On The Multiparty Communication Complexity of Testing Triangle-Freeness

Orr Fischer
Computer Science Department,
Tel-Aviv University
orrfischer@mtau.ac.il

Shay Gershtein
Computer Science Department,
Tel-Aviv University
shayg1@tau.ac.il

Rotem Oshman
Computer Science Department,
Tel-Aviv University
roshman@tau.ac.il

ABSTRACT

In this paper we initiate the study of property testing in multi-party communication complexity, focusing on testing triangle-freeness in graphs. We consider the *coordinator* model, where we have k players receiving private inputs, and a coordinator who receives no input; the coordinator can communicate with all the players, but the players cannot communicate with each other. In this model, we ask: if an input graph is divided between the players, with each player receiving some of the edges, how many bits do the players and the coordinator need to exchange to determine if the graph is triangle-free, or *far* from triangle-free? We are especially interested in *simultaneous* communication protocols, where there is only one communication round.

For general communication protocols, we show that $\tilde{O}(k(nd)^{1/4} + k^2)$ bits are sufficient to test triangle-freeness in graphs of size n with average degree d. We also give a simultaneous protocol using $\tilde{O}(k\sqrt{n})$ bits when $d = O(\sqrt{n})$ and $\tilde{O}(k(nd)^{1/3})$ when $d = \Omega(\sqrt{n})$. We show that for average degree $d = O(1)$, our simultaneous protocol is asymptotically optimal up to logarithmic factors. For higher degrees, we are not able to give lower bounds on testing triangle-freeness, but we give evidence that the problem is hard by showing that finding an edge that participates in a triangle is hard, even when promised that the graph is far from triangle-free.

1 INTRODUCTION

The field of property testing asks the following question: for a given property P, how hard is it to test whether an input *satisfies* P, or is ϵ*-far from* P, in the sense that an ϵ-fraction of its representation would need to be changed to obtain an object satisfying P? Property-testing has received extensive attention, including graph properties such as connectivity and bipartiteness [20], properties of Boolean functions (monotonicity, linearity, etc.), properties of distributions, and many others [15, 19, 32]. The usual model in which property-testing is studied is the *query* model, in which the tester cannot "see" the entire input, and accesses it by asking local queries, that is by only viewing a single entry in the object representation at a time. For example, for graphs represented by their adjacency matrix, the tester might ask whether a given edge is in the graph, or what is the degree of some vertex. The efficiency of a property tester is measured by the number of queries it needs to make. One can also distinguish between *oblivious* testers, which decide in advance on the set of queries, and *adaptive* testers, which decide on the next query after observing the answers to the previous queries. It is known that for many graph properties, one-sided oblivious testers are no more than quadratically more expensive than adaptive testers [22].

In this paper we study property testing from a different perspective, that of *communication complexity*. We focus on property testing for graphs, and we assume that the input graph is divided between several players, who must communicate in order to determine whether it satisfies the property or is far from satisfying it. Each player can operate on its own part of the input "for free", without needing to make queries; we charge only for the number of bits that the players exchange between them. This is on one hand easier than the query model, because players are not restricted to making local queries, and on the other hand harder, because the query model is centralized while here we are in a distributed setting. This leads us to questions such as: does the fact that players are not restricted to local queries make the problem easier, or even trivial? Which useful "building blocks" from the world of property testing can be implemented efficiently by multi-party protocols? Does *interaction* between the players help, or can we adopt the "oblivious approach" represented by *simultaneous* communication protocols?

Beyond the intrinsic interest of these questions, our work is motivated by two recent lines of research. First, [9, 18], study property testing in the CONGEST model, and show that many graph property-testing problems can be solved efficiently in the distributed setting. As pointed out in [18], existing techniques for proving lower bounds in the CONGEST model seem ill-suited to proving lower bounds for property testing. It seems that such lower bounds will require some advances on the communication complexity side, and in this paper we make initial steps in this direction. Second, recent work has shown that many *exact* problems are hard in the setting of multi-party communication complexity: Woodruff et al. [35] proved that for several natural graph properties, such as triangle-freeness, bipartiteness and connectivity, determining whether a graph satisfies the property essentially requires each player to send its entire input. We therefore ask whether weakening our requirements by turning to property testing can help.

In this paper we focus mostly on the specific graph property of *triangle-freeness*, an important property which has received a wealth of attention in the property testing literature. It is known that in dense graphs (average degree $\Theta(n)$) there is an *oblivious* tester for

PODC '17, July 25-27, 2017, Washington, DC, USA

ACM ISBN 978-1-4503-4992-5/17/07…$15.00
https://doi.org/10.1145/3087801.3087821

triangle-freeness which is asymptotically optimal in terms of the size of the graph (i.e., adaptivity does not help)[2, 17], and [3] also gives an oblivious tester for graphs with average degree $\Omega(\sqrt{n})$. The closest parallel to oblivious testers in the world of communication complexity is *simultaneous communication protocols*, where the players each send a single message to a referee, and the referee then outputs the answer. We devote special attention to the question of the simultaneous communication-complexity of testing triangle-freeness.

1.1 Our contributions

Our results are summarized in Table 1 below.

	$d = \Theta(1)$	$d = O(\sqrt{n})$	$d = \Omega(\sqrt{n})$
Unrestricted	$\tilde{O}(k\sqrt[4]{nd} + k^2)$		
Simultaneous	$\tilde{O}(k\sqrt{n})$, $\Omega(k\sqrt[6]{nd} + \sqrt[3]{nd})$		$\tilde{O}(k\sqrt[3]{nd})$
"Extended one-way"	$\Omega(\sqrt[6]{nd})$		$-$

Table 1: Summary of our results for the various models of communication complexity and settings of the average degree d.

Basic building-blocks. We show that many useful building-blocks from the property testing world can be implemented efficiently in the multi-player setting, allowing us to use existing property testers in our setting as well. For some primitives — e.g., sampling a random set of vertices — this is immediate. However, in some cases it is less obvious, especially when *edge duplication* is allowed (so that several players can receive the same edge from the input graph). We show that even with edge duplication the players can efficiently simulate a random walk, estimate the degree of a node, and implement other building blocks. For lack of space, this part of the paper is relegated to the appendix.

Upper bounds on testing triangle-freeness. For unrestricted communication protocols, we show that $\tilde{O}(k\sqrt[4]{nd}+k^2)$ bits are sufficient to test triangle-freeness, where n is the size of the graph, d is its average degree (which is not known in advance), and k is the number of players. When interaction is not allowed (simultaneous protocols), we give a protocol that uses $\tilde{O}(k\sqrt{n})$ bits when $d = O(\sqrt{n})$, and another protocol using $\tilde{O}(k\sqrt[3]{nd})$ bits for the case $d = \Omega(\sqrt{n})$. We also combine these protocols into a single *degree-oblivious protocol*, which does not need to know the average degree in advance. (This is not as simple as might sound, since we are working with *simultaneous* protocols, where we cannot first estimate the degree and then use the appropriate protocol for it.)

Lower bounds. Our lower bounds are mostly restricted to simultaneous protocols, although we first prove lower bounds for one-way protocols for two or three players, and then then "lift" the results to simultaneous protocols for $k \geq 3$ players using the symmetrization technique [30].

We show that for average degree $d = O(1)$, $\Omega(k\sqrt{n})$ bits are required to simultaneously test triangle-freeness, matching our upper bound. For higher degrees, we are not able to give a lower bound on testing triangle-freeness, but we give evidence that the problem is hard: we show that it is hard to *find an edge that participates in a triangle*, even in graphs that are ϵ-far from triangle free (for constant ϵ), and where every edge participates in a triangle with (small) constant marginal probability.

2 RELATED WORK

Property testing is an important notion in many areas of theoretical computer science; see the surveys [15, 19, 32] for more background.

Triangle-freeness, the problem we consider in this paper, is one of the most extensively studied properties in the world of property testing; many different graph densities and restrictions have been investigated (e.g., [1, 2, 4, 21]). Of particular relevance to us is triangle-freeness in the *general model* of property testing, where the average degree of the graph is known in advance, but no other restrictions are imposed. For this model, [3] showed an upper bound of $\tilde{O}(\min\{\sqrt{nd}, n^{\frac{4}{3}}/d^{\frac{2}{3}}\})$ on testing triangle-freeness, and a lower bound of $\Omega(\max\{\sqrt{n}/d, \min\{d, n/d\}, \min\{\sqrt{d}, n^{2/3}\}n^{-o(1)}\})$, both for graphs with a average degree d ranging from $\Omega(1)$ up to $n^{1-o(1)}$. For specific ranges of d, [31] and [23] improved these upper and lower bounds, by showing a $O(\max\{(nd)^{4/9}, n^{2/3}/d^{1/3}\})$ upper bound and a $\Omega(\min\{(nd)^{1/3}, n/d), n/d\})$ lower bound.

