The Topology of Local Computing in Networks

Pierre Fraigniaud
Institut de Recherche en Informatique Fondamentale, CNRS, Université de Paris, France

Ami Paz
Faculty of Computer Science, Universität Wien, Austria

Abstract

Modeling distributed computing in a way enabling the use of formal methods is a challenge that has been approached from different angles, among which two techniques emerged at the turn of the century: protocol complexes, and directed algebraic topology. In both cases, the considered computational model generally assumes communication via shared objects (typically a shared memory consisting of a collection of read-write registers), or message-passing enabling direct communication between any pair of processes. Our paper is concerned with network computing, where the processes are located at the nodes of a network, and communicate by exchanging messages along the edges of that network (only neighboring processes can communicate directly).

Applying the topological approach for verification in network computing is a considerable challenge, mainly because the presence of identifiers assigned to the nodes yields protocol complexes whose size grows exponentially with the size of the underlying network. However, many of the problems studied in this context are of local nature, and their definitions do not depend on the identifiers or on the size of the network. We leverage this independence in order to meet the above challenge, and present *local* protocol complexes, whose sizes do not depend on the size of the network. As an application of the design of "compacted" protocol complexes, we reformulate the celebrated lower bound of $\Omega(\log^* n)$ rounds for 3-coloring the n-node ring, in the algebraic topology framework.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases Distributed computing, distributed graph algorithms, combinatorial topology

Digital Object Identifier 10.4230/LIPIcs.ICALP.2020.128

Category Track B: Automata, Logic, Semantics, and Theory of Programming

Related Version `https://arxiv.org/pdf/2003.03255.pdf`

Funding *Pierre Fraigniaud*: Additional support from ANR projects DESCARTES and FREDDA, and INRIA project GANG.
Ami Paz: This research was done during the stay of the second author at Institut de Recherche en Informatique Fondamentale (IRIF), supported by Fondation des Sciences Mathématiques de Paris (FSMP).

Acknowledgements Both authors are thankful to Juho Hirvonen for his help with the figures.

1 Context and Objective

Several techniques for formalizing distributed computing based on algebraic topology have emerged in the last decades, including the study of complexes capturing all possible global states of the systems at a given time [11], and the study of the (di)homotopy classes of directed paths representing the execution traces of concurrent programs [7]. We refer to [10] for a recent attempt to reconcile the two approaches. This paper is focusing on the approach based on the study of complexes.

A generic methodology for studying distributed computing through the lens of topology has been set by Herlihy and Shavit [12]. This methodology has played an important role in distributed computing, mostly for establishing lower bounds and impossibility results [5,12,18], but also for the design of algorithms [6]. It is based on viewing distributed computation as a topological deformation of an input space. More specifically, recall that a *simplicial complex* $\mathcal{K}$ is a collection of non-empty subsets of a finite set V, downward closed under inclusion, i.e., for every $\sigma \in \mathcal{K}$, and every non-empty $\sigma' \subset \sigma$, it holds that $\sigma' \in \mathcal{K}$. Every $\sigma \in \mathcal{K}$ is called a *simplex*, and every $v \in V$ is called a *vertex*. For instance, a graph $G = (V, E)$ with $E \subseteq \binom{V}{2}$, can be viewed as the complex $\mathcal{K} = \{\{v\} : v \in V\} \cup E$ on the set V of vertices. A *sub-complex* of a complex $\mathcal{K}$ is a subset of simplices of $\mathcal{K}$ forming a complex. The dimension of a simplex is one less than the number of its elements. A *facet* of a complex $\mathcal{K}$ is a maximal simplex of $\mathcal{K}$, that is, a simplex not contained in any other simplex. E.g., the facets of a graph with no isolated nodes are its edges. We note that a set of facets uniquely defines a complex.

The set of all possible input (resp., output) configurations of a distributed system can be viewed as a simplicial complex, called *input complex* (resp., *output complex*), and denoted by $\mathcal{I}$ (resp., $\mathcal{O}$). A vertex of $\mathcal{I}$ (resp., $\mathcal{O}$) is a pair (p, x) where p is a process name, and x is an input (resp., output) value. For instance, the input complex of binary consensus in an n-process system with process names $p_1, \ldots, p_n$ is:

$$\mathcal{I}_{\|} = \Big\{\big\{(p_i, x_i) : i \in I, x_i \in \{0,1\} \text{ for every } i \in I\big\} : I \subseteq [n], I \neq \varnothing\Big\},$$

with $[n] = \{1, \ldots, n\}$, and the output complex is:

$$\mathcal{O}_{\|} = \Big\{\big\{(p_i, y) : i \in I\big\}, I \subseteq [n], I \neq \varnothing, y \in \{0,1\}\Big\}.$$

One can check that $\mathcal{I}_{\|}$ and $\mathcal{O}_{\|}$ are indeed collections of non-empty subsets of a finite set, downward closed under inclusion. A distributed computing *task* is then specified as a *carrier map* $\Delta : \mathcal{I} \to 2^{\mathcal{O}}$, i.e., a function Δ that maps every input simplex $\sigma \in \mathcal{I}$ to a sub-complex $\Delta(\sigma)$ of the output complex, satisfying that, for every $\sigma, \sigma' \in \mathcal{I}$, if $\sigma \subseteq \sigma'$ then $\Delta(\sigma)$ is a sub-complex of $\Delta(\sigma')$. The carrier map Δ is describing the output configurations that are legal with respect to the input configuration σ. For instance, the specification of consensus is, for every $\sigma = \{(p_i, x_i) : i \in I, x_i \in \{0,1\}\} \in \mathcal{I}_{\|}$,

$$\Delta_{\|}(\sigma) = \begin{cases} \{\{(p_i, 0) : i \in I\},\ \{(p_i, 1) : i \in I\}\} & \text{if } \exists\, i, j \in I,\ x_i \neq x_j; \\ \{\{(p_i, y) : i \in I\}\} & \text{if } \forall\, i \in I,\ x_i = y. \end{cases}$$

Note that the specification of consensus given here is very general, i.e., Δ is specified for every simplex $\sigma \in \mathcal{I}_{\|}$. This enables, e.g., to handle crash failures. In absence of failures, the specification of a task can be done just by specifying Δ for the facets in the input complex.

In the topological framework, computation is modeled by a *protocol complex* that evolves with time, where the notion of "time" depends on the computational model at hand. The protocol complex at time t, denoted by $\mathcal{P}^{(t)}$, captures all possible states of the system at time t. Typically, a vertex of $\mathcal{P}^{(t)}$ is a pair (p, s) where p is a process name, and s is a possible state of p at time t. A set $\{(p_i, s_i) : i \in I\}$ of such vertices, for $\emptyset \neq I \subseteq [n]$, forms a simplex of $\mathcal{P}^{(t)}$ if the states s_i, $i \in I$, are mutually compatible, that is, if $\{s_i : i \in I\}$ forms a possible global state for the processes in the set $\{p_i : i \in I\}$ at time t.

A crucial point is that an algorithm that outputs in time t induces a mapping $\delta : \mathcal{P}^{(t)} \to \mathcal{O}$. Specifically, if the process p_i with state s_i at time t outputs y_i, then δ maps the vertex $(p_i, s_i) \in \mathcal{P}^{(t)}$ to the vertex $\delta(p_i, s_i) = (p_i, y_i)$ in $\mathcal{O}$. For the task to be correctly solved, the mapping δ must preserve the simplices of $\mathcal{P}^{(t)}$, and must agree with the specification Δ of the task. That is, δ must map simplices to simplices, and if the configuration $\{(p_i, s_i), i \in I\}$ of a distributed system is reachable at time t starting from the input configuration $\{(p_i, x_i), i \in I\}$, then it must be the case that $\{\delta(p_i, s_i), i \in I\} \in \Delta(\{(p_i, x_i), i \in I\})$. The set of configurations reachable in time t stating from an input configuration $\sigma \in \mathcal{I}$ is denoted by $\Xi_t(\sigma)$. In particular, $\Xi_t : \mathcal{I} \to 2^{\mathcal{P}^{(t)}}$ is a carrier map.

Fundamental Lemma. The framework defined by Herlihy and Shavit [12] enables to characterize the power and limitation of distributed computing, thanks to the following generic result, which can be viewed as the basis of distributed computing within the topological framework. Let us consider some (deterministic) distributed computing model, assumed to be *full information*, that is, every process communicates its entire history at each of its communication step. The following result connects solvability of a task by an algorithm in a given model with the existence of a mapping of a specific form between the topological complexes corresponding to this task and this model (see [4, 11, 12] for instantiations of this result for different computational models).

▶ **Lemma 1.** *A task $(\mathcal{I}, \mathcal{O}, \Delta)$ is solvable in time t if and only if there exists a simplicial map $\delta : \mathcal{P}^{(t)} \to \mathcal{O}$ such that, for every $\sigma \in \mathcal{I}$, $\delta(\Xi_t(\sigma)) \subseteq \Delta(\sigma)$.*

Again, beware that the notion of *time* in the above lemma depends on the computational model. The topology of the protocol complex $\mathcal{P}^{(t)}$, and the nature of the carrier map Ξ_t, depend on the input complex $\mathcal{I}$, and on the computing model at hand. For instance, wait-free computing in asynchronous shared memory systems induces protocol complexes by a deformation of the input complex, called *chromatic subdivisions* [11]. Similarly, t-resilient computing may introduce *holes* in the protocol complex, in addition to chromatic subdivisions. More generally, the topological deformation Ξ_t of the input complex caused by the execution of a full information protocol in the considered computing model entirely determines the existence of a decision map $\delta : \mathcal{P}^{(t)} \to \mathcal{O}$, which makes the task $(\mathcal{I}, \mathcal{O}, \Delta)$ solvable or not in that model.

