A

Lower Bounds for Local Approximation

MIKA GÖÖS, Department of Computer Science, University of Toronto
JUHO HIRVONEN, and JUKKA SUOMELA, Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki

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, we derive a tight lower bound on the local approximability of the *minimum edge dominating set problem*. By prior work, there is a deterministic local algorithm that achieves the approximation factor of $4 - 1/\lfloor \Delta/2 \rfloor$ in graphs of maximum degree Δ. This approximation ratio is known to be optimal in the port-numbering model—our main theorem implies that it is optimal also in the standard model in which each node has a unique identifier.

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*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Approximation algorithms, deterministic distributed algorithms, edge dominating set, local algorithms, unique identifiers

Authors' addresses: M. Göös, Department of Computer Science, University of Toronto, 3302-10 King's College Road, Toronto, Ontario M5S 3G4, Canada; J. Hirvonen and J. Suomela, Helsinki Institute for Information Technology HIIT, University of Helsinki, P.O. Box 68, FI-00014 University of Helsinki, Finland. Email: mika.goos@mail.utoronto.ca, juho.hirvonen@cs.helsinki.fi, and jukka.suomela@cs.helsinki.fi

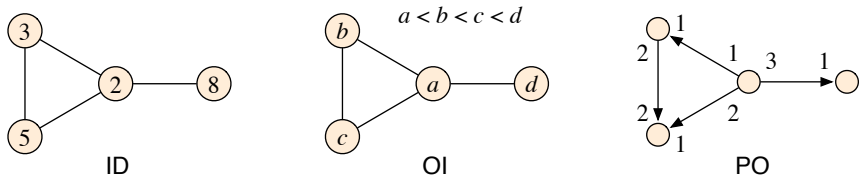


Fig. 1. Three models of distributed computing.

1. INTRODUCTION

We study deterministic distributed graph algorithms under three different assumptions; see Figure 1 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 the 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, in this work, we 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.

Put succinctly, we prove that ID $=$ OI $=$ PO for *local approximation*.

Our work provides a new, general technique for *proving lower bounds* in the field of distributed computing. Proving tight lower bounds in the ID model is challenging. However, the PO model is very restricted, and for many graph problems it is fairly easy to prove tight lower bounds for local algorithms in the PO model. Our work can be used to amplify such results to obtain the same lower bound in the ID model.

This work also shows that strictly constant-time distributed algorithms are genuinely different from almost-constant-time algorithms, and this has practical consequences. At first sight, the difference between distributed algorithms with running times of $O(1)$ vs. $O(\log^* n)$ is only of theoretical interest—in practice, both algorithms are fast for any reasonable value of n. However, our work shows that $O(1)$-time algorithms admit a *simple implementation*: even if the algorithm is originally designed for the ID model, we can construct an algorithm that solves the same problem without transmitting any unique identifiers. This is not the case with $O(\log^* n)$-time algorithms, which typically make an essential use of the unique identifiers and cannot be used in anonymous networks at all.

1.1. Graph Problems

Distributed graph algorithms solve 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

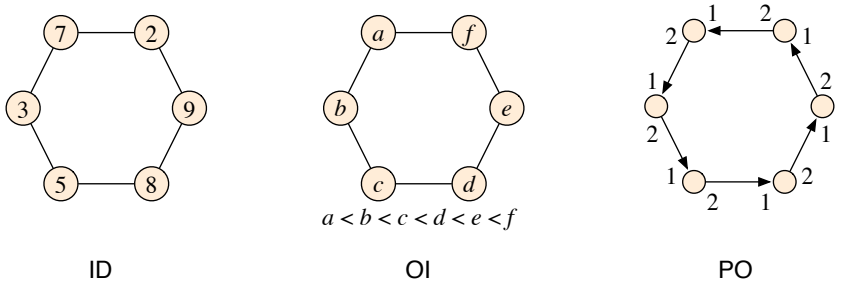


Fig. 2. In model ID, the numerical identifiers break symmetry everywhere—for example, in a cycle, a maximal independent set can be found in $O(\log^* n)$ rounds. In model OI, we can have a cycle with only one "seam", and in model PO we can have a completely symmetric cycle.

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 (we will make the model formally precise in Section 2.1).

If we do not restrict the run-time of an algorithm, the models ID, OI, and PO are easy to separate. Consider, for example, the problem of finding a maximal independent set (MIS) in an n-cycle. In the ID model the problem can be solved in $O(\log^* n)$ rounds [Cole and Vishkin 1986], while in the OI model we need $\Theta(n)$ rounds, and the problem is not solvable at all in PO, as we cannot break symmetry—see Figure 2. Hence ID is strictly stronger than OI, which is strictly stronger than PO.

1.2. Deterministic Local Algorithms

In this work we focus on deterministic *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 [Naor and Stockmeyer 1995; Suomela 2013b].

With the run-time restricted to a constant, the MIS problem above no longer serves as an example separating ID, OI, and PO: the MIS problem cannot be solved in constant-time in any of these models [Linial 1992]. Nevertheless, some *artificial* problems do exist that are solvable in OI but not in PO. One example is given by Mayer et al. [1995, §7], and more examples can be derived from the fact that PO-algorithms cannot detect small cycles in the network, by definition (see Section 2.5).

However, there has been a conspicuous lack of *natural* graph problems that would separate ID, OI, and PO from the perspective of local algorithms. In fact, as we discuss next, there are results that show that many natural problems that can be solved with a local algorithm in ID also admit a local algorithm in OI or PO.

1.3. LCL Problems

The seminal paper by Naor and Stockmeyer [1995] studies so-called *locally checkable labellings*, or LCL problems for short, that include problems such as graph colouring and maximal matchings on bounded-degree graphs. In particular, the authors show that local algorithms in ID and OI are indeed equally expressive among LCL problems.

The followup work by Mayer et al. [1995] hints of a stronger property—some problems that admit local algorithms in ID have also local algorithms in PO:

— *Weak 2-colouring* is an LCL problem that can be solved (in certain graph families) with a local algorithm in the ID model [Naor and Stockmeyer 1995]. It turns out that this problem can be solved in constant time in the PO model as well [Mayer et al. 1995].

As we will see in the following sections, we can find many similar examples also outside the class of LCL problems.

1.4. Global Optimisation vs. Local Approximation

Most of the classical graph problems that are studied in the field of distributed computing are *optimisation problems*. These problems go beyond *local* checkability: we are not only producing a feasible solution—which is an LCL problem—but preferably one that has a small cost—which is a *global* problem. Given that many graph optimisation problems are hard to solve exactly even in the centralised setting, the best we can hope to achieve in the distributed setting is an *approximation* of the global optimum.

As was shown by Kuhn et al. [2004], few graphs problems admit constant-time constant-factor approximation algorithms on general graphs. Therefore the study of local algorithms has focused on specific graph families, in particular on *bounded-degree graphs*. We follow this practice throughout this work. More precisely, we assume that there is a known constant Δ such that the degree of any node in any graph that we may encounter is at most Δ, and we are interested in deterministic algorithms that have a running time that may depend on Δ but is independent of the number of nodes in the network.

In this setting, we have a wide selection of *tight* results on the local approximability—and surprisingly the best possible approximation ratios are very similar in ID, OI, and PO. For example, the following hold for any given $\Delta \geq 2$ and $\epsilon > 0$; here we use the shorthand notation $\Delta' = 2\lfloor \Delta/2 \rfloor$:

— *Minimum vertex cover* can be approximated to within factor 2 in each of these models [Åstrand et al. 2009; Åstrand and Suomela 2010]. This is tight: $(2-\epsilon)$-approximation is not possible in any of these models [Czygrinow et al. 2008; Lenzen and Wattenhofer 2008; Suomela 2013b].
— *Minimum edge cover* can be approximated to within factor 2 in each of these models [Suomela 2013b]. This is tight: $(2-\epsilon)$-approximation is not possible in any of these models [Czygrinow et al. 2008; Lenzen and Wattenhofer 2008; Suomela 2013b].
— *Minimum dominating set* can be approximated to within factor $\Delta' + 1$ in each of these models [Åstrand et al. 2010]. This is tight: $(\Delta' + 1 - \epsilon)$-approximation is not possible in any of these models [Czygrinow et al. 2008; Lenzen and Wattenhofer 2008; Suomela 2013b].
— *Maximum independent set* cannot be approximated to within any constant factor in any of these models [Czygrinow et al. 2008; Lenzen and Wattenhofer 2008].
— *Maximum matching* cannot be approximated to within any constant factor in any of these models [Czygrinow et al. 2008; Lenzen and Wattenhofer 2008].

1.5. State of the Art

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:

— *Minimum edge dominating set* can be approximated to within factor $4 - 2/\Delta'$ in each of these models [Suomela 2010]. This is tight for PO but only near-tight for ID and OI: $(4 - 2/\Delta' - \epsilon)$-approximation is not possible in PO [Suomela 2010], and $(3-\epsilon)$-approximation is not possible in ID and OI [Czygrinow et al. 2008; Lenzen and Wattenhofer 2008; Suomela 2013b].

