ordering. However, to ensure large girth, the groups we consider must be sufficiently far from being abelian, i.e., they must have large derived length [7].

3.2 Homogeneous Lifts

We fix some notation towards a proof of Theorem 1. By Theorem 3 we let $(\mathcal{H}_\epsilon, <_\epsilon)$, $\epsilon > 0$, be a family of $2|L|$-regular $(1-\epsilon, r)$-homogeneous connected graphs of girth $> 2r+1$ interpreted as L-digraphs. The homogeneity type τ^* that is shared by all $\mathcal{H}_\epsilon$ is then of the form $\tau^* = (\mathcal{T}^*, <^*, \lambda)$, where $\mathcal{T}^*$ is the complete L-labelled tree of Section 2.3.

We use the graphs $\mathcal{H}_\epsilon$ to prove the following theorem.

THEOREM 4. *Let $\mathcal{G}$ be an L-digraph. For every $\epsilon > 0$ there exists a lift $(\mathcal{G}_\epsilon, <_{\mathcal{G}\epsilon})$ of $\mathcal{G}$ such that a $(1-\epsilon)$ fraction of the vertices in $(\mathcal{G}_\epsilon, <_{\mathcal{G}\epsilon})$ have r-neighbourhoods isomorphic to a subtree of $\tau^* = (\mathcal{T}^*, <^*, \lambda)$. Moreover, if $\mathcal{G}$ is connected, $\mathcal{G}_\epsilon$ can be made connected.*

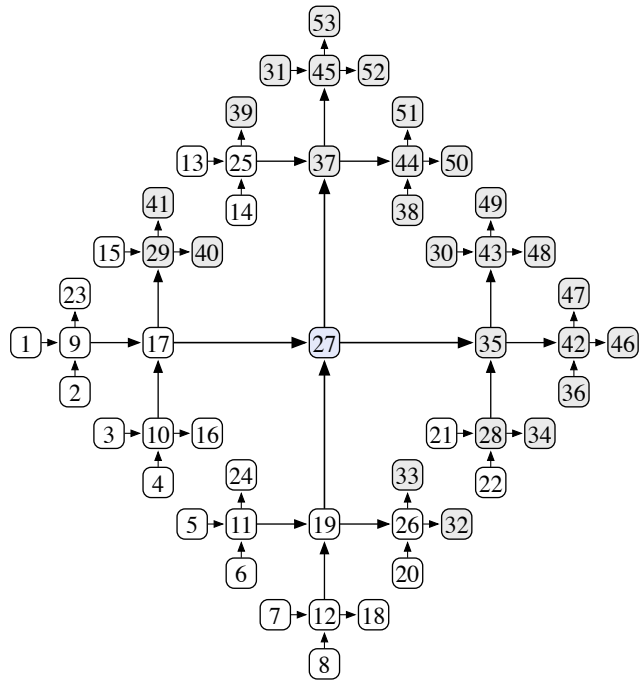

Figure 5: A fragment of a 4-regular infinite ordered tree $(\mathcal{G}, <)$. The numbering of the nodes indicates a $(1, r)$-homogeneous linear order in the neighbourhood of node 27; grey nodes are larger than 27 and white nodes are smaller than 27.

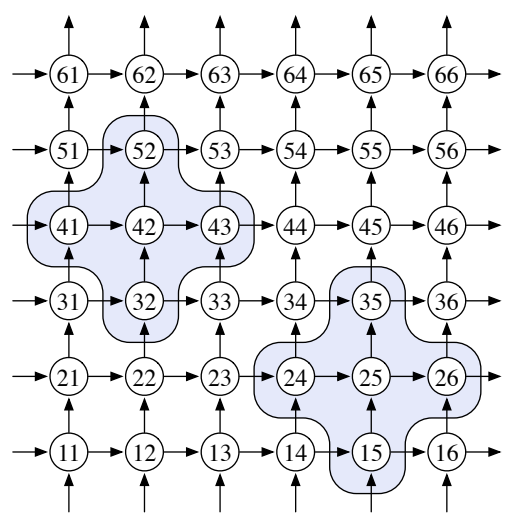

Figure 6: A 4-regular graph $\mathcal{G}$ constructed as the cartesian product of two directed 6-cycles. We define the ordered graph $(\mathcal{G}, <)$ by choosing the linear order $11 < 12 < \cdots < 16 < 21 < 22 < \cdots < 66$. The radius-1 neighbourhood of node 25 is isomorphic to the radius-1 neighbourhood of node 42. In general, there are 16 nodes (fraction 4/9 of all nodes) that have isomorphic radius-1 neighbourhoods; hence $(\mathcal{G}, <)$ is $(4/9, 1)$-homogeneous. It is also $(1/9, 2)$-homogeneous.

