The Distributed Computing Column

Seth Gilbert
National University of Singapore
seth.gilbert@comp.nus.edu.sg

This month, in the Distributed Computing Column, Francesco d'Amore surveys some recent exciting results in quantum distributed computing, focusing on the question of when a distributed quantum advantage is possible. That is, imagine you have a network of quantum computers connected via quantum communication channels; when can such a network solve a problem faster than in the classical LOCAL model? While general versions of the questions remain open, there has been significant and compelling progress in the last several years.

To answer this question of quantum advantage, d'Amore begins by examining the fundamental principles that enable fast quantum distributed computation, e.g., issues related to causality and independence of output distributions. He explores the landscape of intermediate models, including those both stroger and weaker than the "quantum-LOCAL" model, including discussions of "non-signalling" models, "bounded dependence" models, and online local models. These models provide key technical tools for excluding quantum advantage—or showing when it may help. The survey then describes a recently discovered concrete example of a local problem that is provably faster in the quantum models: it can be solved in $O(1)$ time in quantum-LOCAL, but requires $\Omega(\Delta)$ time in classical LOCAL (where Δ is the maximum degree of the network graph).

Overall, then, this article provides a comprehensive overview of the state-of-the-art for quantuam distributed algorithms today, along with some nice insight into the tools and techniques needed to understand the performance of these algorithms.

The Distributed Computing Column is particularly interested in contributions that propose interesting new directions and summarize important open problems in areas of interest. If you would like to write such a column, please contact me.

On the limits of distributed quantum computing

Francesco d'Amore*

*Bocconi University, BIDSA, Italy

Abstract

Quantum advantage is well-established in centralized computing, where quantum algorithms can solve certain problems exponentially faster than classical ones. In the distributed setting, significant progress has been made in bandwidth-limited networks, where quantum distributed networks have shown computational advantages over classical counterparts. However, the potential of quantum computing in networks that are constrained only by large distances is not yet understood. We focus on the LOCAL model of computation (Linial, FOCS 1987), a distributed computational model where computational power and communication bandwidth are unconstrained, and its quantum generalization. In this brief survey, we summarize recent progress on the quantum-LOCAL model outlining its limitations with respect to its classical counterpart: we discuss emerging techniques, and highlight open research questions that could guide future efforts in the field.

1 Introduction

Since the advent of quantum computing, extensive research has been conducted to explore its potential, revealing its advantage over classical computing, at least from a theoretical point of view: there are problems that classical algorithms solve in super-polynomial time, while quantum algorithms can solve them in polynomial time [Aar22]. But what about *distributed* quantum advantage?

There is a large body of research investigating this question in *bandwidth-limited* networks [Cen+22; LM18; WY22; WWY21; IL19; ILM20; AV22; Fra+24; MN22]. Such networks are captured by (possibly variants of) the CONGEST model of computing. In essence, the question is: can a synchronous network of quantum computers that send b quantum qubits per time unit to each neighbor outperform a synchronous network of classical machines that send b classical bits

per time unit? It turns out that in many cases, the answer is *yes*. There are computational tasks that are asymptotically easier to solve in the quantum-CONGEST model (and related variants): see, e.g., [IL19; ILM20; LM18].

However, if we consider instead networks that are constrained only by *large distances*, the scenario is much less understood. Such networks typically model distributed systems where network latency and the time required for information to propagate play key roles. We emphasize that large distances are a fundamental physical limitation of distributed networks: information, whether classical or quantum, cannot travel faster than the speed of light. This limitation cannot be overcome by technological progress, unlike bandwidth constraints, which can benefit from innovations (say, the installation of multiple parallel communication channels by, e.g., increasing the number of fiber-optic links between nodes).

Such distance-constrained networks are modeled by the famous LOCAL model of computation, first introduced in the seminal work by Linial [Lin87] (see [Lin92] for the journal version). In the LOCAL model, a distributed network is represented as a graph $G = (V, E)$, where the nodes of G are processors capable of *unbounded local computation*, and the edges represent communication links. Time proceeds synchronously: in each round, nodes send and receive messages of *arbitrarily large size* from their neighbors and perform local computations to update their state variables. Eventually, all nodes announce their outputs, marking the end of the computation. The complexity measure in this model is the number of communication rounds required to solve a problem, captured by the notion of *locality*. Specifically, T rounds of communication allow a node to gather information about the topology and input within its radius-T neighborhood. Thus, T is also called the *locality* of the algorithm, reflecting why distances are the only limitation in this model.

The LOCAL model has been extensively studied [Lin92; Nao91; NS95; CKP16; CKP19; GKM09; CP19; GHK18; FG17; GKM17; Bra+16; Hir+17; KMW04; KSV13]. A specific class of problems of particular interest in distributed computing is *locally checkable labeling* (LCL) problems, introduced by Naor and Stockmeyer [NS95]. LCL problems are defined via local constraints (e.g., graph coloring). For LCL problems, a solution may be *hard to find* but is *easy to verify* with a distributed algorithm. As such, LCL problems can bee seen as the distributed analogue of the FNP class in centralized computation. Nowadays, we have a good understanding of complexity landscape of LCL problems in the LOCAL model [NS95; CKP19; CKP16; CP19; Bal+18; Bal+19a; Bal+19b; Bal+21a; Bal+20; Bal+21b; Bal+22b; Bal+22a; Bal+23a; Bal+23b; Bra+17; Bra19; Akb+23].

To date there is no clear understanding of the impact of quantum computation and communication in the LOCAL model, especially concerning LCL problems. The main challenge lies in the lack of tools for directly tackling quantum-LOCAL, particularly for establishing lower bounds. Nevertheless, general arguments based

on physical principles, such as *causality* and *independence* of output distributions, provide some insights.

In this brief survey, we summarize previous knowledge and recent results, introducing new techniques for investigating the role of quantum computation and communication in distributed settings and shedding light on potential research directions.

2 Preliminaries

In order to proceed, we need to provide the mathematical framework in which we work. We start with some basic graph notations and definitions, introducing the class of problems we consider and the computational model that is the heart of our investigation. Section by section, we will give other definitions that are useful for that section and the subsequent ones. We denote the set of natural numbers (starting from 0) by $\mathbb{N}$, and also define $\mathbb{N}_+ = \mathbb{N} \setminus \{0\}$.

Graphs. We work with simple graphs unless otherwise specified. Given any set S, we denote by $\binom{S}{k}$ the set whose elements are all sets of k different elements of S. We will consider both undirected and directed graphs. A graph is a pair $G = (V, E)$ where V is the set of nodes and E is the set of edges. In case of undirected graphs, $E \subseteq \binom{V}{2}$, whereas $E \subseteq V \times V$ in case of directed graphs. For any graph G, we also denote its set of nodes by $V(G)$ and its set of edges by $E(G)$.

The distance between two nodes u, v of any graph G is the number of edges in any shortest path between u and v (note that the shortest path is not necessarily an oriented path), and is denoted by $\mathrm{dist}_G(u, v)$. The notion of distance can be easily extended to subset of nodes: Given any node $u \in V$ and any two subsets $A, B \subseteq V$, the distance between u and A is $\mathrm{dist}_G(u, A) = \min_{v \in A}\{\mathrm{dist}_G(u, v)\}$ and the distance between A and B is $\mathrm{dist}_G(A, B) = \min_{u \in A, v \in B}\{\mathrm{dist}_G(u, v)\}$. When the graph is clear from the context, we omit the suffix and write only dist() instead of $\mathrm{dist}_G()$. Through the notion of distance, we define the diameter of a graph G to be $\mathrm{diam}(G) = \max_{u,v \in V(G)} \mathrm{dist}_G(u, v)$.

For any non-negative integer T, the radius-T (closed) neighborhood of a node u in a graph G is the set $\mathcal{N}_T[u] = \{v \in V \mid \mathrm{dist}(u, v) \leq T\}$. Throughout this survey, we will only make use of closed neighborhoods. More in general, the radius-T neighborhood of any subset of node $A \subseteq V$ is $\mathcal{N}_T[A] = \cup_{u \in A} \mathcal{N}_T[u]$. We also define the *ring neighborhood* between T_1 and T_2 of a node $u \in V(G)$ (for $T_1 \leq T_2$) as $\mathcal{N}^{T_1}_{T_2}[u] = \mathcal{N}_{T_2}[u] \setminus \mathcal{N}_{T_1}[u]$. We use an analogous notation for subsets of nodes.

The degree of a node v in an undirected graph G is the number of edges the node belongs to, i.e., $\deg_G(v) = |\{\{u, v\} \in E(G) \mid u \in V(G)\}|$. In a directed graph G, we define the *indegree* and the *outdegree* of a node v as follows: $\mathrm{indeg}_G(v) =$

$|\{(u,v) \in E(G) \mid u \in V(G)\}|$ and $\text{outdeg}_G(v) = |\{(v,u) \in E(G) \mid u \in V(G)\}|$. Then the degree of v in G is just $\deg_G(v) = \text{indeg}_G(v) + \text{outdeg}_G(v)$. In all these notations, we omit the suffix when the graph is clear from the context. Finally, the degree of a graph G is just the maximum degree of any node, i.e., $\deg(G) = \max_{v \in V(G)}\{\deg_G(v)\}$.

For any graph $G = (V, E)$ and any subset of nodes $A \subseteq V$, the subgraph of G induced by A is denoted by $G[A]$. Consider any node $u \in V$ (or any subset $S \subseteq V$): with an abuse of notation, we define the *open induced subgraph* as the set $\mathring{G}[\mathcal{N}_T[u]] = G[\mathcal{N}_T[u]] \setminus G[\mathcal{N}_T^{T-1}[u]]$. (or $\mathring{G}[\mathcal{N}_T[S]] = G[\mathcal{N}_T[S]] \setminus G[\mathcal{N}_T^{T-1}[S]]$):. In practice, in this definition we are removing from the classical notion of neighborhood the edges that connect nodes that are at distance T from u (or S) as the graph that u (or S) *sees* by moving T hops away does not include them.

Now we define some graph operations. Given any two graphs G and H, the intersection of G and H is the graph $G \cap H = (V(G) \cap V(H), E(G) \cap E(H))$. The union of G and H is the graph $G \cup H = (V(G) \cup V(H), E(G) \cup E(H))$, while the difference between G and H is the graph $G \setminus H = (V(G) \setminus V(H), E(G) \setminus E(H))$.

An isomorphism between two graphs G and H is a function $\varphi : V(G) \to V(H)$ that is bijective and such that $\{u, v\} \in E(G)$ if and only if $\{\varphi(u), \varphi(v)\} \in E(H)$. In case of directed graphs, we also require the isomorphism to keep edge orientation.

Labeling problems. In this brief survey we consider labeling problems, namely, graph problems that ask to output some labels on the nodes of the graph. A formal definition follows.

Definition 2.1 (Labeling problem)**.** Let $\Sigma_{\text{in}}, \Sigma_{\text{out}}$ be two alphabets, and I a set of indices. A labeling problem Π is a mapping $(G, \text{in}) \mapsto \{\text{out}_i\}_{i \in I}$ that maps every input graph $G = (V, E)$ where nodes are labelled by any input function $\text{in} : V \to \Sigma_{\text{in}}$ to a family of suitable output functions $\text{out}_i : V \to \Sigma_{\text{out}}$ indexed by I. The mapping is closed under graph isomorphism, that is, for any isomorphism $\varphi : V(G) \to V(H)$ between two graphs G and H, $\text{out} \in \Pi((H, \text{in}))$ if and only if $\text{out} \circ \varphi \in \Pi((G, \text{in} \circ \varphi))$.

The reader may notice that we defined labeling problems for *any input graph*, despite the fact that some labeling problems might be defined only for some specific graph family like, for example, 3-coloring bipartite graphs. However, Definition 2.1 is general enough to capture all such problems, because one can just say that, for graphs that are outside the right graph family (e.g., non-bipartite graphs), all outputs are admissible. Examples of graph labeling problems are leader election, consensus, diameter approximation, etc.

A subclass of labeling problems that is of particular interest in the distributed computing community is that of locally checkable labeling (LCL) problems, already mentioned in the introduction. LCL problems were first introduced by Naor

and Stockmeyer [NS93] (for the journal version we refer to [NS95]). Here, we report the original definition.

For any function $f : A \to B$ and any subset $S \subseteq A$, we denote the restriction of f to S by $f \restriction_S : S \to B$. We define a *centered graph* to be any pair (G, v_G), where G is a graph and $v_G \in V(G)$ is a node of G that we call the *center* of G. The *radius* of a centered graph (G, v_G) is the maximum distance between v_G and any other node of G. We are now ready to state the definition of LCL problems.

Definition 2.2 (Locally Checkable Labeling problem)**.** Let r, Δ be non-negative integers. Let $\Sigma_{\text{in}}, \Sigma_{\text{out}}$ be two finite alphabets, and I a finite set of indices. Consider a labeling problem Π defined on $\Sigma_{\text{in}}, \Sigma_{\text{out}}, I$. Π is *locally checkable* with checking radius r and maximum degree Δ if there exists a family $\mathcal{S} = \{((H, v_H), \text{in}, \text{out})_i\}_{i \in I}$ where each tuple $((H, v_H), \text{in}, \text{out})_i$ contains a centered graph (H, v_H) of radius at most r and degree at most Δ, an input labeling function $\text{in} : V(H) \to \Sigma_{\text{in}}$ and an output labeling function $\text{out} : V(H) \to \Sigma_{\text{out}}$ with the following property:

- For every input (G, in) to Π with $\deg(G) \leq \Delta$, an output vector $\text{out} : V(G) \to \Sigma_{\text{out}}$ is admissible (i.e., $\text{out} \in \Pi((G, \text{in}))$) if and only if, for each node $v \in V(G)$, the tuple $((G[\mathcal{N}_T[v]], v), \text{in} \restriction_{\mathcal{N}_T[v]}, \text{out} \restriction_{\mathcal{N}_T[v]})$ belongs to $\mathcal{S}$.

