


# A Two-Sided Error Distributed Property Tester For Conductance

**Hendrik Fichtenberger**

TU Dortmund, Dortmund, Germany


hendrik.fichtenberger@tu-dortmund.de

 <https://orcid.org/0000-0003-3246-5323>

**Yadu Vasudev**

Indian Institute of Technology Madras, Chennai, India

yadu@cse.iitm.ac.in

 <https://orcid.org/0000-0001-7918-7194>

---

## Abstract

We study property testing in the distributed model and extend its setting from testing with one-sided error to testing with two-sided error. In particular, we develop a two-sided error property tester for general graphs with round complexity  $\mathcal{O}(\log(n)/(\epsilon\Phi^2))$  in the CONGEST model, which accepts graphs with conductance  $\Phi$  and rejects graphs that are  $\epsilon$ -far from having conductance at least  $\Phi^2/1000$  with constant probability. Our main insight is that one can start  $\text{poly}(n)$  random walks from a few random vertices without violating the congestion and unite the results to obtain a consistent answer from all vertices. For connected graphs, this is even possible when the number of vertices is unknown. We also obtain a matching  $\Omega(\log n)$  lower bound for the LOCAL and CONGEST models by an indistinguishability argument. Although the power of vertex labels that arises from two-sided error might seem to be much stronger than in the sequential query model, we can show that this is not the case.

**2012 ACM Subject Classification** Theory of computation → Distributed algorithms

**Keywords and phrases** property testing, distributed algorithms, conductance

**Digital Object Identifier** 10.4230/LIPIcs.MFCS.2018.19

**Related Version** <https://arxiv.org/abs/1707.06126>

**Funding** The research leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP7/2007-2013)/ ERC grant agreement n° 307696.

**Acknowledgements** We would like to thank Gopal Pandurangan for pointing out related work [7, 21], and we would like to thank Pan Peng for inspiring discussions on spectral graph theory. We are grateful for the helpful comments of anonymous reviewers.

## 1 Introduction

Property testing algorithms derive approximate decisions by probing a sublinear part of the input only. A tester for a graph property  $\mathcal{P}$  is a randomized algorithm that, with high constant probability, accepts graphs that have the property  $\mathcal{P}$  and rejects graphs that are  $\epsilon$ -far from having the property  $\mathcal{P}$ , that is, at least an  $\epsilon$ -fraction of the edges have to be modified to make the graph have the property  $\mathcal{P}$ . Two-sided error testers may err on all graphs, while one-sided error testers have to present a witness when rejecting a graph. See [12, 13, 14] for introductions and surveys.



© Hendrik Fichtenberger and Yadu Vasudev;  
licensed under Creative Commons License CC-BY

43rd International Symposium on Mathematical Foundations of Computer Science (MFCS 2018).

Editors: Igor Potapov, Paul Spirakis, and James Worrell; Article No. 19; pp. 19:1–19:15

Leibniz International Proceedings in Informatics



LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Testing graph properties in the classic, sequential computing model has been studied quite extensively. Property testing in the distributed CONGEST model was first studied by Brakerski and Patt-Shamir [1] and later more thoroughly by Censor-Hillel et al. [2]. In this model, each vertex of the graph is equipped with a processor that has a unique identifier of size  $\mathcal{O}(\log n)$  (alternatively, one may define the model such that vertices pick their identifier randomly) and it knows only its neighboring vertices. The vertices of the graph communicate with each other in synchronized rounds such that in each round only communication of length  $\mathcal{O}(\log n)$  is allowed on every edge. Finally, every vertex casts a vote and a decision rule is applied on all votes to derive the answer of the tester. The complexity measure is the amount of rounds required to test the property. Edge congestion and round complexity strictly limit the amount of information on the whole graph that a single vertex can gather. In [2], it is shown that many one-sided error testers for dense graphs carry over from the sequential to the distributed setting. Furthermore, tight logarithmic bounds for testing bipartiteness and cycle-freeness in bounded degree graphs are proved. In [8, 11], subgraph-freeness is studied for subgraphs on at most five vertices, trees and cliques.

## 1.1 Our Results

We extend the study of distributed property testing to testers with two-sided error. In particular, we present a two-sided error distributed testing algorithm in the CONGEST model for conductance.

► **Theorem 1.** *Testing whether a graph  $G = (V, E)$  has conductance at least  $\Phi$  or is  $\epsilon$ -far from having conductance at least  $\Phi^2/1000$  with two-sided error has complexity  $\mathcal{O}(\log(|V| + |E|)/(\epsilon\Phi^2))$  in the CONGEST model.*

In contrast to previous one-sided error testers, our tester can be implemented such that all vertices accept or all vertices reject (see also the discussion of decision rules in Section 5). Furthermore, we prove that the size of the input graph is not required to be known a priori to perform the test if it is connected. On the other hand, there exists no tester for disconnected graphs if no prior knowledge of the graph is assumed at all. Since communication between two connected components is not allowed, we cannot distinguish a graph  $G$  with high conductance from a graph  $G'$  that is composed of two isolated copies of  $G$ .

In the setting of property testing, we aim for an efficient method to check whether the input has the desired property or is at least close to it. For example, we might run the tester for conductance once in a while on a peer-to-peer network to check whether the topology has changed significantly such that efficient communication is no longer possible. Therefore, we think of  $\Phi$  as a parameter with no or only weak dependence on  $n$ .

We complement this result by showing that any distributed tester with this gap requires  $\Omega(\log(n + m))$  rounds of communication regardless of the final decision rule.

► **Theorem 2.** *Let  $\epsilon, d > 0$  be constants, and let  $\Phi, c$  be constants that depend on  $d$ . Testing whether a  $d$ -regular graph  $G = (V, E)$  has conductance at least  $\Phi$  or is  $\epsilon$ -far from having conductance at least  $c\Phi^2$  requires  $\Omega(\log(|V| + |E|))$  rounds of communication in the LOCAL and CONGEST models.*

## 1.2 Related Work

In the classic, sequential setting of property testing, the problem of testing conductance in bounded degree graphs was first studied in [6, 15]. Kale and Seshadhri [16] and Nachmias and Shapira [22] give  $\tilde{\mathcal{O}}(\Phi^{-2}\sqrt{n})$ -query two-sided error testers that accept graphs that have

conductance at least  $\Phi$  and reject graphs that are  $\epsilon$ -far from having conductance at least  $\Omega(\Phi^2)$ . An algorithm for testing the cluster structure of graphs has been given in [5]. Testing conductance in unbounded degree graphs in the stronger rotation map model with query complexity roughly  $\tilde{O}(\Phi^{-2}\sqrt{m})$  was studied in [17], and testing conductance properties restricted to small sets has been studied in [18]. However, the optimal query complexity for testing conductance in general graphs of unbounded degree is still open.

In the CONGEST model, random walks have been analyzed by Censor-Hillel et al. [2] to design a tester for bipartiteness. The idea there is to perform a constant number of random walks from every vertex and to test if two such walks intersect in a cycle of odd length. It is crucial to bound the number of random walks that traverse an edge in one step. In contrast, we can perform polynomially many random walks from a constant number of vertices in the graph and we can afford that all walks traverse the same edge simultaneously.