Our simultaneous protocols use ideas, and in one case an entire tester, from [3], but implementing them in our model presents different challenges and opportunities. (Our unrestricted-round protocol does not bear much similarity to existing testers.) As for lower bounds, we cannot use the techniques from [3] or other property-testing lower bounds, because they rely on the fact that the tester only has query access to the graph. For example, [3] uses the fact that a triangle-freeness tester with one-sided error must *find* a triangle before it can announce that the graph is far from triangle-free ([3] also gives a reduction lifting their results to two-sided error). In the communication complexity setting this is no longer true; there is no obvious reason why the players need to find a triangle in order to learn that the graph is not triangle-free.

Property testing in other contexts. Recently, property testing explored in distributed computing [6, 9, 18]. Among their other results, Censor-Hillel et al. [9] showed that triangle-freeness can be tested in $O(\frac{1}{\epsilon^2})$ rounds in the CONGEST model; expanding this,[18] showed that testing H-freeness for any 4 node graph H can be done in $O(\frac{1}{\epsilon^2})$ rounds, and showed that their BFS and DFS approach fails for K_5 and C_5-freeness, respectively; [18] does not give a general lower bound. There has also been work on property testing in the streaming model [24]. The related problem of computing the exact or approximate number of triangles has also been studied in many contexts, including distributed computing [10, 12, 13, 27], sublinear-time algorithms (see [14] and the references therein), and streaming (e.g., [25]). Specifically, [25] shows a lower bound on the space complexity of approximating the number of triangles in the streaming model; we apply their reduction here to show the hardness of testing triangle-freeness, by reducing from a different variant of the problem they use.

Communication complexity. Multi-party number-in-hand communication complexity has received significant attention recently. In [35] it is shown that several graph problems, including exact triangle-detection, are hard in this model. Many other exact and approximation problems have also been studied, including [5, 7, 8, 28, 34, 36] and others. Unfortunately, it seems that canonical lower bounds and techniques in communication complexity cannot be leveraged to obtain property-testing lower bounds.

3 PRELIMINARIES

Multi-player number-in-hand communication models. In *number-in-hand* multi-party communication protocols, we have k playerswith private inputs $X_1, \ldots, X_k$. The players communicate with each other to accomplish some shared goal.

There are three popular models for how the players communicate: the *blackboard model*, where communication is by broadcast, and a message by any player is seen by everyone; the *message-passing model*, where every two players have a private communication channel and each message has a specific recipient; and the *coordinator model*, where we have a special player called the *coordinator*, and players can communicate back-and-forth with the coordinator but not directly with each other. More formally, a protocol for the coordinator model is divided into communication rounds: in each such round, the coordinator sends a message of arbitrary size to one of the players, who then responds back with a message. Eventually the coordinator outputs the answer.

The *communication complexity* of a protocol Π, denoted $\mathrm{CC}(\Pi)$, is the maximum over inputs of the expected number of bits exchanged between the players and the coordinator in the protocol's run. For a problem P, we let $\mathrm{CC}_{k,\delta}(P)$ denote the best communication complexity of any protocol that solves P with worst-case error probability δ on any input.

In terms of communication complexity, the coordinator model is roughly equivalent to the message-passing model: any protocol for one model can be simulated in the other with a multiplicative overhead of $O(\log k)$ in the total communication cost.

For convenience, we assume that the players and the coordinator have access to *shared randomness* instead of private randomness; our protocols use shared randomness to simplify tasks like sampling a uniformly random vertex from the graph. For protocols that use more than one round, it is possible to get rid of this assumption and use private randomness instead, via Newman's Theorem [29], extended to multiple players. This costs at most additional $O(k \log n)$ bits.

Simultaneous communication. Of particular interest to us in this work are *simultaneous protocols*, which are, in a sense, the analog of oblivious property testers. This is the second primary model we investigate in addition to unrestricted communication. In a simultaneous protocol, there is only one communication round, where each player, after seeing its input, sends a *single message* to the coordinator (usually called the *referee* in this context). The coordinator then outputs the answer. Any oblivious graph property tester which uses only edge queries (which test whether a given edge is in the graph or not) can be implemented by a simultaneous protocol, but the converse is not necessarily true.

Communication complexity of property testing in graphs. we are given a graph $G = (V, E)$ on n vertices, which is divided between the k players, with each player j receiving some subset $E_j \subseteq E$ of edges. More concretely, each player, j, receives the characteristic vector of E_j, where each entry corresponds to a single edge, such that if the bit is 1 then that edge exists in E, and if the bit is 0 it is unknown to the player whether it exists or not, as this entry might be 1 in the input of a different player. The logical OR of all inputs results in the characteristic vector of the graph edges, E. Note that there is no guarantee for any vertex for a single player to have all its adjacent edges in its input, as is the case in models like CONGEST. To make our results as broad as possible we follow the *general model* of property testing in graphs (see, e.g., [3]): we do not assume that the graph is regular or that there is an upper bound on the degree of individual nodes. As in [35], edges may be duplicated, that is, the sets $E_1, \ldots, E_k$ are not necessarily disjoint.

The goal of a property tester for property P is to distinguish the case where G satisfies P from the case where G is ϵ-far from satisfying P, that is, at least $\epsilon|E|$ edges would need to be added or removed from G to obtain a graph satisfying P. An important parameter in our algorithms is the average degree, d, of the graph (also referred to as *density*); for our upper bounds, we do not assume that d is known, but our lower bounds can assume that it is known to the protocol up to a tight multiplicative factor of $(1 \pm o(1))$. Moreover, as in [3], we focus on $d = \Omega(1)$ and $d \leq n^{1-\nu(n)}$, where $\nu(n) = o(1)$, since for graphs of average degree $d = \Theta(n)$ there is a known solution whose complexity is independent on n in the property-testing query model and consequently in our model as well. The case of $d = o(1)$, although not principally different, is ignored for simplicity, as its extreme sparsity makes it of less interest than any degree which is $\Omega(1)$.

Graph definitions and notation. We let $\deg(v)$ denote the degree of a vertex v in the input graph, and for a player $j \in [k]$, we denote by $d^j(v)$ the degree of v in player j's input (the subgraph (V, E_j)). We also let $\deg_S(v)$ denote the degree of vertex v with respect to a subset $S \subseteq E$ of the graph edges.

We say that a pair of edges $\{\{u,v\}, \{v,w\}\} \subseteq E$ is a *triangle-vee* if $\{u,w\} \in E$, and in this case we call v the *source* of the triangle-vee. We say that an edge $e \in E$ is a *triangle edge* if it participates in some triangle in the graph.

4 UPPER BOUNDS

4.1 Unrestricted Communication

The first protocol we present requires *interaction* between the players, and exploits the following advantage we have over the query model: suppose that the players have managed to find a set $S \subseteq E$ of edges that contains a "triangle-vee" — a pair of edges $\{u,v\}, \{v,w\} \in S$ such that $\{u,w\} \in E$ (but $\{u,w\}$ is not necessarily in S). Then even if S is very large, the players can easily conclude that the graph contains a triangle: each player examines its own input and checks if it has an edge that closes a triangle together with some vee in S, and in the next round informs the other players. Thus, in our model, finding a triangle boils down to finding a triangle-vee. (In contrast, in the query model we would need to query $\{u,w\}$ for every 2-path $\{u,v\}, \{v,w\} \in S$, and this could be expensive if S is large.) Notice that if G is ϵ-far from triangle-free, then it must contain at least $(\epsilon nd/3)$ pairwise disjoint triangles, each contributing three unique triangle-vees, hence the graph contains at least ϵnd different triangle-vees.

Let us say that a vertex v is *full* if it is the source of at least $\Omega(\epsilon \deg(v)/\log n)$ different triangle-vees. (Throughout this section, asymptotic notation is used for clarity, see full paper for concrete constants. [16]) If we can find a full vertex v, then we can use it to find a triangle-vee:

Lemma 4.1. *If v is a full vertex, then sampling each of its edges with probability $p_{d(v)} = \Theta(\sqrt{\log n / d(v)})$ will reveal a triangle-vee with constant probability.*

(This follows from the birthday paradox.) Note that $\deg(v)$ may be significantly higher than the average degree d in the graph, so we cannot necessarily afford to sample each of v's edges with probability $p_{d(v)}$; we need to find a *low-degree* vertex which is full. However, because there cannot be many nodes with "very high degree" ($\Omega(\sqrt{nd})$), this is not a major concern.

How can the players find a full vertex? A uniformly random vertex is not always likely to be full — there might be a small dense subgraph of relatively high-degree nodes which contains all the triangles. In order to target such dense subgraphs, we use *bucketing*: we partition the vertices into buckets, with each bucket B_i containing the vertices with degrees in the range $[2^i, 2^{i+1})$[1]. By a pigeonhole argument, there is some bucket B_i whose nodes are the sources of at least $\epsilon nd/\log n$ different triangle-vees. The size of B_i cannot exceed $nd/2^i$, as each node in B_i contributes at least 2^i edges, and the graph contains a total of nd edges. Therefore, the average vertex in B_i is the source of $\Omega(\epsilon 2^i/\log n)$ different triangle-vees, and since the degree in B_i is bounded by 2^{i+1}, this means the average vertex in B_i is full. So if we sample a uniform vertex $v \in B_i$, and sample sufficiently many neighbors of v, according to Lemma 4.1 we stand a good chance of uncovering a triangle-vee, and the protocol will then identify a triangle in the next round. Of course, we cannot know in advance which bucket is full; we must try all the buckets.