Topological Invariants. The typical approach for determining whether a task (e.g., consensus) is solvable in t rounds consists of identifying topological *invariants*, i.e., properties of complexes that are preserved by simplicial maps. Specifically, the approach consists in:

1. Identifying a topological invariant, i.e., a property satisfied by the input complex $\mathcal{I}$, and preserved by Ξ_t;
2. Checking whether this invariant, which must be satisfied by the sub-complex $\delta(\mathcal{P}^{(t)})$ of the output complex $\mathcal{O}$, does not contradict the specification Δ of the task.

For instance, in the case of binary consensus, the input complex $\mathcal{I}_{\|}$ is a *sphere*. One basic property of spheres is being *path-connected* (i.e., there is a path in $\mathcal{I}_{\|}$ between any two vertices). As mentioned earlier, shared-memory wait-free computing corresponds to subdividing the input complex [11]. Therefore, independently from the length t of the execution, the protocol complex $\mathcal{P}^{(t)}$ is a chromatic subdivision of the sphere $\mathcal{I}_{\|}$, and thus it remains path-connected. On the other hand, the output complex $\mathcal{O}_{\|}$ of binary consensus is the disjoint union of two complexes $\mathcal{O}_0$ and $\mathcal{O}_1$, where $\mathcal{O}_y = \big\{\{(i, y) : i \in I\}, I \subseteq [n], I \neq \varnothing\big\}$ for $y \in \{0, 1\}$. Since simplicial maps preserve connectivity, it follows that $\delta(\mathcal{P}^{(t)}) \subseteq \mathcal{O}_0$ or $\delta(\mathcal{P}^{(t)}) \subseteq \mathcal{O}_1$. As a consequence, δ cannot agree with $\Delta_{\|}$, as the latter maps the simplex $\{(i, 0), i \in [n]\}$ to $\mathcal{O}_0$, and the simplex $\{(i, 1), i \in [n]\}$ to $\mathcal{O}_1$. Therefore, consensus cannot be achieved wait-free, regardless of the number t of rounds.

The fact that connectivity plays a significant role in the inability to solve consensus in the presence of asynchrony and crash failures is known since the original proof of the FLP theorem [8] in the early 1980s. However, the relation between k-set agreement and higher dimensional forms of connectivity (i.e., the ability to contract high dimensional spheres) was only established ten years later [12, 18]. We refer to [11] for numerous applications of Lemma 1 to various models of distributed computing, including asynchronous crash-prone shared-memory or fully-connected message passing models. In particular, for tasks such as renaming, identifying the minimal number t of rounds enabling a simplicial map δ to exist is currently the only known technique for upper bounding their time complexities [1].

Network Computing. Recently, Castañeda et al. [4] applied Lemma 1 to synchronous fault-free computing in networks, that is, to the framework in which processes are located at the vertices of a simple (no multiple edges, no loops) n-node undirected graph G, and can exchange messages only along the edges of that graph. They mostly focus on *input-output tasks* such as consensus and set-agreement, in a simplified computing model, called KNOW-ALL, specifying that every process is initially aware of the name and the location of all the other processes in the network. As observed in [4], synchronous fault-free computing in the KNOW-ALL model preserves the *facets* of the input complex, and does not subdivide them. However, *scissor cuts* may occur between adjacent facets during the course of the computation, that is, the protocol complex $\mathcal{P}^{(t)}$ is obtained from the input complex $\mathcal{I}$ by partially separating facets that initially shared a simplex. Figure 1 illustrates two types of scissor cuts applied to the sphere, corresponding to two different communication networks. The positions of the cuts depend on the structure of the graph G in which the computation takes place, and determining the precise impact of the structure of G on the topology of the protocol complex is a nontrivial challenge, even in the KNOW-ALL model.

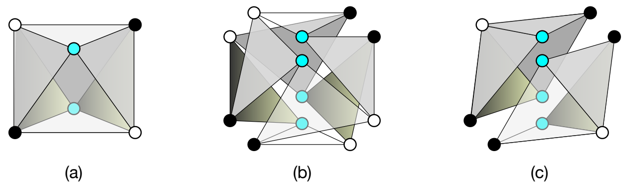

Figure 1 (a) The input complex of binary consensus for three processes; (b) The scissor cuts for the consistently directed 3-process cycle C_3 after one round; (c) The scissor cuts for the directed 3-process star S_3, where edges are directed from the center to the leaves, after one round.

Instead, we aim at analyzing classical *graph problems* (e.g., coloring, independent set, etc.) in the standard LOCAL model [17] of network computing, which is weaker than the KNOW-ALL model, and thus allows for more complicated topological deformations. In the LOCAL model, every node is initially aware of solely its identifier (which is unique in the network), and its input (e.g., for minimum weight vertex cover or for list-coloring), all nodes wake up synchronously, and compute in locksteps. The LOCAL model is an ideal model for studying *locality* in the context of network computing [17].

In addition to the fact that the topological deformations of the protocol complexes strongly depend on the structure of the network, another obstacle that makes applying the topological approach to the LOCAL model even more challenging is the presence of process *identifiers*. Indeed, the model typically assumes that the node IDs are taken in a range $[N]$ where $N = \text{poly}(n)$. As a consequence, independently from the potential presence of other input values, the size of the complexes (i.e., their number of vertices) may become as large as $\binom{N}{n} n!$, since there are $\binom{N}{n}$ ways of choosing n IDs, and $n!$ ways of assigning the n chosen IDs to the n nodes of G (unless G presents symmetries). For instance, Figure 1 assumes the KNOW-ALL model, hence fixed IDs. Redrawing these complexes assuming that the three processes can pick arbitrary distinct IDs as in the LOCAL model, even in the small domain $\{1, 2, 3, 4\}$, would yield a cumbersome figure with 24 nodes. Note that the presence of IDs also results in input complexes that may be topologically more complicated than pseudospheres, even for tasks such as consensus.

Importantly, the fact that the IDs are not fixed a priori, and may even be taken in a range exceeding $[n]$, is inherent to distributed network computing. Indeed, this framework aims at understanding the power and limitation of computing in large networks, from LANs to the whole Internet, where the processing nodes are assigned arbitrary IDs taken from a range of values which may significantly exceed the number of nodes in the network.

Objective. To sum up, while the study of protocol complexes has found numerous applications in the context of fault-tolerant message-passing or shared-memory computing, extending this theory to network computing faces a difficulty caused by the presence of arbitrary IDs, which are often the only inputs to the processes [17]. The objective of this paper is to show how the combinatorial blowup caused by the presence of IDs in network computing can be bypassed, at least as far as local computing is concerned.

2 Our Results

We show how to bypass the aforementioned exponential blowup in the size of the complexes, that would result from a straightforward application of Lemma 1 for analyzing the complexity of tasks in networks. Our result holds for a variety of problems, including classical graph problems such as vertex and edge-coloring, maximal independent set (MIS), maximal matching, etc. More specifically, it holds for the large class of *locally checkable labeling* (LCL) tasks [16] on bounded-degree graphs. These are tasks for which it is possible to verify locally the correctness of a solution, and thus they are sometimes viewed as the analog of NP in the context of computing in networks. An LCL task is described by a finite set of labels, and a local description of how these labels can be legally assigned to the nodes of a network. Our local characterization theorem is strongly based on a seminal result by Naor and Stockmeyer [16] who showed that the *values* of the IDs do not actually matter much for solving LCL tasks in networks, but only their *relative order* matters.

We prove an analog of Lemma 1, but where the size of the complexes involved in the statement is independent of the size of the networks. Specifically, the size of the complexes in our characterization theorem depends solely on the maximum degree d of the network, the number of labels used for the description of the task, and the number of rounds of the considered algorithm for solving that task. In particular, the identifiers are taken from a bounded-size set, even if the theorem applies to tasks defined on n-node networks with arbitrarily large n, and for identifiers taken in an arbitrarily large range $[N]$. We denote by $\mathcal{K}_{x,[y]}$ the fact that the facets of $\mathcal{K}$ have dimension x, and that the IDs are taken in the set $\{1, \dots, y\}$, and we let $\mathcal{K}_x = \mathcal{K}_{x,\varnothing}$. Also $\pi : \mathcal{K}_{x,[y]} \to \mathcal{K}_x$ denotes the mapping that removes the IDs of the vertices. Every LCL task in networks with maximum degree d can be expressed topologically as a task $(\mathcal{I}_d, \mathcal{O}_d, \Delta)$ where $\mathcal{I}_d$ and $\mathcal{O}_d$ are complexes of dimension d. Our main result is the following.

▶ **Theorem 2** (A simplified version of Theorem 3). *For every LCL task $T = (\mathcal{I}_d, \mathcal{O}_d, \Delta)$ on graphs of maximum degree d, and for every $t \geq 0$, there exists $R \in \mathbb{N}$ such that the following holds. The task T is solvable in t rounds in the LOCAL model if and only if there is a simplicial map $\delta : \mathcal{P}^{(t)}_{d,[R]} \to \mathcal{O}_d$ such that, for every facet $\sigma \in \mathcal{I}_{d,[R]}$, $\delta(\Xi_t(\sigma)) \subseteq \Delta(\pi(\sigma))$.*

Figure 2 provides a rough description of the commutative diagram corresponding to the brute force application of Lemma 1 to LCL tasks, and of the commutative diagram corresponding to Theorem 2. Note that Lemma 1, which corresponds to the left diagram in Figure 2, involves *global* complexes with $(n-1)$-dimensional facets, whose vertices are labeled by IDs in an arbitrarily large set $[N]$. In contrast, the complexes corresponding to Theorem 2, which correspond to the right diagram, are *local* complexes, with facets of constant dimension, and vertices labeled with IDs in a finite set whose size is constant w.r.t. the number of nodes n in the network.