$\mathcal{S}$ is also called the family of *permissible outputs*, and is finite (up to graph isomorphisms) since Δ, r are finite and also $\Sigma_{\text{in}}, \Sigma_{\text{out}}$ are finite sets. Notice that in Definitions 2.1 and 2.2 we assumed that input and output labels are given to and from nodes. We might similarly assume that they are given to and from edges, and come up with other problems. In this survey, we will interchangeably use both possibilities when it is convenient. LCL problems capture all problems defined via local constraints, e.g., graph coloring, maximal independent set, maximal matching, sinkless orientation, triangle freeness, etc.

We now introduce the models of computations we are interested in. We begin with the port-numbering model, and on top of it we define the LOCAL model of computation [Lin92].

The port-numbering model. A port-numbered network is a triple $N = (V, P, p)$ where V is the set of nodes, P is the set of *ports*, and $p : P \to P$ is a function specifying connections between ports. Each element $x \in P$ is a pair (v, i) where $v \in V$, $i \in \mathbb{N}_+$. The connection function p between ports is an involution, that is, $p(p(x)) = x$ for all $x \in P$. If $(v, i) \in P$, we say that (v, i) is port number i in node v. With an abuse of notation, we say that the degree of a node v in the network N is the number of ports in v and is denoted by $\deg_N(v)$. We assume that port numbers are consecutive, i.e., the ports of any node $v \in V$ are $(v, 1), \ldots, (v, \deg_N(v))$. Clearly, a port-numbered network identifies an *underlying graph* $G = (V, E)$ where, for any two nodes $u, v \in V$, $\{u, v\} \in E$ if and only if there

exists ports $x_u, x_v \in P$ such that $p(x_u) = x_v$. Here, the degree of a node $\deg_N(v)$ corresponds to $\deg_G(v)$.

In the port-numbering model we are given a distributed system consisting of a port-numbered network of $|V| = n$ *processors* (or *nodes*) that operates in a sequence of synchronous rounds. In each round the processors may perform unbounded computations on their respective local state variables and subsequently exchange messages of arbitrary size along the links given by the underlying input graph. Nodes identify their neighbors by using ports as defined before, where the port assignment may be done adversarially. Barring their degree, all nodes are identical and operate according to the same local computation procedures. Initially all local state variables have the same value for all processors; the sole exception is a distinguished local variable $\mathrm{x}(v)$ of each processor v that encodes input data (that is, port numbers, degree, possible input from the problem itself, etc.). Usually, we assume that $\mathrm{x}(v)$ also encodes the number of nodes n composing the distributed system.

Let Σ_{in} be a set of input labels. The input of a problem is defined in the form of a labeled graph (G, in) where $G = (V, E)$ is the system graph, V is the set of processors (hence it is specified as part of the input), and $\text{in}\colon V \to \Sigma_{\text{in}}$ is an assignment of an input label $\text{in}(v) \in \Sigma_{\text{in}}$ to each processor v and is encoded in $\mathrm{x}(v)$. The output of the algorithm is given in the form of a function of output labels $\text{out}\colon V \to \Sigma_{\text{out}}$, and the algorithm is assumed to terminate once all labels $\text{out}(v)$ are definitely fixed. We assume that nodes and their links are fault-free. The local computation procedures may be randomized by giving each processor access to its own set of random variables; in this case, we are in the *randomized* port-numbering model as opposed to the *deterministic* port-numbering model.

The running time of an algorithm is the number of synchronous rounds required by all nodes to produce output labels. If an algorithm running time is T, we also say that the algorithm has locality T. Notice that T can be a function of the size (or other parameters) of the input graph. We say that a problem Π over some graph family $\mathcal{F}$ has complexity (or locality) T in the port-numbering model if there is a port-numbering algorithm running in time T that solves Π over $\mathcal{F}$, and $T = T(n)$ is the minimum running time (among all possible algorithms that solve Π over $\mathcal{F}$) in the worst case instance of size n. If the algorithm is randomized, we also require that the failure probability is at most $1/\text{poly}(n)$, where n is the size of the input graph.

We remark that the notion of an (LCL) problem is a graph problem, and does not depend on the specific model of computation we consider (hence, the problem definition cannot depend on, e.g., port numbers).

The LOCAL model. The LOCAL model of computation is just the port-numbering model augmented with an assignment of unique identifiers to nodes. Let $c \geq 1$ be a constant. The nodes of the input graph $G = (V, E)$ are given as input also unique identifiers specified by an injective function $\mathrm{id} : V \to [n^c]$. This assignment might be adversarial and is stored in the local state variable $\mathrm{x}(v)$, and nodes can exploit these values during their local computation.

The local computation procedures may be randomized by giving each processor access to its own set of random variables; in this case, we are in the *randomized* LOCAL (rand-LOCAL) model as opposed to the *deterministic* LOCAL (det-LOCAL) model. If the algorithm is randomized, we also require that the failure probability while solving any problem is at most $1/\mathrm{poly}(n)$, where n is the size of the input graph. The definition of running time, locality and complexity easily extends from the port-numbering model to the LOCAL model.

On top of the LOCAL model, it is easy to describe its quantum generalization. In order to avoid the math of quantum mechanics, we only provide an informal definition of the quantum-LOCAL model. For a formal definition, we defer the reader to [GKM09].

The quantum-LOCAL model. The quantum-LOCAL of computing is similar to the deterministic LOCAL model above, but now with quantum computers and quantum communication links. More precisely, the quantum computers manipulate local states consisting of an unbounded number of qubits with arbitrary unitary transformations, the communication links are quantum communication channels (adjacent nodes can exchange any number of qubits), and the local outputs can be the result of any quantum measurement.

Relations between models. We say that a computational model A is *stronger* than a computational model B if an algorithm with locality T running in A can be simulated by an $O(T)$-round algorithm in B. Clearly, det-LOCAL is stronger than the port-numbering model, rand-LOCAL is stronger than det-LOCAL, and quantum-LOCAL is stronger than rand-LOCAL. We suggest the reader has Fig. 1 at hand to keep track of the models and their relations while we introduce them. We will define the other models present in Fig. 1 later in the related sections.

3 The non-signaling model

As mentioned in the introduction, to date we do not have direct ways to prove lower bounds in quantum-LOCAL. Specific procedures that are commonly used in the LOCAL model such as round elimination [Bra19] do not generalize to quantum-LOCAL (regarding this, we argue more later in Section 6.2). However,

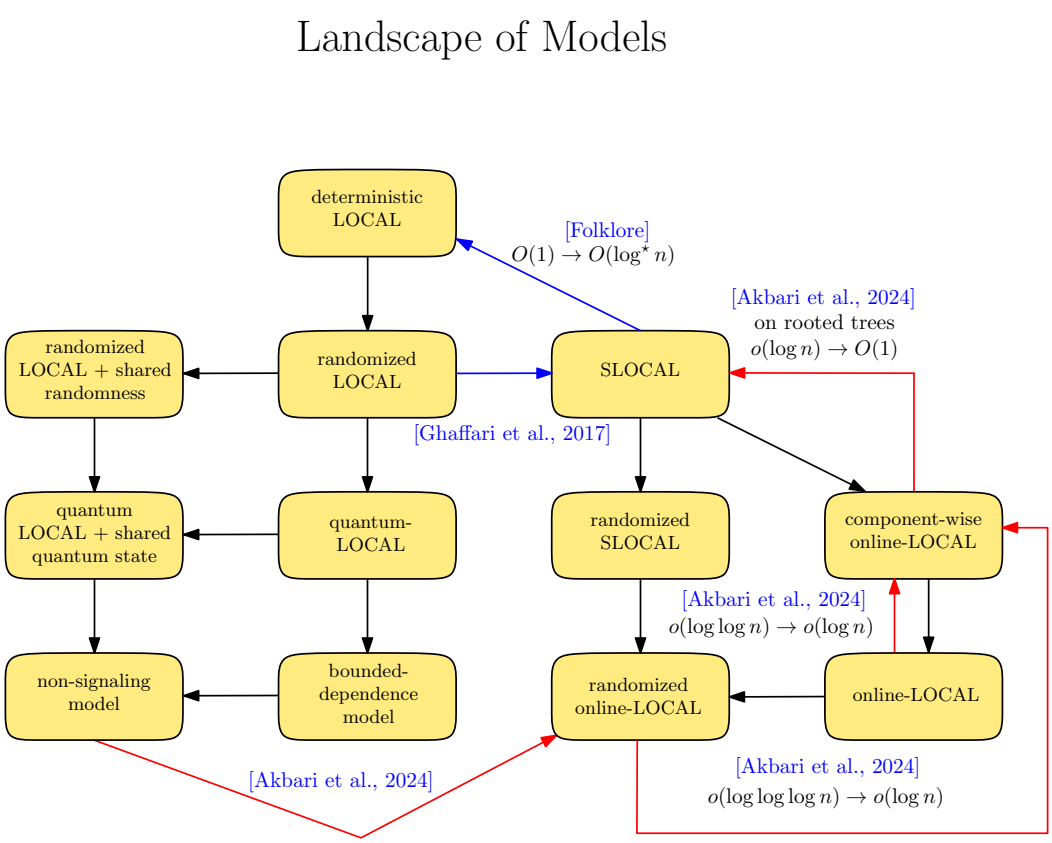


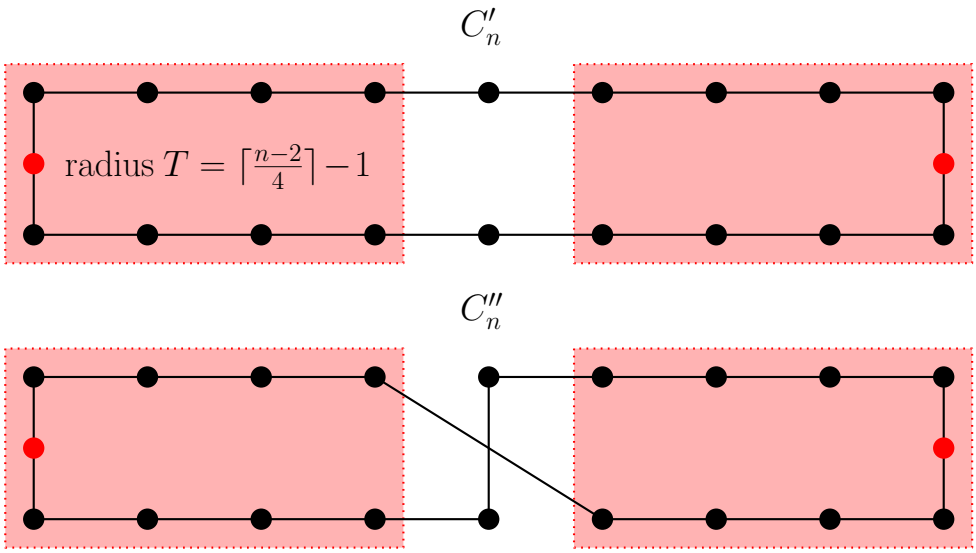
Figure 1: Landscape of computational models. An arrow between model X and Y, that is $X \to Y$ means that model Y is stronger than model X, unless otherwise specified. Black arrows are trivial implications (by construction), blue arrows are known results, and red arrows are recent results.

some general arguments based on the *non-signaling* principle do generalize to quantum-LOCAL and actually holds more in general. Before going through specific definition, let us introduce the no-signaling principle via two examples on the same problem: 2-coloring even cycles both in det-LOCAL and in rand-LOCAL.

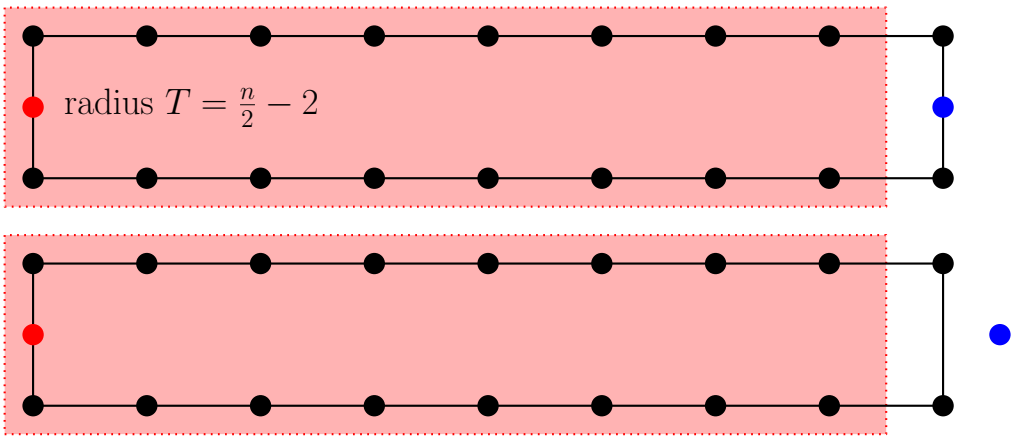
3.1 Warm-up: 2-coloring cycles in classical LOCAL

Let $n \in \mathbb{N}$ be an even number and consider a cycle C_n with n nodes.

First example: indistinguishability argument (*within* the input graph family). The indistinguishability argument in the LOCAL model relies on the fact that a node, by accessing its local view, cannot distinguish between *proper* inputs that are indistinguishable in the local neighborhood but differ outside and, for which,



(a) Indistinguishability argument *within* the same input graph family. The two highlighted nodes cannot distinguish between the case in which they have even (in C'_n) or odd (in C''_n) distance between each other.



(b) Graph-existential argument: *outside* the input graph family. The red nodes cannot distinguish whether they are in an even or an odd cycle.

Figure 2: The no-signaling principle illustrated in the problem of 2-coloring cycles.

the node must behave in different ways. For example, the bipartitness of an even cycle is a very rigid property: if the radius-T neighborhoods of two distinct nodes in a cycle do not intersect, the nodes cannot guess if they are in the same set of the bipartition or not. More formally, assume that there is a det-LOCAL algorithm $\mathcal{A}$ that 2-colors C_n in time $T = \lceil (n-2)/4 \rceil - 1$. Consider two nodes u_1, u_2 that are at distance at least $2\lceil (n-2)/4 \rceil - 1$ between each other. Their radius-T neighborhoods do not intersect, hence there is no way to coordinate in case their distance is odd (which implies different colors) or even (which implies the same color). One can instantiate two input graphs C'_n and C''_n in which the distance between u_1 and u_2 is even and odd, respectively. However, when running $\mathcal{A}$ the output must be the same, leading to a contradiction (see Fig. 2a). This argument gives a lower bound of $\lceil (n-2)/4 \rceil - 1$ to the problem. As for rand-LOCAL, simply notice that, in the worst case, the two nodes u_1 and u_2 cannot produce the correct output with probability more than $1/2$.