It remains to describe, given B_i is a full bucket, how we can sample a random vertex from it, that is, a random vertex with degree in the range $[2^i, 2^{i+1})$. We cannot do that, precisely, but we can come close. Because the edges are divided between the players, no single player initially knows the degree of any given vertex. However, by the pigeonhole principle, for each vertex v there is some player that has at least $\deg(v)/k$ of v's edges, and of course no player has more than $\deg(v)$ edges for v.

Let $\tilde{B}_i^j := \{v \in V \mid 2^i/k \le d^j(v) \le 2^{i+1}\}$ be the set of vertices that player j can "reasonably suspect" belong to bucket i, where d^j denotes the degree of vertex v in the input of player j, and let $\tilde{B}_i := \bigcup_j \tilde{B}_j$. By the argument above, $B_i \subseteq \tilde{B}_i$. Also, $\tilde{B}_i \subseteq \bigcup_{i'=i-\log k}^{i+\log k} B_{i'}$, since the total degree of any vertex selected cannot exceed $k \cdot 2^{i+1}$ or be smaller than $2^i/k$. Therefore, sampling uniformly from $\tilde{B}_i$ is a good proxy for sampling from B_i, although we may also hit adjacent buckets. Nevertheless, since those adjacent buckets are roughly comparable to B_i in size (up to a factor of k), a uniformly random sample will yield a vertex from B_i with probability at least $\tilde{\Omega}(1/k)$, and $\tilde{\Theta}(k)$ samples yield a vertex from B_i w.h.p.

To implement the sampling procedure we need two components: first, we need to be able to sample uniformly from $\tilde{B}_i$. The difficulty here is that each vertex $v \in \tilde{B}_i$ can be known to a different number of players — possibly only one player j has $v \in \tilde{B}_i^j$, possibly all players do. If we try a naive approach, such as having each player j post a random sample from $\tilde{B}_i^j$, then our sample will be biased in favor of vertices that belong to $\tilde{B}_i^j$ for many players j. Our solution is to impose a random order on the nodes in $\tilde{B}_i$ by publicly sampling a permutation π on V (this is done by interpreting the shared random bits as a random permutation), and we then choose the *smallest* node in $\tilde{B}_i$ with respect to π. This yields a uniformly random sample, unbiased by the number of players that know of a given node. We call this procedure TrySampleFromBucket(i) (the code appears in full paper [16]).

The second component verifies that a sampled node indeed belongs to B_i. We approximate the degree of the sampled node to within a constant, and discard vertices whose degree does not match bucket B_i (we might falsely keep vertices from its two adjacent buckets, but this increases the sampled set by at most a small constant). We call this procedure ApproxDegree(v).

The protocol for player j is sketched in Algorithm 1. Here $N = \tilde{\Theta}(k)$ is the number of samples from $\tilde{B}_i$ required to produce a sample from B_i with good probability. After the procedure described in Algorithm 1, the coordinator sends all the edges he received to all the players, and the players then check their own inputs for an edge that closes a triangle with some triangle-vee sent by the coordinator. With high probability, a triangle-vee *is* discovered, and the protocol ends in the next round.

Algorithm 1 Code for player j

```
For each i = 0, ..., log n:
ℓ ← 0
Repeat until ℓ ≥ N:
    v ← TrySampleFromBucket(i)
    d̄(v) ← ApproxDegree(v)
    If d⁻(B_i)/√3 ≤ d̄(v) ≤ √3 d⁺(B_i):
        ℓ ← ℓ + 1
        Jointly generate a public random set S ⊆ V, where
each u ∈ S with iid probability p_d̄(v)
        Send E_j ∩ ({v} × S) to the coordinator
```

Theorem 4.2. *The communication complexity in the model of unrestricted communication of triangle detection in graphs of average degree d, that are ϵ-far from being triangle free, is $\tilde{O}(k \cdot \sqrt[4]{nd} + k^2)$.*

4.2 Simultaneous Protocols

In the simultaneous model, the players cannot interact with each other — they send only one message to the referee, and the referee then outputs the answer. This rules out our previous approach, as exposing a triangle-vee does not help us if the players cannot then check their inputs for an edge that completes the triangle. Indeed, the simultaneous model is closer to the query model in spirit. Accordingly, we will use the triangle-freeness testers of [3], but show that we can implement them more efficiently in our model. Moreover, we achieve roughly the same complexity without knowing the average degree in advance.

[1]We use in the appendix $[3^i, 3^{i+1})$ for technical reasons, although both are correct

We begin by describing how to adapt the testers of [3] to our model when the average degree is known. We then sketch the main ideas for dealing with an unknown degree; the details appear in the appendix.

High-degree graphs. For graphs with average degree $\Omega(\sqrt{n})$, the tester from [3] samples a uniformly random set $S \subseteq V$ of $\Theta(\sqrt[3]{n^2/d})$ vertices, queries all edges in S^2, and checks if the subgraph exposed contains a triangle. It is shown in [3] that if the graph is ϵ-far from triangle-free, then the subgraph induced by S will contain $\Theta(1)$ triangles in expectation, and the variance is small enough to ensure small error.

We can implement this tester easily, and in our model it is less expensive: instead of querying all pairs in S^2, the players simply send all the edges from S^2 in their input, paying only for edges that exist and not for edges that do not exist in the graph. The set S is large enough that the number of edges in the subgraph does not deviate significantly from its expected value, $\Theta\left((nd)^{1/3}\right)$. In Section 5 we show that for average degree $\Theta(\sqrt{n})$ this is tight for 3 players.

Low-degree graphs. For density $d = o(\sqrt{n})$, the approach above no longer works, as the variance is too large. To illustrate this, consider a graph with d vertices of degree $\Theta(n)$, which are the sources of $\Theta(nd)$ triangle-vees, and the remaining nodes are all of constant degree. If we sample vertices uniformly at random, we need to sample $\Theta(n/d)$ vertices in order to find one of the d sources, hence we need at least $\Omega(n/d)$ to find a triangle. However, whereas in the query model we would need to make $\Theta(n^2/d^2)$ queries to learn the entire subgraph induced by the set we sampled, in our model we proceed as follows (using ideas from [3], which require adaptivity there, and deploying them in a different way): let S be the set of $\Theta(n/d)$ uniformly-random vertices. We sample another set, R, of $\Theta(\sqrt{n})$ vertices, and we send all edges in $R \times (S \cup R)$. If indeed there is a small set of high-degree vertices participating in most of the triangles, then with good probability we will have one of them in S, and by the birthday paradox, one of its triangles will have its other two vertices in R. On the other hand, if the triangles are spread out "evenly", then the subgraph $R \times R$ will probably contain one. The expected size of $R \times (S \cup R)$ is $O(\sqrt{n})$, and we show that w.h.p. the total communication is $\tilde{O}(k\sqrt{n})$.

Note that both our solutions work for $d = \Theta(\sqrt{n})$, and for this density they are essentially the same: both sets, S and R, are of size $\Theta(n/d) = \Theta(d) = \Theta\left((nd)^{1/3}\right)$, so for $d = \Theta(\sqrt{n})$ the second protocol is not very different from the first. We can also show that if edge duplication is not allowed, a factor of k is saved in the communication complexity with high probability.

A degree-oblivious protocol.
Finally, let us give a high-level overview of how we combine the protocols above and modify them so that they can be used without advance knowledge of the degree. The challenge here is that no single player can get a good estimate of the degree from their input, and since the protocol is simultaneous, the players must decide what to do without consulting each other. The natural approach is to use $\log n$ exponentially-increasing "guesses" for the density, covering the range $[1, n]$, and try them all; however, if we do this we will incur a high cost for guesses that imply examining a larger sample than needed. We therefore take a more fine-grained approach.

Our first observation is that some players *can* make a reasonable estimate of the global density, although they do not *know* that they can. Let $\bar{d}^i$ denote the average degree in player i's input E_i, and let us say that player i is *relevant* if $\bar{d}^i \geq (\epsilon/(4k))d$. If we eliminate all the irrelevant players and their inputs, the graph still remains $(\epsilon/2)$-far from triangle-free, so we can afford to ignore the irrelevant players in our analysis — except for making sure that their messages are not too large.

Since players cannot know if they are relevant, all players assume that they are. Based on the degree $\bar{d}^i$ that player i observes, it knows that *if* it is relevant, then the average degree in the graph is in the range $D_i = [\bar{d}^i, \Theta(k\bar{d}^i)]$. We fix in advance an exponential scale $\left\{2^j\right\}_{j=0}^{\log n}$ of guesses for the density, and execute in parallel $\log n$ instances of triangle-freeness protocols, one for each degree 2^j. However, each player i only participates in instances corresponding to density guesses that fall in D_i, and sends nothing for the other instances. For relevant players, we know that the true density falls in their range D_i, so they will participate in the "correct" instance. For irrelevant players, we do not care, and their message size is also not an issue: their density estimate is too low, and the communication complexity of each instance increases with the density it corresponds to.