Figure 2 The commutative diagrams of Lemma 1 (left), and Theorem 2 (right).

As an application of Theorem 2, we reformulate the celebrated lower bound $\Omega(\log^* n)$ rounds for 3-coloring the n-node ring by Linial [15], in the algebraic topology framework (see Corollary 4).

3 Models and Definitions

The LOCAL model. The LOCAL model was introduced more than a quarter of a century ago (see, e.g., [15, 16]) for studying which tasks can be solved *locally* in networks, that is, which tasks can be solved when every node is bounded to collect information only from nodes in its vicinity. Specifically, the LOCAL model [17] states that the processors are located at the nodes of a simple connected graph $G = (V, E)$ modeling a network. All nodes are fault-free, they wake up simultaneously, and they execute the same algorithm. Computation proceeds in synchronous rounds, where a round consists of the following three steps performed by every node: (1) sending a message to each neighbor in G, (2) receiving the messages sent by the neighbors, and (3) performing local computation. There are no bounds on the size of the messages exchanged at every round between neighbors, and there are no limits on the individual computational power or memory of the nodes. These assumptions enable the design of unconditional lower bounds on the number of rounds required for performing some task (e.g., for providing the nodes with a proper coloring), while the vast majority of the algorithms solving these tasks do not abuse of these assumptions [19], that is, they exchange small (i.e., polylogarithmic size) messages, and perform efficient (i.e., poly-time) individual computations.

Every node in the network has an identifier (ID) which is supposed to be unique in the network. In n-node networks, the IDs are supposed to be in a range $1, \ldots, N$ where $N \gg n$ typically holds (most often, $N = \text{poly}(n)$). The absence of limits on the amount of communication and computation that can be performed at every round implies that the LOCAL model enables *full-information* protocols, that is, protocols in which, at every round, every node sends all the information it acquired during the previous rounds to its neighbors. Therefore, for every $t \geq 0$, and every graph G, a t-round algorithm allows every node in G to acquires a local *view* of G, which is a ball in G centered at that node, and of radius t. A view includes the inputs and the IDs of the nodes in the corresponding ball. It follows that a t-round algorithm in the LOCAL model can be considered as a function from the set of views of radius t to the set of output values.

Locally Checkable Labelings (LCL). Let $d \geq 2$, and let $\mathcal{G}_d$ be the class of connected simple undirected d-regular graphs (all nodes have degree d). Recall that, for a positive integer c, c-coloring is the task consisting in providing each node with a color in $\{1,\dots,c\}$ in such a way that no two adjacent nodes are given the same color. Maximal independent set (MIS) is the closely related task consisting in providing each node with a boolean value (0 or 1) such that no two adjacent nodes are given the value 1, and every node with value 0 is adjacent to at least one node with value 1. Proper c-coloring in $\mathcal{G}_d$ can actually be described by the collection of *good stars* of degree d, and with nodes colored by labels in $\{1,\dots,c\}$, such that the center node has a color different from the color of each leaf. Similarly, maximal independent set (MIS) in $\mathcal{G}_d$ can be described by the collection of stars with degree d, and with each node colored by a label in $\{0,1\}$, such that if the center node is labeled 1 then all the leaves are colored 0, and if the center node is labeled 0 then at least one leaf is colored 1. Other tasks such as variants of coloring, or $(2,1)$-ruling set[1] can be described similarly, by a finite number of legal labeled stars.

More generally, given a finite set $\mathcal{L}$ of labels, we denote by $\mathbf{S}_d^{\mathcal{L}}$ the set of all labeled stars resulting from labeling each node of the $(d+1)$-node star by some label in $\mathcal{L}$. A *locally checkable labeling* (LCL) [16] is then defined by a finite set $\mathcal{L}$ of labels, and a set $\mathcal{S} \subseteq \mathbf{S}_d^{\mathcal{L}}$. Every star in $\mathcal{S}$ is called a *good* star, and those in $\mathbf{S}_d^{\mathcal{L}} \setminus \mathcal{S}$ are *bad*. The computing task defined by an LCL $(\mathcal{L},\mathcal{S})$ consists, for every node of every graph $G \in \mathcal{G}_d$, of computing a label in $\mathcal{L}$ such that every resulting labeled radius-1 star in G is isomorphic to a star in $\mathcal{S}$. In other words, the objective of every node is to compute a label in $\mathcal{L}$ such that every resulting labeled radius-1 star in G is good. It is undecidable, in general, whether a given LCL task has an algorithm performing in $O(1)$ rounds in the LOCAL model [16].

In their full generality, LCL tasks include tasks in which nodes have inputs, potentially of some restricted format. For instance, this is the case of the task consisting of reducing c-coloring to MIS in the n-node cycle C_n, studied in the next section. Hence, in its full generality, an LCL task is described by a quadruple $(\mathcal{L}_{in},\mathcal{S}_{in},\mathcal{L}_{out},\mathcal{S}_{out})$ where $\mathcal{L}_{in}$ and $\mathcal{L}_{out}$ are the input and output labels, respectively. The set of stars $\mathcal{S}_{in}$ can often be simply viewed as a promise stating that every radius-1 star of the input graph G belongs to $\mathcal{S}_{in}$, and the set $\mathcal{S}_{out}$ is the target set of good radius-1 stars. LCL tasks also capture settings in which the legality of the output stars depends on the inputs. A typical example of such a setting is list-coloring where the output color of each node must belong to a list of colors given to this node as input. The framework of LCL tasks can be extended to balls of radius $r > 1$, and assuming radius 1 is not restrictive, up to increasing the size of the set of labels [3].

4 Warm Up: Coloring and MIS in the Ring

In this section, we exemplify our technique, in a way that resembles the proof of Theorem 2. We consider an LCL task on a ring, where the legal input stars define a proper 3-coloring, and the output stars define a maximal independent set (MIS). That is, we study the time complexity of reducing a 3-coloring to a MIS on a ring. It is known [15] that there is a 2-round algorithm for the problem in the LOCAL model, and we show that this is optimal using topological arguments. This toy example provides the basic concepts and arguments that we use later, when considering general LCL tasks and proving Theorem 2.

[1] Recall that an (α,β)-ruling set in a graph $G=(V,E)$ is a set $R \subseteq V$ such that, for any node $v \in V$ there is a node $u \in R$ in distance at most β from v, and any two nodes in R are at distance at least α from each other.

4.1 Reduction from 3-coloring to MIS

Let us consider three consecutive nodes of the n-node ring C_n, denoted by p_{-1}, p_0, and p_1, as displayed on Figure 3.

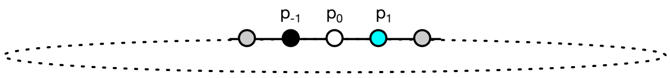

Figure 3 Three consecutive nodes in the n-node ring.

By the independence property, if p_0 is in the MIS, then neither p_{-1} nor p_1 can be in the MIS, and, by the maximality property, if p_0 is not in the MIS, then p_{-1} or p_1, or both, must be in the MIS. These constraints are captured by the complex $\mathcal{M}_2$ displayed on Figure 4, including six vertices (p_i, x), with $i \in \{-1, 0, 1\}$, and $x \in \{0, 1\}$, where $x = 1$ (resp., $x = 0$) indicates that p_i is in the MIS (resp., not in the MIS).

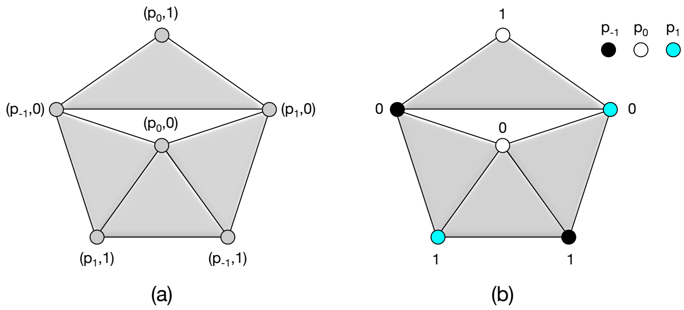

Figure 4 The local complex $\mathcal{M}_2$ of MIS in the ring. (a) the vertices are labeled with the index of the processes and the values; (b) the indexes of the processes are replaced by colors.

The complex $\mathcal{M}_2$ of Figure 4 has four facets of dimension 2: they are triangles. Some triangles intersect along an edge, while some others intersect only at a node. The complex $\mathcal{M}_2$ is called the *local* complex of MIS in the ring (the index 2 refers to the fact that rings have degree 2). Note that the sets $\{(p_{-1}, 0), (p_0, 0), (p_1, 0)\}$ and $\{(p_{-1}, 1), (p_0, 1), (p_1, 1)\}$ do not form simplices of $\mathcal{M}_2$. We call these two sets monochromatic. In the objective of reducing 3-coloring to MIS, $\mathcal{M}_2$ will be the output complex, corresponding to $\mathcal{O}_d$ with $d = 2$ in Figure 2 and in Theorem 2.