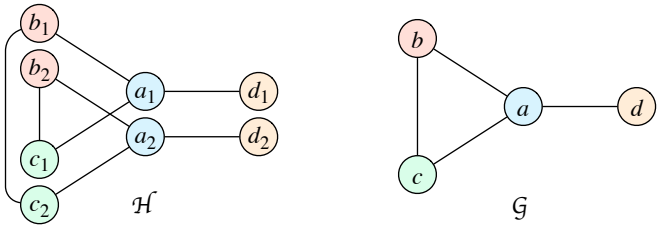


Fig. 3. Graph $\mathcal{H}$ is a lift of $\mathcal{G}$. The covering map $\varphi\colon V(\mathcal{H}) \to V(\mathcal{G})$ maps $a_i \mapsto a$, $b_i \mapsto b$, $c_i \mapsto c$, and $d_i \mapsto d$ for each $i = 1, 2$. The fibre of $a \in V(\mathcal{G})$ is $\{a_1, a_2\} \subseteq V(\mathcal{H})$; all fibres have the same size.

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 lifting the existing lower bound from PO to ID and OI.

1.6. Main Result

We prove that—under some mild assumptions—the best achievable local approximation ratios agree across our three models ID, OI, and PO. Before we state this result precisely, we introduce and recall some terminology.

Simple graph problems. A simple graph problem $\mathbf{\Pi}$ 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 $\mathbf{\Pi}$ is a PO-*checkable graph problem* if there is a local PO-algorithm $\mathbf{A}$ that recognises a feasible solution. That is, $\mathbf{A}(\mathcal{G}, X, v) = 1$ for all nodes $v \in V(\mathcal{G})$ if X is a feasible solution of problem $\mathbf{\Pi}$ in graph $\mathcal{G}$, and $\mathbf{A}(\mathcal{G}, X, v) = 0$ for some node $v \in V(\mathcal{G})$ otherwise—here $\mathbf{A}(\mathcal{G}, X, v)$ is the output of a node v if we run algorithm $\mathbf{A}$ on graph $\mathcal{G}$ and the local inputs form an encoding of X (see Section 2). In particular, a PO-checkable problem is also an LCL problem.

The definition of a simple PO-checkable problem may seem somewhat artificial. However, it is easy to verify that numerous classical graph problems are PO-checkable.

Example 1.1. Minimum vertex cover, minimum edge covers, maximum matching, maximum independent set, minimum dominating sets, and minimum edge dominating set are simple PO-checkable graph problems.

Graph lifts. Let $\varphi\colon V(\mathcal{H}) \to V(\mathcal{G})$ be an onto graph homomorphism from graph $\mathcal{H}$ to graph $\mathcal{G}$. If φ preserves vertex degrees, i.e., $\deg_{\mathcal{H}}(v) = \deg_{\mathcal{G}}(\varphi(v))$, then φ is called a *covering map*, and $\mathcal{H}$ is said to be a *lift* of $\mathcal{G}$. The *fibre* of $v \in V(\mathcal{G})$ is the set $\varphi^{-1}(v)$ of pre-images of v. We usually consider l-lifts that have fibres of the same cardinality l; any connected lift $\mathcal{H}$ of $\mathcal{G}$ is an l-lift for some l. See Figure 3 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 $\mathcal{G}$. Again, the definitions are a bit technical, but many graph families that have been studied in the context of distributed algorithms are closed under lifts.

Example 1.2. Bounded-degree graphs, regular graphs, bounded-arboricity graphs, and cyclic graphs are closed under lifts. The restrictions of these families to connected graphs are closed under connected lifts.

Main theorem. We give two versions of our main result. The first version is easier to state, but the second version is perhaps more interesting, as it lets us derive lower bounds that hold even if we restrict ourselves to connected graphs.

THEOREM 1.3 (MAIN THEOREM, GENERAL VERSION). *Let $\mathbf{\Pi}$ be a simple* PO*-checkable graph problem. Assume that $\mathcal{F}$ is a family of bounded-degree graphs, and it is closed under lifts. If there is an* ID*-algorithm $\mathbf{A}$ with run-time $r = O(1)$ that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$, then there is a* PO*-algorithm $\mathbf{B}$ with run-time r that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$.*

THEOREM 1.4 (MAIN THEOREM, CONNECTED VERSION). *Let $\mathbf{\Pi}$ be a simple* PO*-checkable graph problem. Assume that $\mathcal{F}$ is a family of connected bounded-degree graphs, it does not contain any trees, and it is closed under connected lifts. If there is an* ID*-algorithm $\mathbf{A}$ with run-time $r = O(1)$ that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$, then there is a* PO*-algorithm $\mathbf{B}$ with run-time r that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$.*

In other words, our main theorems say that in order to show run-time or approximability lower bounds in the ID model, it suffices to do so in the easy-to-analyse PO model. Examples 1.1 and 1.2 demonstrate that the result is widely applicable—among others, it applies to all optimisation problems that we discussed in Sections 1.4 and 1.5. We refer to Section 6 for a discussion on possible extensions and generalisations.

Remark 1.5. To separate constant-time algorithms and non-constant-time algorithms, we have to consider arbitrarily large instances. In general, we can use lifts to increase the size of a graph. However, *connected* lifts of a tree are isomorphic to the original tree. This is why the statement of Theorem 1.4 requires that $\mathcal{F}$ does not contain any trees. As we will show in the following section, the requirement is harmless in typical applications of Theorem 1.4. To prove a lower bound, it is sufficient that *some* worst-case instances contain some cycles.

1.7. An Application: Edge Dominating Sets

Theorems 1.3 and 1.4 provide us with powerful tools for proving lower-bound results: we can easily transfer negative results from PO to OI and ID. We demonstrate this by deriving a new lower bound for the minimum edge dominating set problem.

Recall that an *edge dominating set* for graph $\mathcal{G}$ is a set D of edges such that each edge of $\mathcal{G}$ is in D or adjacent to at least one edge in D. From the perspective of centralised algorithms, this is equivalent to the problem of finding a minimum maximal matching—in particular, a minimum maximal matching is a minimum edge dominating set. The problem is NP-hard [Yannakakis and Gavril 1980] and hard to approximate to within factor $7/6 - \epsilon$ [Chlebík and Chlebíková 2006].

Both in a centralised setting and in a distributed setting there is a simple factor-2 approximation algorithm: find any maximal matching. However, this does not result in a local algorithm, as finding a maximal matching requires $\Omega(\log^* n)$ time (even if $\Delta = 2$); this is a corollary of Linial's [1992] seminal result.

As we pointed out in Section 1.5, the local approximability of the edge dominating set problem has been an open question. Now we can settle it as a straightforward corollary of the main theorem.

THEOREM 1.6. *Let $\Delta \geq 2$, and let $\mathbf{A}$ 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 [Suomela 2010], 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.

(1) We have $\mathcal{F}_0 \subseteq \mathcal{F}$; by assumption, $\mathbf{A}$ finds an α-approximation in $\mathcal{F}_0$.
(2) 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 $\mathbf{B}$ that finds an α-approximation in $\mathcal{F}_0$.
(3) However, $\mathcal{G}_0 \in \mathcal{F}_0$, and hence $\alpha \geq \alpha_0$.

The matching upper bound is presented in prior work [Suomela 2010]. □

1.8. Overview of Proof

Roughly speaking, the proof of the main theorem consists of showing that, in the worst case, an input graph $\mathcal{G}$ can be assigned unique identifiers in a way that reveals virtually no additional symmetry breaking information to that already contained in the PO-structure of $\mathcal{G}$. Here, OI will serve as an intermediate model that splits up our proof of ID = PO into two steps: OI = PO and ID = OI.

The first part of our proof introduces new techniques:

> OI = PO. In Section 3, we introduce tools to control the order structure of a local neighbourhood in the OI model. For each input graph $\mathcal{G} \in \mathcal{F}$ we construct a *high-girth* lift $\mathcal{G}_\epsilon$ of $\mathcal{G}$ and a linear order $<_\epsilon$ on the nodes of $\mathcal{G}_\epsilon$ such that there is a near-perfect correspondence between the ordered neighbourhoods of $(\mathcal{G}_\epsilon, <_\epsilon)$ and the port-numbered neighbourhoods of $\mathcal{G}_\epsilon$.
>
> In Section 4.1, we use this correspondence to prove that for any OI-algorithm $\mathbf{A}$ there is a PO-algorithm $\mathbf{B}$ that can simulate $\mathbf{A}$ on at least a $1 - \epsilon$ fraction of the nodes of $\mathcal{G}_\epsilon$. Hence, the algorithms $\mathbf{A}$ and $\mathbf{B}$ produce almost the same solutions on the lift $\mathcal{G}_\epsilon$. From this we deduce that $\mathbf{B}$ produces a good approximation on the original graph $\mathcal{G}$, as the output of any PO-algorithm is invariant under lifts.

For example, if $\mathcal{G}$ was a directed cycle the construction would be standard [Czygrinow et al. 2008]: the lift $\mathcal{G}_\epsilon$ would simply be a long cycle, and $<_\epsilon$ would order the nodes along the cycle. Moreover, there would be only one "seam" in $(\mathcal{G}_\epsilon, <_\epsilon)$ that could potentially help $\mathbf{A}$ in comparison with $\mathbf{B}$, but only an ϵ fraction of the nodes are near the seam.

More generally, we give a construction that works for any $\mathcal{G}$. Our main technical tool is the construction of *homogeneous graphs*. 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 graphs of large degree and 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.