If we are not careful, we may still incur a blow-up of $k^{1/3}$ in communication, as relevant players may use guesses lower than the true density by a factor of k, which increases the size of the sample beyond what is necessary. However, by carefully assigning each player i a communication budget depending on $\bar{d}^i$, we can eliminate the blow-up, and match the degree-aware protocol up to polylogarithmic factors.

Theorem 4.3. *The simultaneous communication complexity of testing triangle-freeness in graphs of average degree d is* $\tilde{O}(k \max\left\{\sqrt{n}, (nd)^{1/3)}\right\})$, *even if d is not known to the players.*

5 LOWER BOUNDS

5.1 Testing Triangle-Freeness with $d = \Theta(1)$

For graphs of average degree $\Theta(1)$, we show that for a sufficiently small constant ϵ, *one-sided* two-player protocols require $\Omega(\sqrt{n})$ communication to distinguish triangle-free graphs from graphs that are ϵ-far from triangle-free. We use the same reduction used in [25] to show the hardness of approximating the number of triangles in the streaming model, but we reduce from a different variant of the Boolean Hypermatching problem [26, 33]. Using symmetrization, we then lift the bound to a lower bound of $\Omega(k\sqrt{n})$ for k-player simultaneous protocols. The details appear in the full version of the paper [16].

5.2 Finding a Triangle Edge in ϵ-far from Triangle-Free Graphs

Our main result in this section is the following:

Theorem 5.1. *For any $d = O(\sqrt{n})$, let $T^{\epsilon}_{n,d}$ be the task of finding a triangle edge in graphs of size n and average degree d which are*

ϵ-far from triangle-free. Then for sufficiently small constant error probability δ we have:

(1) For $k > 3$ players: $\mathrm{CC}^{sim}_{k,\delta}(T^{\epsilon}_{n,d}) = \Omega\left(k(nd)^{1/6}\right)$.

(2) For 3 players: $\mathrm{CC}^{sim}_{3,\delta}(T^{\epsilon}_{n,d}) = \Omega\left((nd)^{1/3}\right)$.

To show both results, we first prove them for average degree $d = \Theta(\sqrt{n})$, and then easily obtain the result for lower degrees by embedding a dense subgraph of degree $\Theta(\sqrt{n})$ in a larger graph with lower overall average degree and many isolated nodes. For lack of space, we describe here the lower bounds for graphs of average degree $\Theta(\sqrt{n})$, and defer the embedding into lower degree graphs to the full version of the paper.

To prove (1), we begin by proving that for graphs of average degree $\Theta(\sqrt{n})$, *three* players require $\Omega(n^{1/4})$ bits of communication to solve $T^{\epsilon}_{n,\sqrt{n}}$ in the *one-way* communication model, where Alice and Bob send messages to Charlie, and then Charlie outputs the answer. In fact, our lower bound is more general, and allows Alice and Bob to communicate back-and-forth for as many rounds as they like, with Charlie observing the transcript. We then "lift" the result to $k > 3$ players communicating simultaneously, using symmetrization [30].

To prove (2), we show directly that in the simultaneous communication model, three players require $\Omega(\sqrt{n})$ bits to solve $T^{\epsilon}_{n,d}$ in graphs of average degree $\Theta(\sqrt{n})$.

Our lower bounds actually bound the *distributional* hardness of the problems: we show an input distribution μ on which any protocol that has a small probability of error *on inputs drawn from* μ requires high communication. This is stronger than worst-case hardness, which would only assert that any protocol that has small error probability *on all inputs* requires high communication.

The hard distribution. We use a simple distribution μ to generate graphs that with high probability have average degree $\Theta(\sqrt{n})$ and are ϵ-far from triangle-free: we let $G = (U \cup V_1 \cup V_2, \mathbf{E})$ be a tripartite graph, where $|U| = |V_1| = |V_2| = n$, and each edge of the tripartite graph is included in the edge set $\mathbf{E}$ iid with probability $\gamma/\sqrt{n}$, for a small constant $\gamma \in (0,1)$. We give each player the edges on one side of the graph: Alice receives $\mathbf{E}_1 = \mathbf{E} \cap (U \times V_1)$, Bob receives $\mathbf{E}_2 = \mathbf{E} \cap (U \times V_2)$, and Charlie receives $\mathbf{E}_3 = \mathbf{E} \cap (V_1 \times V_2)$.

Clearly the distribution μ is not *guaranteed* to produce graphs with average degree $\Theta(\sqrt{n})$, or graphs that are ϵ-far from triangle-free, but we show that it does so with constant probability:

Lemma 5.2. *When γ is sufficiently small, a graph sampled from μ is $O(1)$-far from triangle-free with probability at least $1/2$.*

Therefore, any protocol that succeeds with constant probability on graphs that have average degree $\Theta(\sqrt{n})$ and are ϵ-far from triangle-free, must also succeed with (smaller) constant probability on μ.

5.3 Information theory

Our lower bounds use information theory to argue that using a small number of communication bits, the players cannot convey much information about their inputs. For lack of space, we give here only the essential definitions and properties we need.

Let $(\mathbf{X}, \mathbf{Y}) \sim \mu$ be random variables. (For clarity, from now on we denote random variables with bold-face letters.) To measure the information we learned about $\mathbf{X}$ after observing $\mathbf{Y}$, we examine the difference between the *prior* distribution of $\mathbf{X}$, denoted $\mu(\mathbf{X})$, and the *posterior* distribution of $\mathbf{X}$ after seeing $\mathbf{Y} = y$, which we denote $\mu(\mathbf{X}|\mathbf{Y} = y)$. We use KL divergence to quantify this difference:

Definition 5.3 (KL Divergence). For distributions $\mu, \eta : \mathcal{X} \to [0,1]$, the *KL divergence* between μ and η is

$$\mathrm{D}(\mu \parallel \eta) := \sum_{x \in \mathcal{X}} \mu(x) \log(\mu(x)/\eta(x)).$$

We require the following property, which follows from the superadditivity of information [11]: if $(\mathbf{X}_1, \ldots, \mathbf{X}_n, \mathbf{Y}) \sim \mu$ are such that $\mathbf{X}_1, \ldots, \mathbf{X}_n$ are independent, and $\mathbf{Y}$ can be represented using m bits (that is, its *entropy* is at most m), then

$$\mathop{\mathrm{E}}_{y \sim \mu(\mathbf{Y})}\left[\sum_{i=1}^{n} \mathrm{D}(\mu(\mathbf{X}_i|\mathbf{Y} = y) \parallel \mu(\mathbf{X}_i))\right] \le m.$$

We use superadditivity as follows: let $\{\mathbf{X}_e\}$ be a collection of indicators for the presence of edge e in the input. In the input distribution μ defined above, the edges appear independently, so we can use superadditivity to argue that after observing the transcript $\mathbf{\Pi}$ of the protocol, the *sum* of what we learn about each edge is at most $\mathrm{CC}(\Pi)$, the communication complexity of the protocol. In particular, if $\mathrm{CC}(\Pi)$ is small, then the average edge in the input has a *posterior probability* given the transcript that is close to its *prior probability* of appearing, $\gamma/\sqrt{n}$.

5.4 "Extended One-Way" 3-Player Protocols

Suppose we have a protocl Π for three players, where Alice and Bob communicate back-and-forth for some number of rounds, and eventually Charlie, having observed the conversation between Alice and Bob, outputs an edge. If Charlie outputs an edge that is not in $\mathbf{E}_3$, this is an error, so let us assume for simplicity that he never does this (the full proof does not make this assumption).

Let π be the distribution of Π's transcripts on inputs drawn from μ, and let $\mathbf{\Pi} \sim \pi$ be a random variable for Π's transcript. Given a transcript t, let $\mu|t$ denote the conditional distribution of the input given $\mathbf{\Pi} = t$.

Outline of the lower bound. Intuitively, when Charlie outputs an edge $e = \{v_1, v_2\} \in \mathbf{E}_3$, he should *believe* that e is part of a triangle, that is, he should "think" that there is a node $u \in U$ such that $\{u, v_1\} \in \mathbf{E}_1$ and $\{u, v_2\} \in \mathbf{E}_2$. We say that such an edge $e \in V_1 \times V_2$ is *covered*. Since Charlie cannot see $\mathbf{E}_1$ or $\mathbf{E}_2$, the set of covered edges is a function of the messages sent by Alice and Bob; we let $\mathrm{C}(t) \subseteq V_1 \times V_2$ denote the set of covered edges when t is the transcript of Alice and Bob's communication. The edge output by Charlie should be an edge in $\mathrm{C}(t) \cap \mathbf{E}_3$, that is, an edge in Charlie's input that he believes is part of a triangle. The crux of our lower bound consists of showing that if Alice and Bob use only $O(n^{1/4})$ bits of communication, then typically we have $|\mathrm{C}(t)| = O(\sqrt{n})$; and since each edge in $\mathrm{C}(t)$ appears in $\mathbf{E}_3$ independently with probability $\gamma/\sqrt{n}$, the probability that $\mathrm{C}(t) \cap \mathbf{E}_3 \neq \emptyset$ is too small, leaving Charlie with no good edge to output.