Similarly, let us focus on 3-coloring, with the same three processes p_{-1}, p_0, and p_1. The neighborhood of p_0 cannot include the same color as its own color, and thus there are twelve possible colorings of the nodes in the star centered at p_0. Each of these stars corresponds to a 2-dimensional simplex, forming the facets of the local complex $\mathcal{C}_2$ of 3-coloring in the ring, depicted in Figure 5. This complex contains nine vertices of the form (p_i, c), with $i \in \{-1, 0, 1\}$, and $c \in \{1, 2, 3\}$, and twelve facets. Note that the vertices $(p_{-1}, 3)$ and $(p_1, 3)$ appear twice in the figure, since the leftmost and rightmost edges are identified, but in opposite direction, forming a Möbius strip. $\mathcal{C}_2$ is a manifold (with boundary). When reducing 3-coloring to MIS, $\mathcal{C}_2$ will be the input complex, corresponding to $\mathcal{I}_d$ with $d = 2$ in Figure 2.

Remark. It is crucial to note that the complexes displayed in Figures 4 and 5 are not the ones used in the standard settings (e.g., [4, 11]), for which Lemma 1 would use vertices of the form (p, x) for $p \in [n]$, or even $p \in [N]$ assuming IDs in a range of N values. As

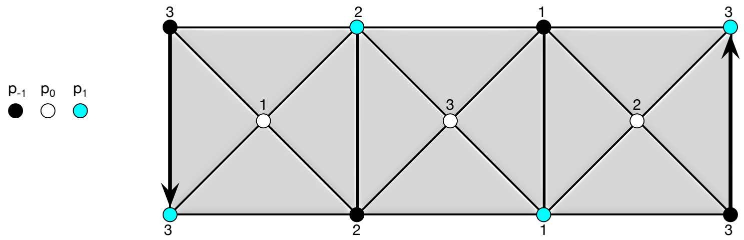

Figure 5 Local complex $\mathcal{C}_2$ of 3-coloring in the ring.

a consequence, these complexes have 6 vertices instead of $2n!\binom{N}{n}$ for MIS, and 9 vertices instead of $3n!\binom{N}{n}$ for coloring, where n can be arbitrarily large. Even if the IDs would have been fixed, the approach of Lemma 1 would yield complexes with a number of vertices linear in n, while the complexes of Figs. 4 and 5 are of constant size.

As it is well know since the early work by Linial [15], a properly 3-colored ring can be "recolored" into a MIS in just two rounds. First, the nodes colored 3 recolor themselves 1 if they have no neighbors originally colored 1. Then, the nodes colored 2 do the same, i.e., they recolor themselves 1 if they have no neighbors colored 1 (whether it be neighbors originally colored 1, or nodes that recolored themselves 1 during the first round). The nodes colored 1 output 1, and the other nodes output 0. The set of nodes colored 1 forms a MIS. Note that this algorithm is *ID-oblivious*, i.e., it can run in an anonymous network.

Task specification. The specification of reducing 3-coloring to MIS can be given by the trivial carrier map $\Delta : \mathcal{C}_2 \to 2^{\mathcal{M}_2}$ defined by $\Delta(F) = \{F' : F' \text{ is a facet of } \mathcal{M}_2\}$ for every facet F of $\mathcal{C}_2$. (As the LOCAL model is failure-free, it is enough to describe all maps at the level of facets.) Note that the initial coloring of a facet in $\mathcal{C}_2$ does not induce constraints on the facet of $\mathcal{M}_2$ to which it should be mapped. Figure 6 displays some of the various commutative diagrams that will be considered in this section. In all of them, Δ is the carrier map specifying reduction from 3-coloring to MIS in the ring, and none of the simplicial maps δ exist. Also recall that π is the map removing IDs.

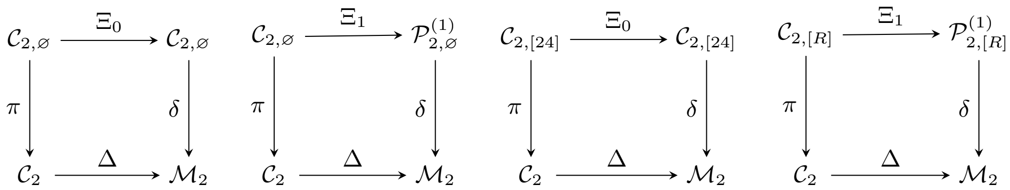

Figure 6 Complexes corresponding to reduction from 3-coloring to MIS in the n-node ring. From left to right: 0 rounds without IDs, 1-round without IDs, 0 rounds with ID, and 1-round with IDs.

4.2 ID-Oblivious Algorithms

Impossibility in Zero Rounds. Let us consider an alleged ID-oblivious algorithm ALG which reduces 3-coloring to MIS in zero rounds. Such an algorithm sees only the node's color $c \in \{1,2,3\}$, and must map it to some $x \in \{0,1\}$. This mapping can be extended to a mapping δ that maps every pair (p_i, c) with $i \in \{-1,0,1\}$ and $c \in \{1,2,3\}$ to a pair $\delta(p_i, c) = (p_j, x)$, $j \in \{-1,0,1\}$ and $x \in \{0,1\}$, with the following properties.

- *Name-preservation.* The mapping δ must satisfy that $p_j = p_i$, i.e., δ is *name-preserving*. By the name-preserving property, the algorithm maps the vertices in Figure 5 to the vertices in Figure 4(b) while preserving the names p_{-1}, p_0, p_1 of these vertices. Therefore, the algorithm induces a *chromatic* simplicial map $\delta : \mathcal{C}_2 \to \mathcal{M}_2$. (The "color" of p, i.e., p's name, is preserved).
- *Name-independence.* In addition to name-preservation, the mapping δ must satisfy that, for every $i \neq j$, (p_i, c) and (p_j, c) are mapped to (p_i, x_i) and (p_j, x_j), respectively, with $x_i = x_j$, i.e., δ is *name-independent*. Indeed, the names p_{-1}, p_0, and p_1 given to the nodes are "external", i.e., they are not part of the input to the algorithm ALG.

We are therefore questioning the existence of a name-preserving name-independent simplicial map $\delta : \mathcal{C}_2 \to \mathcal{M}_2$. This is in correspondence to Figure 2 and Theorem 2, in the degenerated case where $t = 0$ and $[R] = \varnothing$, for which $\mathcal{C}_2 = \mathcal{I}_2$, and $\mathcal{C}_{2,\varnothing} = \mathcal{I}_{2,\varnothing} = \mathcal{P}^{(0)}_{2,\varnothing} = \mathcal{C}_2$ – see the leftmost diagram in Figure 6. There cannot exist a name-preserving name-independent simplicial map δ from the manifold $\mathcal{C}_2$ to $\mathcal{M}_2$ (from Figure 5 to Figure 4(b)), which we formally prove in the full version of the paper [9]. The intuition is that if some triangle of $\mathcal{C}_2$ is mapped to the triangle $\{(p_0, 1), (p_{-1}, 0), (p_1, 0)\}$ of $\mathcal{M}_2$ then *all* triangles of $\mathcal{C}_2$ must be mapped to that triangle of $\mathcal{M}_2$, from which it follows by name-independence that all processes output 1, or all processes output 0, which leads to contraction in both cases. The absence of a name-independent name-preserving simplicial map $\delta : \mathcal{C}_2 \to \mathcal{M}_2$ is a witness of the impossibility to construct a MIS from a 3-coloring of the ring in zero rounds, when using an ID-oblivious algorithm.

Impossibility in One Round. For analyzing 1-round algorithms, let us consider the local protocol complex $\mathcal{P}^{(1)}_{2,\varnothing}$, including the views of the three nodes p_{-1}, p_0, and p_1 after one round. The vertices of $\mathcal{P}^{(1)}_{2,\varnothing}$ are of the form (p_i, xyz) with $i \in \{-1, 0, 1\}$, and $x, y, z \in \{1, 2, 3\}$, $x \neq y$, and $y \neq z$. The vertex (p_i, xyz) is representing a process p_i starting with color y, and receiving the input colors x and z from its left and right neighbors, respectively. The facets of $\mathcal{P}^{(1)}_{2,\varnothing}$ are of the form $\{(p_{-1}, x'xy), (p_0, xyz), (p_1, yzz')\}$. Figure 7 displays that complex, which consists of three connected components $\mathcal{K}_1, \mathcal{K}_2$, and $\mathcal{K}_3$ where, for $y = 1, 2, 3$, $\mathcal{K}_y$ includes the four vertices (p_0, xyz) for $x, z \in \{1, 2, 3\} \setminus \{y\}$, and all triangles that include these vertices. Each set of four triangles sharing a vertex (p_0, xyz) forms a cone (see Figure 8). These cones are displayed twisted on Figure 7 to emphasis the "circular structure" of the three components.

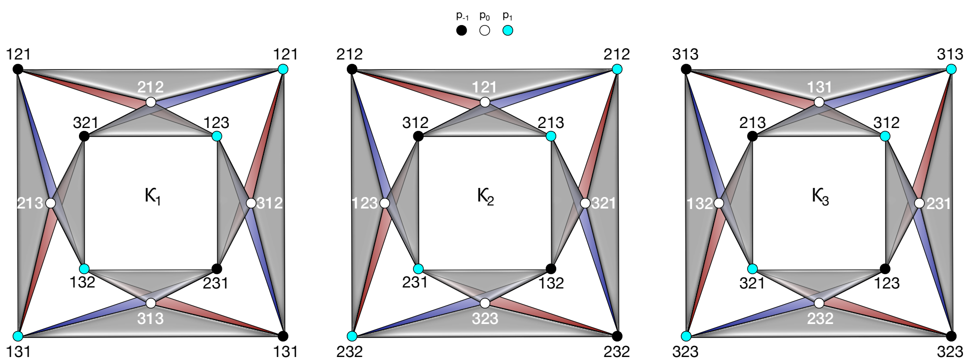

Figure 7 Local protocol complex $\mathcal{P}^{(1)}_{2,\varnothing}$ after 1 round starting from a 3-coloring of the ring.