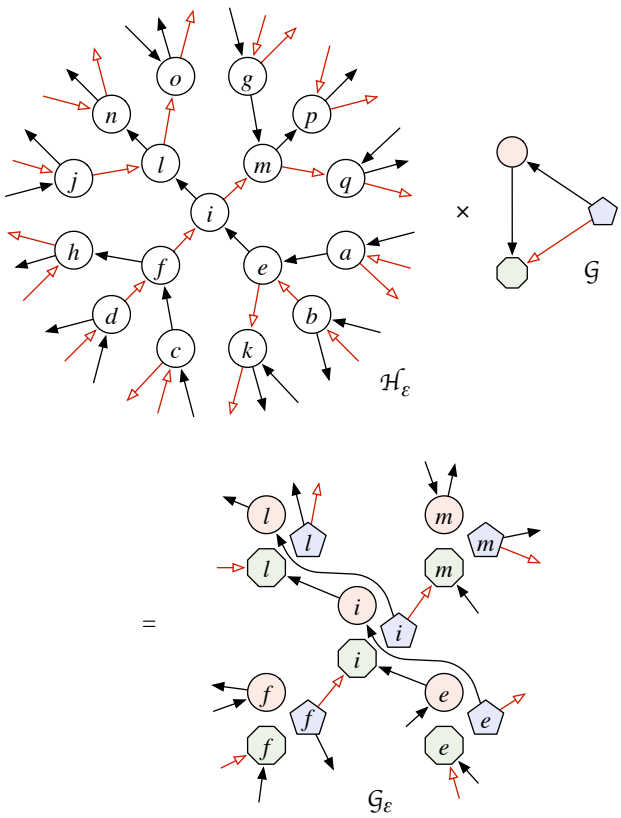

Figure 7: Homogeneous lifts. In this example $L = |2|$, and the two labels are indicated with two different kinds of arrows. Graph $\mathcal{H}_\epsilon$ is a homogeneous $2|L|$-regular ordered L-digraph with a large girth—in particular, the local neighbourhood of a node looks like a tree. Graph $\mathcal{G}$ is an arbitrary L-digraph, not necessarily ordered. Their product $\mathcal{G}_\epsilon$ is a lift of $\mathcal{G}$, but it inherits the desirable properties of $\mathcal{H}_\epsilon$: a high girth and a homogeneous linear order.

Proof. For brevity, write

$$(\mathcal{C}, <_\mathcal{C}) = (\mathcal{G}_\epsilon, <_{\mathcal{G}_\epsilon}), \quad (\mathcal{H}, <_\mathcal{H}) = (\mathcal{H}_\epsilon, <_\epsilon).$$

Our goal is to construct $(\mathcal{C}, <_\mathcal{C})$ as a certain product of $(\mathcal{H}, <_\mathcal{H})$ and $\mathcal{G}$; see Figure 7. This product is a modification of the common lift construction of Angluin and Gardiner [2].

The lift $\mathcal{C}$ is defined on the product set

$$V(\mathcal{C}) = V(\mathcal{H}) \times V(\mathcal{G})$$

by "matching equi-labelled edges": the out-neighbours of $(h, g) \in V(\mathcal{C})$ are vertices $(h', g') \in V(\mathcal{C})$ such that $(h, h') \in E(\mathcal{H})$, $(g, g') \in E(\mathcal{G})$ and $\ell_\mathcal{H}(h, h') = \ell_\mathcal{G}(g, g')$. An edge $((h, g), (h', g')) \in E(\mathcal{C})$ inherits the common label $\ell_\mathcal{H}(h, h') = \ell_\mathcal{G}(g, g')$.

The properties of $\mathcal{C}$ are related to the properties of $\mathcal{G}$ and $\mathcal{H}$ as follows.

(a) The projection $\varphi_\mathcal{G} \colon V(\mathcal{C}) \to V(\mathcal{G})$ mapping $(h, g) \mapsto g$ is a covering map. This follows from the fact that each edge incident to $g \in V(\mathcal{G})$ is always matched against an edge of $\mathcal{H}$ in the fibre $V(\mathcal{H}) \times \{g\}$.

(b) The projection $\varphi_\mathcal{H} \colon V(\mathcal{C}) \to V(\mathcal{H})$ mapping $(h, g) \mapsto h$ is not a covering map in case $\mathcal{G}$ is not $2|L|$-regular. In any case $\varphi_\mathcal{H}$ is a graph homomorphism, and this implies that $\mathcal{C}$ has girth $> 2r + 1$.

Next, we define a partial order $<_p$ on $V(\mathcal{C})$ as $u <_p v \iff \varphi_\mathcal{H}(u) <_\mathcal{H} \varphi_\mathcal{H}(v)$, for $u, v \in V(\mathcal{C})$. Note that this definition leaves only pairs of vertices in a common $\varphi_\mathcal{H}$-fibre incomparable. But since $\mathcal{H}$ has large girth, none of the incomparable pairs appear in an r-neighbourhood of $\mathcal{C}$. We let $<_\mathcal{C}$ be any completion of $<_p$ into a linear order. The previous discussion implies that $<_\mathcal{C}$ satisfies $\tau(\mathcal{C}, <_\mathcal{C}, u) = \tau(\mathcal{C}, <_p, u)$ for all $u \in V(\mathcal{C})$.