The second part of our proof is based on existing techniques:

> ID = OI. In Section 4.2, we apply Ramsey's theorem to force an ID-algorithm into producing an output that depends only on the relative order of the identifiers. This extends the methods of Naor and Stockmeyer [1995] and Czygrinow et al. [2008].

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.

2.1. General Setting

We start by fixing an input graph family $\mathcal{F}$ where every $\mathcal{G} = (V(\mathcal{G}), E(\mathcal{G})) \in \mathcal{F}$ has maximum degree at most some constant $\Delta \in \mathbb{N}$. We consider distributed algorithms $\mathbf{A}$ that operate on graphs in $\mathcal{F}$.

We denote by $\mathbf{A}(\mathcal{G}, v) \in \Omega$ the output of $\mathbf{A}$ on a node $v \in V(\mathcal{G})$. Here, Ω is a finite set of possible outputs of $\mathbf{A}$ in $\mathcal{F}$. If the solutions to $\mathbf{\Pi}$ are sets of vertices, we take $\Omega = \{0, 1\}$ so that the solution produced by $\mathbf{A}$ on $\mathcal{G}$, denoted $\mathbf{A}(\mathcal{G}) \subseteq V(\mathcal{G})$, is the set of nodes v with $\mathbf{A}(\mathcal{G}, v) = 1$. Similarly, if the solutions to $\mathbf{\Pi}$ are sets of edges, we take $\Omega = \{0, 1\}^{\Delta}$ so that the ith component of the vector $\mathbf{A}(\mathcal{G}, v)$ indicates whether the ith edge incident to v is included in the solution $\mathbf{A}(\mathcal{G}) \subseteq E(\mathcal{G})$.

2.2. Locality

A local algorithm $\mathbf{A}$ is characterised by the fact that its running time $r \in \mathbb{N}$ is a constant. This means that a node $v \in V(\mathcal{G})$ can only receive messages from nodes within distance r in $\mathcal{G}$, i.e., from nodes in the radius-r ball

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

Let $\tau(\mathcal{G}, v)$ denote the radius-r neighbourhood of v in $\mathcal{G}$. That is, $\tau(\mathcal{G}, v)$ is the restriction of the structure $(\mathcal{G}, v)$ to the vertices $B_{\mathcal{G}}(v, r)$; in symbols, $\tau(\mathcal{G}, v) := (\mathcal{G}, v) \restriction B_{\mathcal{G}}(v, r)$. The output of a local algorithm is limited to be a function of the data $\tau(\mathcal{G}, v)$ in that

$$\mathbf{A}(\mathcal{G}, v) = \mathbf{A}(\tau(\mathcal{G}, v)). \tag{1}$$

Conversely, any function $\mathbf{A}$ satisfying (1) is a local ID-algorithm. The models OI and PO, by contrast, impose further restrictions on this function.

Remark 2.1. In general, $\tau(\mathcal{G}, v)$ could contain edges that have both endpoints at distance precisely r from v. However, in our proof we will construct graphs of girth at least $2r + 2$, and hence there are no such edges.

2.3. Model ID

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 $n = |\mathcal{G}|$ has $V(\mathcal{G}) \subseteq \{1, 2, \ldots, \mathrm{poly}(n)\}$ where by $\mathrm{poly}(n)$ we denote some fixed polynomial function of n; our presentation assumes $\mathrm{poly}(n) \gg n$.

2.4. Model OI

Local OI-algorithms $\mathbf{A}$ do not directly use the numerical values of the 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 $\tau(\mathcal{G}, <, v)$ the restriction of the structure $(\mathcal{G}, <, v)$ to the radius-r ball $B_{\mathcal{G}}(v, r)$, i.e., in symbols, $\tau(\mathcal{G}, <, v) := (\mathcal{G}, <, v) \restriction B_{\mathcal{G}}(v, r)$. Then, the output $\mathbf{A}(\mathcal{G}, <, v)$ depends only on the *isomorphism type* of $\tau(\mathcal{G}, <, v)$, so that if $\tau(\mathcal{G}, <, v) \cong \tau(\mathcal{G}', <', v')$ then $\mathbf{A}(\mathcal{G}, <, v) = \mathbf{A}(\mathcal{G}', <', v')$.

2.5. Model PO

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

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 $v \in V(\mathcal{G})$ assign the incoming edges $(u, v) \in E(\mathcal{G})$ distinct labels and the outgoing edges $(v, w) \in E(\mathcal{G})$

distinct labels; we allow $\ell_{\mathcal{G}}(u,v) = \ell_{\mathcal{G}}(v,w)$. 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 $\ell_{\mathcal{H}}(v,u) = \ell_{\mathcal{G}}(\varphi(v),\varphi(u))$ for all $(v,u) \in E(\mathcal{H})$.

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 4. 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 $|L| \leq \Delta^2$.

Views. The information available to a PO-algorithm computing on a node $v \in V(\mathcal{G})$ in an L-digraph $\mathcal{G}$ is usually modelled as follows [Angluin 1980; Yamashita and Kameda 1996; Suomela 2013b]. The *view* of $\mathcal{G}$ from v is an L-edge-labelled rooted (possibly infinite) directed tree $\mathcal{T} = \mathcal{T}(\mathcal{G},v)$, where the vertices $V(\mathcal{T})$ correspond to all non-backtracking walks on $\mathcal{G}$ starting at v; see Figure 4c. 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 in which neither $\ell\ell^{-1}$ nor $\ell^{-1}\ell$ appear. If $w \in V(\mathcal{T})$ is a walk on $\mathcal{G}$ from v to u, we define $\varphi(w) := u$. In particular, the root of $\mathcal{T}$ is the *empty word* λ with $\varphi(\lambda) = v$. 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.

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

$$\mathbf{B}(\mathcal{G},v) = \mathbf{B}(\tau(\mathcal{T}(\mathcal{G},v))).$$

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 $\mathbf{B}(\mathcal{H},v) = \mathbf{B}(\mathcal{G},\varphi(v))$. The intuition is that nodes in a common fibre are always in the same state during computation as they see the same view.

Notation. The following formalism will become useful. Denote by $(\mathcal{T}^*,\lambda)$ the complete L-labelled directed radius-r tree rooted at λ whose vertices $V(\mathcal{T}^*)$ correspond to walks of length at most r. Every non-leaf vertex in $\mathcal{T}^*$ has an outgoing edge and an incoming edge for each $\ell \in L$; see Figure 5. The output of $\mathbf{B}$ 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 $(\mathcal{T}^*,\lambda) \restriction W = \tau(\mathcal{T}(\mathcal{G},v))$ for some $\mathcal{G} \in \mathcal{F}$ and $v \in V(\mathcal{G})$. Then a function $\mathbf{B}\colon \mathfrak{W} \to \Omega$ defines a PO-algorithm by identifying

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

Note that $(\mathcal{T}^*,\lambda)$ is a regular tree: all non-leaf nodes have degree $2|L|$. However, the subtree $(\mathcal{T}^*,\lambda) \restriction W$ need not be regular.

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 $\tau(\mathcal{G},<,v)$ to be some canonical representative from its isomorphism class.

Definition 3.1. 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

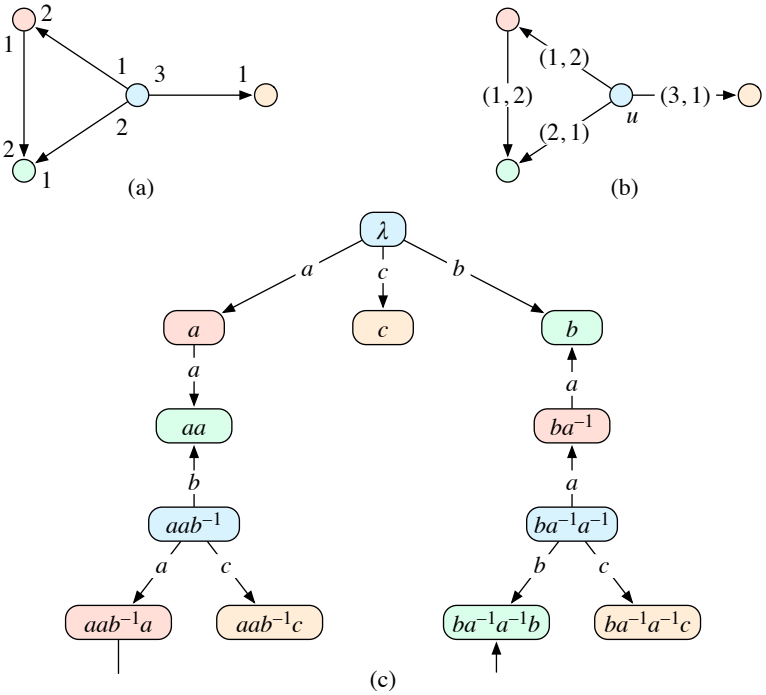


Fig. 4. (a) A graph $\mathcal{G}$ with a port numbering and an orientation. (b) A proper labelling $\ell_{\mathcal{G}}$ that is derived from the port numbering. We have an L-digraph with $L = \{a, b, c\}$, $a = (1, 2)$, $b = (2, 1)$, and $c = (3, 1)$. (c) The view of $\mathcal{G}$ from u is an infinite directed tree $\mathcal{T} = \mathcal{T}(\mathcal{G}, v)$; there is a covering map φ from $\mathcal{T}$ to $\mathcal{G}$ that preserves adjacencies, orientations, and edge labels. For example, $\varphi(\lambda) = \varphi(aab^{-1}) = u$.