Following the same reasoning as for 0-round algorithms, a 1-round algorithm ALG induces a chromatic (i.e., name-preserving) simplicial map $\delta : \mathcal{P}^{(1)}_{2,\varnothing} \to \mathcal{M}_2$, as in the second to left diagram in Figure 6. In the full version, we show that such a mapping cannot exist [9].

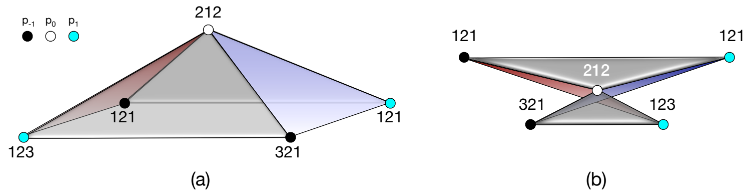

Figure 8 (a) A cone composed of four triangles; (b) The same cone "twisted".

The 2-Round Algorithm. The local protocol complex $\mathcal{P}^{(2)}_{2,\varnothing}$ includes the views of the three nodes p_{-1}, p_0, and p_1 after two rounds. The vertices of $\mathcal{P}^{(2)}_{2,\varnothing}$ are of the form $(p_i, c_1c_2c_3c_4c_5)$ with $i \in \{-1, 0, 1\}$, $c_j \in \{1, 2, 3\}$ for $1 \leq j \leq 5$, and $c_j \neq c_{j+1}$ for $1 \leq j < 5$. Figure 9(a) displays one of the connected components of $\mathcal{P}^{(2)}_{2,\varnothing}$, denoted $\mathcal{K}_{323}$,which includes the four vertices $(p_0, c_1 323 c_5)$, $c_1, c_5 \in \{1, 2\}$. There are 12 disjoint isomorphic copies of this connected component in $\mathcal{P}^{(2)}_{2,\varnothing}$, one for each triplet $c_2, c_3, c_4 \in \{1, 2, 3\}$, $c_2 \neq c_3$, and $c_3 \neq c_4$.

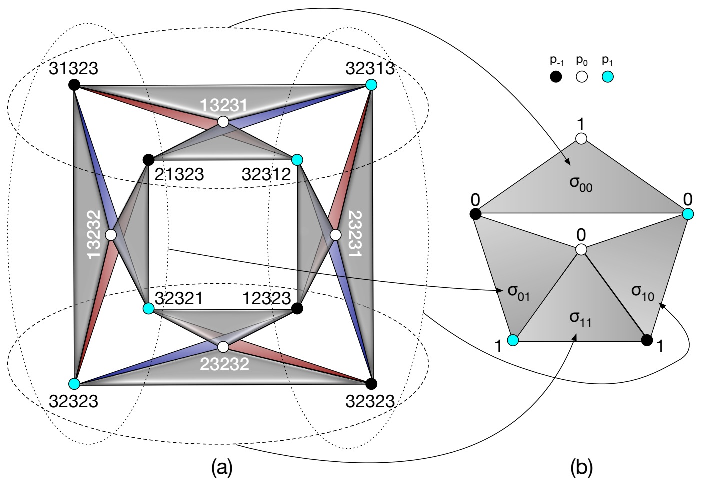

Figure 9 (a) The sub-complex $\mathcal{K}_{323}$ of the local protocol complex $\mathcal{P}^{(2)}_{2,\varnothing}$. (b) The facets of $\mathcal{M}_2$.

Interestingly, each connected component of $\mathcal{P}^{(2)}_{2,\varnothing}$ is isomorphic to each connected component of $\mathcal{P}^{(1)}_{2,\varnothing}$, while there are more connected components in $\mathcal{P}^{(2)}_{2,\varnothing}$ than in $\mathcal{P}^{(1)}_{2,\varnothing}$. However, the larger views of the processes provides more flexibility for the mapping from $\mathcal{P}^{(2)}_{2,\varnothing}$ to $\mathcal{M}_2$ than for the mapping from $\mathcal{P}^{(1)}_{2,\varnothing}$ to $\mathcal{M}_2$. And indeed, the 2-round anonymous algorithm presented at the end of Section 4.1 does induce a chromatic simplicial map $\delta : \mathcal{P}^{(2)}_{2,\varnothing} \to \mathcal{M}_2$. Specifically, the four sub-complexes $\mathcal{K}_{x1y}$, as well as the simplex $\mathcal{K}_{232}$ are entirely mapped to the simplex σ_{00} (see Figure 9(b) for the labeling of the four facets of $\mathcal{M}_2$). The two sub-complexes $\mathcal{K}_{1x1}$ are entirely mapped to the simplex σ_{11}. The two sub-complexes $\mathcal{K}_{321}$ and $\mathcal{K}_{231}$ are entirely mapped to the sub-complex $\sigma_{01} \cup \sigma_{11}$, and the two sub-complexes $\mathcal{K}_{123}$ and $\mathcal{K}_{132}$ are entirely mapped to the sub-complex $\sigma_{10} \cup \sigma_{11}$. The mapping of the remaining sub-complex $\mathcal{K}_{323}$ is more sophisticated, and illustrates that the simple algorithm showing reduction from 3-coloring to MIS in [15] is actually topologically non-trivial. Indeed, $\mathcal{K}_{323}$ is mapped by the algorithm so that it wraps around the hole in $\mathcal{M}_2$, as depicted in Figure 9.

4.3 General Case with IDs

The presence of IDs given to the nodes adds power to the distributed algorithms, as the output of a process is not only a function of the observed colors in its neighborhood, but also of the observed IDs. In particular, after one round, a process p is not only aware of a triplet of colors $(c_1c_2c_3)$, but also of a triplet of distinct IDs $(x_1x_2x_3)$.

Impossibility in Zero Rounds with IDs. Since the simplicial maps δ induced by the potential algorithms are name-preserving, they actually act on pairs (x, c) where x is an ID and c is a color, i.e., $\delta(p,(x,c)) = (p, \hat{\delta}(x,c))$ for some $\hat{\delta}$. For brevity, we identify $\hat{\delta}$ with δ. Let us assume that the IDs are from $\{1,\ldots,R\}$, for some $R \geq 4$. That is, we consider now $\mathcal{C}_{2,[R]}$ for $R \geq 4$. By the pigeon-hole principle, there exists a set $I_1 \subseteq \{1,\ldots,R\}$ with $|I_1| \geq R/2$ such that, for every $x, x' \in I_1$, $\delta(x,1) = \delta(x',1)$. Therefore, again by the pigeon-hole principle, there exists a set $I_2 \subseteq I_1$ with $|I_2| \geq |I_1|/2$ such that, for every $x, x' \in I_2$, $\delta(x,2) = \delta(x',2)$. Finally, there exists a set $I_3 \subseteq I_2$ with $|I_3| \geq |I_2|/2$ such that, for every $x, x' \in I_3$, $\delta(x,3) = \delta(x',3)$. Therefore, there exists a set $I \subseteq \{1,\ldots,R\}$ with $|I| \geq R/8$ such that, for every $x, x' \in I$, $\delta(x,1) = \delta(x',1)$, $\delta(x,2) = \delta(x',2)$, and $\delta(x,3) = \delta(x',3)$. Therefore, whenever $R \geq 24$, the set I has size at least 3. Consider the sub-complex $\mathcal{C}'_{2,[R]}$ of $\mathcal{C}_{2,[R]}$ induced by the three smallest IDs in I – this sub-complex is isomorphic to $\mathcal{C}_{2,\varnothing}$ (Figure 5). More importantly, the mapping from $\mathcal{C}'_{2,[R]}$ to $\mathcal{M}_2$ depends only on the colors and not on the IDs, by the choice of I. Hence, if there was a mapping from $\mathcal{C}'_{2,[R]}$ to $\mathcal{M}_2$, then there would exist a mapping from $\mathcal{C}_{2,\varnothing}$ to $\mathcal{M}_2$, which we know does not exist.

It follows that there are no mappings from $\mathcal{C}_{2,[24]} = \mathcal{P}^{(0)}_{2,[24]}$ to $\mathcal{M}_2$ – see the second to right diagram in Figure 6. In other words, if the IDs are picked from a set of at least 24 values, then 3-coloring cannot be reduced to MIS in zero rounds.