Distributed random walks have also been studied in [7] and [21]. In particular, [21] show that one can approximate the mixing time  $\tau_v$  of a vertex  $v$  in  $\mathcal{O}(\tau_v \log n)$  rounds by running  $\text{poly}(n)$  random walks  $v$  and comparing their endpoint distribution to the stationary distribution. The graph's mixing time  $\tau = \max_v \tau_v$  relates to the conductance by  $c_1 \Phi^2 / \log n \leq 1/\tau \leq c_2 \Phi$ . A straightforward approach based on [21] leads to an  $\mathcal{O}(n \log^2(n) / \Phi^2)$  round algorithm for *approximating*  $\Phi$  with a multiplicative gap of  $\Theta(\Phi / \log n)$ . In comparison, our *tester's* gap does not depend on  $n$  and its complexity is only logarithmic in  $n$ . One reason is that if the graph is *far from* having conductance  $\Omega(\Phi^2)$ , there exist many vertices with large mixing times compared to the case that the graph has conductance  $\Phi$  (see the proof of Theorem 1 for details). This is not necessarily the case if the graph is not  $\epsilon$ -far from having conductance  $\Omega(\Phi^2)$ .

### 1.3 Overview

The classic tester [16] for bounded degree graphs exploits that random walks converge rapidly to the uniform distribution in graphs with high conductance and they mix slowly for at least a small fraction of start vertices in graphs that are  $\epsilon$ -far from having high conductance. This boils down to approximating the collision probability of random walks for a few start vertices. However, in general graphs the stationary distribution is not uniform, and collisions at vertices with high degree are more important than at vertices with low degree.

In the distributed model, one has to take care of edge congestion, too. Simulating  $\omega(1)$  random walks while keeping them distinguishable is very costly. A key observation is that for approximating the discrepancy, it is sufficient to maintain only some statistics of the random walks, which reduces the congestion significantly: one has to transfer only the number of random walks that pass through an edge for each of a constant number of start vertices.

For the lower bound, we construct two distributions on graphs of high and low conductance respectively such that the vertices' neighborhoods of radius  $\Omega(\log n)$  are isomorphic. The idea is that within only  $\mathcal{O}(\log n)$  rounds, all vertices receive the same information up to isomorphism and therefore cannot distinguish between the two distributions. However, since the tester is allowed to have two-sided error and it only needs to distinguish graphs that have the property from graphs that are far, it is plausible that it can glean information about the two distributions from the vertex labels of the subgraphs it has seen. For example, certain sets of labeled subgraphs might be present in (many of) the graphs with high conductance that are absent in (many of) the graphs with low conductance. Lower bounds for sequential testers usually argue that with high probability, one can assume that all probes are independent. Since the local views of the vertices in the distributed model always have a large overlap, we base our argument on a reduction and a random labeling argument instead.

## 2 Preliminaries

Let  $G = (V, E)$  be a graph and let  $S, T \subseteq V, S \cap T = \emptyset$  be sets of vertices. We denote  $|V|$  and  $|E|$  by  $n$  and  $m$  respectively for the graph  $G$  at hand. Let  $d(v)$  be the degree of vertex  $v \in V$ . We write  $\bar{S}$  for the set  $V \setminus S$ . The set of vertices in  $\bar{S}$  that are adjacent to some  $u \in S$  is denoted by  $\Gamma(S)$ . The *volume* of  $S$  is the sum of degrees of vertices in  $S$ , that is,  $\text{vol}(S) := \sum_{v \in S} d(v)$ . The cut between  $S$  and  $T$  is denoted by  $E(S, T) = E \cap (S \times T)$ . For a set  $S \subseteq V$  such that  $\text{vol}(S) \leq \text{vol}(\bar{S})$ , the conductance of  $S$  is  $\text{cond}(S) = |E(S, \bar{S})|/\text{vol}(S)$ . The conductance of  $G$  is defined as  $\Phi(G) = \min_{S \subseteq V, \text{vol}(S) \leq \text{vol}(\bar{S})} \text{cond}(S)$ .

### 2.1 Distributed Computing

In the distributed computational model, a computation network  $G = (V, E)$  with a processor associated to each vertex  $v \in V$  is given. Each processor  $v$  has access to numbered communication channels to its neighbors in  $G$ . Additionally, it may have some specific input  $I(v)$ . The computation operates in synchronized rounds that are divided into three phases. In each round, each processor may do some local computation first, then it may send a message to each of its neighbors, and finally it receives the messages sent from its neighbors.

► **Definition 3** (Distributed Computational Model, DCM). *Let  $G = (V, E)$  be a graph and  $p_G = (p_v)_{v \in V}$  with  $p_v : [d(v)] \rightarrow \Gamma(v)$  be a bijective function, that is, an adjacency list representation of  $G$ . Let  $I : V \rightarrow \{0, 1\}^*$  be a mapping from the set of vertices to bit strings. An instance of the distributed computational model on  $G$ ,  $p_G$  and  $I$ ,  $\text{DCM}(G, p_G, I)$ , is defined as follows. Each vertex  $v \in V$  is a processor that has communication access to its neighbors  $p_v(1), \dots, p_v(d(v))$  by ports numbered  $1, \dots, d(v)$ . The model operates in synchronized rounds, where each round  $r$  consists of three phases: (i) Each vertex performs local computation, (ii) each vertex  $v$  sends a message to its neighbor  $p_v(i)$ , denoted  $s_r(v, i)$ , for all  $i \in d(v)$ , (iii) each vertex  $u$  receives a message from its neighbor  $p_u(j)$ , for all  $j \in d(u)$ . The distributed computational model DCM is the set of all instances  $\text{DCM}(G, p_G, I)$ .*

The LOCAL model is the subset of the DCM such that for each vertex  $v \in V$ , the input  $I(v)$  is only  $n$  and a numerical vertex identifier from  $[n^c]$  for some universal constant  $c$ . The CONGEST model is the subset of the LOCAL model such that the size of each message  $s_r(v, i)$  is restricted to  $c \log n$  bits.

A distributed network decision algorithm  $\text{DNDA}(\mathcal{A}, O)$  is an algorithm  $\mathcal{A}$  that is deployed to the vertices of a DCM to decide a property of an instance of the model. In particular, the output of  $\mathcal{A}$  is a single bit, and the final decision is obtained by applying a function  $O(\cdot)$  to the union of all vertices' answers.

► **Definition 4** (Distributed Network Decision Algorithm). *Let  $\mathcal{A}$  be an algorithm that takes a bit string as input and outputs a single bit, and let  $O : \{0, 1\}^* \rightarrow \{0, 1\}$  be a function. When the distributed network decision algorithm  $\text{DNDA}(\mathcal{A}, O)$  is run on an instance  $\text{DCM}(G, p_G, I)$ , a copy of  $\mathcal{A}$  is deployed to every vertex  $v$  with input  $I(v)$  and run in parallel as described in Definition 3. We refer to the copy of  $\mathcal{A}$  deployed to  $v$  by  $\mathcal{A}_v$ . When every vertex  $v_i$  has terminated its computation with output bit  $b_{v_i}$ , the decision of  $\text{DNDA}(\mathcal{A}, O)$  is  $O(b_{v_1} b_{v_2} \cdots b_{v_n})$ .*

### 2.2 Distributed property testing

A distributed property testing algorithm is a distributed algorithm as defined in Definition 4 that accepts graphs that have a property, and rejects graphs that are  $\epsilon$ -far from the property. We say that a graph  $G$  with  $n$  vertices and  $m$  edges is  $\epsilon$ -far from a property  $\mathcal{P}$  if at least  $\epsilon m$

edges of  $G$  have to be modified to make the new graph have the property  $\mathcal{P}$ . A one-sided error distributed  $\epsilon$ -tester accepts all graphs with property  $\mathcal{P}$ , whereas it rejects, with probability at least  $2/3$ , all graphs that are  $\epsilon$ -far from the property. In this paper, we give a two-sided (error) property tester that is also allowed to err, with probability at most  $1/3$ , when the graph has the property.