Covered edges. We begin by formalizing what it means for Charlie to "believe" that edge $\{v_1, v_2\}$ is part of a triangle.

For $v_1 \in V_1, v_2 \in V_2$, let $\mathsf{Vee}(\{v_1, v_2\})$ denote the event that there is a node $u \in U$ such that $\{u, v_1\} \in \mathbf{E}_1$ and $\{u, v_2\} \in \mathbf{E}_2$. We call $\{u, v_1\}, \{u, v_2\}$ a *vee supported on* $\{v_1, v_2\}$. Also, let $\mathsf{Tri}(e)$ be the event that e participates in a triangle; then $\mathsf{Tri}(e) = (e \in \mathbf{E}_3) \wedge \mathsf{Vee}(e)$, that is, e is in a triangle whenever it is in Charlie's input and there is a vee supported on it.

We formally capture Charlie's "belief" using the *posterior probability* that a vee exists, given the transcript that Charlie observed.

Definition 5.4 (Covered edges). We say that $e \in V_1 \times V_2$ is *covered* by transcript t if $\Pr_{\mathbf{E}\sim\mu|t}[\mathsf{Vee}(e)] \geq 1/10$, and let $\mathrm{C}(t)$ be the set of edges covered by t.

(The constant $1/10$ should, in general, depend on the error probability δ of the protocol, but here we use a specific value for clarity and assume that γ and δ are small enough.)

Now let us show that with good probability Alice and Bob provide Charlie with $\Omega(\sqrt{n})$ covered edges: we show that whenever $|\mathrm{C}(t)| < \alpha\sqrt{n}$ for an appropriately chosen constant α, the protocol incurs a large probability of error.

Recall that that the inputs $\mathbf{E}_1, \mathbf{E}_2$ and $\mathbf{E}_3$ to the three players are independent a-priori; by the properties of a communication protocol, they remain independent conditioned on the transcript, $\mathbf{\Pi} = t$. Thus, each edge in $\mathrm{C}(t)$ appears in $\mathbf{E}_3$ iid with probability $\gamma/\sqrt{n}$. By Markov, whenever $|\mathrm{C}(t)| < \alpha\sqrt{n}$, we have

$$\Pr_{\mathbf{E}\sim\mu|t}[\mathrm{C}(t) \cap \mathbf{E}_3 \neq \emptyset] \leq (\gamma/\sqrt{n}) \cdot (\alpha\sqrt{n}) = \alpha\gamma.$$

This implies that

$$\Pr_{\mathbf{E}\sim\mu}\left[\mathrm{C}(t) \cap \mathbf{E}_3 = \emptyset \,\middle|\, (|\mathrm{C}(t)| < \alpha\sqrt{n})\right] \geq 1 - \alpha\gamma.$$

Since we assumed that Charlie always outputs an edge from his input, whenever $\mathrm{C}(t) \cap \mathbf{E}_3 = \emptyset$, Charlie outputs an edge that is not covered, $e \notin \mathrm{C}(t)$. By definition, if $e \notin \mathrm{C}(t)$,

$$\Pr_{\mathbf{E}\sim\mu|t}[\mathsf{Tri}(e)] \leq \Pr_{\mathbf{E}\sim\mu|t}[\mathsf{Vee}(e)] < 9/10,$$

and therefore,

$$\begin{aligned}
\delta &\geq \Pr_{\pi}[\Pi \text{ errs}] \\
&\geq \Pr_{t\sim\pi}[\Pi \text{ errs} \mid \mathrm{C}(t) \cap \mathbf{E}_3 = \emptyset] \cdot \Pr_{t\sim\pi}[\mathrm{C}(t) \cap \mathbf{E}_3 = \emptyset] \\
&\geq (1/10) \cdot \Pr_{t\sim\pi}[\mathrm{C}(t) \cap \mathbf{E}_3 = \emptyset] \\
&\geq (1/10)\Pr\left[\mathrm{C}(t) \cap \mathbf{E}_3 = \emptyset \,\middle|\, |\mathrm{C}(t)| < \alpha\sqrt{n}\right] \cdot \Pr\left[|\mathrm{C}(t)| < \alpha\sqrt{n}\right] \\
&\geq ((1-\alpha\gamma)/10)\Pr\left[|\mathrm{C}(t)| < \alpha\sqrt{n}\right].
\end{aligned}$$

Assuming that $1 - \alpha\gamma \geq 1/2$, we see that with probability at least $1 - 20\delta$ we have $|\mathrm{C}(t)| \geq \alpha\sqrt{n}$. Next we show that this requires $\Omega(n^{1/4})$ bits of communication from Alice and Bob.

The cost of covering $\Omega(\sqrt{n})$ edges. Let us say that transcript t is *good* if $|\mathrm{C}(t)| \geq \alpha\sqrt{n}$. For simplicity, we assume that no transcript covers *more* than $\lceil\alpha\sqrt{n}\rceil$ edges (if it does, we remove some covered edges from $\mathrm{C}(t)$ arbitrarily; this does not affect the math above).

What do Alice and Bob need to do to raise the posterior probability of $\mathsf{Vee}(e)$? Since Π is a communication protocol, and $\mathbf{E}_1, \mathbf{E}_2$ are independent a-priori, they remain independent given the transcript, that is, given $\mathbf{\Pi} = t$ for any t. Therefore,

$$\begin{aligned}
\Pr_{\mathbf{E}\sim\mu|t}[\mathsf{Vee}(\{v_1, v_2\})] &\leq \sum_{u\in U}\Pr_{\mathbf{E}\sim\mu|t}[\{u, v_1\} \in \mathbf{E}_1 \wedge \{u, v_2\} \in \mathbf{E}_2] \\
&\leq \sum_{u\in U}\left(\Pr_{\mathbf{E}\sim\mu|t}[\{u, v_1\} \in \mathbf{E}]\Pr_{\mathbf{E}\sim\mu|t}[\{u, v_2\} \in \mathbf{E}]\right). \qquad (1)
\end{aligned}$$

Now, let

$$\Delta_t(\{w_1, w_2\}) := \Pr_{\mathbf{E}\sim\mu|t}[\{w_1, w_2\} \in \mathbf{E}] - 2\gamma/\sqrt{n}$$

denote the *increase* from *twice* the prior probability that $\{w_1, w_2\} \in \mathbf{E}$, which is $\gamma/\sqrt{n}$, to the posterior probability that $\{w_1, w_2\} \in \mathbf{E}$ given $\mathbf{\Pi} = 1$. (The reason we need to take twice the prior probability will be revealed shortly.) We can re-write (1) as

$$\begin{aligned}
&\Pr_{\mathbf{E}\sim\mu|t}[\mathsf{Vee}(\{v_1, v_2\})] \leq \\
&\quad\leq \sum_{u\in U}\left[\left(\Delta_t(\{u, v_1\}) + 2\gamma/\sqrt{n}\right) \cdot \left(\Delta_t(\{u, v_2\}) + 2\gamma/\sqrt{n}\right)\right].
\end{aligned}$$

Recall that if t is good, then $\alpha\sqrt{n} \leq |\mathrm{C}(t)| \leq \lceil\alpha\sqrt{n}\rceil$, and each edge $e \in \mathrm{C}(t)$ has posterior probability of $\mathsf{Vee}(e)$ at least $1/10$. Thus, for a good transcript t, summing over the edges in $\mathrm{C}(t)$ and using Cauchy-Schwarz yields

$$\begin{aligned}
\left(1/10 - \gamma^2\right)\alpha\sqrt{n} &\leq \left(\sum_{e\in U\times(V_1\cup V_2)}\Delta_t(e)\right)^2 \\
&\quad + (2\gamma/\sqrt{n}) \cdot \lceil\alpha\sqrt{n}\rceil\left(\sum_{e\in U\times(V_1\cup V_2)}\Delta_t(e)\right) \\
&\leq 2\left(\sum_{e\in U\times(V_1\cup V_2)}\Delta_t(e)\right)^2.
\end{aligned}$$

(On the right-hand side we simplified the expression and added some edges that are not necessarily in $\mathrm{C}(t)$.)

We see that if t is a good transcript, then we must have

$$\sum_{e\in U\times(V_1\cup V_2)}\Delta_t(e) \geq \Omega(n^{1/4}). \qquad (2)$$

Using information theory, we can show that $\Omega(n^{1/4})$ bits of communication are required to achieve this with constant probability.