Second example: graph-existential argument (*outside* the input graph family). Graph-existential arguments are another key technique to prove lower bounds in classical LOCAL. The idea is the following: Suppose we have an LCL problem that assume that the input graph family satisfies some key-property (in this case, bipartite cycles). Assume also that one can find a graph that *locally* looks like a proper input, but lies *outside* the input graph family (e.g., an odd cycle). In this way, one can exploit the fact that, if the locality T of the algorithm solving the problem is not large enough to detect that the graph is not proper, then the algorithm must run and produce some local failure (e.g., two monochromatic nodes). We can now take a copy of the radius-T neighborhood of the failing nodes and construct a proper input that contains this copy. Since the nodes do not distinguish in which input (proper or not) they are in, they must produce the same output, leading to the failure of the algorithm in a proper instance (see Fig. 2b). Formally, assume that there is a det-LOCAL algorithm $\mathcal{A}$ that 2-colors C_n in time $T = n/2 - 2$. Consider now a second graph G of n nodes that is the disjoint union of a cycle C_{n-1} with $n-1$ nodes and a single node with no edges. Run $\mathcal{A}$ in G: the nodes in C_{n-1} will not notice that they are in an odd cycle since the radius-T neighborhood of any node leaves out 2 nodes. Hence, the nodes must output some color and, since C_{n-1} is not 2-colorable, there must be a failure somewhere, that is, two adjacent nodes u_1, u_2 share the same color. We can now construct an even cycle C_n that contains a subgraph isomorphic to $C_{n-1}[\mathcal{N}_T[\{u_1, u_2\}]]$, with the same input data (that is, identifiers and port numbers). Since the isomorphic copies of u_1, u_2 in C_n cannot distinguish if they are in C_n or in C_{n-1} when running $\mathcal{A}$, they must produce the same failure, contradicting the hypothesis that $\mathcal{A}$ was correct. This argument gives a lower bound of $n/2 - 2$ for 2-coloring even cycles with n

nodes.

When we allow randomness, things are a bit more complex. Indeed, the failure in C_{n-1} is random and when we look at two specific adjacent nodes, in the worst case, the probability of a failure (i.e., a monochromatic edge) can be as small as $1/\text{poly}(n)$. However, we can do something different. Let $u_1, \ldots, u_{n-1}$ be the nodes of C_{n-1}, where $\{u_i, u_{i+1}\}$ is an edge for $i = 1, \ldots, n-2$, and $\{u_1, u_{n-1}\}$ is another edge. Consider two subgraphs G, H of C_{n-1} defined as follows: $V(G) = \{u_1, \ldots, u_{n/2}\}$, $V(H) = \{u_{n/2}, \ldots, u_{n-1}\}$. Now assume $\mathcal{A}$ is a rand-LOCAL algorithm that 2-colors even cycles of n nodes in time $T = \lceil (n-2)/4 \rceil - 1$, and run $\mathcal{A}$ in C_{n-1} (plus one disjoint singleton node to make the number of nodes equal to n). Again, since locality T is not sufficient for the nodes of C_{n-1} to understand that they are not in an even cycle, there must be a failure in at least two adjacent nodes u, v at every run of $\mathcal{A}$. Since $G \cup H = C_{n-1}$, in at least one of them there is probability no less than $1/2$ that a failure takes place. Wlog, assume G is such subgraph. Then, one can construct an even cycle of n nodes that contains as induced subgraph a copy of $\mathring{C}_{n-1}[\mathcal{N}_T[V(G)]]$ and give as input the same identifiers and port numbers. Since the view of the nodes of the copy of G in C_n is indistinguishable from the view of the nodes of G in C_{n-1}, they must reproduce the same failure probability. Hence, we get that any algorithm that 2-colors cycles in rand-LOCAL with locality at most $T = \lceil (n-2)/4 \rceil - 1$ fails with probability at least $1/2$. Notice that we could consider more subgraphs of C_n and get lower bounds with higher locality but smaller failure probability.

Randomized LOCAL: boosting the failure probability. In rand-LOCAL, we can also boost the failure probability at the cost of worsening the locality of the lower bound, by simply repeating the experiment many times. We only focus on the graph-existential argument, but a similar approach holds for the other argument as well. We assume that the overall amount of nodes is now $n = m \cdot N$ for two positive integers m, N where m is odd and N is even. Assume now that we have an algorithm $\mathcal{A}$ that 2-colors even cycles of n nodes with locality $T = \lceil (m-2)/4 \rceil - 1 = \lceil (n/N - 2)/4 \rceil - 1$. Now consider N disjoint copies $C_m^{(1)}, \ldots, C_m^{(N)}$ of an odd cycle C_m with m nodes. For each $C_m^{(i)}$, the same argument as before holds, and we can identify a subgraph G_i of each $C_m^{(i)}$ where a failure takes place with probability at least $1/2$ independently of the others, and such that the radius-T neighborhood of $V(G)$ still leaves at least one node of $C_m^{(i)}$ out. Now we can construct a proper input C_n that contains, as induced subgraphs, a copy of each $C_m^{(i)}[\mathcal{N}_T(V(G_i))]$. By independence, the failure probability here is at least $1 - \frac{1}{2^N}$. Hence, any algorithm 2-coloring even cycles with locality $T = \lceil (n/N - 2)/4 \rceil - 1$ fails with probability at least $1 - \frac{1}{2^N}$. See Fig. 3 for a visual explanation.

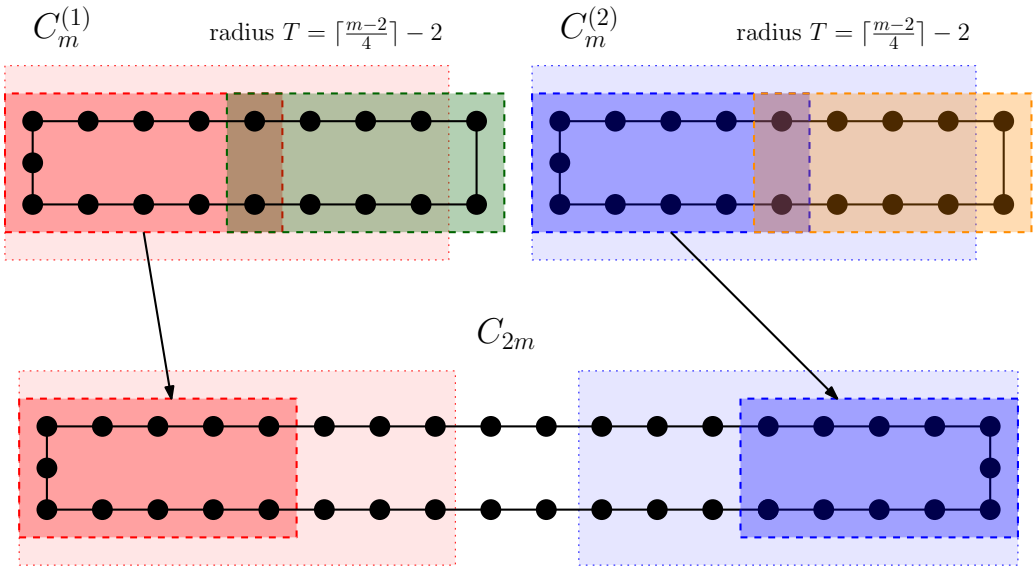


Figure 3: Visual representation of the failure probability boost: We take two odd cycles $C_m^{(1)}$ and $C_m^{(2)}$ and subdivide the nodes in two slightly overlapping regions (the colored dashed regions). For $T = \left\lceil \frac{m-2}{4} \right\rceil - 2$, a T-round algorithm $\mathcal{A}$ must fail both in $C_m^{(1)}$ and $C_m^{(2)}$ (it does not catch that we are in odd cycles). For each cycle, in at least one region $\mathcal{A}$ must fail with probability at least 1/2. Wlog, we assume that this happens in the red and the blue regions. Now we create a new cycle C_n with $n = 2m$ nodes where we copy the radius$T$ neighborhoods of the red and the blue region, and we add the remaining nodes. The failing probability of $\mathcal{A}$ over C_n is, by independence, at least 3/4, and we get a lower bound on the locality of magnitude $T = \left\lceil \frac{n-4}{8} \right\rceil - 2$.

3.2 The no-signaling principle

The lower bound techniques in Section 3.1 rely on a crucial assumption, which is quite intuitive when dealing with classical LOCAL. First, let us introduce the notion of *view* of a subset of nodes.

For any input distributed network (G, x) to any problem, and any subset of nodes $S \subseteq V(G)$, the radius-T view of S is $\mathcal{V}_T(S) = (\mathring{G}[\mathcal{N}_T[S]], \mathrm{x} \restriction_{\mathcal{N}_T[S]})$. Basically, $\mathcal{V}_T(S)$ includes everything that can be *seen* by the nodes in S with T rounds of communication, including input data (degree, ports, identifiers and input labels—if any, etc.). Suppose G has n nodes, and fix any subset of nodes $S \subseteq V(G)$. Given any two graphs G, H with inputs $\mathrm{x}_G, \mathrm{x}_H$ and any two subset of nodes $S_G \subseteq V(G)$ and $S_H \subseteq V(H)$, it is natural to define the notion of *isomorphism between views*. We say that $\mathcal{V}_T(S_G)$ is isomorphic to $\mathcal{V}_T(S_H)$ if there exists a function $\varphi : V(G) \to V(H)$ such that the folowwing holds:

1. $\varphi \restriction_{S_G}$ is an isomorphism between $G[S_G]$ and $H[S_H]$

2. $\varphi \restriction_{\mathcal{N}_T[S_G]}$ is an isomorphism between $\mathring{G}[\mathcal{N}_T[S_G]]$ and $\mathring{H}[\mathcal{N}_T[S_H]]$;

3. $\mathrm{x}_G(u) = \mathrm{x}_H(\varphi(u))$.

Consider any T-round (deterministic or randomized) LOCAL algorithm $\mathcal{A}$ run by the nodes of G. Imagine that the distributed network is split in two different laboratories, Alice's and Bob's. Alice's lab contains $\mathcal{V}_0(S)$, while Bob's lab contains everything that is not contained in $\mathcal{V}_T(S)$. When performing T rounds of communication, Alice's lab receive no information from Bob's lab. Hence, the behavior of the output distribution over nodes in Alice's lab cannot change whatever Bob does in his lab, including rearranging links between nodes or manipulating inputs. The *cause* of outputs in Alice's lab *cannot be influenced* by any action of Bob in his lab. See Fig. 4 for a visual representation of the property.[1] This property is the so-known *no-signaling from the future* principle in physics, which states that no signals can be sent from the future to the past, and is equivalent to the *causality principle* [DCP16]. Such principle holds in *every physical distributed network* running any kind of synchronous distributed algorithm, including quantum ones.

To see how this principle formally translates in our setting, let us define the notion of *outcome*, which is some kind of generalization of an algorithm. Here, we restrict to finite sets of input and output labels since we focus on LCL problems.

Definition 3.1 (Outcome)**.** Let $\Sigma_{\text{in}}, \Sigma_{\text{out}}$ be two finite sets of labels, and I a finite set of indices. An *outcome* is a mapping $\mathrm{O} : (G, \mathrm{x}) \mapsto \{(\text{out}_i : V(G) \to \Sigma_{\text{out}}, p_i)\}_{i \in I}$ that assigns to every input distributed network (G, x) (with any input labeling in : $V(G) \to \Sigma_{\text{in}}$) a distribution over output labelings $\{(\text{out}_i : V(G) \to \Sigma_{\text{out}}, p_i)\}_{i \in I}$, where p_i is the probability that out_i occurs; in particular, $0 \leq p_i \leq 1$ and $\sum_{i \in I} p_i = 1$.

This definition is easily generalizable to the case of infinite label sets, but we avoid it for the sake of simplicity. Notice that all synchronous distributed algorithms gives r outcomes: it is just the assignment of an output distribution (the one that is the result of the algorithmic procedure) to the input graph. We remark we have defined the domain set of outcomes to include *all possible graphs*. This is not restrictive: Even classical (or quantum) algorithms can run on every possible graph. One can just introduce some garbage output label so that whenever a node

[1] Formally, Bob's lab also contains edges that are in $E(G[\mathcal{N}_T[S]]) \setminus E(\mathring{G}[\mathcal{N}_T[S]])$, but does not contain the nodes in $\mathcal{N}_T^{T-1}[S]$. We can imagine that Bob sees the ports of edges that are not in $\mathcal{V}_T(S)$, but nothing else, and is constrained to assign edges to those ports. In Fig. 4 Bob would have the freedom to modify such edges in the red region as well, but we avoid representing this aspect for the sake of simplicity.

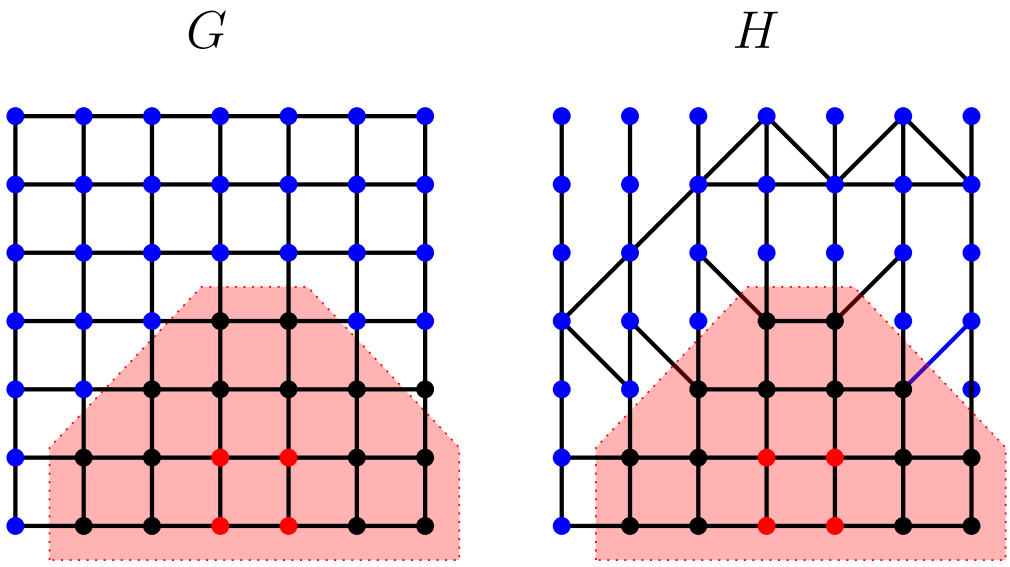


Figure 4: No-signaling property. Alice's lab contains the red nodes, Bob's lab contains the blue nodes. By running any 2-rounds synchronous distributed algorithm, the red nodes cannot distinguish between G and H: Bob has freedom to change the topology outside the red region without being detected by Alice.

running an algorithm needs to do something that is not well-defined (given its neighborhood), it can just output the garbage label.