► **Definition 5** (Two-sided tester). *A two-sided (error) distributed  $\epsilon$ -tester for a property  $\mathcal{P}$  is a DNDA( $\mathcal{A}, O$ ), where  $O(b_{v_1}b_{v_2}\cdots b_{v_n}) = 1$  iff  $b_{v_i} = 1$  for all  $v_i \in V$  such that the following conditions hold: (i) if  $G$  has the property  $\mathcal{P}$ , then, with probability at least  $2/3$ ,  $b_{v_i} = 1$  for all  $v_i \in V$ , (ii) if  $G$  is  $\epsilon$ -far from  $\mathcal{P}$ , then, with probability at least  $2/3$ , there exists a  $v_i \in V$  such that  $b_{v_i} = 0$ .*

The guarantees given by our tester are actually a bit stronger in the sense that the tester can be modified such that either  $b_v = 0$  or  $b_v = 1$  for all  $v \in V$  simultaneously.

### 3 Testing Using Random Walks

In this section we will present the distributed algorithm for testing whether a graph has conductance at least  $\Phi$  or is  $\epsilon$ -far from having conductance at least  $\Phi^2/1000$ . The core idea of the algorithm is to perform random walks from a small set of vertices and test whether these walks converge to the stationary distribution rapidly, which is the case for graphs with high conductance. It is based on the ideas of Kale and Seshadhri [16] and Goldreich and Ron[15].

Before we describe the algorithm, we give a few useful definitions and lemmas. A *lazy random walk* on a graph  $G = (V, E)$  on  $n$  vertices is a random walk on the graph, where at each vertex  $v$  the walk chooses to stay at  $v$  with probability  $1/2$  and chooses a neighbor  $u$  with probability  $1/(2d(v))$ . The walk matrix  $W = [w_{uv}]_{u,v \in [n]}$  is defined by  $w_{uv} := 1/2$  if  $u = v$ ,  $w_{uv} := 1/(2d(v))$  if  $u \neq v, (u, v) \in E$  and  $w_{uv} := 0$  otherwise. Notice that for irregular graphs,  $W$  is not symmetric. To analyze these random walks, one can draw on the *normalized walk matrix*, which is a symmetric matrix similar to  $W$ . The normalized walk matrix  $N$  of  $G$  is  $D^{-1/2}WD^{1/2}$ , where  $D$  is the diagonal matrix with  $D(u, u) := d(u)$ .

Since  $N$  is a real symmetric matrix, it has real eigenvalues. Let  $1 = \mu_1, \dots, \mu_n \geq 0$  be its eigenvalues, and let  $\{\vec{f}_i\}_{i \in [n]}$  be its orthonormal eigenbasis. We have  $\vec{f}_1 = \sqrt{\vec{\pi}}$ , where  $\vec{\pi}$  is the random walk's *stationary distribution*. In particular, it is well known that  $\vec{\pi}_v = d(v)/(2m)$ . For more details on spectral graph theory, refer to [4].

It is well known that graphs with high conductance have small diameter.

► **Lemma 6** ([3, cf. Theorem 2]). *Let  $G = (V, E)$  be a graph with conductance  $\Phi$ . The diameter of  $G$  is at most  $(3/\Phi) \ln(m)$ .*

Sinclair [24] proved that there is a tight connection between the conductance and the mixing time of random walks. In particular, the  $L_2$  distance of any starting distribution  $\vec{\pi}'$  to  $\vec{\pi}$  after  $\Phi^{-2} \log n$  steps is  $\mathcal{O}(1/n)$ .

► **Lemma 7** ([24, cf. Theorem 2.5]). *Let  $G = (V, E)$  be a graph with conductance  $\Phi$ . For any starting distribution  $\vec{\pi}'$ , it holds that  $\|W^\ell \vec{\pi}' - \vec{\pi}\|_2 \leq (1 - \Phi^2/2)^\ell$ .*

#### 3.1 Algorithm

We discuss the algorithm from a global point of view instead of describing an algorithm  $\mathcal{A}$  for a single vertex to provide a better explanation of the interactions between vertices.

Lemma 6 implies that if the graph has high conductance, then it has diameter  $\mathcal{O}(\log n/\Phi)$ , which we want to use as an assumption in the algorithm later. To test the diameter, we perform a BFS of depth  $\mathcal{O}(\log n/\Phi)$  of the graph starting from an arbitrary vertex. Initially, every vertex chooses itself as root of the BFS and announces itself as root to all its neighbors. To break the symmetry between the vertices, a vertex accepts every vertex with a lower identifier than its current root as new root and forwards its messages. If the diameter is  $\mathcal{O}(\log n/\Phi)$ , a unique root has been chosen after  $\mathcal{O}(\log n/\Phi)$  rounds and every vertex knows its parent and its children in the BFS tree. Otherwise, at least one of the remaining candidates will reject. Algorithm 3 gives a formal description of the BFS.

From now on, assume that the diameter is  $\mathcal{O}(\log n/\Phi)$ . Using the previously computed BFS tree, we can compute the number of edges in the graph by summing up vertex degrees from the leaves to the root and transmitting this number to all vertices afterwards. Algorithm 4 describes the procedure in detail.

The key technical lemma from [16] for bounded degree graphs states that if a graph is  $\epsilon$ -far from having conductance  $\Omega(\Phi^2)$ , then there exists a  $\Omega(\epsilon)$ -fraction of *weak* vertices such that random walks starting from these vertices converge only slowly to the stationary distribution. Therefore, a sample  $S \subset V$  of size  $\mathcal{O}(1)$  will likely contain a weak vertex (technically, we sample each vertex  $v$  independently into  $S$  with probability  $\Theta(d(v)/\epsilon m)$ . By Markov's inequality, we may reject if  $S$  is much larger than its expected size.). We extend this lemma to unbounded degree graphs. Then, we perform  $N = n^{100}$  random walks of length  $\ell = 40/\Phi^2 \cdot \log n = \mathcal{O}(\log n/\Phi^2)$  starting from each of the vertices in  $S$  to approximate the rate of convergence.

The crucial point here is that in each round of the algorithm, we do not send the full trace of every random walk. Instead, for every origin  $v \in S$ , every vertex  $u \in V$  only transmits the total number of random walks that are leaving it through an edge  $(u, w)$  to its neighbor  $w \in \Gamma(u)$ . Since the size of  $S$  is constant, we require  $\mathcal{O}(\log n)$  bits per edge to communicate this. On the other hand, this information is sufficient because we are only interested in the distribution of endpoints of the lazy random walks for every  $v \in S$ . Algorithm 2 gives a formal description of this procedure. Finally, the estimated distribution of endpoints is used to approximate the distance to the stationary distribution for each  $v \in S$ . The whole algorithm is summarized in Algorithm 1.

First, we show that either the estimates  $\widehat{W}_{v,u}^\ell$  of Algorithm 2 are good or the algorithm rejects in line 14 because  $G$  has low conductance.