Let $\mathbf{X}_e$ be an indicator for the event that edge e is in the input, $e \in \mathbf{E}$. Because the edges appear independently of each other, we can use the superadditivity of information to get:

$$\mathrm{CC}(\Pi) \geq \sum_e \mathop{\mathrm{E}}_{t\sim\pi}\left[\mathrm{D}\left(\mu|t(\mathbf{X}_e) \parallel \mu(\mathbf{X}_e)\right)\right]. \qquad (3)$$

In order to relate the KL divergence to Δ_t, we prove the following lemma:

Lemma 5.5. *Let $p, q \in (0, 1)$, and let $\mathrm{D}(p \parallel q)$ denote the KL divergence between* Bernoulli(p) *and* Bernoulli(q). *Then for any $q < 1/2$ we have $\mathrm{D}(p \parallel q) \geq p - 2q$.*

(This relationship is, in general, *quadratic*: Pinsker's inequality asserts that $\mathrm{D}(p \parallel q) \geq |p - q|$, and this is known to be tight in some cases. However, quadratic behavior occurs only when the prior p is close to $1/2$; our lemma shows that for small priors the relationship is roughly linear.)

Plugging Lemma 5.5 into (3) yields $\mathrm{CC}(\Pi) \geq \sum_e \mathrm{E}_t [\Delta_t(e)]$. Together with (2) and the fact that the transcript must be good with probability at least $1 - 20\delta$, we see that $\Omega(n^{1/4})$ bits of communication are required.

For this particular input distribution, the lower bound is tight: suppose Alice and Bob choose in advance some vertex $u \in U$, and each of them sends Charlie $O(n^{1/4})$ nodes adjacent to u in their input (with high probability, the degree of u is $\Theta(\sqrt{n})$, so they have enough edges to send). This has the effect of covering $\Theta(\sqrt{n})$ edges in $V_1 \times V_2$ — for every combination v_1, v_2 of nodes sent by Alice and Bob respectively, the edge $\{v_1, v_2\}$ is covered. With good probability Charlie will find one of the covered edges in his input and he can then output it (and also the triangle containing it).

5.5 Simultaneous 3-Player Lower Bound

We now turn our attention to *simultaneous* protocols, where we have a referee who does not see any of the input; Alice, Bob and Charlie send messages to the referee, and the referee then outputs the answer. We assume for simplicity that the edge output is always from Charlie's side of the graph (this is not essential).

In our one-way lower bound we argued that Alice and Bob must provide Charlie with $\Omega(\sqrt{n})$ edges that "look like" there is a vee supported on them, so that w.h.p. Charlie will find one of those edges in his input. For simultaneous protocols, this is not enough: the referee does not see Charlie's input, and cannot know if a given edge is in $\mathbf{E}_3$ or not unless Charlie "tells him". Accordingly, we define the set $\mathrm{Rep}(t)$ of edges "reported by Charlie" to be the set of edges in $V_1 \times V_2$ with good posterior probability of being in $\mathbf{E}_3$ given $\mathbf{\Pi} = t$. Moreover, because there is no interaction, the referee cannot look at the set of covered edges and *ask* Charlie which of them is in $\mathbf{E}_3$; Alice, Bob and Charlie need to provide the referee with sets $\mathrm{C}(t)$ and $\mathrm{Rep}(t)$ such that with high probability $\mathrm{C}(t) \cap \mathrm{Rep}(t) \neq \emptyset$, and the referee can then output an edge that has good probability of being in a triangle.

More formally, let $t = (m_1, m_2, m_3)$ be the transcript, where m_1, m_2 and m_3 are the messages of Alice, Bob and Charlie, respectively. We define

$$\mathrm{Rep}(t) = \left\{ e \in V_1 \times V_2 \mid \Pr_{\mathrm{E} \sim \mu | t} [\mathbf{X}_e = 1] \geq 1/10 \right\}.$$

In a simultaneous protocol, the messages sent by the players are independent of each other given the input. In our case, because the inputs are *also* independent of each other, the messages are independent even without conditioning on a particular input. We therefore abuse notation slightly by omitting parts of the transcript that are not relevant to the event at hand; we write $\mathrm{Rep}(m_1)$ for $\mathrm{Rep}(t)$ and $\mathrm{C}(m_1, m_2)$ for $\mathrm{C}(()\, t)$.

As we showed in the one-way case, whenever the edge output is not in $\mathrm{C}(m_1, m_2)$ we incur a large contribution to the error of the protocol. We can now show a similar claim for $\mathrm{Rep}(m_3)$ as well: if the referee outputs an edge with low posterior probability of being in the input, then the error probability is high. Combining, we get:

Lemma 5.6. *The probability that there exists an edge that is both reported by Charlie and covered by Alice and Bob is at least* $1 - 100\delta$. *That is,*

$$\Pr_{(\mathbf{m}_1, \mathbf{m}_2, \mathbf{m}_3) \sim \pi} [\mathrm{Rep}(\mathbf{m}_3) \cap \mathrm{C}(\mathbf{m}_1, \mathbf{m}_2) \neq \emptyset] \geq 1 - 100\delta.$$

Because the inputs and messages are independent, for a fixed message m_3 of Bob, we get by union bound that

$$\Pr_{\mathbf{m}_1, \mathbf{m}_2 \sim \pi} [\mathrm{Rep}(m_3) \cap \mathrm{C}(\mathbf{m}_1, \mathbf{m}_2) \neq \emptyset] \leq \sum_{e \in \mathrm{Rep}(m_3)} \Pr_{\mathbf{m}_1, \mathbf{m}_2 \sim \pi} [e \in \mathrm{C}(\mathbf{m}_1, \mathbf{m}_2)]. \quad (4)$$

Let us define

$$\mathrm{Cov}(e) = \Pr_{\mathbf{m}_1, \mathbf{m}_2 \sim \pi} [e \in \mathrm{C}(\mathbf{m}_1, \mathbf{m}_2)].$$

Using this notation, the combination of Lemma 5.6 and (4) gives:

Corollary 5.7. $\mathrm{E}_{\mathbf{m}_3 \sim \pi} \left[\sum_{e \in \mathrm{Rep}(\mathbf{m}_3)} \mathrm{Cov}(e) \right] \geq 1 - 100\delta.$

We show that if $\mathrm{CC}(\Pi) \leq \alpha\sqrt{n}$ where α is a sufficiently small constant, then this cannot be achieved, that is, the protocol cannot focus enough "cover probability" on the edges reported by Bob.

Bob's "best strategy". By definition, $\mathrm{Cov}(e)$ is a score measuring how likely Alice and Bob are to cover edge e. It depends on the protocol, but not on the specific input to Alice, Bob or Charlie. When Bob examines his input, he *knows* the score $\mathrm{Cov}(e)$ for each edge $e \in \mathbf{E}_3$. He cannot afford to report all the edges in $\mathbf{E}_3$, as this would require too much communication, so he must choose some subset to report. Intuitively, Bob's "best strategy" is to report those edges in $\mathbf{E}_3$ that have the highest $\mathrm{Cov}(e)$, as this maximizes the sum in Corollary 5.7 (i.e., it increases the probability that $\mathrm{C}(t) \cap \mathrm{Rep}(t) \neq \emptyset$, giving the referee a good edge to output).

Let us make this intuition a little more formal. Let $T \subseteq V_1 \times V_2$ be the set of $|T| = \Theta(n)$ edges with the highest $\mathrm{Cov}(e)$ scores in $V_1 \times V_2$. Bob can typically afford to report only $O(\sqrt{n})$ edges; this follows from the superadditivity of information and Lemma 5.5. In addition, each edge in T only appears in Bob's input with probability $\gamma/\sqrt{n}$. Thus, if Bob reports all the edges in T that appear in his input (the set $T \cap \mathbf{E}_3$), this consumes his entire communication budget. If Bob follows this strategy then each edge in T is reported with probability nearly $\gamma/\sqrt{n}$: if it appears in Bob's input, it is reported — unless $T \cap \mathbf{E}_3$ is very large and Bob cannot afford to report all the edges in it, but this happens with very small probability. In the full version, we show that indeed reporting the edges in $T \cap \mathbf{E}_3$ is Bob's "best strategy", in the sense that

$$\mathrm{E}\left[\sum_{e \in \mathrm{Rep}(\mathbf{E}_3)} \mathrm{Cov}(e) \right] \leq \Theta(1/\sqrt{n}) \cdot \sum_{e \in T} \mathrm{Cov}(e). \quad (5)$$

Therefore, it suffices to show that the protocol *for Alice and Bob* can achieve $\sum_{e \in T} \mathrm{Cov}(e) = \Omega(\sqrt{n})$ for a fixed set $T \subseteq V_1 \times V_2$ of $\Theta(n)$. If we can show this, then Corollary 5.7 does not hold, which is a contradiction.

Bounding the cover scores. Our goal now is to show that for any set T of $\Theta(n)$ edges from $V_1 \times V_2$, we have $\sum_{e \in T} \mathrm{Cov}(e) = O(\sqrt{n})$.

Fix a set $T \subseteq V_1 \times V_2$ of $|T| = \xi n$ edges, where ξ is constant. We first partition T into subsets T_1, T_2, where, for some constant $\xi \in (0, 1)$, the set T_1 contains only edges (u, v) such that $\deg_{T_1}(u) \leq$

$\xi\sqrt{n}$ and T_2 contains only edges (u, v) such that $\deg_{T_2}(v) \leq \xi\sqrt{n}$. (Recall that $\deg_S(w)$ denotes the number of edges adjacent to w in edge set S.)