Let $U_\mathcal{H} \subseteq V(\mathcal{H})$, $|U_\mathcal{H}| \geq (1 - \epsilon)|\mathcal{H}|$, be the set of type τ^* vertices in $(\mathcal{H}, <_\mathcal{H})$. Set $U_\mathcal{C} = \varphi_\mathcal{H}^{-1}(U_\mathcal{H})$ so that $|U_\mathcal{C}| \geq (1-\epsilon)|\mathcal{C}|$. Let $u \in U_\mathcal{C}$. By our definition of $<_p$, $\varphi_\mathcal{H}$ maps the r-neighbourhood $\tau_u = \tau(\mathcal{C}, <_\mathcal{C}, u)$ into $\tau(\mathcal{H}, <_\mathcal{H}, \varphi_\mathcal{H}(u)) \simeq \tau^*$ while preserving the order. But because τ^* is a tree, $\varphi_\mathcal{H}$ must be injective on the vertex set of τ_u so that τ_u is isomorphic to a subtree of τ^* as required.

Finally, suppose $\mathcal{G}$ is connected. Then, by averaging, some connected component of $\mathcal{C}$ will have vertices in $U_\mathcal{C}$ with density at least $(1-\epsilon)$. This component satisfies the theorem. □

4. PROOF OF MAIN THEOREM

Next, we use the tools of the previous section to prove Theorem 1. For clarity of exposition we first prove Theorem 1 in the special case where A is an OI-algorithm. The subsequent proof for an ID-algorithm A uses a somewhat technical but well-known Ramsey type argument.

4.1 Proof of Main Theorem for OI-algorithms

We will prove the general and connected versions of Theorem 1 simultaneously; for the proof of the connected version it suffices to consider only connected lifts below. We do not need the assumption that $\mathcal{F}$ does not contain any trees.

Let Π be as in the statement of Theorem 1. Suppose an OI-algorithm A finds an α-approximation of Π in $\mathcal{F}$. We define a PO-algorithm B simply by setting for $W \in \mathfrak{W}$,

$$\mathsf{B}(W) = \mathsf{A}\big((\mathcal{T}^*, <^*, \lambda) \restriction W\big).$$

Now, Theorem 4 translates into saying that for every $\mathcal{G} \in \mathcal{F}$ and $\epsilon > 0$ we have that $\mathsf{A}(\mathcal{G}_\epsilon, <_{\mathcal{G}_\epsilon}, u) = \mathsf{B}(\mathcal{G}_\epsilon, u)$ for at least a $(1 - \epsilon)$ fraction of nodes $u \in V(\mathcal{G}_\epsilon)$. The claim that B works as expected follows essentially from this fact as we argue next.

For simplicity, we assume the solutions to Π are sets of vertices so that $\mathsf{A}(\mathcal{G}) \subseteq V(\mathcal{G})$; solutions that are sets of edges are handled similarly.

Fix $\mathcal{G} \in \mathcal{F}$ and let $\varphi_\epsilon \colon V(\mathcal{G}_\epsilon) \to V(\mathcal{G})$, $\epsilon > 0$, be the associated covering maps.

4.1.1 Feasibility

Let us first show that algorithm B finds a feasible solution of Π on $\mathcal{G}$. Let V be a local PO-algorithm verifying the feasibility of a solution for Π; we may assume V also runs in time r. For $\epsilon > 0$ sufficiently small, each $v \in V(\mathcal{G})$ has a preimage $v' \in \varphi_\epsilon^{-1}(v)$ such that A and B agree on the vertices $\bigcup_{v \in V(\mathcal{G})} B_{\mathcal{G}_\epsilon}(v', r)$. Thus, V accepts the solution $\mathsf{B}(\mathcal{G}_\epsilon)$ on the vertices v'. But because $\varphi_\epsilon(\{v' : v \in V(\mathcal{G})\}) = V(\mathcal{G})$ it follows that V accepts the solution $\mathsf{B}(\mathcal{G}) = \varphi_\epsilon(\mathsf{B}(\mathcal{G}_\epsilon))$ on every node in $\mathcal{G}$.

4.1.2 Approximation

Now we proceed to show that algorithm B finds an α-approximation of Π on $\mathcal{G}$. We assume Π is a minimisation problem; maximisation problems are handled similarly. Let $X \subseteq V(\mathcal{G})$ and $X_\epsilon \subseteq V(\mathcal{G}_\epsilon)$ be some optimal solutions of Π.