► **Lemma 8.** *Consider Algorithm 2. For every  $v, u \in V$ , it holds with probability at least  $1 - m^{-10}$  that (i)  $|\widehat{W}_{v,u}^\ell - W^\ell(v, u)| \leq m^{-20}$  and, conditioned on the previous, (ii) if  $\widehat{W}_{v,u}^\ell < m^{-2}$  then  $G$  has conductance less than  $\Phi$ .*

**Proof.** We have  $E[\widehat{W}_{v,u}^\ell] = W^\ell(v, u)$ . By Hoeffding's inequality, it holds that

$$\Pr[|\widehat{W}_{v,u}^\ell - E[\widehat{W}_{v,u}^\ell]| \geq m^{-10}] \leq 2 \exp\left(-\frac{N}{3m^{40}}\right) \leq m^{-10}. \quad (1)$$

Condition on  $|\widehat{W}_{v,u}^\ell - E[\widehat{W}_{v,u}^\ell]| < m^{-20}$ , which happens with probability at least  $1 - 1/m^{10}$ . If  $\widehat{W}_{v,u}^\ell < m^{-2}$ , then

$$W^\ell(v, u) = E[\widehat{W}_{v,u}^\ell] < \widehat{W}_{v,u}^\ell + m^{-20} = m^{-2} + m^{-10} < 2m^{-2}.$$

Let  $\pi' = \mathbb{1}_v$ . We bound  $\|W^\ell \pi' - \bar{\pi}\|_2$  from below.

$$\|W^\ell \pi' - \bar{\pi}\|_2 \geq |W^\ell(v, u) - d(u)/2m| \geq -(2m^{-2} - 1/(2m)) \geq 1/(4m).$$

By the contrapositive of Lemma 7,  $G$  has conductance less than  $\Phi$ . ◀

---

**Algorithm 1** Conductance tester.
 

---

```

1: procedure TESTCONDUCTANCE( $G = (V, E)$ ,  $n$ ,  $\Phi$ )
2:   BFS( $G$ ,  $6/\Phi \ln n$ )  $\triangleright$  construct BFS of depth  $6/\Phi \log n$ , Algorithm 3
3:   if BFS visited less than  $n$  vertices then reject
4:    $m \leftarrow$  AGGREGATESUM( $G$ ,  $12/\Phi \ln n$ ,  $f$ )  $\triangleright f(v) := d(v)/2$ , Algorithm 4
5:   let every vertex  $v \in V$  do
6:     with probability  $\min\{1, 10^4 d(v)/2em\}$ , mark  $v$ 
7:    $S \leftarrow$  marked  $v$ ,  $r \leftarrow$  root of BFS tree
8:   if  $|S| > 10^5/\epsilon$  then reject
9:   RANDOMWALK( $G$ ,  $S$ ,  $40/\Phi^2 \cdot \log n$ ,  $n^{100}$ )  $\triangleright$  compute local  $s_{v,u}$ , Algorithm 2
10:  for all  $v \in S$  do  $s_v \leftarrow$  AGGREGATESUM( $G$ ,  $12/\Phi \ln n$ ,  $f$ )  $\triangleright f(u) := s_{v,u}$ , Alg. 4
11:  let every vertex  $v \in V$  do
12:    if  $s_v \leq m^{-15}$  for all  $v \in S$  then accept
13:    else reject

```

---



---

**Algorithm 2** Perform random walks.
 

---

```

1: procedure RANDOMWALK( $G, S, \ell, N$ )
2:   let every vertex  $v \in S$  do
3:     sample  $u_1, \dots, u_N$  independently according to  $W\vec{e}_v$ 
4:     for all  $w \in \Gamma(v)$  do send  $(v, v, |\{i \mid w = u_i\}|)$  to  $w$ 
5:   for  $\ell$  rounds, let every vertex  $x$  do
6:     receive  $(u_1, x'_1, k_1), (u_2, x'_2, k_2), \dots$ 
7:     for all  $v \in S$  do
8:       sample  $u_1, \dots, u_{n_v}$  independently according to  $W\vec{e}_x$ , where  $n_v = \sum_{u_i=v} k_i$ 
9:       for all  $w \in \Gamma(v)$  do send  $(v, x, |\{i \mid w = u_i\}|)$  to  $w$ 
10:  let every vertex  $u \in V$  do
11:    receive  $(u_1, x'_1, k_1), (u_2, x'_2, k_2), \dots$ 
12:    for all  $v \in S$  do
13:       $\widehat{W}_{v,u}^\ell \leftarrow \sum_{v_i=v} n_i/N$ 
14:      if  $\widehat{W}_{v,u}^\ell \leq 2m^{-2}$  then reject
15:       $s_{v,u} \leftarrow (\widehat{W}_{v,u}^\ell - \frac{d(v)}{2m})^2$ 

```

---



---

**Algorithm 3** Construct BFS tree.
 

---

```

1: procedure BFS( $G, D$ )
2:   let every vertex  $v$  do
3:      $T_v \leftarrow (v, \cdot)$   $\triangleright$  set root to itself, parent to empty
4:      $minid \leftarrow v$ 
5:     send  $(v, v)$  to every neighbor  $u \in \Gamma(v)$ 
6:   for  $D$  rounds, let every vertex  $w$  do
7:      $R_w \leftarrow \{(v', u') \text{ received} \mid u' \in \Gamma(w)\}$ 
8:      $(v, u) \leftarrow \arg \min_{(v', u') \in R_w} v'$ 
9:     if  $T_w = (\cdot)$  or  $v' < minid$  then
10:       $T_w \leftarrow (v, u)$   $\triangleright$  set root to  $v$ , parent to  $u$ 
11:      send  $(v, w)$  to all neighbors  $\neq u$ 

```

---

---

**Algorithm 4** Aggregate sum of vertex values and propagate it to all vertices.

---

**Require:**  $\forall v : v$  has local information  $f(v)$   
**Ensure:**  $\forall v : v$  has information  $\sum_{u \in V} f(u)$

- 1: **procedure** AGGREGATESUM( $G, D, f : V \rightarrow \mathbb{R}$ )
- 2:     **for**  $D$  rounds, **let** every vertex  $v$  **do**
- 3:         **if**  $v$  received partial sums  $s_u$  from all its children  $u$  in BFS tree **then**
- 4:              $s_v \leftarrow f(v) + \sum_u s_u$
- 5:             send  $s_v$  to parent in BFS tree
- 6:     **let** vertex root  $r$  of BFS tree **do**
- 7:         send total sum  $s = \sum_v f(v)$  to all children
- 8:     **for**  $D$  rounds, **let** every vertex  $v$  **do**
- 9:         **if**  $v$  received total sum  $s$  from its parent **then**
- 10:             send  $s_v$  to all children in BFS tree
- 11:     **return**  $s_v$   $\triangleright$  consider  $s_v$  to be the output of the algorithm

---

Furthermore, Lemma 8 implies that the estimates  $s_v$  in Algorithm 1 (see line 10) are also good if Algorithm 2 has not rejected before.

► **Lemma 9.** *Consider Algorithm 1. With probability at least  $1 - m^{-8}$  it holds for every  $v \in S$  in line 10 that  $|\|W^\ell(v, \cdot) - \bar{\pi}\|_2^2 - s_v| \leq 3m^{-19}$ .*

**Proof.** Let  $v \in S$ . We have the following equality for the discrepancy of the distribution of the random walks' endpoints that start at  $v$  and the stationary distribution:

$$\|W^\ell(v, \cdot) - \bar{\pi}\|_2^2 = \sum_{u \in V} \left( W^\ell(v, u) - \frac{d(u)}{2m} \right)^2. \quad (2)$$

By Lemma 8, we know that for every  $u \in V$  we have  $|\widehat{W}_{u,v}^\ell - W^\ell(v, u)| \leq m^{-20}$  with probability  $1 - 1/m^9$ . Then, by the triangle inequality,  $|(W^\ell(v, u) - \frac{d(u)}{2m})^2 - s_{v,u}| \leq 3m^{-20}$ . Combining this with Eq. (2), a union bound over all  $u \in V$  implies that with probability at least  $1 - n \cdot m^{-10} \geq 1 - m^{-9}$ , we have that  $|\|W^\ell(v, \cdot) - \bar{\pi}\|_2^2 - \sum_{u \in V} s_{v,u}| \leq 3m^{-19}$ . A union bound over all  $v \in S$  gives that with probability at least  $1 - |S|/m^{-9} \geq 1 - m^{-8}$ ,  $|\|W^\ell(v, \cdot) - \bar{\pi}\|_2^2 - \sum_{u \in V} s_{v,u}| \leq 3m^{-19}$  for every  $v \in S$ . ◀

### 3.2 Completeness and Soundness

The proof of completeness is a straightforward application of the results from the previous section.