Impossibility in One Rounds with IDs. We reduce the case with IDs to the case without IDs following the guideline introduced in [16]. We consider the 1-round protocol complex with IDs in a finite set X with at least 5 elements, denoted by $\mathcal{P}^{(1)}_{2,X}$. That is, $\mathcal{P}^{(1)}_{2,X} = \mathcal{P}^{(1)}_{2,[k]}$ with $k = |X|$. The vertices of this complex are of the form $(p_i,(xyz, abc))$ where $i \in \{-1,0,1\}$, $\{x,y,z\} \in \binom{X}{3}$, and $a,b,c \in \{1,2,3\}$ with $a \neq b$ and $b \neq c$. The facets of $\mathcal{P}^{(1)}_{2,X}$ are of the form $F = \{(p_{-1},(x'xy, a'ab)),(p_0,(xyz,abc)),(p_1,(yzz',bcc'))\}$. Let us assume the existence of a name-preserving name-independent simplicial map $\delta : \mathcal{P}^{(1)}_{2,X} \to \mathcal{M}_2$ (see the rightmost diagram in Figure 6). This map induces a labeling of the pairs (xyz, abc) with labels in $\{0,1\}$, where xyz is an ordered triplet of distinct IDs, and abc is an ordered triplet of colors in $\{1,2,3\}$. It follows that δ induces a labeling of the ordered triplets xyz of distinct IDs by labels in $\{0,1\}^{12}$, by applying δ to the 12 possible choices of color triplets. By Ramsey's Theorem [14], by taking the IDs in the set $X = \{1,\ldots,R\}$ with R large enough, there exists a set Y of five IDs such that, for every two sets $\{x,y,z\}$ and $\{x,' y', z'\}$ of IDs in Y, with $x < y < z$ and $x' < y' < z'$, and for every ordered sequence abc of colors, $\delta(p_0,(xyz,abc)) = \delta(p_0,(x'y'z',abc))$. Let $\mathcal{P}^{(1)}_{2,Y}$ be the sub-complex of the 1-round protocol complex $\mathcal{P}^{(1)}_{2,X}$ induced by the vertices with IDs in Y ordered in increasing order. By construction of Y, δ is ID-oblivious on $\mathcal{P}^{(1)}_{2,Y}$. Now, let $\mathcal{P}^{(1)}_{2,\varnothing}$ as displayed on Figure 7. Let us define the map $\delta' : \mathcal{P}^{(1)}_{2,\varnothing} \to \mathcal{M}_2$ by $\delta'(p_i, abc) = \delta(p_i,(xyz,abc))$ where $\{x,y,z\} \subset Y$ and $x < y < z$. Note that δ' is well defined as δ is ID-oblivious on Y. Assuming $\delta : \mathcal{P}^{(1)}_{2,X} \to \mathcal{M}_2$ is simplicial yields that $\delta' : \mathcal{P}^{(1)}_{2,\varnothing} \to \mathcal{M}_2$ is simplicial as well. We have seen in Section 4.2 that such a simplicial mapping does not exist. It follows that there are no name-preserving name-independent simplicial maps $\delta : \mathcal{P}^{(1)}_{2,[R]} \to \mathcal{M}_2$ whenever R is large enough (see Figure 6).

5 Topology of LCL Tasks

Let S_d be the star of $d+1$ nodes, whose center node is named p_0, and the leaves are named p_i, for $i = 1,\dots,d$. We consider algorithms for classes $\mathcal{G} \subseteq \mathcal{G}_d$ of graphs. Let $T = (\mathcal{L}_{in}, \mathcal{S}_{in}, \mathcal{L}_{out}, \mathcal{S}_{out})$ be an LCL task for $\mathcal{G} \subseteq \mathcal{G}_d$. The input complex $\mathcal{I}_d$ (resp., output complex $\mathcal{O}_d$) associated with T is the complex where $\{(p_i, x_i) : i \in \{0,\dots,d\}\}$ is a facet of $\mathcal{I}_d$ (resp., a facet of $\mathcal{O}_d$) if $x_i \in \mathcal{L}_{in}$ (resp., $\mathcal{L}_{out}$) for every $i \in \{0,\dots,d\}$, and the labeled star resulting from assigning label x_i to the node p_i of S_d for every $i \in \{0,\dots,d\}$ is in $\mathcal{S}_{in}$ (resp., $\mathcal{S}_{out}$). If the considered LCL task T imposes constraints on the correctness of the outputs as a function of the inputs, as in list-coloring, then the carrier map $\Delta : \mathcal{I}_d \to 2^{\mathcal{O}_d}$ specifies, for each facet $F \in \mathcal{I}_d$, the facets $\Delta(F)$ which are legal with respect to F. Otherwise, $\Delta(F) = \{\text{all facets of } \mathcal{O}_d\}$, for every facet F of $\mathcal{I}_d$.

Let $t \geq 0$, and let us fix a graph $G = (V, E)$ in $\mathcal{G} \subseteq \mathcal{G}_d$. In t rounds, every node in G acquires a *view* w, whose structure is isomorphic to a radius-t ball in G centered at that node, including the input labels and the IDs of the nodes in the ball. An ordered collection $w_0,\dots,w_d$ of views at distance t forms a collection of *mutually compatible* views for $\mathcal{G}$ if there exists a graph $G \in \mathcal{G}$, an assignment of input labels and IDs to the nodes of G, and a star S in G, with nodes $v_0,\dots,v_d$, centered at v_0, such that w_i is the view of v_i in G after t rounds, for $i = 0,\dots,d$.

Let T be an LCL task for $\mathcal{G} \subseteq \mathcal{G}_d$, and let $t \geq 0$. The t-round protocol complex associated with T for a finite set X of IDs, is the complex $\mathcal{P}^{(t)}_{d,X}$ where $F = \{(p_i, w_i) : i \in \{0,\dots,d\}\}$ is a facet of $\mathcal{P}^{(t)}_{d,X}$ if $w_0,\dots,w_d$ is an ordered collection of mutually compatible views at distance t for $\mathcal{G}$. The special case $t = 0$ corresponds to $\mathcal{P}^{(0)}_{d,X} = \mathcal{I}_{d,X}$ where $\mathcal{I}_{d,X}$ in the input complex $\mathcal{I}_d$ extended with IDs in X. The set X must be large enough for all the nodes in the views w_i, $i = 0,\dots,d$, to be provided with distinct IDs. Namely, $|X| \geq N(d, t+1)$, where $N(d, t+1)$ denotes the maximum number of nodes in the ball of radius t in a graph in $\mathcal{G}$.

Two mappings from $\mathcal{I}_{d,X}$ play a crucial role. The first is the simplicial map $\pi : \mathcal{I}_{d,X} \to \mathcal{I}_d$ defined by $\pi(p_i, (\mathsf{id}, x)) = (p_i, x)$ for every $i = 0,\dots,d$, every $\mathsf{id} \in X$, and every $x \in \mathcal{L}_{in}$. The second is the carrier map $\Xi_t : \mathcal{I}_{d,X} \to 2^{\mathcal{P}^{(t)}_{d,X}}$ that specifies, for each facet $F \in \mathcal{I}_{d,X}$, the set $\Xi_t(F)$ of facets which may result from F after t rounds of computation in graphs in $\mathcal{G}$. Specifically, they are merely the facets of $\mathcal{P}^{(t)}_{d,X}$ for which the views $w_0,\dots,w_d$ are compatible with the IDs and inputs of $p_0,\dots,p_d$ in F.

Our main result is an analog of the generic lemma (see Lemma 1), but involving local complexes, even for tasks defined on arbitrarily large networks, and for arbitrarily large sets of IDs.

▶ **Theorem 3.** *Let $T = (\mathcal{I}_d, \mathcal{O}_d, \Delta)$ be an LCL task for $\mathcal{G} \subseteq \mathcal{G}_d$, and let $t \geq 0$.*

- *If there exists a distributed algorithm solving T in t rounds in the LOCAL model then, for every $R \geq N(d, t+1)$, there is a name-independent and name-preserving simplicial map $\delta : \mathcal{P}^{(t)}_{d,[R]} \to \mathcal{O}_d$ such that, for every facet $F \in \mathcal{I}_{d,[R]}$, $\delta(\Xi_t(F)) \subseteq \Delta(\pi(F))$.*
- *There exists $R \geq N(d, t+1)$ satisfying that, if there is a name-independent and name-preserving simplicial map $\delta : \mathcal{P}^{(t)}_{d,[R]} \to \mathcal{O}_d$ such that, for every facet $F \in \mathcal{I}_{d,[R]}$, $\delta(\Xi_t(F)) \subseteq \Delta(\pi(F))$, then there is a distributed algorithm solving T in t rounds in the LOCAL model.*

Proof. Let us fix an LCL task $T = (\mathcal{L}_{in}, \mathcal{S}_{in}, \mathcal{L}_{out}, \mathcal{S}_{out}) = (\mathcal{I}_d, \mathcal{O}_d, \Delta)$ for $\mathcal{G}$, and $t \geq 0$. Let ALG be a t-round algorithm for T. For any finite set X of IDs, let $\delta_X : \mathcal{P}^{(t)}_{d,X} \to \mathcal{O}_d$ defined by $\delta_X(p_i, w_i) = (p_i, \text{ALG}(w_i))$, for every $i = 0,\dots,d$. By construction, δ_X is name-independent, and name-preserving. To show that δ_X is simplicial, let $F' = \{(p_i, w_i) : i \in \{0,\dots,d\}\}$ be a facet of the protocol complex $\mathcal{P}^{(t)}_{d,X}$. This facet is mapped to $\delta_X(F') = \{(p_i, \text{ALG}(w_i)) :$

$i \in \{0, \ldots, d\}\}$. Since ALG solves T, every output ALG(w_i) is in $\mathcal{L}_{out}$, and the labeled star resulting from assigning label ALG(w_i) to the node p_i of the star S_d, for every $i \in \{0, \ldots, d\}$, is in $\mathcal{S}_{out}$. It follows that $\delta_X(F')$ is a facet of $\mathcal{O}_d$, and thus δ_X is simplicial. Moreover, if the facet F' belongs to the image $\Xi_t(F)$ of a facet F of $\mathcal{I}_{d,X}$, since ALG solves T, it follows that $\delta_X(F') \in \Delta(\pi(F))$ as desired. So, the existence of an algorithm ALG guarantees the existence of a simplicial map δ_X satisfying the requirements of the theorem for *every* large enough set X of IDs.