We say that an outcome O solves a problem Π over a family of graphs $\mathcal{F}$ with probability $q > 0$ if, for every $G \in \mathcal{F}$ and every input data x, it holds that

$$\sum_{\substack{i \in I: \\ \text{out}_i \in \Pi((G,\text{x}))}} p_i \geq q.$$

Let $\text{O} : (G, \text{x}) \mapsto \{(\text{out}_i, p_i)\}_{i \in I}$ be any outcome and fix an input (G, x). Consider any subset of nodes $S \subseteq V(G)$. The restriction of the output distribution $\text{O}((G, \text{x})) = \{(\text{out}_i : V(G) \to \Sigma_{\text{out}}, p_i)\}_{i \in I}$ to S is the distribution $\{(\text{out}_j : S \to \Sigma_{\text{out}}, p'_j)\}_{j \in J}$ such that

$$p'_j = \sum_{\substack{i \in I: \\ \text{out}_j = \text{out}_i \restriction_S}} p_i,$$

and is denoted by $\text{O}((G, \text{x}))[S]$ or $\{(\text{out}_i, p_i)\}_{i \in I}[S]$. We also say that two labeling distributions $\{(\text{out}_i : V(G) \to \Sigma, p_i)\}_{i \in I}, \{(\text{out}_j : V(H) \to \Sigma, p_j)\}_{j \in J}$ over two graphs G, H are isomorphic if there is an isomorphism $\varphi : V(G) \to V(H)$ between G and H such that $\{(\text{out}_i : V(G) \to \Sigma, p_i)\}_{i \in I} = \{(\text{out}_j \circ \varphi : V(G) \to \Sigma, p_j)\}_{j \in I}$.

We are now ready to define *non-signaling outcomes*.

Definition 3.2 (Non-signaling outcome)**.** Let O be any outcome. Fix any two graphs G, H of n nodes and any two input data functions $\mathrm{x}_G, \mathrm{x}_H$. Suppose there exists a non-negative integer $T \geq 0$ with the following property: For any two subsets $S_G \subseteq V(G), S_H \subseteq V(H)$ such that $\varphi : V(G) \to V(H)$ is an isomorphism between $\mathcal{V}_T(S_G)$ and $\mathcal{V}_T(S_H)$, then the restrictions $\mathrm{O}((H, \mathrm{x}_H))[S_H]$ and $\mathrm{O}((G, \mathrm{x}_G))[S_G]$ are isomorphic under φ. We say that O is *non-signaling beyond distance* T or, alternatively, that O has locality T.

With this notion, we can define the non-signaling model.

The non-signaling model. The non-signaling model is a computational model which produces non-signaling outcomes. More specifically, the distributed network in input (G, x) is as in the definition of the deterministic LOCAL model, with $\mathrm{x}(v)$ encoding the degree of a node, port numbers, the identifier and (possibly) an input label expected by the problem of interest. The model can produce non-signaling outcomes where one wants to minimize the locality T to solve the problem. Usually, we require that the success probability of an outcome that solves a problem is at least $1 - 1/\mathrm{poly}(n)$.

The non-signaling model was first introduced by Gavoille, Kosowski, and Markiewicz [GKM09] with a slightly different definition, and then formalized by Arfaoui and Fraigniaud [AF14] in the current form. It is stronger than any *physical* synchronous distributed computing model, as T-rounds synchronous distributed algorithms (both classical and quantum) obey the no-signaling principle and must produce outcomes that are non-signaling beyond distance T. [GKM09] was the first to observe that lower bounds in the non-signaling model must hold in all weaker models, and noticed that lower bound techniques based on the indistinguishability argument *withing* the input graph family still hold in non-signaling. More specifically, [GKM09] revisited some previous lower bound results and noticed that they hold also in the non-signaling model, and it also established a new lower bound for 2-coloring even cycles, revisiting the first argument on indistinguishability shown in Section 3.1.

Theorem 3.3 (Gavoille, Kosowski, and Markiewicz [GKM09])**.** *In the non-signaling model, the following holds:*

1. *Maximal independent requires locality* $\Omega\left(\sqrt{\frac{\log n}{\log\log n}}\right)$ *[KMW04].*
2. *Locally minimal (greedy) coloring requires locality* $\Omega\left(\frac{\log n}{\log\log n}\right)$ *[Gav+07; Gav+09].*
3. *Finding a connected subgraph with* $O\left(n^{1+1/k}\right)$ *edges requires locality* $\Omega(k)$ *[Der+08; Elk07].*

4. *Finding a 2-coloring in even cycles requires locality* $\Omega\left(\left\lceil \frac{n-2}{4} \right\rceil\right)$.

We remark that [GKM09] had a slight different definition of the non-signaling model. In the paper, the model was called φ-LOCAL and defined outcomes to exist only on a specific input graph family. When considering, e.g., the problem of 2-coloring even cycles, the outcome produced by φ-LOCAL were defined *only* for even cycles, leaving out the possibility to play with graphs outside the input graph family. However, as previously mentioned, (classical or quantum) algorithms can be run also on network graphs that lie outside the input graph family: If computation is at any point undefined, we can let a node output some garbage label. This freedom that we have with algorithms is what we exploit in the second lower bound argument in Section 3.1, the graph-existential one. By defining outcomes on *every possible input graph*, we have access to new lower bound techniques. In [Coi+24], we proved that the graph-existential argument can be extended all the way up to the non-signaling model (under some general hypotheses), and used it to prove the following lower bounds.

Theorem 3.4 (Coiteux-Roy et al. [Coi+24])**.** *In the non-signaling model, the following holds:*

1. *The problem of c-coloring χ-chromatic graphs requires locality* $\Omega\left(n^{1/\left\lfloor \frac{c-1}{\chi-1} \right\rfloor}\right)$.
2. *Finding a* 3*-coloring $n \times m$ grids requires locality* $\min\{n, m\}$.
3. *Finding a c-coloring of trees requires locality* $\log_c n$.

The three results of Theorem 3.4 make use of *cheating graphs*, that is, of graphs that are locally everywhere indistinguishable from proper inputs, but globally they are not in the input family. E.g., for c-coloring χ-chromatic graphs we need to find a graph such that the radius-T neighborhood of any node induces a graph that is χ-colorable, for $T = \Theta\left(n^{1/\left\lfloor \frac{c-1}{\chi-1} \right\rfloor}\right)$, but globally the graph has chromatic number strictly greater than c. The existence of such graph immediately implies that deterministic LOCAL algorithms require locality $\Omega\left(n^{1/\left\lfloor \frac{c-1}{\chi-1} \right\rfloor}\right)$ to solve the problem. Such graph is given, for all combinations of c and χ, by Bogdanov [Bog13]: this was the first time that Bogdanov's construction found application in distributed computing and, more in general, theoretical computer science. Interestingly, in [Coi+24] the authors proved an almost matching upper bound for the problem in det-LOCAL, thus excluding any significant quantum advantage over det-LOCAL for this problem. As for 3-coloring grids, we made use of odd quadrangulations of Klein-bottles [MS02], which are everywhere locally indistinguishable from grids, but have global chromatic number, while for c-coloring trees we

revisited Linial's argument in [Lin92] that made use of Ramanujan graphs, which are high-girth, high-chromatic graphs [MST13]. When looking at rand-LOCAL and the non-signaling model, we need to make sure that the graph can be nicely covered by a small amount of subgraphs (whose union form the whole graph) that are slightly overlapping, in order to identify a subgraph where the failure probability is high enough. In general, boosting the failure probability is possible also in the non-signaling model. However, we cannot rely on the same argument of Section 3.1 as a non-signaling outcome does not guarantee *independence* between the output distributions of far-away subsets of nodes (which instead is guaranteed by rand-LOCAL algorithms). The reason is that non-signaling outcomes include the possible use of *global* resources, such as *shared randomness* or *shared quantum states*. Such resources make output distributions of distant nodes dependent of each other, even if their distance is greater than the locality of the algorithm itself. Nonetheless, boosting the failure probability is still possible provided that the cheating graph meets some properties. For more details, we defer the reader to the original article [Coi+24].

All this discussion might suggest that non-signaling argument are sufficient to exclude quantum advantage, and hence gives rise to the following question:

Question 1. *Can we exclude quantum advantage for all LCL problems using non-signaling arguments?*

Unfortunately, the answer to this question is *no*. Indeed, the non-signaling model is too strong to compare with classical LOCAL. For example, very recently Balliu et al. [Bal+24] solved a longstanding open problem, proving that shared randomness gives advantage over private randomness in classical LOCAL when restricting to LCL problems (otherwise, the thesis is trivial). More specifically, there is an LCL problem that requires $\Omega\left(\sqrt{n}\right)$ rounds in rand-LOCAL but can be solved in $O(\log n)$ rounds when nodes have access to shared randomness (e.g., an infinite random bit string). Since the non-signaling model is strong enough to simulate LOCAL with shared randomness, we already know that it is too strong with respect to classical LOCAL and there is no hope to prove something like Question 1. However, it is worth investigating to what extent lower bound arguments based on the non-signaling property apply, since we still miss a characterization.

Question 2 (Open)**.** *For which LCL problems can we exclude quantum advantage using non-signaling arguments?*

4 The bounded-dependence model

To avoid dealing with shared resources, we can *weaken* the non-signaling model by introducing further restrictions on the non-signaling outcomes. Running T-

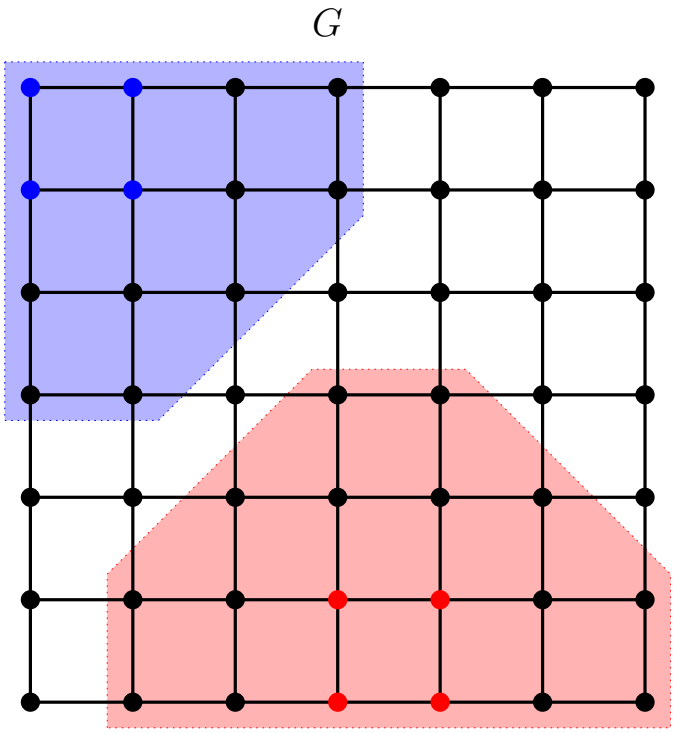


Figure 5: Bounded-dependence property. When running any 2-round synchronous distributed algorithm (which does not rely on shared resources), the output labeling distribution of the red nodes is independent of that of the blue nodes.

rounds classical or quantum-LOCAL algorithms without shared resources, we obtain the following property on the output labeling distribution: for every two subset of nodes A, B such that $\mathrm{dist}(A, B) > 2T$, then the output distributions restricted to A and B are independent (see Fig. 5). Let us formalize this notion.

Definition 4.1 (T-dependent distribution)**.** Let Σ_{out} and I be two sets, and let (G, x) be an input for some labeling problem Π. An output labeling distribution $\{(\text{out}_i : V(G) \to \Sigma_{\text{out}}, p_i)\}_{i \in I}$ is T-dependent if, for any two subsets of nodes $A, B \subseteq V(G)$, we have that $\{(\text{out}_i, p_i)\}_{i \in I}[A]$ is independent of $\{(\text{out}_i : V(G) \to \Sigma_{\text{out}}, p_i)\}_{i \in I}[B]$.

We can think now of non-signaling outcomes that produces such distributions.

Definition 4.2 (Bounded-dependent outcome)**.** Let O be any outcome that is non-signaling beyond distance T. We say that O is bounded-dependent with locality T if for any input (G, x) we have that $\mathrm{O}((G, \mathrm{x}))$ is $2T$-dependent. Furthermore, when $T = O(1)$ (i.e., it does not depend on the input graph), we say that $\mathrm{O}((G, \mathrm{x}))$ is a finitely-dependent distribution. Moreover, if for all inputs (G, x) it holds that $\mathrm{O}((G, \mathrm{x}))$ does not depend on identifiers and port numbers, we say that O is *invariant under subgraph isomorphism*.

With the addition of this further property, we can define the *bounded-dependence model*, first formalized in [Akb+25].

The bounded-dependence model. Similarly to the introduction of the non-signaling model, we can define the *bounded-dependence model* as a model that produces bounded-dependent outcomes. Again, the required success probability then solving a problem should be at least $1 - 1/\mathrm{poly}(n)$.

One might hope that we can prove lower bounds in the bounded-dependence model and matching upper bounds in classical LOCAL.