► **Lemma 10 (Completeness).** *Let  $G(V, E)$  be a graph with conductance at least  $\Phi$ . Then, with probability at least  $2/3$ , each vertex in  $G$  returns **accept** when it runs Algorithm 1.*

**Proof.** The probability that the algorithm rejects in Line 8 of Algorithm 1 is at most  $1/10$ , and we assume, for the remainder of the proof, that this event did not occur. If  $G$  has conductance at least  $\Phi$ , then from Lemma 7 we know that  $\|W^\ell(\cdot, v) - \bar{\pi}\|_2^2 \leq (1 - \Phi^2/2)^{2\ell} \leq \exp(-\Phi^2\ell/2) \leq m^{-20}$  for every vertex  $v$ . Lemma 9 implies that with probability at least  $9/10$ , it holds that  $|\|W^\ell(\cdot, v) - \bar{\pi}\|_2^2 - s_v| \leq 3m^{-19}$ . Conditioning on this event, every vertex accepts in line 12 of Algorithm 1. ◀

To complete the analysis of the tester, we show that whenever the graph is  $\epsilon$ -far from having conductance  $\Omega(\Phi^2)$ , the tester rejects with probability at least  $2/3$ . To this end, we



actually show that if the volume of weak vertices is small, then the graph can be converted to another graph  $G'$  by modifying at most  $\epsilon m$  edges such that the conductance is  $\Omega(\Phi^2)$ . The idea of the analysis is due to Kale and Seshadhri [16], who analyzed a classic property tester for testing expansion in graphs with vertex degrees bounded by a constant. We deviate from their analysis where it becomes necessary to take care of arbitrary vertex degrees.

Let a vertex  $v \in V$  be called *weak* if  $\|W^\ell(v, \cdot) - \vec{\pi}\|_2 > 6m^{-15}$ . The following lemma states that if there exists a set of vertices  $S$  with small conductance, then there exists a set of weak vertices  $T$  whose volume is at least a constant fraction of the volume of  $S$ .

► **Lemma 11.** *Let  $S \subset V$  be such that  $\text{vol}(S) \leq \text{vol}(\bar{S})$  and  $\text{cond}(S) \leq \delta$ . Then, for any  $\ell \in \mathbb{N}$  and any  $0 < \theta \leq 1/10$ , there exists a set  $T \subseteq S$  such that  $\text{vol}(T) \geq \theta \text{vol}(S)$  and for every  $v \in T$ , it holds that  $\|W^\ell(v, \cdot) - \vec{\pi}\|_2^2 > \frac{1}{80m^\tau} (1 - 4\delta)^{2\ell}$ .*

The proof can be found in the arXiv version [10]. We can use Lemma 11 to separate weak vertices from the remaining graph.

► **Lemma 12.** *Let  $G = (V, E)$  be a graph. If the volume of weak vertices in  $G$  is at most  $(1/100)\epsilon m$ , then there is a partition of  $V$  into  $P \cup \bar{P}$  such that  $\text{vol}(P) \leq \epsilon m/10$  and  $\Phi(G[\bar{P}]) \geq \Phi^2/256$ .*

**Proof.** We partition the graph recursively into two sets  $(P, \bar{P})$ . At the beginning,  $P_0 = \emptyset$  and  $\bar{P}_0 = V$ . As long as there is a cut  $(C_i, \bar{C}_i)$  in  $\bar{P}_{i-1}$  in step  $i$  with  $\text{vol}(C_i) \leq \text{vol}(\bar{C}_i)$  and  $E(C_i, \bar{C}_i)/\text{vol}(C_i) \leq \Phi^2/256$ , we set  $P_i = P_{i-1} \cup C_i$  and  $\bar{P}_i = V \setminus P_i$ . We continue this until we don't find such a cut or the condition  $\text{vol}(P_{i+1}) \leq \text{vol}(\bar{P}_{i+1})$  would be violated. The number of edges going across the cut  $(P, \bar{P})$  is at most  $\sum_i |E(C_i, \bar{C}_i)|$ . Therefore,  $|E(P, \bar{P})| \leq \frac{\Phi^2}{256} \sum_i \text{vol}(C_i) \leq \frac{\Phi^2}{256} \text{vol}(P)$ .

Now, assume that  $\text{vol}(P) > (1/10)\epsilon m$ . Lemma 11 implies that there exists  $P' \subseteq P$  such that  $\text{vol}(P') \geq \frac{1}{10} \text{vol}(P) > \epsilon m/100$  (where  $\theta = 1/10$ ) and for all  $v \in P'$  we have  $\|W^\ell(v, \cdot) - \vec{\pi}\|_2 > \frac{1}{80m^\tau} (1 - 4\Phi^2/256)^{2\ell} > \frac{1}{80m^{10}}$ . This means that  $P'$  contains only weak vertices and has volume at least  $\epsilon m/100$ , which contradicts our assumption that the volume of weak vertices in  $G$  is at most  $\epsilon m/100$ . Therefore,  $\text{vol}(P) \leq \epsilon m/10$  when the partitioning terminates. Hence  $\Phi(G[\bar{P}]) \geq \Phi^2/256$ . ◀

Finally, the following lemma states that few edge modifications in a graph with separated weak vertices are sufficient to make it a graph with high conductance.

► **Lemma 13** ([18, Lemma 9]). *Let  $G = (V, E)$  be a graph. If there exists a set  $P \subseteq V$  such that  $\text{vol}(P) \leq \epsilon m/10$  and the subgraph  $G[V \setminus P]$  is a  $\Phi'$ -expander, then there exists an algorithm that modifies at most  $\epsilon m$  edges to get a  $\Phi'/3$ -expander  $G' = (V, E')$ .*

Combining the results on the separation of weak vertices and patching the graph (Lemmas 11 to 13) and approximating the endpoint distribution (Lemmas 8 and 9), we prove the soundness of the algorithm.

► **Lemma 14 (Soundness).** *Let  $G(V, E)$  be a graph. If  $G$  is  $\epsilon$ -far from having conductance at least  $\Phi^2/768$ , then, with probability at least  $2/3$ , each vertex in  $G$  returns **reject** when it runs Algorithm 1.*

**Proof.** First we note that if the volume of weak vertices is less than  $\epsilon m/100$ , then by Lemmas 12 and 13, the graph is  $\epsilon$ -close to having conductance at least  $\Phi^2/768$ . Therefore, the volume of weak vertices is at least  $\epsilon m/100$ . Each vertex  $v$  is contained in  $S$  with probability  $\Theta(d(v)/\epsilon m)$ . Hence, the expected number of weak vertices that are present in

the sample  $S$  is at least 100. Therefore, with probability at least  $9/10$ , at least one weak vertex is sampled in  $S$ .

If  $W^\ell(v, u) < m^{-2}$  for some  $v \in S, u \in V$ , then with probability at least  $9/10$ ,  $\widehat{W}_{v,u}^\ell < 2m^{-2}$  by Lemma 8. In this case, the algorithm will reject in line 14 of Algorithm 2. If  $W^\ell(v, u) \geq m^{-2}$  for all  $v \in S, u \in V$ , then with probability at least  $9/10$ , it holds that  $\|W^\ell(v, \cdot) - \bar{\pi}\|_2 - s_v \leq 3m^{-19}$  for every  $v \in S$  by Lemma 9. Since at least one vertex  $v \in S$  is weak, that is,  $\|W^\ell(v, \cdot) - \bar{\pi}\|_2 > 6m^{-15}$ , the algorithm rejects in line 13 of Algorithm 1. ◀

### 3.3 Unknown Size of the Graph

We describe how to get rid of the assumption that the size  $n$  of the graph  $G$  is known to the tester if  $G$  is connected. Note that without any prior knowledge of  $G$ , no distributed tester can distinguish between a graph with conductance  $\Phi$  and two distinct copies of it (the latter is  $\epsilon$ -far from being a graph with conductance  $\Phi^c$  for  $\epsilon < \Phi^c/2, c \geq 1$ ).