As $\epsilon \to 0$, the solutions $\mathsf{B}(\mathcal{G}_\epsilon)$ and $\mathsf{A}(\mathcal{G}_\epsilon)$ agree on almost all the vertices. Indeed, a simple calculation shows that $|\mathsf{B}(\mathcal{G}_\epsilon)| \leq f(\epsilon) \cdot |\mathsf{A}(\mathcal{G}_\epsilon)|$ for some f with $f(\epsilon) \to 1$ as $\epsilon \to 0$. Furthermore,

$$\frac{|\mathsf{B}(\mathcal{G})|}{|X|} = \frac{|\varphi_\epsilon^{-1}(\mathsf{B}(\mathcal{G}))|}{|\varphi_\epsilon^{-1}(X)|} \leq \frac{|\mathsf{B}(\mathcal{G}_\epsilon)|}{|X_\epsilon|} \leq \frac{f(\epsilon) \cdot |\mathsf{A}(\mathcal{G}_\epsilon)|}{|X_\epsilon|} \leq f(\epsilon)\alpha,$$

where the first equality follows from φ_ϵ being an n-lift, and the first inequality follows from $\varphi_\epsilon^{-1}(\mathsf{B}(\mathcal{G})) = \mathsf{B}(\mathcal{G}_\epsilon)$ and the fact that $\varphi_\epsilon^{-1}(X)$ is a feasible solution so that $|X_\epsilon| \leq |\varphi_\epsilon^{-1}(X)|$. Since the above inequality holds for every $\epsilon > 0$ we must have that $|\mathsf{B}(\mathcal{G})|/|X| \leq \alpha$, as desired.

4.2 Proof of Main Theorem for ID-algorithms

We extend the above proof to the case of local ID-algorithms A by designing "worst-case" vertex identifiers for the instances in $\mathcal{F}$ in order to make A behave similarly to a PO-algorithm on tree neighbourhoods. To do this we use the Ramsey technique of Naor and Stockmeyer [20]; see also Czygrinow et al. [9]. For a reference on Ramsey's theorem see Graham et al. [12].

We use the following notation: if $(X, <_X)$ and $(Y, <_Y)$ are linearly ordered sets with $|X| \leq |Y|$, we write $f\colon (X, <_X) \hookrightarrow (Y, <_Y)$ for the unique order-preserving injection $f\colon X \to Y$ that maps the ith element of X to the ith element of Y. A *t-set* is a set of size t, and the set of t-subsets of X is denoted $X^{(t)}$.

Write $\Omega^{\mathfrak{W}}$ for the family of functions $\mathfrak{W} \to \Omega$; recall that each $\mathsf{B} \in \Omega^{\mathfrak{W}}$ can be interpreted as a PO-algorithm. Set $k = |\Omega^{\mathfrak{W}}|$ and $t = |\mathcal{T}^*|$. We consider every t-subset $A \in \mathbb{N}^{(t)}$ to be ordered by the usual order $<$ on $\mathbb{N}$. For $W \in \mathfrak{W}$ we let $f_{W,A}\colon (W, <^*) \hookrightarrow (A, <)$ so that the vertex-relabelled tree $f_{W,A}((\mathcal{T}^*, \lambda) \restriction W)$ has the $|W|$ smallest numbers in A as vertices. Define a k-colouring $c\colon \mathbb{N}^{(t)} \to \Omega^{\mathfrak{W}}$ by setting

$$c(A)(W) = \mathsf{A}(f_{W,A}((\mathcal{T}^*, \lambda) \restriction W)).$$

For each $m \geq t$ we can use Ramsey's theorem to obtain a number $R(m) \geq m$, so that for every $R(m)$-set $I \subseteq \mathbb{N}$ there exists an m-subset $J \subseteq I$ such that $J^{(t)}$ is monochromatic under c, i.e., all t-subsets of J have the same colour. In particular, for every interval

$$I(m, i) = [(i-1)R(m) + 1,\, iR(m)], \quad i \geq 1,$$

there exist an m-subset $J(m, i) \subseteq I(m, i)$ and a colour (i.e., an algorithm) $\mathsf{B}_{m,i} \in \Omega^{\mathfrak{W}}$ such that $c(A) = \mathsf{B}_{m,i}$ for all t-subsets $A \subseteq J(m, i)$.

This construction has the following property.