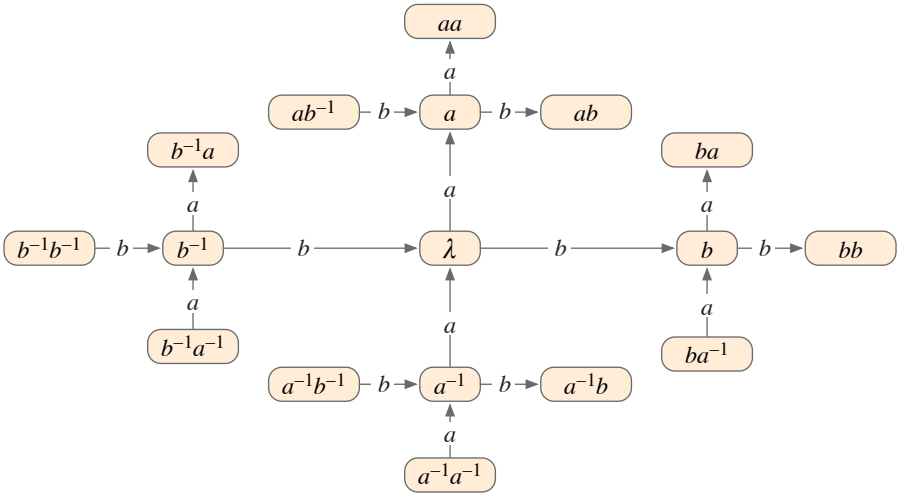


Fig. 5. The complete L-labelled rooted directed tree $(\mathcal{T}^*, \lambda)$ of radius $r = 2$, for $L = \{a, b\}$.

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. For example, if $(\mathcal{G}, <)$ is a non-empty connected ordered graph, and v and u are its smallest and largest vertices, the r-neighbourhoods $\tau(\mathcal{G}, <, v)$ and $\tau(\mathcal{G}, <, u)$ cannot be isomorphic even when $r = 1$. 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 [1995].

3.2. Homogeneous Graphs of Large Girth

As our main technical tool we will use homogeneous graphs of *large girth and degree*. That is, for any setting of the parameters $\epsilon > 0$, r, k, and g we would like to obtain graphs that satisfy the following properties:

(P1) $(1-\epsilon, r)$-homogeneous,
(P2) $2k$-regular,
(P3) girth at least g,
(P4) finite order.

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

(P1, P2, P3) Infinite $2k$-regular trees admit a $(1, r)$-homogeneous linear order; see Figure 6a for an example.
(P1, P2, P4) 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 6b for an example. However, these graphs have girth 4 when $k \geq 2$.
(P1, P3, P4) A sufficiently large directed cycle is $(1-\epsilon, r)$-homogeneous and has large girth. However, all the nodes have degree 2.
(P2, P3, P4) It is well known that regular graphs of arbitrarily high girth exist.

Our construction yields graphs satisfying all four properties simultaneously.

THEOREM 3.2. *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:*

(1) The homogeneity type τ^* of $(\mathcal{H}_\epsilon, <_\epsilon)$ does not depend on ϵ.
(2) The graph $\mathcal{H}_\epsilon$ and the type τ^* are k-edge-labelled digraphs.

The proof of Theorem 3.2 involves constructing a homogeneous infinite locally tree-like graph with *modest growth*. The growth property allows us to cut it down to finite size while introducing only some ϵ-fraction of non-homogenous neighbourhoods. Our algebraic construction is somewhat technical and so we defer it to Section 5.

In what follows, we apply the homogeneous graphs of Theorem 3.2 to control local algorithms—no details of the construction are needed in these applications.

3.3. Homogeneous Lifts

We fix some notation towards a proof of the main theorems. By Theorem 3.2 we let $(\mathcal{H}_\epsilon, <_\epsilon)$, $\epsilon > 0$, be a family of $2|L|$-regular $(1-\epsilon, r)$-homogeneous connected graphs of girth larger than $2r+1$ interpreted as L-digraphs. The homogeneity type τ^* that is

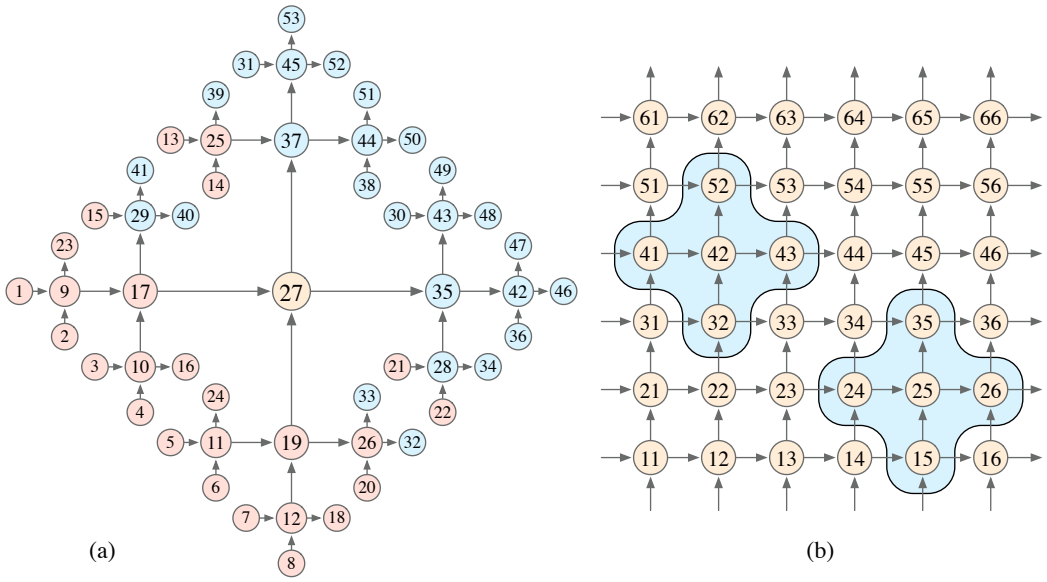


Fig. 6. (a) A fragment of a 4-regular infinite ordered tree $(\mathcal{G}, <)$. The numbering of the nodes indicates a $(1, r)$-homogeneous linear order in the neighbourhood of node 27. (b) A 4-regular graph $\mathcal{G}$ constructed as the cartesian product of two directed 6-cycles. We define the ordered 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.

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 2.5.

We use the graphs $\mathcal{H}_\epsilon$ to construct *homogeneous lifts* of any L-digraph $\mathcal{G}$.

THEOREM 3.3. *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.*

PROOF. For brevity, write $(\mathcal{C}, <_{\mathcal{C}}) = (\mathcal{G}_\epsilon, <_{\mathcal{G}_\epsilon})$ and $(\mathcal{H}, <_{\mathcal{H}}) = (\mathcal{H}_\epsilon, <_\epsilon)$. 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 [1981].

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

$$(h, h') \in E(\mathcal{H}), \quad (g, g') \in E(\mathcal{G}), \quad \ell_{\mathcal{H}}(h, h') = \ell_{\mathcal{G}}(g, g').$$

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.

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

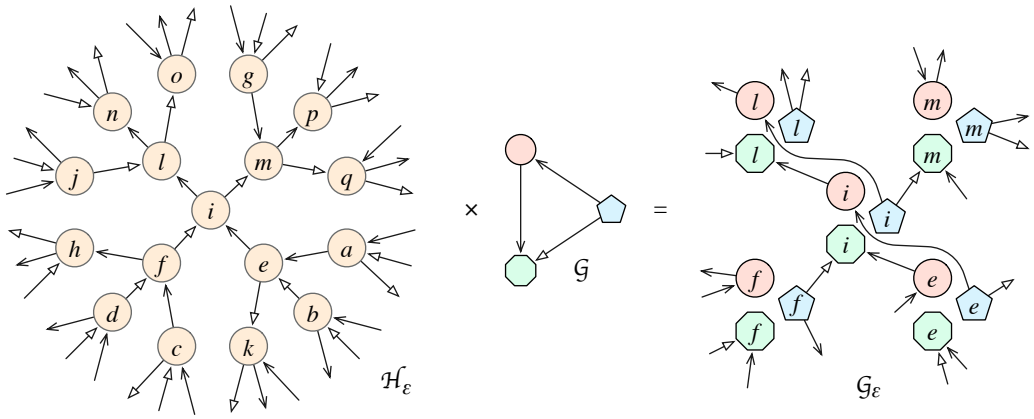


Fig. 7. Homogeneous lifts. In this example $L = |2|$, and the two labels are indicated with two different kinds of arrows. Graph $\mathcal{H}_\epsilon$ is a homogeneous $2|L|$-regular ordered L-digraph with a large girth—in particular, the local neighbourhood of a node looks like a tree. Graph $\mathcal{G}$ 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.

Next, we define a partial order $<_p$ on $V(\mathcal{C})$ as

$$v <_p u \iff \varphi_{\mathcal{H}}(v) <_{\mathcal{H}} \varphi_{\mathcal{H}}(u).$$

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 $\mathcal{C}$. (To see this, let u and v be two incomparable nodes with $\varphi_{\mathcal{H}}(u) = \varphi_{\mathcal{H}}(v) = u'$, and consider a path P of length d from u to v in $\mathcal{C}$. The projection $P' = \varphi_{\mathcal{H}}(P)$ is a non-backtracking walk of length d from u' to u', and we can use P' to construct a cycle of length at most d in $\mathcal{H}$. Hence $d > 2r + 1$.)