Question 3. *Can we always rule out quantum advantage for LCLs using bounded-dependent outcomes?*

Unfortunately, the answer to Question 3 is *no*: Holroyd and Liggett [HL16] and Holroyd, Hutchcroft, and Levy [HHL18] proved it for us. It is well-known that 3-coloring paths and cycles requires locality $\Theta(\log^\star n)$ both in det-LOCAL and in rand-LOCAL [Lin92; CKP16]. [HL16; HHL18] showed that these problems admit finitely-dependent distributions, that is, it is possible to 3-color path and cycles with an $O(1)$-dependent outcome that is invariant under subgraph isomorphism and under permutations of the colors.

Apart from 3-coloring paths and cycles, there are other famous problems in the complexity class $\Theta(\log^\star n)$, such as computing an MIS, $(\Delta + 1)$-coloring graphs of maximum degree Δ, etc.

All such problems are also called *symmetry-breaking* problems, in the sense that they are easily solvable by an $O(1)$-round port-numbering algorithm (and without knowledge of n) if symmetry is locally broken with a constant amount of labels. This fact implies that, given any two LCLs Π_1, Π_2 that have complexity $\Theta(\log^\star n)$ in some graph G, a solution to any of those problems can be converted into a solution of the other in a constant number of rounds. In [Akb+25], the authors proved it is possible to compose port-numbering algorithms (that do not make use of n) and bounded-dependent outcomes obtaining a bounded-dependent outcome (without significant loss in locality). Hence, the results in [HL16; HHL18] immediately imply that in paths and cycles all LCL problems that have classical complexity $\Theta(\log^\star n)$ are solvable by an $O(1)$-dependent outcome in the bounded-dependence model as well: but what about other graphs?

Question 4. *Is there any LCL problem with classical complexity* $\Theta(\log^\star n)$ *for which we can rule out quantum advantage using the bounded-dependence model?*

Unfortunately the answer is, again, *no*. Akbari et al. [Akb+25] proved that all such problems are solvable with locality $O(1)$ in the bounded-dependence model, and the resulting bounded-dependent outcomes are also invariant under subgraph isomorphism.

Theorem 4.3 (Akbari et al. [Akb+25])**.** *Let* Π *be an LCL over some input graph family* $\mathcal{F}$ *that has complexity* $O(\log^\star n)$ *in classical LOCAL. Then, there exists an* $O(1)$*-dependent outcome* O *that solves* Π *over* $\mathcal{F}$*. Furthermore,* O *is invariant under subgraph isomorphism.*

Theorem 4.3 is powerful result that shows the power of finitely-dependent distributions: such distributions are able to break symmetry with constant locality, that is something that classical LOCAL cannot do. This leaves us with one of the major open question in the field.

Question 5 (Open)**.** *Is quantum-LOCAL able to break symmetry in LCLs? That is, can quantum-LOCAL solve in time* $o(\log^\star n)$ *any LCL* Π *that has classical complexity* $\Theta(\log^\star n)$*?*

We currently lack tools to analyze directly quantum-LOCAL (especially regarding lower bounds). What we can do is instead focusing on specific graph families and/or specific complexity classes in the bounded-dependence or non-signaling model. In classical (both deterministic and randomized) LOCAL, LCL complexities belong to the following three classes: $O(1)$, $\Theta(\log^\star n)$, and $\Omega(\log\log n)$. After Theorem 4.3, we might wonder what happens in the complexity class $\Omega(\log\log n)$.

Question 6 (Open)**.** *In the bounded-dependence model and the non-signaling, what can we infer on the complexity of LCL problems that have classical complexity* $\Omega(\log\log n)$*?*

One of the most prominent problems is maybe sinkless orientation (SO), a problem that asks each node v to orient its adjacent edges so that $\mathrm{outdeg}(v) \geq 1$. It is known that SO has complexity $\Theta(\log n)$ in det-LOCAL and $\Theta(\log\log n)$ in rand-LOCAL [Bal+23b]. Hence, we can formulate the following question, which is nowadays open.

Question 7 (Open)**.** *What is the complexity of sinkless orientation in the bounded-dependence model or the non-signaling model?*

Currently, we have little insight on these questions, but we managed to give some small partial answers to Question 6, which we address in the following section.

5 The online-LOCAL model

In this section, we describe other models of computation that, at a first glance, seem completely unrelated from the (super)quantum world we have been describing so far, but after a deeper look turn out to be related and extremely useful. In [GKM17], Ghaffari, Kuhn, and Maus introduced the SLOCAL model, that is, a sequential version of the LOCAL model.

The SLOCAL model. The SLOCAL model is similar to the LOCAL model, but sequential. Here, the nodes of the input graph $G = (V, E)$ with $|V| = n$ are processed according to an adversarial order $\sigma = v_1, \ldots, v_n$. While processing a node v_i, a T-round algorithm collects all data contained in and the topology of the radius-T neighborhood of v_i (including the states and the outputs of previously processed nodes in $\mathcal{N}_T[v_i]$, i.e., any node $v_j \in \mathcal{N}_T[v_i]$ for $j < i$): We say that such an algorithm has locality/complexity/running time T. Note that the algorithm might store the whole data contained in $\mathcal{V}_T(v_i)$ inside the memory of v_i (and we always assume this happens, since it can only make the algorithm stronger). This phenomenon gives to a node v_i access to the data contained in $\mathcal{V}_T(v_j)$ if and only if there is a subsequence of nodes $\{v_{h_k}\}_{k\in[m]}$ with $j = h_k < h_{k+1} < \cdots < h_m = i$ such that $v_{h_k} \in \mathcal{N}_T[v_{h_{k+1}}]$ for all $k \in [m-1]$.

If the algorithm is given in input an infinite random bit string, we talk about the randomized SLOCAL model, as opposed to the deterministic SLOCAL model. Notice that the adversarial order in which node are processed is assumed to be *oblivious* to the random bit string. In this case, we require the success probability to be at least $1 - 1/\mathrm{poly}(n)$, with n being the number of nodes of the input graph.

Clearly, the SLOCAL model is stronger than the LOCAL model, since any deterministic T-round LOCAL algorithm can be converted into a deterministic T-round SLOCAL algorithm. Surprisingly, Ghaffari, Harris, and Kuhn [GHK18] proved that also any randomized T-round LOCAL algorithm can be converted into a deterministic $O(T)$-round SLOCAL algorithm through some derandomization technique. Interestingly, [GKM17] proved that, under certain hypotheses, (both randomized and deterministic) SLOCAL algorithms can be converted in (respectively, randomized and deterministic) LOCAL algorithms with some overhead in the complexity. However, for our purposes it is sufficient to know that $O(1)$-round SLOCAL algorithms for LCLs can be turned into $O(\log^\star n)$-round LOCAL algorithms: this is folklore, but a proof can be found in [Akb+25]. See Fig. 1 for a representation of the relations among models.

On top of the SLOCAL model, Akbari et al. [Akb+23] introduced the online-LOCAL model. The online-LOCAL model is simply the SLOCAL model equipped with global memory.

The online-LOCAL model. The (deterministic) online-LOCAL model is basically equivalent to the SLOCAL model with global memory. More specifically, the online-LOCAL model is a centralized model of computing where the algorithm initially knows only the set of nodes of the input graph G. The nodes are processed with respect to an adversarial input sequence $\sigma = v_1, v_2, \ldots, v_n$. The output of v_i depends on $G_i = \cup_{j=1}^{i} \mathcal{V}_T(v_j)$, i.e., the radius-$T$ views of of $v_1, v_2, \ldots, v_i$ (which includes all input data), plus all the outputs of previously processed nodes

(we can imagine that the views get updated at each step of the algorithm).

In [Akb+25], the authors defined the randomized online-LOCAL model as a randomized variant of the online-LOCAL model where the label assigned by the algorithm to v_i might depend on arbitrarily large portions of an infinite random bit string. Note that this model is oblivious to the randomness used by the algorithm. In particular this means that the graph outside G_i cannot be changed depending on the label assigned to v_i. One could also define the randomized online-LOCAL model in an adaptive manner, but it turns out that this is equivalent to the deterministic online-LOCAL model (as proved in [Akb+25]). The notion of complexity of a problem can be easily extended from the LOCAL model. If the algorithm is randomized, we also require that the failure probability is at most $1/\mathrm{poly}(n)$, where n is the size of the input graph.

Interestingly, [Akb+23] proved that in rooted regular trees LCLs have roughly the same complexity across the three deterministic models LOCAL, SLOCAL, and online-LOCAL.

So, why should we care about the online-LOCAL model? Akbari et al. [Akb+25] found a very important connection with the non-signaling model, which we report here with the following theorem: in [Akb+25] the result is stated only for LCL problems, but the proof holds also for any labeling problem.

Theorem 5.1 (Akbari et al. [Akb+25])**.** *Let* Π *be any labeling problem and* O *any non-signaling outcome with locality* T *solving* Π *with probability* $p > 0$*. Then, there is a randomized online-LOCAL algorithm* $\mathcal{A}$ *with locality* T *that solves* Π *with probability* p*. Furthermore, the output distribution of* $\mathcal{A}$ *over any input* (G, x) *is exactly* $\mathrm{O}((G, \mathrm{x}))$*.*

Theorem 5.1 is very powerful, as it allows us to focus on classical, centralized models of computing: every lower bound in randomized online-LOCAL holds also in non-signaling and, hence, in quantum-LOCAL. Surprisingly, [Akb+25] proved even more results regarding the online-LOCAL model that connects it with the classical LOCAL model.

In order to introduce it, we need to define *component-wise* online-LOCAL algorithms.

The component-wise online-LOCAL model. The component-wise online-LOCAL model is exactly the same as the deterministic online-LOCAL model but when the algorithm processes a node v_i according to the adversarial order $\sigma = v_1, \ldots, v_n$, v_i does not have access to the whole G_i. Rather, it has access only to its connected component in G_i. Clearly a component-wise online-LOCAL algorithm is also a standard online-LOCAL algorithm, hence the model is weaker than the deterministic online-LOCAL model (see Fig. 1 for a landscape of all the computational models). However, we have the following result.

Theorem 5.2 (Akbari et al. [Akb+25])**.** *Let* Π *be any LCL problem, and let* $\mathcal{A}$ *be any online-LOCAL algorithm solving* Π *with locality* $T(n)$ *for graphs of* n *nodes. Then the following holds:*

1. *If* $\mathcal{A}$ *is deterministic, then there exists a deterministic component-wise online-LOCAL algorithm solving* Π *with locality* $T(2^{O(n^3)})$.
2. *If* $\mathcal{A}$ *is randomized and has success probability* $p(n) > 0$*, then there exists a deterministic component-wise online-LOCAL algorithm solving* Π *with locality* $T\left(2^{O(n^3)} + 2^{O(2^{n^2})} \cdot \log \frac{1}{p(n)}\right)$.

5.1 Implications in rooted trees

Why is Theorem 5.2 so meaningful? The difference between online-LOCAL and SLOCAL is *local* vs *global memory*. Component-wise online-LOCAL algorithms lie somewhere in the middle: a node gets access only to the memory contained in the currently explored connected component. Interestingly, in some topologies, the SLOCAL model is able to "simulate" component-wise online-LOCAL algorithms. We remind the reader that a rooted tree is a directed tree where all nodes have outdegree 1 except for a single node v that has outdegree 0 and is called the *root* of the tree.

Theorem 5.3 (Akbari et al. [Akb+25])**.** *Let* Π *be any LCL problem over rooted trees. Assume* $\mathcal{A}$ *is a component-wise online-LOCAL algorithm solving* Π *with locality* $T(n)$*. Then, there exists a deterministic SLOCAL algorithm solving* Π *with locality* $O(1) + T(O(n))$.

Now we can go from (deterministic or randomized) online-LOCAL to SLOCAL in rooted trees. Interestingly, [Akb+25] also proved that LCLs over rooted trees in SLOCAL have complexity either $O(1)$ or $\Omega(\log n)$.

Theorem 5.4 (Akbari et al. [Akb+25])**.** *Let* Π *be any LCL problem over rooted trees. Assume* $\mathcal{A}$ *is an SLOCAL algorithm solving* Π *with locality* $o(\log n)$*. Then, there exists another SLOCAL algorithm* $\mathcal{B}$ *solving* Π *with locality* $O(1)$.

It is folklore that LCLs that have complexity $O(1)$ in SLOCAL over any topology translate in complexity $O(\log^\star n)$ in classical LOCAL. Altogether, we have the following.

Corollary 5.5 (Akbari et al. [Akb+25])**.** *Let* Π *be any LCL problem over rooted trees. Then the following holds:*

1. *If* Π *has complexity* $o(\log \log n)$ *in deterministic online-LOCAL, then it has complexity* $O(\log^\star n)$ *in LOCAL.*

2. *If* Π *has complexity* $o(\log\log\log n)$ *in randomized online-LOCAL, then it has complexity* $O(\log^\star n)$ *in LOCAL.*

See also Fig. 1 for a drawing of all the implications among models. Corollary 5.5 is very powerful in a twofold sense. On one hand, it implies that the LCL complexity class $O(\log^\star n)$ in quantum-LOCAL and in LOCAL coincide over rooted trees, excluding significant quantum advantage (it might still be that $O(1)$ locality in quantum-LOCAL becomes $\Theta(\log^\star n)$ in LOCAL). Also, the LCL complexity class $O(1)$ in the bounded-dependence and non-signaling models becomes $O(\log^\star n)$ in LOCAL over rooted trees, while we know that $O(\log^\star n)$ in LOCAL becomes $O(1)$ in the bounded-dependence and non-signaling models over any topology by Theorem 4.3. On the other hand, Corollary 5.5 allows us to obtain lower bounds in online-LOCAL (and, hence, quantum-LOCAL, bounded-dependence, and non-signaling) through lower bounds in LOCAL: We know that if an LCL over rooted trees cannot be solved in time $O(\log^\star n)$ in LOCAL, then it needs locality $\Omega(\log\log n)$ in deterministic online-LOCAL and $\Omega(\log\log\log n)$ in randomized online-LOCAL. Hence, in rooted trees we also know that the LCL complexity class $\omega(\log^\star n)$ in LOCAL becomes $\Omega(\log\log\log n)$ in quantum-LOCAL, bounded-dependence, and non-signaling.