First, we describe a slightly simpler version of the final algorithm. In the setting of the simpler algorithm, we mark a single vertex that will initiate the test and will also give the final answer of the tester. We call this vertex the maintainer (of the graph). The algorithm can be easily adapted to the CONGEST model.

Let  $v \in V$  be a fixed vertex. The algorithm either makes  $n$  available at all vertices and runs Algorithm 1 afterwards or  $v$  rejects because  $G$  does not have conductance  $\Phi$ . If  $G$  has conductance  $\Phi$ , the algorithm never rejects.

We start with an initial set  $S = \{v\}$  that is grown in two phases. In the first phase, we extend  $S$  to  $S \cup \Gamma(S)$  as long as  $\text{cond}(S) \geq \Phi$ . In particular,  $v$  starts a BFS and in every round, the vertices in the last level report their degree and the number of neighbors outside of  $S$  to their parents. Similar to Algorithm 1, these are aggregated and sent to  $v$  along the edges of the BFS tree. If  $\text{cond}(S) < \Phi$  for the first time, the algorithm proceeds to the second phase. It continues the BFS for  $-\log(\text{vol}(S))/\log(1-\Phi)$  rounds and stops. If any vertex in the graph notices a neighbor that is not in  $S$  after these rounds, then  $S \neq V$  and the algorithm rejects. Otherwise, we have obtained the value of  $n = |S|$  that can be sent to all vertices, and we continue by executing Algorithm 1.

► **Lemma 15.** *Let  $G = (V, E)$  be a graph and  $\Phi \in [0, 1]$ . There is an algorithm that computes  $n$  if  $G$  has conductance at least  $\Phi$ . Otherwise, it either computes  $n$  or rejects. The round complexity is  $\mathcal{O}(\log m / \log(1 - \Phi))$ .*

**Proof.** It is easy to see that if the algorithm explores the whole graph, it computes  $n$  correctly, and else it rejects. Without loss of generality, let  $G$  have conductance  $\Phi$ . Let  $S_i$  be the set  $S$  after  $i$  rounds and let  $\bar{S}_i = V \setminus S_i$ . We denote the last round of the first (second) phase by  $k$  ( $\ell$ ).

In the first phase, we have that  $\text{vol}(S_i) \geq (1 + \Phi) \cdot \text{vol}(S_{i-1})$  for every round  $i$  and by induction,  $k \leq \log \text{vol}(S_k) / \log(1 + \Phi) \leq \log m / \log(1 + \Phi)$ . We also have that  $\text{vol}(S_k) \geq m/2 \geq \text{vol}(\bar{S}_k)$  because  $G$  has conductance  $\Phi$ . In the second phase, we have that  $\text{vol}(\bar{S}_i) \leq (1 - \Phi) \cdot \text{vol}(\bar{S}_{i-1})$  for every round  $i$ . By induction,  $\ell - k \geq -\log m / \log(1 - \Phi) \geq \log \text{vol}(\bar{S}_k)^{-1} / \log(1 - \Phi)$  implies that  $\text{vol}(\bar{S}_\ell) = 0$ . Therefore, the algorithm has explored the whole graph. Clearly,  $\ell \in \mathcal{O}(\log m / \log(1 - \Phi))$ . ◀

To transform the algorithm into a tester in the CONGEST model, we start with each vertex being a maintainer initially. In every round every vertex chooses the vertex with the smallest id it has ever received a message from to be the maintainer and it forwards only this vertex' messages (the latter maintains the congestion bound). At the end of the algorithm, if

$G$  has conductance  $\Phi$ , then there is only one maintainer (the vertex with the smallest id) and the algorithm continues by executing Algorithm 1. Otherwise, there might be multiple vertices that are still maintainers. However, none of these vertices has explored the whole graph, so all of them send a broadcast message to reject.

#### 4 Lower Bound

In this section, we prove a lower bound of  $\Omega(\log(n+m))$  on the round complexity for testing the conductance of a graph in the LOCAL model regardless of how the final decision of the tester is derived from the single votes of the vertices.

For any  $v \in V$ , the  $k$ -disc of  $v$ , denoted by  $\text{disc}_k(G, v)$ , is defined as the subgraph that is induced by the vertices that are at distance at most  $k$  to  $v$  without the edges between vertices at distance exactly  $k$ , and it is rooted at  $v$ . We refer to the isomorphism type of  $\text{disc}_k(G, v)$ , that is, the set of all rooted graphs isomorphic to  $\text{disc}_k(G, v)$ , by  $\text{disc}_k^*(G, v)$ . Let  $\text{girth}(G)$  denote the length of the shortest cycle in  $G$ . We need the following two lemmas to obtain the distribution over graphs to prove the lower bound.

► **Lemma 16** ([20]; cf. [23, Section 16.8.3]). *For every  $n' \in \mathbb{N}$  and every  $d' \in \mathbb{N}$  there exists a  $d$ -regular graph  $G$  of size  $n$  such that  $G$  has conductance  $\Phi(G) = 1/\sqrt{2d}$  and girth  $2 \log n / \log d$ , and  $n \geq n'$ ,  $d \geq d'$ .*

The second lemma states that we can sparsify an arbitrary cut  $E(V_1, V_2)$  in a  $d$ -regular graph with girth  $3k$  without changing  $\text{disc}_k^*(G, v)$  for any  $v \in V$ . In particular, it states that we can remove two edges in the cut and add them somewhere else, or the cut has size  $\text{poly}(d^k)$  only. It is obtained as a special case by observing that we can assume  $L = 1$  and  $\lambda = 0$  in [9, Lemma 8].

► **Lemma 17** ([9, Lemma 8]). *Let  $G = (V, E)$  be a  $d$ -regular graph with  $\text{girth}(G) \geq 3k$  for  $k \geq 2$  and let  $V_1 \dot{\cup} V_2 = V$  be a partitioning of  $V$ . Then either there exists a graph  $H = (V, F)$  such that (i)  $\text{girth}(H) \geq 3k$ , (ii)  $|F \cap (V_1 \times V_2)| \leq |E \cap (V_1 \times V_2)| - 2$ , and (iii)  $\text{disc}_k^*(H, w) = \text{disc}_k^*(G, w) \forall w \in V$  (that is,  $H$  is  $d$ -regular), or  $|E(V_1, V_2)| \leq 6d^{3k}$ .*

To prove the lower bound, we use an auxiliary model we call the ISO-LOCAL model. In this model, the input  $I(\cdot)$  is empty but an additional oracle provides every vertex  $v$  with the ability to construct  $\text{disc}_r^*(G, v)$  in round  $r$  if it knows  $\text{disc}_{r-1}^*(G, u_i)$  of its neighbors  $u_1, \dots, u_{d(v)}$ . It should be noted that the ISO-LOCAL model is not a DCM due to the additional oracle.

► **Definition 18** (ISO-LOCAL model). *Let  $\text{DCM}(G, p_G, I)$  be a DCM instance such that  $I(\cdot)$  maps the whole domain to the empty string. In addition to sending and receiving messages, in every round  $r$  every vertex  $v$  is provided access to a function  $e_{r,v} : (\mathbb{N} \cup \{\star\})^r \times (\mathbb{N} \cup \{\star\})^r \rightarrow \{0, 1\}$  during the local computation phase. The value of  $e_{r,v}((i_1, \dots, i_{r'}), (j_1, \dots, j_{r''}))$  is 1 iff  $p'_v(i_1, \dots, i_{r'}) = p'_v(j_1, \dots, j_{r''})$ , where*

$$p'_v(i_1, \dots, i_r) := \begin{cases} v & \text{if } i_r = \star \\ p_{p'_v(\star, i_1, \dots, i_{r-1})}(i_r) & \text{otherwise.} \end{cases}$$

The instance  $\text{DCM}(G, p_G, I)$  equipped with such an oracle is called ISO-LOCAL.