PROPOSITION 1. *Suppose $m \geq |\mathcal{G}_\epsilon| + t$. Algorithms A and $\mathsf{B}_{m,i}$ produce the same output on at least a $(1-\epsilon)$ fraction of the vertices in the vertex-relabelled L-digraph $f_{m,i}(\mathcal{G}_\epsilon)$, where*

$$f_{m,i}\colon (V(\mathcal{G}_\epsilon), <_{\mathcal{G}_\epsilon}) \hookrightarrow (J(m, i), <).$$

PROOF. By Theorem 4, let $U \subseteq V(f_{m,i}(\mathcal{G}_\epsilon))$, $|U| \geq (1-\epsilon)|\mathcal{G}_\epsilon|$, be the set of vertices u with $\tau(f_{m,i}(\mathcal{G}_\epsilon), <, u)$ isomorphic to a subtree of τ^*. In particular, for a fixed $u \in U$ we can choose $W \in \mathfrak{W}$ such that $\tau(f_{m,i}(\mathcal{G}_\epsilon), <, u) \simeq (\mathcal{T}^*, <^*, \lambda) \restriction W$. Now, as m is large, there exists a t-set $A \subseteq J(m, i)$ such that $\tau(f_{m,i}(\mathcal{G}_\epsilon), u) = f_{W,A}((\mathcal{T}^*, \lambda) \restriction W)$. Thus, A and $\mathsf{B}_{m,i}$ agree on u by the definition of $\mathsf{B}_{m,i}$. $\square$

For every $n \in \mathbb{N}$ some colour appears with density at least $1/k$ (i.e., appears at least n/k times) in the sequence $\mathsf{B}_{m,1}, \mathsf{B}_{m,2}, \ldots, \mathsf{B}_{m,n}$. Hence, let B_m be a colour that appears with density at least $1/k$ among these sequences for infinitely many n. Let B be a colour appearing among the B_m for infinitely many m. We claim B satisfies Theorem 1. In fact, Theorem 1 follows from the following proposition together with the considerations of Section 4.1.

PROPOSITION 2. *For every $\mathcal{G}_\epsilon$ there exists an n-lift $\mathcal{H}$ of $\mathcal{G}_\epsilon$ such that $V(\mathcal{H}) \subseteq \{1, 2, \ldots, s(|\mathcal{H}|)\}$ and $\mathsf{A}(\mathcal{H}, u) = \mathsf{B}(\mathcal{H}, u)$ for a $(1-\epsilon)$ fraction of nodes $u \in V(\mathcal{H})$. Moreover, if $\mathcal{G}_\epsilon$ is connected and not a tree, $\mathcal{H}$ can be made connected.*

PROOF. Let m be such that $m \geq |\mathcal{G}_\epsilon| + t$ and $\mathsf{B} = \mathsf{B}_m$. For infinitely many n there exists an n-set $I \subseteq [nk]$ of indices such that $\mathsf{B} = \mathsf{B}_{m,i}$ for $i \in I$. Consider the following n-lift of $\mathcal{G}_\epsilon$ obtained by taking disjoint unions:

$$\mathcal{H} = \bigcup_{i \in I} f_{m,i}(\mathcal{G}_\epsilon).$$

Algorithms A and B agree on a $(1-\epsilon)$ fraction of the nodes in $\mathcal{H}$ by Proposition 1. Furthermore, we have $|\mathcal{H}| = n|\mathcal{G}_\epsilon|$ and $V(\mathcal{H}) \subseteq \{1, 2, \ldots, nkR(m)\}$. We are assuming that $s(n) = \omega(n)$ so choosing a large enough n proves the non-connected version of the claim.

Finally, suppose $\mathcal{G}_\epsilon$ is connected and not a tree. We may assume that there is an edge $e = (u, v) \in E(\mathcal{G}_\epsilon)$ so that $\mathcal{G}_\epsilon$ remains connected when e is removed and that a $(1-\epsilon)$ fraction of vertices in $\mathcal{G}_\epsilon$ have r-neighbourhoods not containing e that are isomorphic into τ^*. Now $\mathcal{H}$ above is easily modified into a connected graph by redefining the directed matching between the fibre $\{u_i\}_{i \in I}$ of u and the fibre $\{v_i\}_{i \in I}$ of v. Namely, let π be a cyclic permutation on I and set

$$E' = \big(E(\mathcal{H}) \smallsetminus \{(u_i, v_i)\}_{i \in I}\big) \cup \{(u_i, v_{\pi(i)})\}_{i \in I}.$$

Then $\mathcal{H}' = (V(\mathcal{H}), E')$ is easily seen to be a connected n-lift of $\mathcal{G}_\epsilon$ satisfying the claim. $\square$

Remark 1. Above, we assumed that instances $\mathcal{G}$ have node identifiers $V(\mathcal{G}) \subseteq \{1, 2, \ldots, s(n)\}$, $n = |\mathcal{G}|$, for $s(n) = \omega(n)$. By choosing identifiers more economically as in the work of Czygrinow et al. [9] one can show lower bounds for the graph problems of Section 1.3 even when $s(n) = n$.

5. CONSTRUCTION OF HOMOGENEOUS GRAPHS OF LARGE GIRTH

In this section we prove Theorem 3. Our construction uses Cayley graphs of semi-direct products of groups. First, we recall the terminology in use here; for a standard reference on group theory see, e.g., Rotman [21], and for more background information, see the full version of this work [11].