We now show that, to guarantee the existence of an algorithm, it is sufficient to guarantee the existence of a simplicial map δ_X just for *one* specific set $X = [R]$. In order to identify R, we follow the same guideline as the specific impossibility proof in Section 4.3, using Ramsey's theorem. Note that the number of possible balls of radius t in graphs of $\mathcal{G}$ is finite, and depends only on t and d. Given such a ball B, there are finitely many ways of assigning input labels to the vertices of B. The number of assignments depends only on the structure of B, and on $|\mathcal{L}_{in}|$. (It may also depend on $\mathcal{S}_{in}$, but in the worst case, all assignments are possible.) Let us enumerate all the labeled balls in $\mathcal{G}$ as $B^{(1)}, \ldots, B^{(k)}$. The number k of such labeled balls depends only on d, t, and $|\mathcal{L}_{in}|$. (It may also depend on $\mathcal{G}$, but it is upper bounded by a function of d, t, and $|\mathcal{L}_{in}|$.)

For every labeled ball $B^{(i)}$, $i = 1, \ldots, k$, let $\nu_i = |B^{(i)}|$. Let us rank the vertices of $B^{(i)}$ arbitrarily from 1 to ν_i, and let Σ_i be the set of all permutations of $\{1, \ldots, \nu_i\}$. To every $\pi \in \Sigma_i$ corresponds a labeled ball $B^{(i)}_\pi$ in which the rank of the vertices is determined by π. Now, let X be a finite set of IDs with $|X| \geq N(d, t+1)$. We consider all possible identity-assignments with IDs in X to the nodes of the labeled balls with ranked vertices, $B^{(i)}_\pi$, $i = 1, \ldots, k$, $\pi \in \Sigma_i$, as follows. For every $S \subseteq X$ with $|S| = N(d,t)$, let us order the IDs in S in increasing order. Given a ranked labeled ball $B^{(i)}_\pi$, i.e., a labeled ball $B^{(i)}$ whose vertices are ranked by some permutation $\pi \in \Sigma_i$, the IDs in S are assigned to the nodes of $B^{(i)}_\pi$ by assigning the jth smallest ID in S to the node ranked $\pi(j)$ in $B^{(i)}_\pi$, for $j = 1, \ldots, \nu_i$. By picking all $i = 1, \ldots, k$, all $\pi \in \Sigma_i$, and all $S \subseteq X$, we obtain all possible views resulting from performing a t-round algorithm in $\mathcal{G}$ with IDs taken from X. Let us order these views as $w^{(1)}, \ldots, w^{(h)}$, where the views induced by $B^{(1)}$ are listed first, then the views induced by $B^{(2)}$, etc., until the views induced by $B^{(k)}$. Moreover, for a given $i \in \{1, \ldots, k\}$, the views corresponding to the labeled ball $B^{(i)}$ are listed according to the lexicographic order of the permutations in Σ_i. Note that the number h of views depends only on d, t, $|\mathcal{L}_{in}|$, and $|X|$. Each set S is then "colored" by

$$c(S) = (\delta_X(p_0, w^{(1)}), \ldots, \delta_X(p_0, w^{(h)})) \in \{1, \ldots, |\mathcal{L}_{out}|\}^h.$$

In this way, the set $\binom{X}{N(d,t)}$ is partitioned into $|\mathcal{L}_{out}|^h$ classes. Thanks to Ramsey's Theorem, by taking set

$$X = [R] \text{ with } R = R(a, b, c) \text{ for } a = |\mathcal{L}_{out}|^h, \text{ and } b = c = N(d, t+1),$$

we are guaranteed that there exists a set Y of at least $N(d, t+1)$ IDs such that every two sets S and S' of $N(d,t)$ IDs in Y are given the same color $c(S) = c(S')$. In other words, for any ball B of radius t in a graph from $\mathcal{G}$, and for every valid assignment of inputs values to the nodes of B, if one assigns the IDs in S and S' in the same manner (i.e., the ith smallest ID of S is assigned to the same node as the ith smallest ID of S'), then $\delta_X(p_0, w) = \delta_X(p_0, w')$, where w and w' are the views resulting from assigning IDs from S and S' to the nodes, respectively.

Now, let us define the following t-round algorithm ALG for T. Actually, this is precisely the order-invariant algorithm constructed in [16]. Every node v collects the data available in its centered ball $B = B_G(v,t)$ of radius t in the actual graph $G \in \mathcal{G}$. Note that B contains IDs, and input values. Node v reassigns the IDs to the nodes of B by using the $|B|$ smallest IDs in Y, and assigning these IDs to the nodes of B in the order respecting the order of the actual IDs assigned to the nodes of B. Then node v considers the view w after reassignment of the IDs, and outputs ALG$(w) = \delta_X(p_0, w)$. Note that δ_X returns values in $\mathcal{L}_{out}$, and thus ALG is well defined.

To show correctness, let us consider a star $v_0, \dots, v_d$ centered at v_0 in some graph $G \in \mathcal{G}$. Performing ALG in G, each of these $d+1$ nodes acquires a view of radius t. These views are mutually compatible. Let us reassign the IDs in the ball of radius $t+1$ centered at v_0 in G, using the at most $N(d, t+1)$ smallest IDs in Y, and assigning these IDs to the nodes of the ball B of radius $t+1$ centered at v_0, in the order respecting the order of the actual IDs assigned to the nodes of B. The resulting views $w_0, \dots, w_d$ of the $d+1$ nodes $v_0, \dots, v_d$ remain mutually compatible. It follows that if these $d+1$ nodes would output $\delta_X(p_0, w_0), \dots, \delta_X(p_d, w_d)$, respectively, then the resulting star would be good. We claim that this is exactly what occurs with ALG. Indeed, first, δ_X is name-independent, and thus $\delta_X(p_0, w) = \delta_X(p_i, w)$ for every $i = 1, \dots, d$. Second, and more importantly, by the construction of Y, the actual *values* of the IDs do not matter, but solely their *relative order*. The reassignment of IDs performed at each of the nodes $v_0, \dots, v_d$ is different from the reassignment of IDs in the ball B of radius $t+1$ around v_0, but the relative order of these IDs is preserved as it is governed by the relative order of the original IDs in B. As a consequence, the nodes of S_d correctly output $\delta_X(p_0, w_0), \dots, \delta_X(p_d, w_d)$ in ALG, as desired. ◀

To illustrate Theorem 3, we reprove the celebrated result by Linial [15] regarding 3-coloring the n-node ring in at least $\frac{1}{2}\log^* n - 1$ rounds (see also [2, 3, 13]), which can be obtained by iterating Corollary 4.

▶ **Corollary 4.** *Let $t \geq 1$, $k \geq 2$, $n \geq 1$, and $N \geq n$. If there is a t-round algorithm for k-coloring $C_n = (v_1, \dots, v_n)$ whenever the IDs in $[N]$ are assigned to consecutive nodes v_i, v_{i+1}, $i \in \{1, \dots, n-1\}$, in increasing order of their indices, then there is a $(t-1)$-round algorithm for 2^{2^k}-coloring C_n under the same constraints of the identity assignment.*

Proof. Our aim is to find $\delta_{t-1} : \mathcal{P}^{(t-1)}_{[R]} \to \mathcal{O}_{2^{2^k}}$ where $\mathcal{O}_{2^{2^k}}$ is the output complex for 2^{2^k}-coloring C_n. For this purpose, we follows the approach illustrated on Figure 10. That is, first, we identify a functor Φ on a category corresponding to a subclass of simplicial complexes. From the simplicial map $\delta_t : \mathcal{P}^{(t)}_{[R]} \to \mathcal{O}_k$, we derive the simplicial map $\Phi(\delta_t) : \Phi(\mathcal{P}^{(t)}_{[R]}) \to \Phi(\mathcal{O}_k)$. Then we show that $\Phi(\mathcal{O}_k) \subseteq \mathcal{O}_{2^{2^k}}$, and therefore $\Phi(\delta_t)$ maps $\Phi(\mathcal{P}^{(t)}_{[R]})$ to $\mathcal{O}_{2^{2^k}}$. Finally, we identify a simplicial map $f : \mathcal{P}^{(t-1)}_{[R]} \to \Phi(\mathcal{P}^{(t)}_{[R]})$ enabling to conclude that $\delta_{t-1} : \mathcal{P}^{(t-1)}_{[R]} \to \mathcal{O}_{2^{2^k}}$ defined by $\delta_{t-1} = \Phi(\delta_t) \circ f$ satisfies the hypotheses of Theorem 3, showing the existence of a $(t-1)$-round algorithm for 2^{2^k}-coloring C_n.

More specifically, given any complex $\mathcal{K}$ with vertices (p_i, v) with $i \in \{-1, 0, 1\}$, and $v \in V$ where V is a finite set of values, we define the functor Φ as follows. The complex $\Phi(\mathcal{K})$ is on the set of vertices $(p_i, \mathbf{S})$ where $\mathbf{S} = \{S_1, \dots, S_\ell\}$ for some $\ell \geq 0$, and $S_i \subseteq V$ for every $i = 1, \dots, \ell$. A set $\{(p_{-1}, \mathbf{S}_{-1}), (p_0, \mathbf{S}_0), (p_1, \mathbf{S}_1)\}$ forms a facet of $\Phi(\mathcal{K})$ if for every $i \in \{0, 1\}$,

$$\exists S \in \mathbf{S}_{i-1} \;\; \forall S' \in \mathbf{S}_i \;\; \exists v' \in S' \;\; \forall v \in S \; : \; \{(p_{i-1}, v), (p_i, s')\} \in \mathcal{K}. \qquad (1)$$

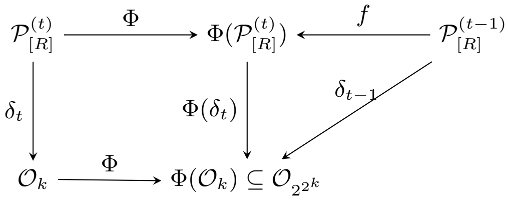

Figure 10 Commutative diagrams in the proof of Corollary 4.