In other words,  $p'_v(\cdot)$  takes a path of length at most  $r$  that starts at  $v$  and that is defined by a sequence of port numbers as input. Then, it maps the path to its endpoint in  $V$ . Finally,  $e_{r,v}(\cdot)$  tells whether two such paths end at the same vertex.

The ISO-LOCAL model is a graph where the nodes are not labeled by any strings. To argue the lower bound, we need to prove the existence of graphs that have same local neighborhoods such that one is a good expander and the other is far from having good conductance. Now it is possible that the algorithm can glean information about the different graphs based on the vertex labels even if the local neighborhoods are identical. Without ISO-LOCAL, we would need to argue that a randomized algorithm cannot deduce information from the vertex labels in the LOCAL model directly. This would be easy if for every good expander  $G$  we use, there is a bad expander  $H$  with exactly the same set of (labeled)  $k$ -discs. However, this is not the case as it would imply that  $G$  and  $H$  are isomorphic. The ISO-LOCAL model formalizes the intuition that, still, isomorphic  $k$ -discs should be sufficient to establish the lower bound even for randomized algorithms.

It is a basic observation that a distributed algorithm can only depend on information that has reached it until the moment it performs the computation in question.

► **Lemma 19** (folklore; cf. [19, Section 2]). *Let  $\text{DNDA}(\mathcal{A}, O)$  be a DNDA. After  $r$  rounds, the state of  $\mathcal{A}_v$  may depend only on  $d(v)$ ,  $I(v)$ , the state of  $\mathcal{A}_u$  at time  $r - \text{dist}(v, u)$  for vertices  $u$  with  $\text{dist}(v, u) < r$  and the random coins of  $\mathcal{A}$ .*

#### 4.1 Proof of the Lower Bound

Let  $G = (V, E)$  be an expander graph obtained from applying Lemma 16 and let  $k = \Theta(\log n)$ . Observe that if a graph is  $d$ -regular and it has girth  $3k$ , then all its  $k$ -discs are pairwise isomorphic. In particular, all  $k$ -discs are full  $d$ -ary trees of depth  $k$ .

We will prove that a distributed algorithm  $\text{DNDA}(\mathcal{A}, O)$  with round complexity  $r$  in the ISO-LOCAL model decides based on the set of views  $\text{disc}_r^*(G, v)$  that the different instances of  $\mathcal{A}$  have (see Lemma 20). Using Lemma 17, it will be easy to come up with a graph  $H$  that is a bad expander but whose  $k$ -discs are isomorphic to the ones of  $G$ . This implies a lower bound of  $k = \Theta(\log n)$  for testing conductance in the ISO-LOCAL model (see Proposition 21). Finally, we prove that a lower bound on the round complexity of a tester in the ISO-LOCAL model implies the same bound in the LOCAL model. Actually, we prove the contrapositive: a tester in the LOCAL model implies a tester in the ISO-LOCAL model (see Proposition 22).

► **Lemma 20.** *Let  $\text{DNDA}(\mathcal{A}, O)$  be a deterministic DNDA in the ISO-LOCAL model. The output of  $\mathcal{A}_v$  depends only on  $\text{disc}_r^*(G, v)$  and the port numbering  $(p_v)_{v \in V}$ .*

**Proof.** Instead of analyzing  $\text{DNDA}(\mathcal{A}, O)$ , we analyze a canonical algorithm  $\text{DNDA}(\mathcal{B}, O)$  that simulates  $\text{DNDA}(\mathcal{A}, O)$  depending only on  $\text{disc}_r^*(G, v)$ . Employing  $\mathcal{B}$ , we prove the following statement by induction: After the local computation phase of round  $r$ , the state of  $\mathcal{A}_v$  depends only on  $\text{disc}_r^*(G, v)$ .

The first local computation phase of  $\mathcal{A}_v$  can only depend on the port numbering and  $I(v)$  (the empty string). Therefore,  $\mathcal{B}_v$  can simulate the execution of the first round of  $\mathcal{A}_v$ .

Let the current round be  $r > 1$ . Algorithm  $\mathcal{B}_v$  maintains a rooted graph  $H_v$  that resembles  $\text{disc}_r^*(G, v)$ . The adjacency lists of  $H_v$  are ordered according to  $(p_v)_{v \in V}$ . Let  $H_v(r)$  be the value of  $H_v$  after the computation phase of round  $r$ . In the send phase, vertex  $v$  sends  $H_v(r)$  to each of its neighbors. In the receive phase, vertex  $v$  receives graphs  $H_{u_1}(r), \dots, H_{u_{d(v)}}(r)$  from its neighbors  $u_1, \dots, u_{d(v)}$ . In the subsequent computation phase of round  $r + 1$ , vertex  $v$  extends  $H_v(r) = \text{disc}_r^*(G, v)$  to  $\text{disc}_{r+1}^*(G, v) = H_v(r + 1)$  by querying  $e_{r,v}$  on all pairs of vertices of  $V(H_v(r)) \cup V(H_{u_1}(r)) \cup \dots \cup V(H_{u_{d(v)}}(r))$  to identify vertices and patching the different views together.

Note that  $H_v(r + 1)$  also provides the isomorphism type of  $\text{disc}_{r-\text{dist}(v,u)}^*(G, u)$  for every vertex  $u$  at distance at most  $r$  from  $v$ . Since the adjacency lists of  $H_v$  are ordered according

to the port numbering, it is also possible to reconstruct  $e_{r-\text{dist}(v,u),u}(\cdot)$ . By the induction hypothesis,  $\mathcal{B}_v$  can now simulate round  $r - \text{dist}(v,u)$  of  $\mathcal{A}_u$  for every such  $u$ . By Lemma 19, this is enough to simulate the local computation phase of round  $r$  of  $\mathcal{A}_u$ . ◀

We show that there is no tester for conductance in the ISO-LOCAL model.

▶ **Proposition 21.** *Let  $G = (V, E)$  be  $d$ -regular graph on  $n$  vertices, and let  $\Phi = 1/\sqrt{2d}$  be a constant. Any algorithm for testing if  $G$  has conductance at least  $\Phi$  or is  $\epsilon$ -far from having conductance at least  $c\Phi^2$  (for constants  $c$  and  $\epsilon$ ) in the ISO-LOCAL model that succeeds with probability  $2/3$  requires  $\Omega(\log n)$  rounds of communication.*

**Proof.** Let  $G = (V, E)$  be a  $d$ -regular graph provided by Lemma 16 and choose  $k = \frac{1}{3} \log_d \left( \frac{c\Phi^2 - \epsilon}{6} dn \right)$ . Without loss of generality assume that  $n$  is even, and let  $S \subset V$  be a set of size  $n/2$ . Apply Lemma 17 (with  $V_1 = S$  and  $V_2 = V \setminus S$ ) repeatedly to  $G$  until  $|E(S, V \setminus S)| \leq 6d^{3k}$  holds. Let  $H = (V, E')$  be the resulting graph. We have that  $|E'(S, V \setminus S)| \leq (c\Phi^2 - \epsilon)dn$ , and  $\text{vol}(S) = nd/2$ . Therefore,  $H$  is  $\epsilon$ -far from having conductance  $c\Phi^2$ . Let  $\mathcal{D}_G$  ( $\mathcal{D}_H$ ) be the uniform distribution over all ISO-LOCAL models  $\text{DCM}(G, p_G, I)$  ( $\text{DCM}(H, p_H, I)$ ) such that  $p_G$  ( $p_H$ ) ranges over all possible mappings, that is, port numberings.