We let $<_{\mathcal{C}}$ be any completion of $<_p$ into a linear order. The previous discussion implies that $<_{\mathcal{C}}$ satisfies

$$\tau(\mathcal{C}, <_{\mathcal{C}}, v) = \tau(\mathcal{C}, <_p, v) \quad \text{for all } v \in V(\mathcal{C}).$$

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 $v \in U_{\mathcal{C}}$. By our definition of $<_p$, projection $\varphi_{\mathcal{H}}$ maps the r-neighbourhood $\tau_v := \tau(\mathcal{C}, <_{\mathcal{C}}, v)$ into

$$\tau\big(\mathcal{H}, <_{\mathcal{H}}, \varphi_{\mathcal{H}}(v)\big) \cong \tau^*$$

while preserving the order. But because τ^* is a tree, $\varphi_{\mathcal{H}}$ must be injective on the vertex set of τ_v so that τ_v 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 Theorems 1.3 and 1.4 (see Section 1.6). For clarity of exposition we first prove the theorems in the special case where $\mathbf{A}$ is an OI-algorithm. The subsequent proof for an ID-algorithm $\mathbf{A}$ uses a somewhat technical but well-known Ramsey type argument.

4.1. Special Case of OI-algorithms

We will prove the general version (Theorem 1.3) and connected version (Theorem 1.4) simultaneously; for the proof of the connected version it suffices to consider only connected lifts below. In this section, we do not need the assumption that $\mathcal{F}$ does not contain any trees.

THEOREM 4.1 (MAIN THEOREM FOR OI-ALGORITHMS). *Let $\mathbf{\Pi}$ be a simple* PO*-checkable graph problem. Assume that $\mathcal{F}$ is a family of (connected) bounded-degree graphs, and it is closed under (connected) lifts. If there is an* OI*-algorithm $\mathbf{A}$ with run-time $r = O(1)$ that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$, then there is a* PO*-algorithm $\mathbf{B}$ with run-time r that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$.*

PROOF. Suppose $\mathbf{A}$ is an OI-algorithm with run-time r that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$. Recall that $(\mathcal{T}^*, <^*, \lambda)$ is a complete L-labelled homogeneous tree; it only depends on run-time r and maximum degree Δ. We define the PO-algorithm $\mathbf{B} \colon \mathfrak{W} \to \Omega$ simply by setting

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

Now, Theorem 3.3 translates into saying that for every $\mathcal{G} \in \mathcal{F}$, the new algorithm $\mathbf{B}$ simulates $\mathbf{A}$ almost perfectly on the lift $\mathcal{G}_\epsilon \in \mathcal{F}$:

Fact 4.2. $\mathbf{A}(\mathcal{G}_\epsilon, <_{\mathcal{G}_\epsilon}, v) = \mathbf{B}(\mathcal{G}_\epsilon, v)$ for at least a $1 - \epsilon$ fraction of nodes $v \in V(\mathcal{G}_\epsilon)$.

The claim that $\mathbf{B}$ works as expected follows essentially from this fact as we argue next. For simplicity, we assume the solutions to $\mathbf{\Pi}$ are sets of vertices so that $\mathbf{A}(\mathcal{G}) \subseteq V(\mathcal{G})$; 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.

Feasibility. Let us first show that algorithm $\mathbf{B}$ finds a feasible solution of $\mathbf{\Pi}$ on $\mathcal{G}$. Let $\mathbf{V}$ be a local PO-algorithm with run-time $t \in \mathbb{N}$ that verifies the feasibility of a solution for $\mathbf{\Pi}$. For $\epsilon > 0$ sufficiently small, each $v \in V(\mathcal{G})$ has a pre-image $v' \in \varphi_\epsilon^{-1}(v)$ such that $\mathbf{A}$ and $\mathbf{B}$ agree on the vertices

$$\bigcup_{v \in V(\mathcal{G})} B_{\mathcal{G}_\epsilon}(v', t).$$

Thus, $\mathbf{V}$ accepts the solution $\mathbf{B}(\mathcal{G}_\epsilon)$ on the vertices v'. But because $\varphi_\epsilon(\{v' : v \in V(\mathcal{G})\}) = V(\mathcal{G})$ it follows that $\mathbf{V}$ accepts the solution $\mathbf{B}(\mathcal{G}) = \varphi_\epsilon(\mathbf{B}(\mathcal{G}_\epsilon))$ on every node in $\mathcal{G}$.

Approximation. Next, we show that algorithm $\mathbf{B}$ finds an α-approximation of $\mathbf{\Pi}$ on $\mathcal{G}$. We assume $\mathbf{\Pi}$ 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 $\mathbf{\Pi}$.

The solution $\mathbf{B}(\mathcal{G}_\epsilon)$ is a disjoint union of some fibres $\varphi_\epsilon^{-1}(v)$, each fibre comprising of exactly a $1/|\mathcal{G}|$ fraction of the nodes of $\mathcal{G}_\epsilon$. It is useful to express Fact 4.2 fibre-wise as

Fact 4.3. $\mathbf{A}(\mathcal{G}_\epsilon, <_{\mathcal{G}_\epsilon}, u) = \mathbf{B}(\mathcal{G}_\epsilon, u)$ for at least a $(1 - \epsilon|\mathcal{G}|)$-fraction of nodes $u \in \varphi_\epsilon^{-1}(v)$.

Thus, for each fibre $\varphi_\epsilon^{-1}(v) \subseteq \mathbf{B}(\mathcal{G}_\epsilon)$, we have that at least a $(1 - \epsilon|\mathcal{G}|)$-fraction of the nodes $\varphi_\epsilon^{-1}(v)$ are included in $\mathbf{A}(\mathcal{G}_\epsilon)$, too. Summing up over all the fibres in $\mathbf{B}(\mathcal{G}_\epsilon)$, we get that

$$|\mathbf{A}(\mathcal{G}_\epsilon)| \geq (1 - \epsilon|\mathcal{G}|) \cdot |\mathbf{B}(\mathcal{G}_\epsilon)|.$$

We can now calculate

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

where the first equality is true because all fibres have the same size, and the first inequality follows from $\varphi_\epsilon^{-1}(\mathbf{B}(\mathcal{G})) = \mathbf{B}(\mathcal{G}_\epsilon)$ and the fact that $\varphi_\epsilon^{-1}(X)$ is a feasible solution so that $|X_\epsilon| \le |\varphi_\epsilon^{-1}(X)|$. Since the above inequality holds for every $\epsilon > 0$ we must have that $|\mathbf{B}(\mathcal{G})|/|X| \le \alpha$, as desired. □

4.2. General Case of ID-algorithms

Next, we will extend the above proof of to handle the case when $\mathbf{A}$ is an ID-algorithm.

To do this we use the Ramsey technique of Naor and Stockmeyer [1995]; see also Czygrinow et al. [2008]. In this technique the algorithm $\mathbf{A}$ is forced into producing an output that depends only on the relative order of the identifiers—this is an ID = OI type result—while maintaining the values of the identifiers below $\mathrm{poly}(n)$. In our case, all of this analysis is done on the locally tree-like lifts $\mathcal{G}_\epsilon$ in which an OI = PO type result holds as we already established above.

For a reference on Ramsey's theorem see Graham et al. [1980].

Notation. If $(X, <_X)$ and $(Y, <_Y)$ are linearly ordered sets with $|X| \le |Y|$, 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.

We also write $\Omega^{\mathfrak{W}}$ for the family of functions $\mathfrak{W} \to \Omega$; recall that each $\mathbf{B} \in \Omega^{\mathfrak{W}}$ can be interpreted as a PO-algorithm. We set $k := |\Omega^{\mathfrak{W}}|$ and $t := |\mathcal{T}^*|$. Also, we denote by $\mathbb{N}^{(t)}$ the subsets of $\mathbb{N}$ of size t; every t-subset $S \in \mathbb{N}^{(t)}$ is ordered by the usual order $<$ on $\mathbb{N}$.

Proof of main theorems. Suppose $\mathbf{A}$ is an ID-algorithm with run-time r that finds an α-approximation of $\mathbf{\Pi}$ in $\mathcal{F}$.

For $W \in \mathfrak{W}$ and $S \in \mathbb{N}^{(t)}$ we let

$$f_{W,S}\colon (W, <^*) \hookrightarrow (S, <)$$

so that the vertex-relabelled tree $f_{W,S}((\mathcal{T}^*, \lambda) \restriction W)$ has the $|W|$ smallest numbers in S as vertices. Define a k-colouring $c\colon \mathbb{N}^{(t)} \to \Omega^{\mathfrak{W}}$ by setting

$$c(S)(W) := \mathbf{A}\big(f_{W,S}\big((\mathcal{T}^*, \lambda) \restriction W\big)\big).$$

For each $m \ge t$ we can use Ramsey's theorem to obtain a number $R(m) \ge m$, 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

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

there exist an m-subset $J(m,i) \subseteq I(m,i)$ and a colour $\mathbf{B}_{m,i} \in \Omega^{\mathfrak{W}}$ such that $c(S) = \mathbf{B}_{m,i}$ for all t-subsets $S \subseteq J(m,i)$. Each $\mathbf{B}_{m,i} \in \Omega^{\mathfrak{W}}$ can be interpreted as a PO-algorithm.