5.2 LCL complexity landscape in general trees

Recently, Dhar et al. [Dha+24] analyzed more carefully the relation between the randomized online-LOCAL model and the det-LOCAL model in various kind of trees: rooted, unrooted, regular, etc. They proved results for the LCL complexity class $\omega(\log n)$ by extending previous results all the way up to the randomized online-LOCAL model [Akb+23; Bal+22a; Bal+23a; Bal+21a; Cha20; CP19; GRB22]. By these previous works, it was known that in the LOCAL model, LCL problems over regular trees that have complexity $\omega(\log n)$ fall into one of the following classes: $\Theta\left(n^{1/k}\right)$ for some $k \in \mathbb{N}_+$. Furthermore, if the tree is also rooted, we know that the only possible complexities in det-LOCAL are $O(1)$, $\Theta(\log^\star n)$, and $\Theta(\log n)$. In [Akb+23] this result was extended all the way up to deterministic online-LOCAL (but only for the case of rooted trees, and did not find an *exact* correspondence between LOCAL and online-LOCAL). The authors of [Dha+24] proved the following two theorems.

Theorem 5.6 (Dhar et al. [Dha+24])**.** *Let* Π *be an LCL problem on unrooted regular trees. If* Π *has complexity* $\Theta\left(n^{1/k}\right)$ *for any* $k \in \mathbb{N}_+$ *in det-LOCAL, then it has complexity* $\Theta\left(n^{1/k}\right)$ *in randomized online-LOCAL, and vice-versa.*

Theorem 5.7 (Dhar et al. [Dha+24])**.** *Let* Π *be an LCL problem on rooted regular trees. The following holds:*

1. *If* Π *has complexity* $\Theta\left(n^{1/k}\right)$ *for any* $k \in \mathbb{N}_+$ *in det-LOCAL, then it has complexity* $\Theta\left(n^{1/k}\right)$ *in randomized online-LOCAL, and vice-versa.*

2. *If* Π *has complexity* $\Theta(\log n)$ *in det-LOCAL, then it has complexity* $\Theta(\log n)$ *in randomized online-LOCAL, and vice-versa.*

Combining Theorem 5.7 with Corollary 5.5, we obtain that in the LCL complexity region $O(\log n)$ over rooted regular trees, the following classes contain the same problems in LOCAL and randomized online-LOCAL, and are the only possible complexity classes: $O(\log^\star n)$ in det-LOCAL and $O(1)$ in randomized online-LOCAL, or $\Theta(\log n)$ both in det-LOCAL and randomized online-LOCAL. We remind the reader that all these equivalences between complexity classes also hold between LOCAL and quantum-LOCAL, as well as the bounded-dependence and the non-signaling models, as they are "sandwiched" between det-LOCAL and randomized online-LOCAL. [Dha+24] also provided a general result on trees extending speedup arguments in [Bal+21a].

Theorem 5.8 (Dhar et al. [Dha+24])**.** *Let* Π *be an LCL problem on general trees. Then, either* Π *has complexity* $\Theta(n)$ *in the (deterministic or randomized) LOCAL model and in the (deterministic or randomized, respectively) online-LOCAL model, or the complexity in both models is* $O\left(\sqrt{n}\right)$.

Theorem 5.8 trivially holds for any intermediate model instead of randomized online-LOCAL. We invite the reader to have a look at Fig. 6 for a representation of the LCL complexity landscape in trees after the results of [Akb+25; Dha+24].[2] We conclude the section observing that, perhaps, the most difficult complexity range is that between $\omega(\log^\star n)$ (hence, by known results, $\Omega(\log\log n)$) and $O(\log n)$ in general (regular or non-regular) trees. The reason is that we don't have techniques that we can refer to which we can extend to these (super)quantum models.

Question 8 (Open)**.** *Let* Π *be an LCL problem over (regular or non-regular) trees that has classical complexity between* $\Omega(\log\log n)$ *and* $O(\log n)$*. Can we find non-trivial upper or lower bounds on the complexity of* Π *in randomized online-LOCAL?*

Again, sinkless orientation is an example of such problem. The lower bound of sinkless orientation (and often of other LCLs with similar complexity) is proved in classical LOCAL via a famous lower bound technique known as round elimination [Bra+16; Bra19]. As we argue in the next section, round elimination is

[2] We reproduced Figure 2 in [Dha+24], for which we give credits to the authors.

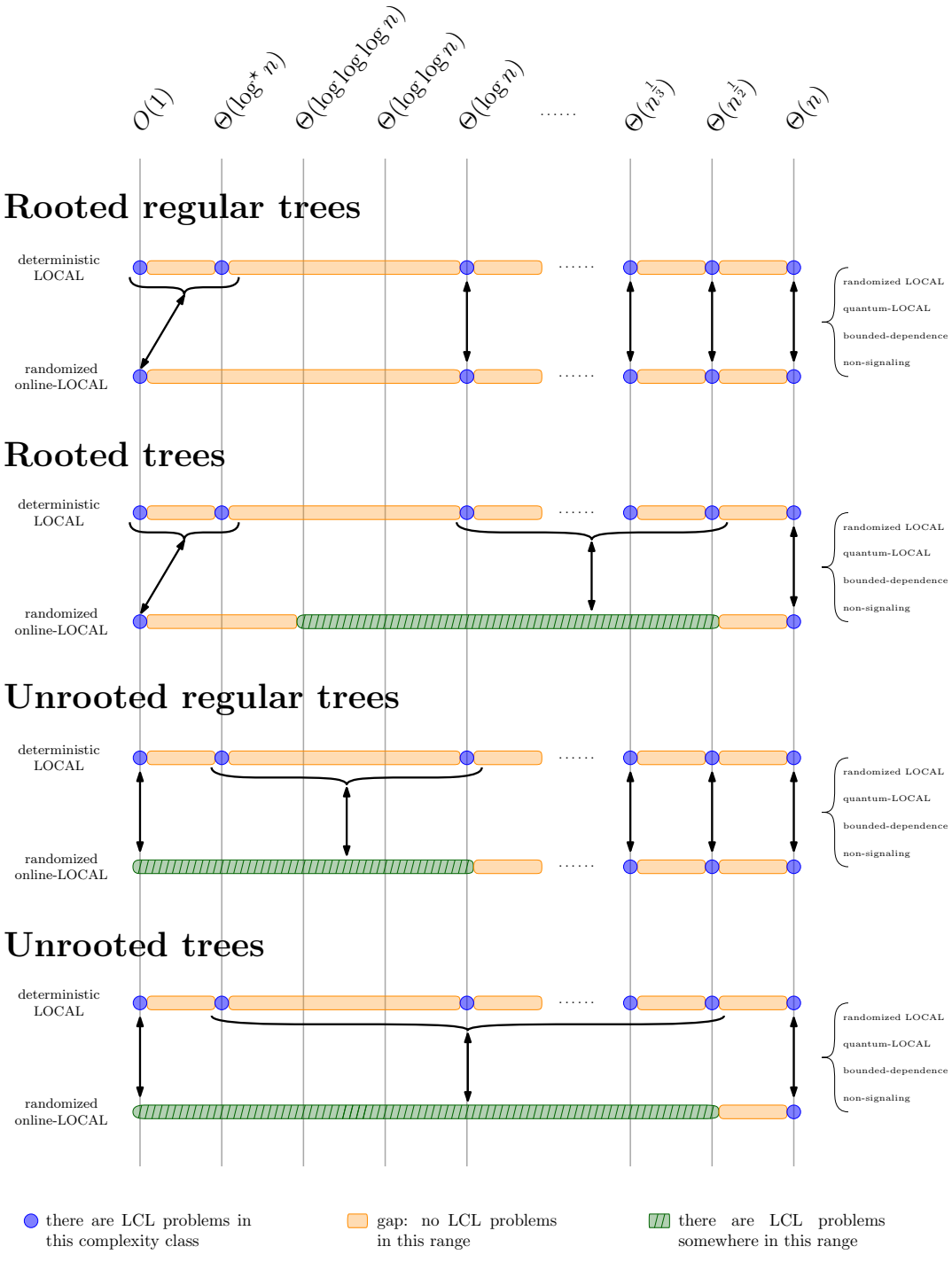


Figure 6: Landscape of LCL complexities in trees.

unfortunately not generalizable to (super)quantum models. Other candidate problems are: 3-coloring trees, 2-2-3 in 3-regular trees (that is, 2-coloring trees so that each node of any color x has at least 2 neighbors colored with a color that is different from x), Δ-coloring trees of maximum degree Δ, etc.

6 Quantum advantage for a local problem

What do we know about quantum advantage in the LOCAL model? Before last year, there was only one example of quantum advantage: Le Gall, Nishimura, and Rosmanis [LNR19] showed that there exists a problem that requires $\Omega(n)$ communication rounds in input graphs with n nodes in classical LOCAL, but can be solved in $O(1)$ rounds in quantum-LOCAL. This problem, however, has an inherent global and artificial definition, and the winning condition depends on the joint output of nodes that are at distances $\Omega(n)$ between one another, which makes it very far from problems that are usually interesting for the distributed computing community, especially from LCLs. Last year, Balliu et al. [Bal+25] exhibited the first *local problem* Π that admits quantum advantage in the LOCAL model. The authors proved that Π is solvable in $O(1)$ rounds in quantum-LOCAL, but requires $\Theta(\Delta)$ rounds in classical LOCAL, where Δ is the degree of the input graph. Before describing the problem in details, let us remark that Π is locally checkable in all senses except that the maximum degree of the input graph is not bounded, hence it is not an LCL in the strict sense of Definition 2.2. Indeed, in case $\Delta = O(1)$, then there would be no asymptotic difference between the complexities in LOCAL and quantum-LOCAL. Let us now introduce the problem in multiple steps: first we introduce the GHZ game between three players, then we use it to build Π.

6.1 The GHZ game

The GHZ game is a game between three players that works as follows: Alice, Bob, and Charlie all receive by an adversary one input bit. The input (x, y, z) is drawn from the set $\{(0, 0, 0), (0, 1, 1), (1, 0, 1), (1, 1, 0)\}$, that is, the promise is that $x \oplus y \oplus z = 0$ ($\oplus$ is the notation for the XOR logical operator). The players must produce one output bit each, resulting in a tuple (a, b, c) such that $a \oplus b \oplus c = 0$ if and only if $(x, y, z) = (0, 0, 0)$ (see Table 1). Now, the game allows players to agree on a strategy *before* receiving the input (x, y, z), but *after* it they cannot communicate anymore. The best classical strategy for the three players to win this game is to deterministically output a tuple that wins in case $(x, y, z) \neq (0, 0, 0)$. However, in the quantum world there is a strategy that *always* wins the game. An uninterested reader might just skip the rest of this subsection, as the only notion that is useful for the rest of this article is that there exists a quantum strategy

that always wins the game. Otherwise, we assume the reader is familiar with basic notions of quantum computing: if not, we defer the reader to introductory surveys like, e.g., [BS98]. The players can share a tripartite entangled state $|\psi\rangle = \frac{1}{\sqrt{2}}(|000\rangle + |111\rangle)$, known as the GHZ state. If a player receives 0 as input, it makes a measurement of the entangled state in the basis $\{|+\rangle, |-\rangle\}$. Otherwise, it makes a measurement in the basis $\{\frac{1}{\sqrt{2}}(|0\rangle + i|1\rangle), \frac{1}{\sqrt{2}}(|0\rangle - i|1\rangle)\}$. In both cases, the player outputs 0 if the result of the measurement is the first state, and 1 if it is the second state [BBT05].

Alice's (x) Input (x)	**Bob's (y) Input (y)**	**Charlie's (z) Input (z)**	**Winning Condition (Outputs)**
0	0	0	$a \oplus b \oplus c = 0$
0	1	1	$a \oplus b \oplus c = 1$
1	0	1	$a \oplus b \oplus c = 1$
1	1	0	$a \oplus b \oplus c = 1$

Table 1: GHZ game definition: the tuple (x, y, z) is the input and the output (a, b, c) must satisfy the winning condition.

6.2 Iterated GHZ games

The GHZ game usually comes with a promise that $x \oplus y \oplus z = 0$. In order to define our labeling problem, we need to remove this promise. We do so by relaxing the definition of the game: when $x \oplus y \oplus z = 1$ we allow any combination of outputs from Alice, Bob, and Charlie: clearly,this modification cannot make the problem harder. Now, our input is a bipartite graph where the two sets of the bipartition are the set W of *white nodes* and the set B of *black nodes*. White nodes are the real players of the games, while black nodes represent the games that the players are playing. More specifically, we assume that each white node has degree Δ and each black node has degree 3. We also assume that black nodes are colored with colors from $[\Delta]$, and that each white node has a unique neighbor colored with color c for all $c \in [\Delta]$. The colors of the black nodes specify the order according to which a white node must play the games. Every white node v must output a vector $[v_1, \ldots, v_\Delta]$ of Δ bits, where $v_i \in \{0, 1\}$ is its output to the game represented by its neighboring black node of color i. Let v be any black node colored with color $c \geq 1$, and s, t, u its three neighboring white nodes. The problem is defined as follows:

1. If $c = 1$, then exactly one among s_1, t_1, and u_1 must be equal to 1, and the others must be equal to 0.

2. If $c > 1$, then:

   (a) If $s_{c-1} \oplus t_{c-1} \oplus u_{c-1} = 0$, then s_c, t_c, and u_c must solve the GHZ game with input $(s_{c-1}, t_{c-1}, u_{c-1})$, that is, $s_c \oplus t_c \oplus u_c = 0$ if $s_{c-1}, t_{c-1}, u_{c-1} = (0, 0, 0)$, and $s_c \oplus t_c \oplus u_c = 1$ otherwise.

   (b) If $s_{c-1} \oplus t_{c-1} \oplus u_{c-1} = 1$, then s_c, t_c, and u_c can be arbitrarily chosen.