To do this, we partition the nodes of V_1, V_2 into "low degree nodes", with degree at most $\sqrt{n}$ in T, and "high degree nodes", with degree more than $\sqrt{n}$ in T. All edges of T that are adjacent to "low degree nodes" from V_1 go into T_1, and any remaining edges that are adjacent to "low degree nodes" from V_2 go into V_2. This leaves us with edges where both endpoints have high degree in T, more than $\sqrt{n}$. But since $|T| = O(n)$, there can be only $O(\sqrt{n})$ nodes with such high degree, so any such edge can go into either T_1 or T_2 arbitrarily.

We separately bound $\sum_{e \in T_1} \mathrm{Cov}(e)$ and $\sum_{e \in T_2} \mathrm{Cov}(e)$, and show that neither exceeds $O(\sqrt{n})$. This implies $\sum_{e \in T} \mathrm{Cov}(e) = O(\sqrt{n})$ as well. The proof is symmetric for the two sets, so we show here ony the lower bound for T_1.

To show that $\sum_{e \in T_1} \mathrm{Cov}(e) = O(\sqrt{n})$ we need to rule out the "quadratic effect" that we saw in the one-way protocol, where Alice and Bob each tell us about some number t of edges in their input, and yet the sum of the cover probabilities reaches t^2. This would be catastrophic for us, because now Alice and Bob can afford to tell us about $\tilde{\Theta}(\sqrt{n})$ edges each, so if we use a naive analysis we might conclude that the sum of the cover probabilities can go as high as $O(n)$; to bound the sum at $O(\sqrt{n})$ we need to show that this cannot happen, and the key lies in the fact that we consider nodes with degree at most $\sqrt{n}$ in T_1. (The difference from the one-way case is that there we did not have a *fixed* set T of $O(n)$ edges over which Alice and Bob tried to maximize the sum of the cover probabilities; and indeed, the optimal strategy for the one-way case produces a uniformly random set of edges with high cover probabilities, not a fixed set.)

Intuitively, suppose that $v_1 \in V_1$ has degree $O(\sqrt{n})$ in T_1, and let $S \in V_2$ be the neighbors of v_1 with respect to the edges in T_1. Because $|S| = \deg_{T_1}(v_1) = O(\sqrt{n})$, in expectation only $(\gamma/\sqrt{n}) \cdot O(\sqrt{n}) = O(1)$ edges connect U and S. Thus, whatever effort Alice spends to tell the referree about edges in $U \times \{v_1\}$ is "multiplied" only by $O(1)$, even if Bob was able to tell us about *all* the edges in $U \times S$. We get that the sum of the cover probabilities is *linear* in Alice's communication budget, not quadratic.

We now make this intuition formal. Let $\mathbf{C}_e$ be an indicator for the event that $e \in \mathrm{C}(\mathbf{m}_1, \mathbf{m}_2)$. Then $\mathrm{Cov}(e) = \Pr_\mu[\mathbf{C}_e]$.

For fixed messages m_1, m_2 of Alice and Bob, if $e = (v_1, v_2) \in \mathrm{C}(m_1, m_2)$, then by definition of $\mathrm{C}(m_1, m_2)$ and by union bound,

$$\begin{aligned} 1/10 &\leq \Pr_{\mathbf{E} \sim \mu | m_1, m_2} [\exists u : (u, v_1) \in \mathbf{E}_1 \wedge (u, v_2) \in \mathbf{E}_2] \\ &\leq \sum_{u \in U} \Pr_{\mathbf{E} \sim \mu | m_1, m_2} [(u, v_1) \in \mathbf{E}_1 \wedge (u, v_2) \in \mathbf{E}_2]. \end{aligned}$$

Because $\mathbf{E}_1$ and $\mathbf{E}_2$ remain independent given $\mathbf{m}_1 = m_1, \mathbf{m}_2 = m_2$, we can re-write this as

$$\begin{aligned} 1/10 &\leq \sum_{u \in U} \left(\Pr_{\mathbf{E} \sim \mu | m_1} [(u, v_1) \in \mathbf{E}_1] \cdot \Pr_{\mathbf{E} \sim \mu | m_2} [(u, v_2) \in \mathbf{E}_2] \right) \\ &\leq \sum_{u \in U} \left(\left(\Delta_{m_1}(u, v_1) + 2\gamma/\sqrt{n}\right) \Pr_{\mathbf{E} \sim \mu | m_2} [(u, v_2) \in \mathbf{E}_2] \right), \end{aligned}$$

where here, as in the previous section, we let

$$\Delta_{m_1}(u, v_1) = \Pr_{\mathbf{E} \sim \mu | m_1} [(u, v_1) \in \mathbf{E}_1] - 2\gamma/\sqrt{n}.$$

Consider first the edges in T_1. Let M_2 be the set of Bob's messages m_2 such that $(v_1, v_2) \in \mathrm{C}(m_1, m_2)$ (keeping m_1 fixed), and let $\pi_2(m_2)$ be the probability that Bob sends a specific message m_2. Multiplying both sides by $\pi_2(m_2)$ and summing over all $m_2 \in M_2$, we get

$$\begin{aligned} &\sum_{m_2 \in M_2} \sum_{u \in U} \left(\pi_2(m_2) \Pr_{\mathbf{E} \sim \mu | m_2} [(u, v_2) \in \mathbf{E}_2] \left(\Delta_{m_1}(u, v_1) + 2\gamma/\sqrt{n}\right) \right) \\ &\geq (1/10) \sum_{m_2 \in M_2} \pi_2(m_2) = (1/10) \Pr_{\mu | m_1} \left[\mathbf{C}_{(v_1, v_2)}\right]. \end{aligned} \tag{6}$$

Notice that for any $u \in U$ and m_1,

$$\begin{aligned} &\sum_{m_2 \in M_2} \left(\pi_2(m_2) \Pr_{\mathbf{E} \sim \mu | m_2} [(u, v_2) \in \mathbf{E}_2] \right) \\ &= \Pr_{\mathbf{E} \sim \mu | m_1} \left[(u, v_2) \in \mathbf{E}_2 \wedge \mathbf{C}_{(u, v_2)}\right] \\ &\leq \Pr_{\mathbf{E} \sim \mu} [(u, v_2) \in \mathbf{E}_2] = \gamma/\sqrt{n}. \end{aligned}$$

Plugging this into (6) yields

$$(\gamma/\sqrt{n}) \sum_{u \in U} \left(\Delta_{m_1}(u, v_1) + 2\gamma/\sqrt{n}\right) \geq (1/10) \Pr_{\mu | m_1} \left[\mathbf{C}_{(v_1, v_2)}\right].$$

Now, taking the expectation over all m_1, and summing across all v_2 such that $(v_1, v_2) \in T_1$ (while keeping v_1 fixed),

$$\begin{aligned} &\deg_{T_1}(v_1)(\gamma/\sqrt{n}) \mathop{\mathrm{E}}_{\mathbf{m}_1} \left[\sum_{u \in U} (\Delta_{\mathbf{m}_1}(u, v_1) + 2\gamma/\sqrt{n}) \right] \\ &\geq (1/10) \sum_{v_2 : (v_1, v_2) \in T_1} \mathop{\mathrm{E}}_{\mathbf{m}_1} \left[\Pr_{\mu | \mathbf{m}_1} \left[\mathbf{C}_{(v_1, v_2)}\right] \right] \\ &\geq (1/10) \sum_{v_2 : (v_1, v_2) \in T_1} \mathrm{Cov}(v_1, v_2). \end{aligned}$$

Finally, summing over all $v_1 \in V_1$, and plugging in $\deg_{T_1}(v_1) \leq \xi\sqrt{n}$, we get

$$\begin{aligned} &\xi\gamma \sum_{v_1 \in V_1} \mathop{\mathrm{E}}_{\mathbf{m}_1} \left[\sum_{v_1 \in V_1} \sum_{u \in U} \Delta_{\mathbf{m}_1}(u, v_1) + 2\gamma/\sqrt{n} \right] \\ &= \xi\gamma \left(\mathop{\mathrm{E}}_{\mathbf{m}_1} \left[\sum_{v_1 \in V_1} \sum_{u \in U} \Delta_{\mathbf{m}_1}(u, v_1) \right] + 2\gamma\sqrt{n} \right) \\ &\geq (1/10) \sum_{(v_1, v_2) \in T_1)} \mathrm{Cov}(v_1, v_2). \end{aligned}$$

As we saw previously, using Lemma 5.5 and the subadditivity of information we get that $\mathrm{E}_{\mathbf{m}_1} \left[\sum_{v_1 \in V_1} \sum_{u \in U} \Delta_{\mathbf{m}_1}(u, v_1)\right] \leq \mathrm{CC}(\Pi) = O(\sqrt{n})$, so the left-hand side is bounded by $O(\sqrt{n})$, as desired: the sum of the cover scores of edges in T_1 cannot exceed $O(\sqrt{n})$.