This construction has the following property.

PROPOSITION 4.4. *Suppose $m \ge |\mathcal{G}_\epsilon| + t$. Algorithms $\mathbf{A}$ and $\mathbf{B}_{m,i}$ agree 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 3.3, let $U \subseteq V(f_{m,i}(\mathcal{G}_\epsilon))$, $|U| \ge (1-\epsilon)|\mathcal{G}_\epsilon|$, be the set of vertices v with $\tau(f_{m,i}(\mathcal{G}_\epsilon), <, v)$ isomorphic to a subtree of τ^*. In particular, for a fixed $v \in U$ we can choose $W \in \mathfrak{W}$ such that

$$\tau(f_{m,i}(\mathcal{G}_\epsilon), <, v) \cong (\mathcal{T}^*, <^*, \lambda) \restriction W.$$

Now, as m is large, there exists a t-set $S \subseteq J(m,i)$ such that

$$\tau(f_{m,i}(\mathcal{G}_\epsilon), v) = f_{W,S}((\mathcal{T}^*, \lambda) \restriction W).$$

Thus, $\mathbf{A}$ and $\mathbf{B}_{m,i}$ agree on v by the definition of $\mathbf{B}_{m,i}$. $\square$

For every $l \in \mathbb{N}$ some algorithm appears with density at least $1/k$ (i.e., appears at least l/k times) in the sequence

$$\mathbf{B}_{m,1}, \mathbf{B}_{m,2}, \ldots, \mathbf{B}_{m,l}.$$

Hence, let $\mathbf{B}_m$ be an algorithm that appears with density at least $1/k$ among these sequences for infinitely many l. Let $\mathbf{B}$ be an algorithm appearing among the $\mathbf{B}_m$ for infinitely many m. We claim $\mathbf{B}$ satisfies Theorems 1.3 and 1.4. In fact, the theorems follow from the considerations of Section 4.1 when Fact 4.2 is replaced with the following analogous proposition.

PROPOSITION 4.5. *For every $\mathcal{G}_\epsilon$ there exists an n-node lift $\mathcal{H}$ of $\mathcal{G}_\epsilon$ such that*

— *$V(\mathcal{H}) \subseteq \{1, 2, \ldots, \mathrm{poly}(n)\}$, and*
— *$\mathbf{A}(\mathcal{H}, v) = \mathbf{B}(\mathcal{H}, v)$ for a $1 - \epsilon$ fraction of nodes $v \in V(\mathcal{H})$.*

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 $\mathbf{B} = \mathbf{B}_m$. For infinitely many l there exists an l-set $I \subseteq [lk]$ of indices such that $\mathbf{B} = \mathbf{B}_{m,i}$ for $i \in I$. Consider the following l-lift of $\mathcal{G}_\epsilon$ obtained by taking disjoint unions:

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

Algorithms $\mathbf{A}$ and $\mathbf{B}$ agree on a $1 - \epsilon$ fraction of the nodes in $\mathcal{H}$ by Proposition 4.4. Furthermore, we have that

$$n = |\mathcal{H}| = l|\mathcal{G}_\epsilon| \qquad \text{and} \qquad V(\mathcal{H}) \subseteq \{1, 2, \ldots, lkR(m)\}.$$

Recall from Section 2.3 that we are assuming $\mathrm{poly}(n) \gg n$, so choosing a large enough l 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 $e = (v, u) \in E(\mathcal{G}_\epsilon)$ 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 $\{v_i\}_{i \in I}$ of v and the fibre $\{u_i\}_{i \in I}$ of u. Namely, let π be a cyclic permutation on I and set

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

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

Remark 4.6. Above, we assumed that instances $\mathcal{G}$ have node identifiers $V(\mathcal{G}) \subseteq \{1, 2, \ldots, \mathrm{poly}(n)\}$, for $\mathrm{poly}(n) \gg n$. By choosing identifiers more economically as in the work of Czygrinow et al. [2008] one can show lower bounds for the graph problems in Sections 1.4 and 1.5 even when $V(\mathcal{G}) = \{1, 2, \ldots, n\}$.

5. CONSTRUCTION OF HOMOGENEOUS GRAPHS OF LARGE GIRTH

In this section we prove Theorem 3.2 (see Section 3.2). 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 [1995].

5.1. Group Terminology

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

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 $\mathcal{C}(G,\varphi(S))$.

5.2. The Construction

Our strategy for constructing a $(1-\epsilon, r)$-homogeneous graph $(\mathcal{H}, <)$ is as follows.

(1) We start with an infinite $(1,\infty)$-homogeneous graph $(\mathcal{U}, <)$ of large girth and degree (recall properties P1, P2, and P3 from Section 3.2).
(2) Then, we delete all but a finite fragment of $\mathcal{U}$ to end up with a subgraph $\mathcal{H} \subseteq \mathcal{U}$ (thus, making property P4 hold).

In cutting away $\mathcal{H}$ we necessarily introduce r-neighbourhoods that witness the *boundary* between $\mathcal{H}$ and the rest of $\mathcal{U}$. To ensure the $(1-\epsilon, r)$-homogeneity of $\mathcal{H}$ only an ϵ fraction of the nodes in $\mathcal{H}$ can lie near this boundary. For example, if $\mathcal{U} = \mathcal{T}_k$ is the infinite $2k$-regular digraph (corresponding to a Cayley graph of the free group on k generators) we can never cut away a subgraph $\mathcal{H} \subseteq \mathcal{T}_k$ without leaving too large a boundary. This is because $\mathcal{T}_k$ is too good an expander: the radius-r balls $B_{\mathcal{T}_k}(v,r)$ grow at an exponential rate (the free group has exponential growth). Hence, to implement our strategy, we should choose $\mathcal{U}$ to be a Cayley graph of (a group of) polynomial growth.

The famous characterisation of polynomial growth groups due to Gromov [1981] narrows our search; we end up choosing as $\mathcal{U}$ a Cayley graph of a soluble group that has an abelian subgroup of finite index. For example, the toroidal graphs (of girth 4) mentioned in Section 3.2 are finite versions of Cayley graphs of the free abelian groups $\mathbb{Z}^k$. Analogously, we will use the decomposition of our soluble group into cyclic factors to guarantee the presence of a homogeneous ordering. In addition, to ensure large girth, our soluble groups must be sufficiently far from being abelian, i.e., they must have large derived length [Conder et al. 2010].

PROOF OF THEOREM 3.2. Let $m \in \mathbb{N}$ 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:

$$\begin{aligned} H_1 &:= \mathbb{Z}_m, & W_1 &:= \mathbb{Z}_2, & U_1 &:= \mathbb{Z}, \\ H_{i+1} &:= H_i^2 \rtimes \mathbb{Z}_m, & W_{i+1} &:= W_i^2 \rtimes \mathbb{Z}_2, & U_{i+1} &:= U_i^2 \rtimes \mathbb{Z}. \end{aligned}$$

Here, the cyclic group $\mathbb{Z}_m = \{0,1,\ldots,m-1\}$ acts on the direct product $H_i^2 = H_i \times H_i$ by cyclically permuting the coordinates, i.e., the subgroup $2\mathbb{Z}_m \leq \mathbb{Z}_m$ acts trivially and the elements in $1 + 2\mathbb{Z}_m$ 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.

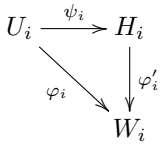
The groups W_i are i-fold iterated regular wreath products of the cyclic group $\mathbb{Z}_2$. These groups have order $|W_i| = 2^{d(i)}$ and they are sometimes called *symmetric 2-groups*; they

are isomorphic to the Sylow 2-subgroups of the symmetric group on 2^i letters [Rotman 1995, p. 176].

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 $(2\mathbb{Z})^{d(i)} \cong \mathbb{Z}^{d(i)}$.

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 m 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:



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.

Girth. Gamburd et al. [2009] 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 [Gamburd et al. 2009, Theorem 6].

THEOREM 5.1 (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.

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., Conrad 1959], but it is easy enough to verify that setting

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

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 $(\mathcal{U}, <, v)$, $v \in U$, are pairwise iso-

morphic. A fortiori, the r-neighbourhoods $\tau(\mathcal{U}, <, v)$, $v \in U$, are all pairwise isomorphic. Let τ^* be this common r-neighbourhood isomorphism type.

Transferring the linear order on U to $\mathcal{H}$. Let $V(\mathcal{H})$ be ordered by restricting the order $<$ on U to the set $V(\mathcal{H}) = \mathbb{Z}_m^d$ 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 use the polynomial growth property of U to check that, as $m \to \infty$, almost all $v \in V(\mathcal{H})$ have r-neighbourhoods of type τ^*.