5.1 Semi-Direct Products

Let G and H be groups with H acting on G as a group of automorphisms. We write $h \cdot g$ for the action of $h \in H$ on $g \in G$ so that the mapping $g \mapsto h \cdot g$ is an automorphism of G. The *semi-direct product* $G \rtimes H$ is defined to be the set $G \times H$ with the group operation given by

$$(g, h)(g', h') = (g(h \cdot g'), hh').$$

5.2 Cayley Graphs

The *Cayley graph* $\mathcal{C}(G,S)$ of a group G with respect to a finite set $S \subseteq G$ is an S-digraph on the vertex set G such that each $g \in G$ has an outgoing edge (g, gs) labelled s for each $s \in S$. We require that $1 \notin S$ so as not to have any self-loops. We do not require that S is a generating set for G, i.e., the graph $\mathcal{C}(G,S)$ need not be connected.

If $\varphi\colon H \to G$ is an onto group homomorphism and $S \subseteq H$ is a set such that the mapping φ is injective on $S \cup \{1\}$, then φ naturally induces a covering map of digraphs $\mathcal{C}(H,S)$ and $\mathcal{C}(G,\varphi(S))$.

5.3 Proof of Theorem 3

Let $n \in \mathbb{N}$ be an even number. We consider three families of groups, $\{H_i\}_{i\geq 1}$, $\{W_i\}_{i\geq 1}$, and $\{U_i\}_{i\geq 1}$, that are variations on a common theme. The families are defined iteratively as follows:

$$H_1 = \mathbb{Z}_n, \qquad W_1 = \mathbb{Z}_2, \qquad U_1 = \mathbb{Z},$$
$$H_{i+1} = H_i^2 \rtimes \mathbb{Z}_n, \quad W_{i+1} = W_i^2 \rtimes \mathbb{Z}_2, \quad U_{i+1} = U_i^2 \rtimes \mathbb{Z}.$$

Here, the cyclic group $\mathbb{Z}_n = \{0, 1, \ldots, n-1\}$ acts on the direct product $H_i^2 = H_i \times H_i$ by cyclically permuting the coordinates, i.e., the subgroup $2\mathbb{Z}_n \leq \mathbb{Z}_n$ acts trivially and the elements in $1 + 2\mathbb{Z}_n$ swap the two coordinates. The groups $\mathbb{Z}_2$ and $\mathbb{Z}$ act analogously in the definitions of W_i and U_i.

The underlying sets of the groups H_i, W_i, and U_i consist of $d(i)$-tuples of elements in $\mathbb{Z}$, for $d(i) = 2^i - 1$, so that $W_i \subseteq H_i \subseteq U_i$ *as sets*. Interpreting these tuples as points in $\mathbb{R}^{d(i)}$ we immediately get a natural embedding of every Cayley graph of these groups in $\mathbb{R}^{d(i)}$. This geometric intuition will become useful later.

(a) The groups W_i are i-fold iterated regular wreath products of the cyclic group $\mathbb{Z}_2$. These groups have order $|W_i| = 2^{d(i)}$ and they are sometimes called *symmetric 2-groups*; they are isomorphic to the Sylow 2-subgroups of the symmetric group on 2^i letters [21, p. 176].

(b) The groups U_i are natural extensions of the groups W_i by the free abelian group of rank $d(i)$: the mapping $\varphi_i\colon U_i \to W_i$ that reduces each coordinate modulo 2 is easily seen to be an onto homomorphism with abelian kernel $(2\mathbb{Z})^{d(i)} \simeq \mathbb{Z}^{d(i)}$.

(c) The groups H_i are intermediate between U_i and W_i in that the mapping $\psi_i\colon U_i \to H_i$ that reduces each coordinate modulo n is an onto homomorphism, and the mapping $\varphi_i'\colon H_i \to W_i$ that reduces each coordinate modulo 2 is an onto homomorphism. In summary, the following diagram commutes:

$$\begin{tikzcd} U_i \arrow[r, "\psi_i"] \arrow[dr, "\varphi_i"'] & H_i \arrow[d, "\varphi_i'"] \\ & W_i \end{tikzcd}$$

Our goal will be to construct a suitable Cayley graph $\mathcal{H}$ of some H_i. We will use the groups W_i to ensure $\mathcal{H}$ has large girth, whereas the groups U_i will guarantee that $\mathcal{H}$ has an almost-everywhere homogeneous linear ordering.

5.3.1 Girth

Gamburd et al. [10] study the girth of random Cayley graphs and prove, in particular, that a random k-subset of W_i generates a Cayley graph of large girth with high probability when $i \gg k$ is large. We only need the following weaker version of their result [10, Theorem 6]; see the full version of this work [11] for an alternative, constructive proof.