5.6 Lifting 3-Player Lower Bounds to k Players

To lift our lower bounds to $k > 3$ players we use *symmetrization*, a technique developed in [30] for embedding hard two-player problems in a multi-player model. Here, we have a hard *3-player* problem, and we also work with a restricted communication pattern, either simultaneous or one-way; we adapt the technique from [30] to these changes, obtaining the following:

Theorem 5.8 (Informal). *Let μ be a 3-player input distribution, where each of the three players has the same marginal input distribution. Let C be the one-way, δ-error communication complexity of problem P under μ. Then there is a k-player input distribution η such that the simultaneous δ-error communication complexity of P under η is $\Omega(k \cdot C)$.*

Notice that we assumed that the problem was hard for 3-player *one-way* protocols, but the k-player lower bound we obtained was for *simultaneous* protocols. This behavior is inherent for a large number of players, $k = \Omega(n)$: a simultaneous k-player protocol can emulate a one-way protocol by having each of the k players send their entire input to the referee with probability $\Theta(1/k)$, and otherwise send the message it would send under the one-way protocol. With good probability, the referee receives the input of at least one player, and it can then compute the output of the one-way protocol. When $k = \Omega(n)$, the overall cost of simulating a one-way protocol with communication cost C is $O(n^2/k + k \cdot C) = O(k \cdot C)$, so Theorem 5.8 is tight in this sense. (For smaller k, it is possible that the result can be tightened.)

For *deterministic* and *symmetric* protocols we can do a little better: we show that if P is hard for deterministic 3-player simultaneous protocols, then it is k times harder for deterministic k-player simultaneous protocols.

6 CONCLUSIONS

We showed that in the setting of communication complexity, property testing can be significantly easier than exactly testing if the input satisfies the property: exactly determining if the input graph is triangle-free requires $\Omega(knd)$ bits in [35], but we showed that weakening the requirement to property-testing improves the complexity, and in fact even simultaneous communication protocols can do better than the best exact algorithm with unrestricted communication. Nevertheless, the problem does not appear to become completely trivial, as shown by our lower bounds for simultaneous and restricted one-way protocols.

Acknowledgements. We thank Noga Alon, Eldar Fischer and Dana Ron for fruitful discussions.

REFERENCES

[1] Noga Alon. 2002. Testing Subgraphs in Large Graphs. *Random Struct. Algorithms* 21, 3-4 (Oct. 2002), 359–370.
[2] Noga Alon, Eldar Fischer, Michael Krivelevich, and Mario Szegedy. 2000. Efficient Testing of Large Graphs. *Combinatorica* 20, 4 (2000), 451–476.
[3] Noga Alon, Tali Kaufman, Michael Krivelevich, and Dana Ron. 2006. Testing Triangle-freeness in General Graphs. In *Proceedings of the Seventeenth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 279–288.
[4] Noga Alon and Asaf Shapira. 2003. Testing Subgraphs in Directed Graphs. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing (STOC '03)*. 700–709.
[5] Christos Boutsidis, David P. Woodruff, and Peilin Zhong. 2016. Optimal Principal Component Analysis in Distributed and Streaming Models. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, (STOC)*. 236–249.
[6] Zvika Brakerski and Boaz Patt-Shamir. 2011. Distributed Discovery of Large Near-Cliques. *Distributed Computing* 24, 2 (2011), 79–89.
[7] Mark Braverman, Faith Ellen, Rotem Oshman, Toniann Pitassi, and Vinod Vaikuntanathan. 2013. A Tight Bound for Set Disjointness in the Message-Passing Model. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS)*. 668–677.
[8] Mark Braverman, Ankit Garg, Tengyu Ma, Huy L. Nguyen, and David P. Woodruff. 2016. Communication Lower Bounds for Statistical Estimation Problems via a Distributed Data Processing Inequality. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing (STOC)*. 1011–1020.
[9] Keren Censor-Hillel, Eldar Fischer, Gregory Schwartzman, and Yadu Vasudev. 2016. Fast Distributed Algorithms for Testing Graph Properties. In *Distributed Computing: 30th International Symposium (DISC)*. 43–56.
[10] Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. 2015. Algebraic Methods in the Congested Clique. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC '15)*. 143–152.
[11] Thomas M. Cover and Joy A. Thomas. 2006. *Elements of Information Theory (Wiley Series in Telecommunications and Signal Processing)*.
[12] Danny Dolev, Christoph Lenzen, and Shir Peled. 2012. "Tri, Tri Again": Finding Triangles and Small Subgraphs in a Distributed Setting. In *Proceedings of the 26th International Conference on Distributed Computing (DISC'12)*. 195–209.
[13] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. 2014. On the Power of the Congested Clique Model. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing (PODC '14)*. 367–376.
[14] Talya Eden, Amit Levi, Dana Ron, and C. Seshadhri. 2015. Approximately Counting Triangles in Sublinear Time. In *IEEE 56th Annual Symposium on Foundations of Computer Science, (FOCS)*. 614–633.
[15] Eldar Fischer. 2001. The Art of Uninformed Decisions. *Bulletin of the EATCS* 75 (2001), 97.
[16] Orr Fischer, Shay Gershtein, and Rotem Oshman. 2017. On The Multiparty Communication Complexity of Testing Triangle-Freeness (Full Version). *CoRR* abs/1705.08438 (2017). http://arxiv.org/abs/1705.08438
[17] Jacob Fox. 2011. A New Proof of the Graph Removal Lemma. *Annals of Mathematics* (2011), 561–579.
[18] Pierre Fraigniaud, Ivan Rapaport, Ville Salo, and Ioan Todinca. 2016. Distributed Testing of Excluded Subgraphs. In *Distributed Computing - 30th International Symposium, (DISC)*. 342–356.
[19] Oded Goldreich. 1998. Combinatorial Property Testing – a Survey. *Randomization Methods in Algorithm Design* (1998), 45–60.
[20] Oded Goldreich, Shari Goldwasser, and Dana Ron. 1998. Property Testing and Its Connection to Learning and Approximation. *J. ACM* 45, 4 (1998), 653–750.
[21] Oded Goldreich and Dana Ron. 1997. Property Testing in Bounded Degree Graphs. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing (STOC '97)*. 406–415.
[22] Oded Goldreich and Luca Trevisan. 2003. Three Theorems Regarding Testing Graph Properties. *Random Struct. Algorithms* 23, 1 (2003), 23–57.
[23] L. Gugelmann. 2006. Testing Trinagle-Freeness in General Graphs: Lower Bounds. Bachelor thesis, Dept. of Mathematics, ETH, Zurich. (2006).
[24] Zengfeng Huang and Pan Peng. 2016. Dynamic Graph Stream Algorithms in o(n) Space. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP)*. 18:1–18:16.
[25] John Kallaugher and Eric Price. 2017. A Hybrid Sampling Scheme for Triangle Counting. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017*. 1778–1797.
[26] Iordanis Kerenidis and Ran Raz. 2006. The One-way Communication Complexity of the Boolean Hidden Matching Problem. *Electronic Colloquium on Computational Complexity (ECCC)* 13 (2006).
[27] François Le Gall. 2016. *Further Algebraic Algorithms in the Congested Clique Model and Applications to Graph-Theoretic Problems*. 57–70.
[28] Yi Li, Xiaoming Sun, Chengu Wang, and David P. Woodruff. 2014. *On the Communication Complexity of Linear Algebraic Problems in the Message Passing Model*. 499–513.
[29] Ilan Newman. 1991. Private vs. Common Random Bits in Communication Complexity. *Inf. Process. Lett.* 39, 2 (1991), 67–71.
[30] Jeff M. Phillips, Elad Verbin, and Qin Zhang. 2012. Lower Bounds for Number-in-hand Multiparty Communication Complexity, Made Easy. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '12)*. 486–501.
[31] T. Rast. 2006. Testing Trinagle-Freeness in General Graphs: Upper Bounds. Bachelor thesis, Dept. of Mathematics, ETH, Zurich. (2006).
[32] Dana Ron. 2009. Algorithmic and Analysis Techniques in Property Testing. *Foundations and Trends in Theoretical Computer Science* 5, 2 (2009), 73–205.
[33] Elad Verbin and Wei Yu. 2011. The Streaming Complexity of Cycle Counting, Sorting by Reversals, and Other Problems. In *Proceedings of the Twenty-second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '11)*. 11–25.
[34] David P. Woodruff and Qin Zhang. 2012. Tight Bounds for Distributed Functional Monitoring. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing (STOC '12)*. 941–960.
[35] David P. Woodruff and Qin Zhang. 2013. When Distributed Computation Is Communication Expensive. In *Distributed Computing: 27th International Symposium, DISC 2013, Jerusalem, Israel, October 14-18, 2013. Proceedings*. 16–30.
[36] David P. Woodruff and Qin Zhang. 2014. An Optimal Lower Bound for Distinct Elements in the Message Passing Model. In *Proceedings of the Twenty-Fifth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*. 718–733.