We use Yao's principle to prove the lower bound. Let  $\text{DNDA}(\mathcal{A}, O)$  be a tester for conductance that has round complexity smaller than  $k$  in the ISO-LOCAL model. Since  $G$  is  $d$ -regular and  $\text{girth}(G) \geq 3k$ ,  $\text{disc}_k^*(G, v)$  is a full  $d$ -ary tree of depth  $k$  for every  $v \in V$ . For any pair  $u, v \in V$ , we have that  $\text{disc}_k^*(G, u)$  is equal to  $\text{disc}_k^*(H, v)$  by Lemma 17. Since the port numberings of two vertices are independent of each other,  $(p_v)_{v \in V}$  is a valid port numbering for  $G \in \mathcal{D}_G$  iff it is valid for  $H \in \mathcal{D}_H$ . By Lemma 20,  $\text{DNDA}(\mathcal{A}, O)$  cannot distinguish between  $G$  and  $H$ . ◀

To complete the proof of the lower bound, we show that each vertex in the graph in the ISO-LOCAL model can choose an id randomly.

▶ **Proposition 22.** *Let  $\text{DNDA}(\mathcal{A}, O)$  be a randomized tester in the LOCAL model that succeeds with probability  $p$ . Then, there is a randomized tester  $\text{DNDA}(\mathcal{B}, O)$  in the ISO-LOCAL model that succeeds with probability at least  $p - o(1)$ , and has the same round complexity.*

**Proof.** We make a simple modification to  $\mathcal{A}$  to obtain  $\mathcal{B}$ : In the first local computation phase,  $\mathcal{B}_v$  draws a random number  $id_v$  uniformly from  $\{1, \dots, n^3\}$  and feeds it into  $\mathcal{A}_v$  as  $I(v)$ . Then,  $\mathcal{A}_v$  is executed as normal. For  $u, v \in V$ , the probability that  $id_u$  and  $id_v$  are equal is  $1/n^3$ . Applying a union bound, with probability  $1 - o(1)$ , it holds that  $id_u \neq id_v$  for every  $u, v \in V$ . We then run algorithm  $\mathcal{A}$  on this new instance and output the result. ◀

## 5 Open Problems

In the case of one-sided distributed testers, it is natural to define the decision rule  $O(\cdot)$  of a distributed tester such that all vertices have to accept or at least one vertex has to reject. This is because in the case of rejection, the tester is required to observe a witness. However, for two-sided testers no such requirement exists. Requiring that all vertices either accept or reject simultaneously seems to be quite strong. For example, if all should vertices accept or at least one vertex should reject, one may overcome the lack of communication between connected components and test, e. g., whether a graph has more than two connected components with two-sided error. On the other hand, it might not always be possible to obtain a lower bound that is independent of the decision rule as in Theorem 2. To this end, it would be interesting to compare the power of different rules.

## References

- 1 Zvika Brakerski and Boaz Patt-Shamir. Distributed discovery of large near-cliques. *Distributed Computing*, 24(2), 2011. doi:10.1007/s00446-011-0132-x.
- 2 Keren Censor-Hillel, Eldar Fischer, Gregory Schwartzman, and Yadu Vasudev. Fast distributed algorithms for testing graph properties. In *Proceedings of the 30th International Symposium on Distributed Computing (DISC)*, 2016.
- 3 Fan R. K. Chung. Diameters and eigenvalues. *Journal of the American Mathematical Society*, 2(2), 1989.
- 4 Fan R. K. Chung. *Spectral Graph Theory*. American Mathematical Society, 1997.
- 5 Artur Czumaj, Pan Peng, and Christian Sohler. Testing cluster structure of graphs. In *Proceedings of the 47th ACM Symposium on Theory of Computing (STOC)*, 2015.
- 6 Artur Czumaj and Christian Sohler. Testing expansion in bounded-degree graphs. In *Proceedings of the 48th IEEE Symposium on Foundations of Computer Science (FOCS)*, 2007.
- 7 Atish Das Sarma, Danupon Nanongkai, Gopal Pandurangan, and Prasad Tetali. Distributed random walks. *Journal of the ACM (JACM)*, 60(1), 2013.
- 8 Guy Even, Orr Fischer, Pierre Fraigniaud, Tzlil Gonen, Reut Levi, Moti Medina, Pedro Montealegre, Dennis Olivetti, Rotem Oshman, Ivan Rapaport, and Ioan Todinca. Three Notes on Distributed Property Testing. In *Proceedings of the 31st International Symposium on Distributed Computing (DISC)*, 2017.
- 9 Hendrik Fichtenberger, Pan Peng, and Christian Sohler. On constant-size graphs that preserve the local structure of high-girth graphs. In *Proceedings of the 19th International Workshop on Randomization and Computation (RANDOM)*, 2015.
- 10 Hendrik Fichtenberger and Yadu Vasudev. A Two-Sided Error Distributed Property Tester For Conductance. *arXiv:1705.08174*, 2018. URL: <http://arxiv.org/abs/1705.08174>.
- 11 Pierre Fraigniaud, Ivan Rapaport, Ville Salo, and Ioan Todinca. Distributed testing of excluded subgraphs. In *Proceedings of the 30th International Symposium on Distributed Computing (DISC)*, 2016.
- 12 Oded Goldreich. Introduction to testing graph properties. In *Property Testing*. Springer, 2010.
- 13 Oded Goldreich. *Introduction to Property Testing*. Cambridge University Press, 2017.
- 14 Oded Goldreich, Shari Goldwasser, and Dana Ron. Property Testing and Its Connection to Learning and Approximation. *Journal of the ACM (JACM)*, 45(4), 1998. doi:10.1145/285055.285060.
- 15 Oded Goldreich and Dana Ron. On Testing Expansion in Bounded-Degree Graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 2000. URL: <https://eccc.weizmann.ac.il/report/2000/020/>.
- 16 Satyen Kale and C. Seshadhri. An expansion tester for bounded degree graphs. *SIAM Journal on Computing (SIAMP)*, 40(3), 2011. doi:10.1137/100802980.
- 17 Angsheng Li, Yicheng Pan, and Pan Peng. Testing Conductance in General Graphs. *Electronic Colloquium on Computational Complexity (ECCC)*, 18(101), 2011. URL: <http://eccc.hpi-web.de/report/2011/101/>.
- 18 Angsheng Li and Pan Peng. Testing Small Set Expansion in General Graphs. In *Proceedings of the 32nd International Symposium on Theoretical Aspects of Computer Science (STACS)*, 2015.
- 19 Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing (SIAMP)*, 21(1), 1992. doi:10.1137/0221015.
- 20 Alexander Lubotzky, Ralph Phillips, and Peter Sarnak. Ramanujan graphs. *Combinatorica*, 8(3), 1988. doi:10.1007/BF02126799.

- 21 Anisur Rahaman Molla and Gopal Pandurangan. Distributed Computation of Mixing Time. In *Proceedings of the 18th International Conference on Distributed Computing and Networking (ICDCN)*, 2017.
- 22 Asaf Nachmias and Asaf Shapira. Testing the expansion of a graph. *Information and Computation*, 208(4), 2010. doi:10.1016/j.ic.2009.09.002.
- 23 Uwe Naumann and Olaf Schenk, editors. *Combinatorial Scientific Computing*. CRC Press, 2012.
- 24 Alistair Sinclair. *Algorithms for Random Generation and Counting: A Markov Chain Approach*. Birkhauser Verlag, 1993.