THEOREM 5 (GAMBURD ET AL.). *Let $k, r \in \mathbb{N}$. There exists an $i \in \mathbb{N}$ and a set $S \subseteq W_i$, $|S| = k$, such that the girth of the Cayley graph $\mathcal{C}(W_i, S)$ is larger than $2r+1$.*

Fix a large enough $j \in \mathbb{N}$ and a k-set $S \subseteq W_j$ so that $\mathcal{C}(W_j, S)$ has a girth larger than $2r+1$. Henceforth, we omit the subscript j and write H, W, U, φ, ψ and d in place of H_j, W_j, U_j, φ_j, ψ_j and $d(j)$. Interpreting S as a set of elements of H and U (so that $\varphi(S) = \psi(S) = S$) we construct the Cayley graphs

$$\mathcal{H} = \mathcal{C}(H,S), \quad \mathcal{W} = \mathcal{C}(W,S), \quad \text{and} \quad \mathcal{U} = \mathcal{C}(U,S).$$

As each of these graphs is a lift of $\mathcal{W}$, none have cycles of length at most $2r+1$ and their r-neighbourhoods are trees.

5.3.2 Linear Order

Next, we introduce a *left-invariant* linear order $<$ on U satisfying

$$u < v \implies wu < wv, \qquad \text{for all } u, v, w \in U.$$

Such a relation can be defined by specifying a *positive cone* $P \subseteq U$ of elements that are greater than the identity $1 = 1_U$ so that

$$u < v \iff 1 < u^{-1}v \iff u^{-1}v \in P.$$

A relation $<$ defined this way is automatically left-invariant; it is transitive iff $u, v \in P$ implies $uv \in P$; and every pair $u \neq v$ is comparable iff for all $w \neq 1$, either $w \in P$ or $w^{-1} \in P$. The existence of a P satisfying these conditions follows from the fact that U is a torsion-free soluble group (e.g., [8]), but it is easy enough to verify that setting

$$P = \big\{(u_1, u_2, \ldots, u_i, 0, 0, \ldots, 0) \in U : 1 \leq i \leq d \text{ and } u_i > 0\big\} \tag{1}$$

satisfies the required conditions above.

Because U acts (by multiplication on the left) on $\mathcal{U}$ as a vertex-transitive group of graph automorphisms, it follows that the structures $(\mathcal{U}, <, u)$, $u \in U$, are pairwise isomorphic. A fortiori, the r-neighbourhoods $\tau(\mathcal{U}, <, u)$, $u \in U$, are all pairwise isomorphic. Let τ^* be this common r-neighbourhood isomorphism type.

5.3.3 Transferring the Linear Order on U to $\mathcal{H}$

Let $V(\mathcal{H})$ be ordered by restricting the order $<$ on U to the set $V(\mathcal{H}) = \mathbb{Z}_n^d$ underlying the group H. Note that $<$ is not a left-invariant order on H (indeed, no non-trivial finite group can be left-invariantly ordered). Nevertheless, we will argue that, as $n \to \infty$, almost all $u \in V(\mathcal{H})$ have r-neighbourhoods of type τ^*.

The neighbours of a vertex $u \in V(\mathcal{U})$ are elements us where

$$s \in S \cup S^{-1} \subseteq [-1, 1]^d.$$

The right multiplication action of $s \in S \cup S^{-1}$ on u can be described in two steps as follows: First, the coordinates of

s are permuted (as determined by u) to obtain a vector s'. Then, us is given as the standard addition of the vectors u and s' in $\mathbb{Z}^d \subseteq \mathbb{R}^d$. Hence, $us \in u + [-1,1]^d$, and moreover,

$$B_{\mathcal{U}}(u,r) \subseteq u + [-r,r]^d. \tag{2}$$

This means that vertices close to u in the graph $\mathcal{U}$ are also close in the associated geometric $\mathbb{R}^d$-embedding.

Consider the set of inner nodes $I = [r,(n-1)-r]^d$. Let $u \in I$. By (2), the vertex set $B_{\mathcal{U}}(u,r)$ is contained in $\mathbb{Z}_n^d$. This implies that the cover map ψ is the identity on $B_{\mathcal{U}}(u,r)$ and consequently the r-neighbourhood $\tau(\mathcal{H},<,u)$ *contains* the ordered tree $\tau(\mathcal{U},<,u) \simeq \tau^*$. If $\tau(\mathcal{H},<,u)$ had any additional edges to those of $\tau(\mathcal{U},<,u)$, this would entail a cycle of length at most $2r+1$ in $\mathcal{H}$, which is not possible. Thus, $\tau(\mathcal{H},<,u) \simeq \tau^*$. The density of elements in $\mathcal{H}$ having r-neighbourhood type τ^* is therefore at least

$$\frac{|I|}{|\mathcal{H}|} = \frac{(n-2r)^d}{n^d} \geq 1-\epsilon,$$

for large n.

Finally, to establish Theorem 3 it remains to address $\mathcal{H}$'s connectedness. But if $\mathcal{H}$ is not connected, an averaging argument shows that some connected component must have the desired density of at least $(1-\epsilon)$ of type τ^* vertices.