Clearly, in 1 communication round the problem can be solved in quantum-LOCAL as follows: The black nodes of color $c = 1$ inform their three neighbors s, t, and u on how to set s_1, t_1, and u_1 so that exactly one of these outputs is equal to 1. Black nodes of color $c > 1$ prepare 3 entangled GHZ states as described in Section 6.1 and send them to their three white neighbors, along with their colors. The white node will now have $\Delta - 1$ entangled states $|\psi_2\rangle, \ldots, |\psi_\Delta\rangle$, where $|\psi_c\rangle$ comes from the black neighbor of color c. Then, it can locally measure the entangled states, in order, setting all correct outputs without communicating further, as output number i depends only on $|\psi_i\rangle$ and output number $i - 1$.

In the classical setting, a trivial strategy would be the following: In round 1, black nodes of color 1 inform their white neighbors about which one should set its output to 1, while the others set their output to 0. In round 2, the white nodes inform their neighbors of color 2 about their output that has been set in the previous round. In this way, the three white neighbors s, t, and u of a black node v of color 2 have specified the inputs s_1, t_1, and u_1 for another GHZ game. Now v can locally solve the GHZ game with the new inputs and inform s, t, and u about their new outputs s_2, t_2, and u_2. Iterating this procedure, we have a solution to the problem in 2Δ rounds.

Interestingly, Balliu et al. [Bal+25] proved the following result.

Theorem 6.1 (Balliu et al. [Bal+25])**.** *The iterated GHZ problem can be solved in* 1 *round in quantum-LOCAL but requires* $\Omega(\min\{\Delta, \log_\Delta \log n\})$ *rounds in classical (deterministic or randomized) LOCAL.*

The classical lower bound in Theorem 6.1 is obtained through *round elimination* (RE), one of the most prominent lower bound techniques. Round elimination was formalized by Brandt [Bra19] but already used in a less general form to get an $\Omega(\log^\star n)$ lower bound for 3-coloring cycles by Linial [Lin92]. Nowadays, we even have an automated software that guides us into the round elimination procedure (i.e., REtor [Oli19]). Round elimination works as follows: Suppose an LCL problem Π has some complexity T, that is, there exists a T-round LOCAL algorithm $\mathcal{A}$ that solves Π, and T is the minimum integer with such property. Imagine

running $\mathcal{A}$ for $T-1$ rounds on a graph G. Nodes of G now will contain enough knowledge to be able to solve Π with just one more round of communication. Hence, one can describe exactly what this knowledge is and come up with the most general LCL problem Π_1 nodes can solve in $T-1$ rounds with $\mathcal{A}$. Now suppose we iterate this procedure for T rounds: We end up with an LCL Π_T that is solvable in 0 rounds of communication. However, T is usually unknown and the object of our investigation. Hence, we can guess the magnitude of T (say, T') and analyze $\Pi_{T'}$. If the description of $\Pi_{T'}$ is simple enough, it might be incredibly easier to understand if $\Pi_{T'}$ is solvable in 0 rounds: if not, then we get a lower bound T' on the complexity of Π. The usual challenge while performing round elimination is that the description of Π_i grows exponentially fast at each iteration, and thus it is fundamental to find *relaxations* of the problem that make Π_i easier to solve but hopefully with a simpler description. Note that while relaxing the description of a problem, one might exaggerate and end up with a problem that is just trivial to solve thwarting all the efforts to understand the complexity of the original problem. The whole procedure specific to the iterated GHZ problem can be found in [Bal+25].

We remark that round elimination cannot be used in the quantum world: apart from our result (which separates quantum-LOCAL and classical LOCAL), the *no-cloning* principle (that states that quantum states cannot be cloned [DCP16]) forbids us any kind of generalization of RE.

6.3 Networks of non-signaling games

After we established quantum advantage via Theorem 6.1, one may wonder whether with more refined games we could achieve even a stronger separation. In this section we show that, unfortunately, *this is not possible*.

Definition 6.2 (Game)**.** Let Σ be a finite set, and let $m \in \mathbb{N}_+$ be the number of players. We call $\mathfrak{g} \subseteq \Sigma^m \times \Sigma^m$ a *game*. Each player i receives one input $x_i \in \Sigma$ and produces one output $y_i \in \Sigma$. A *move* $\mu = (x, y) \in \Sigma^m \times \Sigma^m$ is valid if $\mu \in \mathfrak{g}$. We overload the notation so that $\mathfrak{g}(x) = \{y \in \Sigma^m \mid (x, y) \in \mathfrak{g}\}$. We say $\mathfrak{g}$ is *solvable* if, for every x, $\mathfrak{g}(x)$ is non-empty.

Non-signaling games. Let $\mathfrak{g}$ be any game of m players. We can define the following labeling problem Π on any graph $G = (V, E)$ with $|V| \geq m$. In input, a subset $P \subseteq V$ with $|P| = m$ is chosen to represent the set of players. Those nodes receive an input bit and must output another bit so that their joint output solves $\mathfrak{g}$. All other nodes can output any bit. We say that $\mathfrak{g}$ is non-signaling if there exists a non-signaling outcome that solves Π with locality 0. Clearly, Π is not locally checkable, but if we add the further constraint that the diameter of $G[P]$ is

bounded by a constant, then we can make it locally checkable (even if the degree of the graph might be unbounded).

Network of non-signaling games. We are given a bipartite graph where the two sets of the bipartition are the set W of *white nodes* and the set B of *black nodes*. White nodes are the real players of the games, while black nodes represent the games that the players are playing. We assume that each white node has degree Δ and each black node has degree m. Each black node represents a non-signaling game that its white neighbors must play. Also, white nodes can contain arbitrarily complex arithmetic circuits according to which they need to play the games: in a sense, the input of a game might depend arbitrarily complex arithmetic operations on the outputs of previous games. Such a network is called *network of non-signaling games*: it includes networks that one can build with quantum games such as the network in the iterated GHZ problem, and is locally checkable (possibly with unbounded degree).

[Bal+25] proved the following theorem.

Theorem 6.3 (Balliu et al. [Bal+25])**.** *Let $\Delta \in \mathbb{N}_+$ be a constant. For any network of non-signaling games of maximum degree Δ, there exists a classical LOCAL algorithm solving the games in time $O(1)$, where the constant might depend on Δ.*

The key ingredient for Theorem 6.3 is that any non-signaling game is completable: for example, for a 2-player game this means that for any Alice's input x there is some Alice's output a such that for any Bob's input y there is still a valid Bob's output b, and vice versa. We can exploit completability to solve any network of non-signaling games in a distributed manner: Each white node starts to process its own arithmetic circuit in a topological order. As soon as it encounters a step that involves a game, it sends a message to the black neighbor responsible for that specific game, together with its own input for that game. The black nodes keep track of the inputs they have seen so far, and they always pick safe outputs for those players. In this way, in two rounds of communication all white nodes can learn their own output for the game that appears first in their own circuit. We can then repeat this for each game in a sequential order—thanks to completability, while black nodes will be always able to assign valid outputs also for players that join the game late. The running time of this algorithm is proportional to the size of the circuit held by a single white node: for a fixed LCL (with a finite set of possible local circuits) it will be bounded by some constant. With this kind of arguments all we can hope for is a separation that is a function $f(\Delta)$ of the degree of the graph, but we still do not have an LCL problem (in the strict sense) separating LOCAL and quantum-LOCAL by a function $f(n)$ of the number of nodes of the graph n. We conclude this brief survey with the major open question in the field.

Question 9 (Open)**.** *Is there any LCL problem that quantum-LOCAL can solve asymptotically faster (as a function of the number of nodes) than classical LOCAL?*

References

[Aar22] Scott Aaronson. "How Much Structure Is Needed for Huge Quantum Speedups?" In: *CoRR* abs/2209.06930 (2022). DOI: 10.48550/ARXIV.2209.06930.

[Akb+25] Amirreza Akbari, Xavier Coiteux-Roy, Francesco D'Amore, François Le Gall, Henrik Lievonen, Darya Melnyk, Augusto Modanese, Shreyas Pai, Marc-Olivier Renou, Václav Rozhon, and Jukka Suomela. "Online Locality Meets Distributed Quantum Computing". In: *Proceedings of the 57th Annual ACM Symposium on Theory of Computing, STOC 2025, Prague, CZ, Czech Republic, June 23-27, 2025*. ACM, 2025.

[Akb+23] Amirreza Akbari, Navid Eslami, Henrik Lievonen, Darya Melnyk, Joona Särkijärvi, and Jukka Suomela. "Locality in Online, Dynamic, Sequential, and Distributed Graph Algorithms". In: *50th International Colloquium on Automata, Languages, and Programming, ICALP 2023, July 10-14, 2023, Paderborn, Germany*. Ed. by Kousha Etessami, Uriel Feige, and Gabriele Puppis. Vol. 261. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023, 10:1–10:20. DOI: 10.4230/LIPICS.ICALP.2023.10.

[AV22] Joran van Apeldoorn and Tijn de Vos. "A Framework for Distributed Quantum Queries in the CONGEST Model". In: *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*. Ed. by Alessia Milani and Philipp Woelfel. ACM, 2022, pp. 109–119. DOI: 10.1145/3519270.3538413.

[AF14] Heger Arfaoui and Pierre Fraigniaud. "What can be computed without communications?" In: *SIGACT News* 45.3 (2014), pp. 82–104. DOI: 10.1145/2670418.2670440.

[Bal+19a] Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Mikaël Rabie, and Jukka Suomela. "The Distributed Complexity of Locally Checkable Problems on Paths is Decidable". In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*. Ed. by Peter Robinson and Faith Ellen. ACM, 2019, pp. 262–271. DOI: 10.1145/3293611.3331606.

[Bal+22a] Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Jan Studený, and Jukka Suomela. "Efficient Classification of Locally Checkable Problems in Regular Trees". In: *36th International Symposium on Distributed Computing, DISC 2022, October 25-27, 2022, Augusta, Georgia, USA*. Ed. by Christian Scheideler. Vol. 246. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 8:1–8:19. DOI: 10.4230/LIPICS.DISC.2022.8.

[Bal+23a] Alkida Balliu, Sebastian Brandt, Yi-Jun Chang, Dennis Olivetti, Jan Studený, Jukka Suomela, and Aleksandr Tereshchenko. "Locally checkable problems in rooted trees". In: *Distributed Computing* 36.3 (2023), pp. 277–311. DOI: 10.1007/S00446-022-00435-9.

[Bal+25] Alkida Balliu, Sebastian Brandt, Xavier Coiteux-Roy, Francesco D'Amore, Massimo Equi, François Le Gall, Henrik Lievonen, Augusto Modanese, Dennis Olivetti, Marc-Olivier Renou, Jukka Suomela, Lucas Tendick, and Isadora Veeren. "Distributed Quantum Advantage for Local Problems". In: (2025).

[Bal+19b] 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 2019, Baltimore, Maryland, USA, November 9-12, 2019*. Ed. by David Zuckerman. IEEE Computer Society, 2019, pp. 481–497. DOI: 10.1109/FOCS.2019.00037.

[Bal+20] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. "How much does randomness help with locally checkable problems?" In: *PODC '20: ACM Symposium on Principles of Distributed Computing, Virtual Event, Italy, August 3-7, 2020*. Ed. by Yuval Emek and Christian Cachin. ACM, 2020, pp. 299–308. DOI: 10.1145/3382734.3405715.

[Bal+21a] Alkida Balliu, Sebastian Brandt, Dennis Olivetti, and Jukka Suomela. "Almost global problems in the LOCAL model". In: *Distributed Computing* 34.4 (2021), pp. 259–281. DOI: 10.1007/S00446-020-00375-2.

[Bal+21b] Alkida Balliu, Keren Censor-Hillel, Yannic Maus, Dennis Olivetti, and Jukka Suomela. "Locally Checkable Labelings with Small Messages". In: *35th International Symposium on Distributed Computing, DISC 2021, October 4-8, 2021, Freiburg, Germany (Virtual Conference)*. Ed. by Seth Gilbert. Vol. 209. LIPIcs. Schloss Dagstuhl

- Leibniz-Zentrum für Informatik, 2021, 8:1–8:18. DOI: 10.4230/LIPICS.DISC.2021.8.

[Bal+24] Alkida Balliu, Mohsen Ghaffari, Fabian Kuhn, Augusto Modanese, Dennis Olivetti, Mikaël Rabie, Jukka Suomela, and Jara Uitto. "Shared Randomness Helps with Local Distributed Problems". In: *CoRR* abs/2407.05445 (2024). DOI: 10.48550/ARXIV.2407.05445.

[Bal+18] Alkida Balliu, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Dennis Olivetti, and Jukka Suomela. "New classes of distributed time complexity". In: *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2018, Los Angeles, CA, USA, June 25-29, 2018*. Ed. by Ilias Diakonikolas, David Kempe, and Monika Henzinger. ACM, 2018, pp. 1307–1318. DOI: 10.1145/3188745.3188860.

[Bal+22b] Alkida Balliu, Juho Hirvonen, Darya Melnyk, Dennis Olivetti, Joel Rybicki, and Jukka Suomela. "Local Mending". In: *Structural Information and Communication Complexity - 29th International Colloquium, SIROCCO 2022, Paderborn, Germany, June 27-29, 2022, Proceedings*. Ed. by Merav Parter. Vol. 13298. Lecture Notes in Computer Science. Springer, 2022, pp. 1–20. DOI: 10.1007/978-3-031-09993-9_1.

[Bal+23b] Alkida Balliu, Janne H. Korhonen, Fabian Kuhn, Henrik Lievonen, Dennis Olivetti, Shreyas Pai, Ami Paz, Joel Rybicki, Stefan Schmid, Jan Studený, Jukka Suomela, and Jara Uitto. "Sinkless Orientation Made Simple". In: *2023 Symposium on Simplicity in Algorithms, SOSA 2023, Florence, Italy, January 23-25, 2023*. Ed. by Telikepalli Kavitha and Kurt Mehlhorn. SIAM, 2023, pp. 175–191. DOI: 10.1137/1.9781611977585.CH17.

[BS98] Charles H. Bennett and Peter W. Shor. "Quantum Information Theory". In: *IEEE Trans. Inf. Theory* 44.6 (1998), pp. 2724–2742. DOI: 10.1109/18.720553.

[Bog13] Ilya I. Bogdanov. "Examples of topologically highly chromatic graphs with locally small chromatic number". In: *CoRR* abs/1311.2844 (2013). DOI: 10.48550/arXiv.1311.2844. arXiv: 1311.2844.