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

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

The right multiplication action of $s \in S \cup S^{-1}$ on v can be described in two steps as follows: First, the coordinates of s are permuted (as determined by v) to obtain a vector s'. Then, vs is given as the standard addition of the vectors v and s' in $\mathbb{Z}^d \subseteq \mathbb{R}^d$. Hence, $vs \in v + [-1, 1]^d$, and moreover,

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

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

Consider the set of inner nodes $I := [r, (m-1) - r]^d$. Let $v \in I$. By (2), the vertex set $B_{\mathcal{U}}(v, r)$ is contained in $\mathbb{Z}_m^d$. This implies that the cover map ψ is the identity on $B_{\mathcal{U}}(v, r)$ and consequently the r-neighbourhood $\tau(\mathcal{H}, <, v)$ contains the ordered tree

$$\tau(\mathcal{U}, <, v) \cong \tau^*.$$

If $\tau(\mathcal{H}, <, v)$ had any edges in addition to those of $\tau(\mathcal{U}, <, v)$, this would entail a cycle of length at most $2r + 1$ in $\mathcal{H}$, which is not possible. Thus, $\tau(\mathcal{H}, <, v) \cong \tau^*$. The density of elements in $\mathcal{H}$ having r-neighbourhood type τ^* is therefore at least

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

for a large m.

Finally, to establish Theorem 3.2 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. DISCUSSION

We have shown that, under mild assumptions, deterministic local algorithms do not benefit from unique identifiers: in order to show run-time or approximability lower bounds for local algorithms in the ID model, it suffices to do so in the PO model.

To conclude this work, we give some further references to related work and explore what happens when we step outside our scope of deterministic local algorithms and simple PO-checkable graph problems.

6.1. Models of Computation: Going Below PO

The ID model was introduced by Linial [1992] and it is commonly known as the LOCAL model, following the terminology introduced by Peleg [2000]. The OI model was introduced in the context of local algorithms by Naor and Stockmeyer [1995], and the PO model has been used, e.g., in Mayer et al. [1995].

There are also models of computation that are strictly weaker than PO. The most widely-studied such example is the port-numbering model (without orientations); let us call it PN. The concept of a port numbering was introduced by Angluin [1980], albeit in the context of a different model of computation. Distributed algorithms in the PN model

have been studied, e.g., by Attiya et al. [1988] and Yamashita and Kameda [1996]. It is easy to show that our main theorem cannot be extended from the PO model to the PN model. To see this, consider the following graph family: 3-regular graphs that admit an edge colouring with three colours. In such a graph we can use the edge colouring to assign the port numbers so that all local views are isomorphic; in particular, we cannot produce a nontrivial dominating set. However, in the PO model any orientation will always break symmetry—we can use the local algorithm by Naor and Stockmeyer [1995] to find a weak colouring, which allows us to produce a nontrivial dominating set.

The separation of the PO model and the PN model is only the first step in a natural hierarchy of models that are strictly weaker than PO. We refer to Boldi et al. [1996] and Yamashita and Kameda [1999] for pioneering work in this area, and to the recent work by Hella et al. [2012] for a classification of the weak models from the perspective of local algorithms.

6.2. Running Time: Parameters n and Δ

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.

The study of local algorithms—distributed algorithms that have running time independent of n—was initiated by Naor and Stockmeyer [1995]. The research area has been very active during the past ten years, and the results that we mentioned in Section 1.4 are merely a selection of upper and lower bounds related to local algorithms. For up-to-date information on local algorithms, we refer to the survey [Suomela 2013b] and its online supplement [Suomela 2013a].

Dependence on n. The requirement that the running time is strictly independent of n cannot be relaxed too much in our main theorem. As we mentioned in Section 1.1, the models ID, OI, and PO are easy to separate as soon as we allow a running time of $\Theta(\log^* n)$. With this running time, we can interpret the unique node identifiers in the ID model as a graph colouring with $\mathrm{poly}(n)$ colours, and we can then apply fast *colour reduction* algorithms to reduce the colour space, e.g., from $\mathrm{poly}(n)$ to $\mathrm{poly}(\Delta)$. For example, in the case of a cycle, the colour reduction techniques by Cole and Vishkin [1986] and Naor and Stockmeyer [1995] can be used to find a 3-colouring in $O(\log^* n)$ rounds, and once we have a 3-colouring, we can easily find a non-trivial approximation of a vertex cover, edge cover, independent set, dominating set, matching, or edge dominating set in constant time. Equally good approximations require linear time in the OI model and they are not possible at all in the PO model.

There are many $\Theta(\log^* n)$-time algorithms that are based on some variant of the colour reduction idea. It is an open question whether there is a simple graph problem that can be solved with a $\Theta(\log^* n)$-time algorithm that is not based on a colour reduction technique, but nevertheless makes an essential use of the numerical values of the identifiers.

Dependence on Δ. Prior work that studies the running time of a local algorithm as a function of Δ includes the upper and lower bounds by Kuhn et al. [2004; 2006b; 2010]. These bounds on approximation ratios apply to algorithms that have, for example, a running time sublogarithmic in Δ. There are also recent lower bounds that apply to algorithms with a running time sublinear in Δ [Hirvonen and Suomela 2012].

In this work, the equivalence of ID and PO was established via a simulation argument that preserves the running time of a local algorithm. We hope this opens up a new avenue for proving run-time lower bounds as a function of Δ in the standard ID model. For example, we note that the exact time-complexity of the weak 2-colouring problem is still open.

6.3. Beyond Simple Graph Problems

Our work, as well as the prior work by Naor and Stockmeyer [1995], focuses on problems in which the *size of the local output is bounded*. To keep the presentation easy to follow, we focused on simple graph problems in which there is only one bit of output per node or edge. It is straightforward to generalise the results to cover labellings in which we have multiple bits of output per node and/or edge; however, the number of bits has to be bounded in order to apply the Ramsey technique.

This restriction is not merely a proof artefact. The recent work by Hasemann et al. [2012] gives an example of a natural graph problem—fractional graph colouring—that admits a local $O(1)$-approximation algorithm in the ID model but not in the OI or PO models. The work circumvents the assumptions of our main theorem by exploiting the fact the local outputs can be arbitrarily long: in the fractional graph colouring problem, each node produces a schedule that indicates when the node is active, and the schedule can be as fine-grained as necessary.

While our focus is on *optimisation problems*, the role of unique identifiers is also being explored in the context of *decision problems*. When we study decision problems in a distributed setting, we say that a local algorithm solves a decision problem $\mathbf{\Pi}$ if each yes-instance is accepted by all nodes and each no-instance is rejected by at least one node (recall our definition of a PO-checkable graph problem). Fraigniaud et al. [2012] study two models, LD and LD*, which are closely related to the decision versions of our models ID and OI, respectively. In many cases, models LD and LD* turn out to be equally expressive.

6.4. Beyond Lift-closed Families

Our techniques rely extensively on the use of graph lifts; we fooled ID-algorithms by presenting them with locally tree-like unfoldings of the original input graphs.

However, many interesting graph families are not closed under lifts. Planar graphs are a case in point: there is a local $O(1)$-approximation algorithm for the minimum dominating set problem in the ID and OI models [Czygrinow et al. 2008; Lenzen 2011, §13], but no such PO-algorithm is known.

Open problem 6.1. Is there a PO-algorithm for computing an $O(1)$-approximation of the minimum dominating set problem on planar graphs?

6.5. Determinism vs. Randomness

One direction that we have not discussed at all is the case of randomised algorithms. It is well-known that equipping each node with a random source strictly increases the power of local algorithms.

Upper bounds. Many of the lower bounds cited in Section 1.4 do not apply for randomised local algorithms. Non-trivial approximations (in expectation or w.h.p.) can be achieved for matchings [Wattenhofer and Wattenhofer 2004; Hoepman et al. 2006; Nguyen and Onak 2008] and independent sets [Czygrinow et al. 2008], while no such deterministic algorithms exist.

One very general tool in this context is randomised LP rounding [Kuhn 2005; Kuhn et al. 2006a; Kuhn et al. 2006b; Kuhn and Wattenhofer 2005]. By combining local LP approximation schemes and randomised LP rounding, we can find, for example, an $O(\log \Delta)$-approximation of a minimum dominating set, beating the $\Omega(\Delta)$ lower bound for deterministic algorithms.

Lower bounds. In the presence of randomness there is no longer any distinction between the models ID, OI, and PO, as the random bits can be used to generate unique identifiers w.h.p. (at least if the nodes know an upper bound on n). However, this simple

observation does little to help us in understanding the limitations of randomised local algorithms. In fact, for many problems, tight randomised local approximability bounds are still missing.

Open problem 6.2. Prove lower bounds against randomised local algorithms on bounded-degree graphs, e.g., for the problems in Sections 1.4 and 1.5.

There are also graph problems in which randomness does not help at all. Vertex colouring in cycles is one example of such a problem: Naor [1991] generalised Linial's [1992] tight lower bound to randomised algorithms. It is also known that any randomised local algorithm for linear programs can be derandomised: in essence, a deterministic local algorithm can compute the expected output value of a randomised local algorithm [Kuhn 2005, §2.7.1].

ACKNOWLEDGMENTS

This is an extended and revised version of a preliminary conference report [Göös et al. 2012]. We thank the anonymous reviewers for their helpful feedback on the conference version, 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.

REFERENCES

ANGLUIN, D. 1980. Local and global properties in networks of processors. In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing* (STOC, Los Angeles, CA, USA, April 1980). ACM Press, New York, 82–93.

ANGLUIN, D. AND GARDINER, A. 1981. Finite common coverings of pairs of regular graphs. *Journal of Combinatorial Theory, Series B 30,* 2, 184–187.