6. ACKNOWLEDGEMENTS

We thank anonymous reviewers for their helpful feedback, and Christoph Lenzen and Roger Wattenhofer for discussions. This work was supported in part by the Academy of Finland, Grants 132380 and 252018, the Research Funds of the University of Helsinki, and the Finnish Cultural Foundation.

The full version of this work is available online [11].

7. REFERENCES

[1] Dana Angluin. Local and global properties in networks of processors. In *Proc. 12th Symposium on Theory of Computing (STOC 1980)*, pages 82–93. ACM Press, New York, 1980.

[2] Dana Angluin and A. Gardiner. Finite common coverings of pairs of regular graphs. *Journal of Combinatorial Theory, Series B*, 30(2):184–187, 1981.

[3] Matti Åstrand, Patrik Floréen, Valentin Polishchuk, Joel Rybicki, Jukka Suomela, and Jara Uitto. A local 2-approximation algorithm for the vertex cover problem. In *Proc. 23rd Symposium on Distributed Computing (DISC 2009)*, volume 5805 of *LNCS*, pages 191–205. Springer, Berlin, 2009.

[4] Matti Åstrand, Valentin Polishchuk, Joel Rybicki, Jukka Suomela, and Jara Uitto. Local algorithms in (weakly) coloured graphs, 2010. Manuscript, arXiv:1002.0125 [cs.DC].

[5] Matti Åstrand and Jukka Suomela. Fast distributed approximation algorithms for vertex cover and set cover in anonymous networks. In *Proc. 22nd Symposium on Parallelism in Algorithms and Architectures (SPAA 2010)*, pages 294–302. ACM Press, New York, 2010.

[6] Richard Cole and Uzi Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control*, 70(1):32–53, 1986.

[7] Marston Conder, Geoffrey Exoo, and Robert Jajcay. On the limitations of the use of solvable groups in Cayley graph cage constructions. *European Journal of Combinatorics*, 31:1819–1828, 2010.

[8] Paul Conrad. Right-ordered groups. *The Michigan Mathematical Journal*, 6:267–275, 1959.

[9] Andrzej Czygrinow, Michał Hańćkowiak, and Wojciech Wawrzyniak. Fast distributed approximations in planar graphs. In *Proc. 22nd Symposium on Distributed Computing (DISC 2008)*, volume 5218 of *LNCS*, pages 78–92. Springer, Berlin, 2008.

[10] Alex Gamburd, Shlomo Hoory, Mehrdad Shahshahani, Aner Shalev, and Balint Virág. On the girth of random Cayley graphs. *Random Structures & Algorithms*, 35(1):100–117, 2009.

[11] Mika Göös, Juho Hirvonen, and Jukka Suomela. Lower bounds for local approximation, 2012. Manuscript, arXiv:1201.6675 [cs.DC].

[12] Ronald L. Graham, Bruce L. Rothschild, and Joel H. Spencer. *Ramsey Theory*. John Wiley & Sons, New York, 1980.

[13] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. What cannot be computed locally! In *Proc. 23rd Symposium on Principles of Distributed Computing (PODC 2004)*, pages 300–309. ACM Press, New York, 2004.

[14] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. The price of being near-sighted. In *Proc. 17th Symposium on Discrete Algorithms (SODA 2006)*, pages 980–989. ACM Press, New York, 2006.

[15] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds, 2010. Manuscript, arXiv:1011.5470 [cs.DC].

[16] Christoph Lenzen. *Synchronization and Symmetry Breaking in Distributed Systems*. PhD thesis, ETH Zurich, January 2011.

[17] Christoph Lenzen and Roger Wattenhofer. Leveraging Linial's locality limit. In *Proc. 22nd Symposium on Distributed Computing (DISC 2008)*, volume 5218 of *LNCS*, pages 394–407. Springer, Berlin, 2008.

[18] Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.

[19] Alain Mayer, Moni Naor, and Larry Stockmeyer. Local computations on static and dynamic graphs. In *Proc. 3rd Israel Symposium on the Theory of Computing and Systems (ISTCS 1995)*, pages 268–278. IEEE, Piscataway, 1995.

[20] Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.

[21] Joseph J. Rotman. *An Introduction to the Theory of Groups*, volume 148 of *Graduate Texts in Mathematics*. Springer, New York, 1995.

[22] Jukka Suomela. Distributed algorithms for edge dominating sets. In *Proc. 29th Symposium on Principles of Distributed Computing (PODC 2010)*, pages 365–374. ACM Press, New York, 2010.

[23] Jukka Suomela. Survey of local algorithms. *ACM Computing Surveys*. To appear.

[24] Masafumi Yamashita and Tsunehiko Kameda. Computing on anonymous networks: Part I—characterizing the solvable cases. *IEEE Transactions on Parallel and Distributed Systems*, 7(1):69–89, 1996.