[Bra19] Sebastian Brandt. "An Automatic Speedup Theorem for Distributed Problems". In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*. Ed. by Peter Robinson and Faith Ellen. ACM, 2019, pp. 379–388. DOI: 10.1145/3293611.3331611.

[Bra+16] Sebastian Brandt, Orr Fischer, Juho Hirvonen, Barbara Keller, Tuomo Lempiäinen, Joel Rybicki, Jukka Suomela, and Jara Uitto. "A lower bound for the distributed Lovász local lemma". In: *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*. Ed. by Daniel Wichs and Yishay Mansour. ACM, 2016, pp. 479–488. DOI: 10.1145/2897518.2897570.

[Bra+17] Sebastian Brandt, Juho Hirvonen, Janne H. Korhonen, Tuomo Lempiäinen, Patric R. J. Östergård, Christopher Purcell, Joel Rybicki, Jukka Suomela, and Przemyslaw Uznanski. "LCL Problems on Grids". In: *Proceedings of the ACM Symposium on Principles of Distributed Computing, PODC 2017, Washington, DC, USA, July 25-27, 2017*. Ed. by Elad Michael Schiller and Alexander A. Schwarzmann. ACM, 2017, pp. 101–110. DOI: 10.1145/3087801.3087833.

[BBT05] Gilles Brassard, Anne Broadbent, and Alain Tapp. "Quantum Pseudo-Telepathy". In: *Foundations of Physics* 35 (2005), pp. 1877–1907. DOI: 10.1007/s10701-005-7353-4.

[Cen+22] Keren Censor-Hillel, Orr Fischer, François Le Gall, Dean Leitersdorf, and Rotem Oshman. "Quantum Distributed Algorithms for Detection of Cliques". In: *13th Innovations in Theoretical Computer Science Conference, ITCS 2022, January 31 - February 3, 2022, Berkeley, CA, USA*. Ed. by Mark Braverman. Vol. 215. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2022, 35:1–35:25. DOI: 10.4230/LIPICS.ITCS.2022.35.

[Cha20] Yi-Jun Chang. "The Complexity Landscape of Distributed Locally Checkable Problems on Trees". In: *34th International Symposium on Distributed Computing, DISC 2020, October 12-16, 2020, Virtual Conference*. Ed. by Hagit Attiya. Vol. 179. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 18:1–18:17. DOI: 10.4230/LIPIcs.DISC.2020.18.

[CKP16] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. "An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model". In: *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA*. Ed. by Irit Dinur. IEEE Computer Society, 2016, pp. 615–624. DOI: 10.1109/FOCS.2016.72.

[CKP19] Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. “An Exponential Separation between Randomized and Deterministic Complexity in the LOCAL Model”. In: *SIAM Journal on Computing* 48.1 (2019), pp. 122–143. DOI: 10.1137/17M1117537.

[CP19] Yi-Jun Chang and Seth Pettie. “A Time Hierarchy Theorem for the LOCAL Model”. In: *SIAM Journal on Computing* 48.1 (2019), pp. 33–69. DOI: 10.1137/17M1157957.

[Coi+24] Xavier Coiteux-Roy, Francesco D’Amore, Rishikesh Gajjala, Fabian Kuhn, François Le Gall, Henrik Lievonen, Augusto Modanese, Marc-Olivier Renou, Gustav Schmid, and Jukka Suomela. “No Distributed Quantum Advantage for Approximate Graph Coloring”. In: *Proceedings of the 56th Annual ACM Symposium on Theory of Computing, STOC 2024, Vancouver, BC, Canada, June 24-28, 2024*. Ed. by Bojan Mohar, Igor Shinkar, and Ryan O’Donnell. ACM, 2024, pp. 1901–1910. DOI: 10.1145/3618260.3649679.

[DCP16] Giacomo Mauro D’Ariano, Giulio Chiribella, and Paolo Perinotti. *Quantum Theory from First Principles: An Informational Approach*. Cambridge University Press, 2016. DOI: 10.1017/9781107338340.

[Der+08] Bilel Derbel, Cyril Gavoille, David Peleg, and Laurent Viennot. “On the locality of distributed sparse spanner construction”. In: *Proceedings of the Twenty-Seventh Annual ACM Symposium on Principles of Distributed Computing, PODC 2008, Toronto, Canada, August 18-21, 2008*. Ed. by Rida A. Bazzi and Boaz Patt-Shamir. ACM, 2008, pp. 273–282. DOI: 10.1145/1400751.1400788.

[Dha+24] Anubhav Dhar, Eli Kujawa, Henrik Lievonen, Augusto Modanese, Mikail Muftuoglu, Jan Studený, and Jukka Suomela. “Local Problems in Trees Across a Wide Range of Distributed Models”. In: *28th International Conference on Principles of Distributed Systems, OPODIS 2024, December 11-13, 2024, Lucca, Italy*. Ed. by Silvia Bonomi, Letterio Galletta, Etienne Rivière, and Valerio Schiavoni. Vol. 324. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2024, 27:1–27:17. DOI: 10.4230/LIPICS.OPODIS.2024.27.

[Elk07] Michael Elkin. “A near-optimal distributed fully dynamic algorithm for maintaining sparse spanners”. In: *Proceedings of the Twenty-Sixth Annual ACM Symposium on Principles of Distributed Computing, PODC 2007, Portland, Oregon, USA, August 12-15, 2007*. Ed. by Indranil Gupta and Roger Wattenhofer. ACM, 2007, pp. 185–194. DOI: 10.1145/1281100.1281128.

[FG17] Manuela Fischer and Mohsen Ghaffari. "Sublogarithmic Distributed Algorithms for Lovász Local Lemma, and the Complexity Hierarchy". In: *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*. Ed. by Andréa W. Richa. Vol. 91. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017, 18:1–18:16. DOI: 10.4230/LIPICS.DISC.2017.18.

[Fra+24] Pierre Fraigniaud, Maël Luce, Frédéric Magniez, and Ioan Todinca. "Even-Cycle Detection in the Randomized and Quantum CONGEST Model". In: *Proceedings of the 43rd ACM Symposium on Principles of Distributed Computing, PODC 2024, Nantes, France, June 17-21, 2024*. Ed. by Ran Gelles, Dennis Olivetti, and Petr Kuznetsov. ACM, 2024, pp. 209–219. DOI: 10.1145/3662158.3662767.

[Gav+09] Cyril Gavoille, Ralf Klasing, Adrian Kosowski, Lukasz Kuszner, and Alfredo Navarra. "On the complexity of distributed graph coloring with local minimality constraints". In: *Networks* 54.1 (2009), pp. 12–19. DOI: 10.1002/NET.20293.

[Gav+07] Cyril Gavoille, Ralf Klasing, Adrian Kosowski, and Alfredo Navarra. "On the Complexity of Distributed Greedy Coloring". In: *Distributed Computing, 21st International Symposium, DISC 2007, Lemesos, Cyprus, September 24-26, 2007, Proceedings*. Ed. by Andrzej Pelc. Vol. 4731. Lecture Notes in Computer Science. Springer, 2007, pp. 482–484. DOI: 10.1007/978-3-540-75142-7_37.

[GKM09] Cyril Gavoille, Adrian Kosowski, and Marcin Markiewicz. "What Can Be Observed Locally?" In: *Distributed Computing, 23rd International Symposium, DISC 2009, Elche, Spain, September 23-25, 2009. Proceedings*. Ed. by Idit Keidar. Vol. 5805. Lecture Notes in Computer Science. Springer, 2009, pp. 243–257. DOI: 10.1007/978-3-642-04355-0_26.

[GHK18] Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. "On Derandomizing Local Distributed Algorithms". In: *59th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2018, Paris, France, October 7-9, 2018*. Ed. by Mikkel Thorup. IEEE Computer Society, 2018, pp. 662–673. DOI: 10.1109/FOCS.2018.00069.

[GKM17] Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. "On the complexity of local distributed graph problems". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*. Ed. by Hamed Hatami,

Pierre McKenzie, and Valerie King. ACM, 2017, pp. 784–797. DOI: 10.1145/3055399.3055471.

[GRB22] Christoph Grunau, Václav Rozhon, and Sebastian Brandt. "The Landscape of Distributed Complexities on Trees and Beyond". In: *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*. Ed. by Alessia Milani and Philipp Woelfel. ACM, 2022, pp. 37–47. DOI: 10.1145/3519270.3538452.

[Hir+17] Juho Hirvonen, Joel Rybicki, Stefan Schmid, and Jukka Suomela. "Large Cuts with Local Algorithms on Triangle-Free Graphs". In: *The Electronic Journal of Combinatorics* 24.4 (2017), p. 4. DOI: 10.37236/6862.

[HHL18] Alexander E. Holroyd, Tom Hutchcroft, and Avi Levy. "Finitely dependent cycle coloring". In: *Electronic Communications in Probability* 23 (2018). DOI: 10.1214/18-ecp118.

[HL16] Alexander E. Holroyd and Thomas M. Liggett. "Finitely Dependent Coloring". In: *Forum of Mathematics, Pi* 4, e9 (2016). DOI: 10.1017/fmp.2016.7.

[IL19] Taisuke Izumi and François Le Gall. "Quantum Distributed Algorithm for the All-Pairs Shortest Path Problem in the CONGEST-CLIQUE Model". In: *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*. Ed. by Peter Robinson and Faith Ellen. ACM, 2019, pp. 84–93. DOI: 10.1145/3293611.3331628.

[ILM20] Taisuke Izumi, François Le Gall, and Frédéric Magniez. "Quantum Distributed Algorithm for Triangle Finding in the CONGEST Model". In: *37th International Symposium on Theoretical Aspects of Computer Science, STACS 2020, March 10-13, 2020, Montpellier, France*. Ed. by Christophe Paul and Markus Bläser. Vol. 154. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2020, 23:1–23:13. DOI: 10.4230/LIPICS.STACS.2020.23.

[KSV13] Amos Korman, Jean-Sébastien Sereni, and Laurent Viennot. "Toward more localized local algorithms: removing assumptions concerning global knowledge". In: *Distributed Computing* 26.5-6 (2013), pp. 289–308. DOI: 10.1007/S00446-012-0174-8.

[KMW04] Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. "What cannot be computed locally!" In: *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC 2004, St. John's, Newfoundland, Canada, July 25-28, 2004*.

Ed. by Soma Chaudhuri and Shay Kutten. ACM, 2004, pp. 300–309. DOI: 10.1145/1011767.1011811.

[LM18] François Le Gall and Frédéric Magniez. "Sublinear-Time Quantum Computation of the Diameter in CONGEST Networks". In: *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing*. ACM, 2018, pp. 337–346. DOI: 10.1145/3212734.3212744.

[LNR19] François Le Gall, Harumichi Nishimura, and Ansis Rosmanis. "Quantum Advantage for the LOCAL Model in Distributed Computing". In: *36th International Symposium on Theoretical Aspects of Computer Science, STACS 2019, March 13-16, 2019, Berlin, Germany*. Ed. by Rolf Niedermeier and Christophe Paul. Vol. 126. LIPIcs. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019, 49:1–49:14. DOI: 10.4230/LIPICS.STACS.2019.49.

[Lin87] Nathan Linial. "Distributive Graph Algorithms-Global Solutions from Local Data". In: *28th Annual Symposium on Foundations of Computer Science, Los Angeles, California, USA, 27-29 October 1987*. IEEE Computer Society, 1987, pp. 331–335. DOI: 10.1109/SFCS.1987.20.

[Lin92] Nathan Linial. "Locality in Distributed Graph Algorithms". In: *SIAM Journal on Computing* 21.1 (1992), pp. 193–201. DOI: 10.1137/0221015.

[MN22] Frédéric Magniez and Ashwin Nayak. "Quantum Distributed Complexity of Set Disjointness on a Line". In: *ACM Transactions on Computation Theory* 14.1 (2022), 5:1–5:22. DOI: 10.1145/3512751.

[MS02] Bojan Mohar and Paul D. Seymour. "Coloring Locally Bipartite Graphs on Surfaces". In: *J. Comb. Theory B* 84.2 (2002), pp. 301–310. DOI: 10.1006/JCTB.2001.2086.

[MST13] Bojan Mohar, Gábor Simonyi, and Gábor Tardos. "Local chromatic number of quadrangulations of surfaces". In: *Combinatorica* 33.4 (2013), pp. 467–495. DOI: 10.1007/S00493-013-2771-Y.

[Nao91] Moni Naor. "A Lower Bound on Probabilistic Algorithms for Distributive Ring Coloring". In: *SIAM Journal on Discrete Mathematics* 4.3 (1991), pp. 409–412. DOI: 10.1137/0404036.

[NS93] Moni Naor and Larry J. Stockmeyer. "What can be computed locally?" In: *Proceedings of the Twenty-Fifth Annual ACM Symposium on Theory of Computing, May 16-18, 1993, San Diego, CA, USA*. Ed. by S. Rao Kosaraju, David S. Johnson, and Alok Aggarwal. ACM, 1993, pp. 184–193. DOI: 10.1145/167088.167149.

[NS95] Moni Naor and Larry J. Stockmeyer. "What Can be Computed Locally?" In: *SIAM Journal on Computing* 24.6 (1995), pp. 1259–1277. DOI: 10.1137/S0097539793254571.

[Oli19] Dennis Olivetti. *Round Eliminator: a tool for automatic speedup simulation*. 2019. URL: https://github.com/olidennis/round-eliminator.

[WWY21] Changsheng Wang, Xudong Wu, and Penghui Yao. "Complexity of Eccentricities and All-Pairs Shortest Paths in the Quantum CONGEST Model". In: *SPIN* 11.03 (2021), p. 2140007. DOI: 10.1142/S2010324721400075. eprint: https://doi.org/10.1142/S2010324721400075.

[WY22] Xudong Wu and Penghui Yao. "Quantum Complexity of Weighted Diameter and Radius in CONGEST Networks". In: *PODC '22: ACM Symposium on Principles of Distributed Computing, Salerno, Italy, July 25 - 29, 2022*. Ed. by Alessia Milani and Philipp Woelfel. ACM, 2022, pp. 120–130. DOI: 10.1145/3519270.3538441.