ÅSTRAND, M., FLORÉEN, P., POLISHCHUK, V., RYBICKI, J., SUOMELA, J., AND UITTO, J. 2009. A local 2-approximation algorithm for the vertex cover problem. In *Proceedings of the 23rd International Symposium on Distributed Computing* (DISC, Elche, Spain, September 2009). LNCS Series, vol. 5805. Springer, Berlin, 191–205.

ÅSTRAND, M., POLISHCHUK, V., RYBICKI, J., SUOMELA, J., AND UITTO, J. 2010. Local algorithms in (weakly) coloured graphs. Manuscript, arXiv:1002.0125 [cs.DC].

ÅSTRAND, M. AND SUOMELA, J. 2010. Fast distributed approximation algorithms for vertex cover and set cover in anonymous networks. In *Proceedings of the 22nd Annual ACM Symposium on Parallelism in Algorithms and Architectures* (SPAA, Santorini, Greece, June 2010). ACM Press, New York, 294–302.

ATTIYA, H., SNIR, M., AND WARMUTH, M. K. 1988. Computing on an anonymous ring. *Journal of the ACM 35,* 4, 845–875.

BOLDI, P., SHAMMAH, S., VIGNA, S., CODENOTTI, B., GEMMELL, P., AND SIMON, J. 1996. Symmetry breaking in anonymous networks: characterizations. In *Proceedings of the 4th Israel Symposium on the Theory of Computing and Systems* (ISTCS, Jerusalem, Israel, June 1996). IEEE, Piscataway, 16–26.

CHLEBÍK, M. AND CHLEBÍKOVÁ, J. 2006. Approximation hardness of edge dominating set problems. *Journal of Combinatorial Optimization 11,* 3, 279–290.

COLE, R. AND VISHKIN, U. 1986. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control 70,* 1, 32–53.

CONDER, M., EXOO, G., AND JAJCAY, R. 2010. On the limitations of the use of solvable groups in Cayley graph cage constructions. *European Journal of Combinatorics 31*, 1819–1828.

CONRAD, P. 1959. Right-ordered groups. *The Michigan Mathematical Journal 6*, 267–275.

CZYGRINOW, A., HAŃĆKOWIAK, M., AND WAWRZYNIAK, W. 2008. Fast distributed approximations in planar graphs. In *Proceedings of the 22nd International Symposium on Distributed Computing* (DISC, Arcachon, France, September 2008). LNCS Series, vol. 5218. Springer, Berlin, 78–92.

FRAIGNIAUD, P., HALLDORSSON, M. M., AND KORMAN, A. 2012. On the impact of identifiers on local decision. In *Proceedings of the 16th International Conference on Principles of Distributed Systems* (OPODIS, Rome, Italy, December 2012). LNCS Series, vol. 7702. Springer, Berlin, 224–238.

GAMBURD, A., HOORY, S., SHAHSHAHANI, M., SHALEV, A., AND VIRÁG, B. 2009. On the girth of random Cayley graphs. *Random Structures & Algorithms 35,* 1, 100–117.

GÖÖS, M., HIRVONEN, J., AND SUOMELA, J. 2012. Lower bounds for local approximation. In *Proceedings of the 31st Annual ACM Symposium on Principles of Distributed Computing* (PODC, Madeira, Portugal, July 2012). ACM Press, New York, 175–184.

GRAHAM, R. L., ROTHSCHILD, B. L., AND SPENCER, J. H. 1980. *Ramsey Theory*. John Wiley & Sons, New York.

GROMOV, M. 1981. Groups of polynomial growth and expanding maps. *Publications Mathématiques de L'IHÉS 53*, 53–78.

HASEMANN, H., HIRVONEN, J., RYBICKI, J., AND SUOMELA, J. 2012. Deterministic local algorithms, unique identifiers, and fractional graph colouring. In *Proceedings of the 19th International Colloquium on Structural Information and Communication Complexity* (SIROCCO, Reykjavík, Iceland, June–July 2012). LNCS Series, vol. 7355. Springer, Berlin, 48–60.

HELLA, L., JÄRVISALO, M., KUUSISTO, A., LAURINHARJU, J., LEMPIÄINEN, T., LUOSTO, K., SUOMELA, J., AND VIRTEMA, J. 2012. Weak models of distributed computing, with connections to modal logic. In *Proceedings of the 31st Annual ACM Symposium on Principles of Distributed Computing* (PODC, Madeira, Portugal, July 2012). ACM Press, New York, 185–194.

HIRVONEN, J. AND SUOMELA, J. 2012. Distributed maximal matching: greedy is optimal. In *Proceedings of the 31st Annual ACM Symposium on Principles of Distributed Computing* (PODC, Madeira, Portugal, July 2012). ACM Press, New York, 165–174.

HOEPMAN, J.-H., KUTTEN, S., AND LOTKER, Z. 2006. Efficient distributed weighted matchings on trees. In *Proceedings of the 13th International Colloquium on Structural Information and Communication Complexity* (SIROCCO, Chester, UK, July 2006). LNCS Series, vol. 4056. Springer, Berlin, 115–129.

KUHN, F. 2005. The price of locality: Exploring the complexity of distributed coordination primitives. Ph.D. thesis, ETH Zurich.

KUHN, F., MOSCIBRODA, T., AND WATTENHOFER, R. 2004. What cannot be computed locally! In *Proceedings of the 23rd Annual ACM Symposium on Principles of Distributed Computing* (PODC, St. John's, Newfoundland, Canada, July 2004). ACM Press, New York, 300–309.

KUHN, F., MOSCIBRODA, T., AND WATTENHOFER, R. 2006a. Fault-tolerant clustering in ad hoc and sensor networks. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems* (ICDCS, Lisboa, Portugal, July 2006). IEEE Computer Society Press, Los Alamitos.

KUHN, F., MOSCIBRODA, T., AND WATTENHOFER, R. 2006b. The price of being near-sighted. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms* (SODA, Miami, FL, USA, January 2006). ACM Press, New York, 980–989.

KUHN, F., MOSCIBRODA, T., AND WATTENHOFER, R. 2010. Local computation: Lower and upper bounds. Manuscript, arXiv:1011.5470 [cs.DC].

KUHN, F. AND WATTENHOFER, R. 2005. Constant-time distributed dominating set approximation. *Distributed Computing 17,* 4, 303–310.

LENZEN, C. 2011. Synchronization and symmetry breaking in distributed systems. Ph.D. thesis, ETH Zurich.

LENZEN, C. AND WATTENHOFER, R. 2008. Leveraging Linial's locality limit. In *Proceedings of the 22nd International Symposium on Distributed Computing* (DISC, Arcachon, France, September 2008). LNCS Series, vol. 5218. Springer, Berlin, 394–407.

LINIAL, N. 1992. Locality in distributed graph algorithms. *SIAM Journal on Computing 21,* 1, 193–201.

MAYER, A., NAOR, M., AND STOCKMEYER, L. 1995. Local computations on static and dynamic graphs. In *Proceedings of the 3rd Israel Symposium on the Theory of Computing and Systems* (ISTCS, Tel Aviv, Israel, January 1995). IEEE, Piscataway, 268–278.

NAOR, M. 1991. A lower bound on probabilistic algorithms for distributive ring coloring. *SIAM Journal on Discrete Mathematics 4,* 3, 409–412.

NAOR, M. AND STOCKMEYER, L. 1995. What can be computed locally? *SIAM Journal on Computing 24,* 6, 1259–1277.

NGUYEN, H. N. AND ONAK, K. 2008. Constant-time approximation algorithms via local improvements. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science* (FOCS, Philadelphia, PA, USA, October 2008). IEEE Computer Society Press, Los Alamitos, 327–336.

PELEG, D. 2000. *Distributed Computing: A Locality-Sensitive Approach*. SIAM Monographs on Discrete Mathematics and Applications. SIAM, Philadelphia.

ROTMAN, J. J. 1995. *An Introduction to the Theory of Groups*. Graduate Texts in Mathematics Series, vol. 148. Springer, New York.

SUOMELA, J. 2010. Distributed algorithms for edge dominating sets. In *Proceedings of the 29th Annual ACM Symposium on Principles of Distributed Computing* (PODC, Zurich, Switzerland, July 2010). ACM Press, New York, 365–374.

SUOMELA, J. 2013a. A bibliography of local algorithms. http://www.cs.helsinki.fi/local-survey/

SUOMELA, J. 2013b. Survey of local algorithms. *ACM Computing Surveys 45,* 2, 24:1–40.

WATTENHOFER, M. AND WATTENHOFER, R. 2004. Distributed weighted matching. In *Proceedings of the 18th International Symposium on Distributed Computing* (DISC, Amsterdam, Netherlands, October 2004). LNCS Series, vol. 3274. Springer, Berlin, 335–348.

YAMASHITA, M. AND KAMEDA, T. 1996. Computing on anonymous networks: Part I—characterizing the solvable cases. *IEEE Transactions on Parallel and Distributed Systems 7,* 1, 69–89.

YAMASHITA, M. AND KAMEDA, T. 1999. Leader election problem on networks in which processor identity numbers are not distinct. *IEEE Transactions on Parallel and Distributed Systems 10,* 9, 878–887.

YANNAKAKIS, M. AND GAVRIL, F. 1980. Edge dominating sets in graphs. *SIAM Journal on Applied Mathematics 38,* 3, 364–372.