Given a simplicial map $\psi : \mathcal{A} \to \mathcal{B}$ the map $\Phi(\psi)$ is defined as

$$\Phi(\psi)(p_i, \mathbf{S}) = \Big(p_i, \Big\{\{\pi_2 \circ \psi(p_i, v_{1,1}), \dots, \pi_2 \circ \psi(v_{1,s_1})\}, \dots, \{\pi_2 \circ \psi(v_{\ell,1}), \dots, \pi_2 \circ \psi(v_{\ell,s_\ell})\}\Big\}\Big)$$

for every $i = \{-1, 0, 1\}$, and every $\mathbf{S} = \{S_1, \dots, S_\ell\}$ with $S_j = \{v_{j,1}, \dots, v_{j,s_j}\}$ and $s_j \geq 0$, where $\pi_2 : \mathcal{B} \to V$ is the mere projection $\pi_2(p_i, v) = v$ for every value v. By construction, $\Phi(\psi) : \Phi(\mathcal{A}) \to \Phi(\mathcal{B})$ is simplicial. Note that if ψ is name-invariant and name-preserving, then $\Phi(\psi)$ as well.

Next, we observe that $\Phi(\mathcal{O}_k)$ is a sub-complex of $\mathcal{O}_{2^{2^k}}$. To see why, note first that Φ maps vertices of $\mathcal{O}_k$ to vertices of $\mathcal{O}_{2^{2^k}}$. Moreover, a facet $F = \{(p_{-1}, \mathbf{S}_{-1}), (p_0, \mathbf{S}_0), (p_1, \mathbf{S}_1)\}$ of $\Phi(\mathcal{O}_k)$ is a facet of $\mathcal{O}_{2^{2^k}}$. Indeed, Eq. (1) guarantees that there exists a set S in $\mathbf{S}_{-1}$ such that for every set S' in $\mathbf{S}_0$, there exists a color v' in S' that is different from all the colors in S. It follows that $S \notin \mathbf{S}_0$, and therefore $\mathbf{S}_{-1} \neq \mathbf{S}_0$. By the same argument, $\mathbf{S}_0 \neq \mathbf{S}_1$, and thus F is a facet of $\mathcal{O}_{2^{2^k}}$, as claimed. Finally, we define the simplicial map $f : \mathcal{P}^{(t-1)}_{[R]} \to \Phi(\mathcal{P}^{(t)}_{[R]})$ as follows. Let us consider a vertex $(p_i, w) \in \mathcal{P}^{(t-1)}_{[R]}$, with $w = (z_{-(t-1)}, \dots, z_{-1}, z_0, z_1, \dots, z_{t-1}) \in [R]^{2t-1}$ with $z_{-(t-1)} < \dots < z_{t-1}$. For every $b \in [R]$ with $b > z_{t-1}$, let $W_i^b = \{awb : a \in [R], a < z_{-(t-1)}\}$, and let $\mathbf{W}_i = \{W_i^b : b \in [R], b > z_{t-1}\}$. We set $f(p_i, w) = (p_i, \mathbf{W}_i)$. This mapping maps every vertex of $\mathcal{P}^{(t-1)}_{[R]}$ to a vertex of $\Phi(\mathcal{P}^{(t)}_{[R]})$. Let us show that f is simplicial. For this purpose, let us consider a facet

$$F = \{(p_{-1}, x'xw), (p_0, xwy), (p_1, wyy')\}$$

of $\mathcal{P}^{(t-1)}_{[R]}$. Here $w = (z_{-(t-2)}, \dots, z_{-1}, z_0, z_1, \dots, z_{t-2}) \in [R]^{2t-3}$ with $x' < x < z_{-(t-2)} < \dots < z_{t-2} < y < y'$. Let us consider the two sets $W_{-1}^y \in \mathbf{W}_{-1}$ and $W_0^{y'} \in \mathbf{W}_0$. We claim that these are the two sets witnessing the validity of Eq. (1) for establishing the fact that $f(F)$ is a facet of $\Phi(\mathcal{P}^{(t)}_{[R]})$. To see why, let $W_0^b \in \mathbf{W}_0$, and let $x'xwyb \in W_0^b$. The view $ax'xwy$ for p_{-1} is compatible with the view $x'xwyb$ for p_0, for every $a < x'$. Therefore, for every set $W_0^b \in \mathbf{W}_0$, there exists a view $x'xwyb \in W_0^b$ such that, for every view $ax'xwy \in W_{-1}^y$,

$$\{(p_{-1}, ax'xwy), (p_0, x'xwyb)\} \in \mathcal{P}^{(t)}_{[R]}.$$

Hence Eq. (1) is satisfied for p_{-1} and p_0. By the same arguments, using $W_0^{y'}$ instead of W_{-1}^y, Eq. (1) is satisfied for p_{-1} and p_0, from which it follows that $f(F)$ is a facet of $\Phi(\mathcal{P}^{(t)}_{[R]})$. We conclude that f is simplicial. Since both f and $\Phi(\delta)$ are simplicial, the map $\delta' = \Phi(\delta) \circ f$ is simplicial too, which completes the proof by application of Theorem 3. ◀

6 Conclusion and Further Work

This paper shows that the study of algorithms for solving LCL tasks in the LOCAL model can be achieved by considering simplicial complexes whose sizes are independent of the number of nodes, and independent of the number of possible IDs that could be assigned to these nodes.

Two main directions for further work can be identified. A first direction is understanding topological properties of the carrier map $\Xi_t : \mathcal{I}_{d,X} \to \mathcal{P}^{(t)}_{d,X}$ occurring in the LOCAL model depending on the network. Another direction is understanding what governs the existence of the simplicial map $\delta : \mathcal{P}^{(t)}_{d,X} \to \mathcal{O}$ depending on the considered task.

References

1 Hagit Attiya, Armando Castañeda, Maurice Herlihy, and Ami Paz. Bounds on the step and namespace complexity of renaming. *SIAM J. Comput.*, 48(1):1–32, 2019. doi:10.1137/16M1081439.

2 Alkida Balliu, Sebastian Brandt, Juho Hirvonen, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. Lower bounds for maximal matchings and maximal independent sets. In *60th IEEE Annual Symposium on Foundations of Computer Science (FOCS)*, pages 481–497, 2019. doi:10.1109/FOCS.2019.00037.

3 Sebastian Brandt. An automatic speedup theorem for distributed problems. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 379–388, 2019. doi:10.1145/3293611.3331611.

4 Armando Castañeda, Pierre Fraigniaud, Ami Paz, Sergio Rajsbaum, Matthieu Roy, and Corentin Travers. A topological perspective on distributed network algorithms. In *26th International Colloquium on Structural Information and Communication Complexity (SIROCCO)*, LNCS 11639, pages 3–18. Springer, 2019. doi:10.1007/978-3-030-24922-9_1.

5 Armando Castañeda and Sergio Rajsbaum. New combinatorial topology bounds for renaming: the lower bound. *Distributed Computing*, 22(5-6):287–301, 2010. doi:10.1007/s00446-010-0108-2.

6 Armando Castañeda and Sergio Rajsbaum. New combinatorial topology bounds for renaming: The upper bound. *J. ACM*, 59(1):3:1–3:49, 2012. doi:10.1145/2108242.2108245.

7 Lisbeth Fajstrup, Eric Goubault, Emmanuel Haucourt, Samuel Mimram, and Martin Raussen. *Directed Algebraic Topology and Concurrency*. Springer, 2016.

8 Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. Impossibility of distributed consensus with one faulty process. *J. ACM*, 32(2):374–382, 1985. doi:10.1145/3149.214121.

9 Pierre Fraigniaud and Ami Paz. The topology of local computing in networks. Tech. Report arXiv abs/2003.03255, 2020. arXiv:2003.03255.

10 Éric Goubault, Samuel Mimram, and Christine Tasson. Geometric and combinatorial views on asynchronous computability. *Distributed Computing*, 31(4):289–316, 2018. doi:10.1007/s00446-018-0328-4.

11 Maurice Herlihy, Dmitry N. Kozlov, and Sergio Rajsbaum. *Distributed Computing Through Combinatorial Topology*. Morgan Kaufmann, 2013. URL: https://store.elsevier.com/product.jsp?isbn=9780124045781.

12 Maurice Herlihy and Nir Shavit. The topological structure of asynchronous computability. *J. ACM*, 46(6):858–923, 1999. doi:10.1145/331524.331529.

13 Juhana Laurinharju and Jukka Suomela. Brief announcement: Linial's lower bound made easy. In *ACM Symposium on Principles of Distributed Computing (PODC)*, pages 377–378, 2014. doi:10.1145/2611462.2611505.

14 Ronald L.Graham, Bruce L. Rothschild, Joel H. Spencer, and József Solymosi. *Ramsey Theory*. John Wiley and Sons, 2015.

15 Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992. doi:10.1137/0221015.

16 Moni Naor and Larry J. Stockmeyer. What can be computed locally? *SIAM J. Comput.*, 24(6):1259–1277, 1995. doi:10.1137/S0097539793254571.

17 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. SIAM, 2001.

18 Michael E. Saks and Fotios Zaharoglou. Wait-free k-set agreement is impossible: The topology of public knowledge. *SIAM J. Comput.*, 29(5):1449–1483, 2000. doi:10.1137/S0097539796307698.

19 Jukka Suomela. Survey of local algorithms. *ACM Comput. Surv.*, 45(2):24:1–24:40, 2013. doi:10.1145/2431211.2431223.