Broadcast Congested Clique: Planted Cliques and Pseudorandom Generators*

Lijie Chen
EECS, MIT
Cambridge, MA
wjmbmr@gmail.com

Ofer Grossman
EECS, MIT
Cambridge, MA
ofer.grossman@gmail.com

ABSTRACT

We develop techniques to prove lower bounds for the BCAST($\log n$) Broadcast Congested Clique model (a distributed message passing model where in each round, each processor can broadcast an $O(\log n)$ -sized message to all other processors). Our techniques are built to prove bounds for natural input distributions. So far, all lower bounds for problems in the model relied on constructing specifically tailored graph families for the specific problem at hand, resulting in lower bounds for artificially constructed inputs, instead of natural input distributions.

One of our results is a lower bound for the directed planted clique problem. In this problem, an input graph is either a random directed graph (each directed edge is included with probability $1/2$), or a random graph with a planted clique of size k . That is, k randomly chosen vertices have all of the edges between them included, and all other edges in the graph appear with probability $1/2$. The goal is to determine whether a clique exists. We show that when $k = O(n^{1/4-\epsilon})$, this problem requires a number of rounds polynomial in n .

Additionally, we construct a pseudo-random generator which fools the Broadcast Congested Clique. This allows us to show that every k round randomized algorithm in which each processor uses up to n random bits can be efficiently transformed into an $O(k)$ -round randomized algorithm in which each processor uses only up to $O(k \log n)$ random bits, while maintaining a high success probability. The pseudo-random generator is simple to describe, computationally very cheap, and its seed size is optimal up to constant factors. However, the analysis is quite involved, and is based on the new technique for proving lower bounds in the model.

The technique also allows us to prove the first average case lower bound for the Broadcast Congested Clique, as well as an average-case time hierarchy. We hope our technique will lead to more lower bounds for problems such as triangle counting, APSP, MST, diameter, and more, for natural input distributions.

CCS CONCEPTS

• Theory of computation → Distributed algorithms.

*The full version of this paper can be found in <https://arxiv.org/abs/1905.07780>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

PODC '19, July 29–August 2, 2019, Toronto, ON, Canada

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6217-7/19/07...\$15.00

<https://doi.org/10.1145/3293611.3331599>

KEYWORDS

broadcast congested clique; multi-party communication complexity; planted clique; pseudo-random generators

ACM Reference Format:

Lijie Chen and Ofer Grossman. 2019. Broadcast Congested Clique: Planted Cliques and Pseudorandom Generators. In *2019 ACM Symposium on Principles of Distributed Computing (PODC '19), July 29–August 2, 2019, Toronto, ON, Canada*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3293611.3331599>

1 INTRODUCTION

Recently there has been a surge of lower bound results in the CONGEST, and Broadcast Congested Clique models ([11, 16, 25, 31] to list a few, though there are many more). In general, these results take the following approach: a specific carefully chosen family of graphs is constructed, and the vertices are partitioned into two parts (or, in some rare cases, three parts), and a reduction from classical two-party communication complexity is used.

A main downside of this approach is that it merely proves worst case lower bounds. For all we know, for more natural input distributions (instead of the artificially constructed graph families used for these communication complexity reductions), many problems which are worst-case hard may become easy. One of our main goals in this paper is to develop techniques that prove lower bounds for more natural input distributions.

The distributed model we consider is the Broadcast Congested Clique model (BCAST(1))¹. In this model, there are n processors, each with unlimited local computational power. Computation proceeds in rounds, and in each round each processor broadcasts a single bit² to all other processors (within a single round, a processor must broadcast the same bit to all other processors).

Our results include a lower bound for the planted clique problem. Additionally, we show how to use our techniques to construct a pseudo-random generator that fools the Broadcast Congested Clique model. This is the first pseudo-random generator which fools a distributed message passing model. As simple corollaries, we obtain an average-case lower bound for the model, and an average-case time hierarchy for the model.

¹Note that every lower bound for BCAST(1) can be extended to a lower bound for BCAST($\log n$) with only a $\log n$ factor loss in the number of rounds

²It is standard to use messages of size $O(\log n)$, but for our purposes it is more natural to consider single-bit messages. All of our results generalize to the setting of logarithmic sized messages.

1.1 Our Approach

How to deal with distributions. Consider the (directed) planted clique problem³. In this problem, the input is a directed graph which is either a random graph (each directed edge is included with probability $1/2$), or a random graph where k of the vertices are chosen at random and all edges within this set of k vertices are included (these k vertices are called the planted clique). The goal is to distinguish between these two distributions (or, one can consider the search version of the problem, in which a graph with a planted clique is given, and the goal is to find the clique). For now, one can think of k as approximately $n^{1/4}$.

When trying to prove a lower bound for the problem, one's first approach may be to try reducing to two-party communication complexity: split the vertices into two parts, and argue that a lot of information must pass between the parts. This approach will not work for the problem, since no matter how the graph is split up, at least one of the parts will be able to detect the presence of the clique, since at least one of the parts must have many of the clique's vertices.

One's next approach may be the following: show that for any Congested Clique algorithm, after t rounds, the distribution of transcripts of the Congested Clique algorithm (the "transcript" is the history of the algorithm; that is, the "transcript" is a list of all messages sent so far as well as who sent which message and when the message was sent) if the input were uniform is close to the distribution of transcripts if the input had a planted clique. To prove this, one may take an inductive approach: show that if the transcripts were similar for $t - 1$ rounds, the next round can distinguish between the distributions with only low probability. In this round, all vertices are broadcasting, so one may try to use another approach: show that each vertex reveals little information about whether the graph has a planted clique or not, and therefore, the whole round reveals little information about whether there is a clique. This way, we would only need to analyze a single broadcast at a time, a much simpler task than considering a whole round at once.

There is an issue, however. The problem is that the inputs of different nodes may not be independent. Therefore, it is possible that while each processor's broadcast on its own will not reveal substantial information about whether the graph contains a clique, when we consider many processors' broadcasted bits the information revealed may be more than the sum of the information of the individual broadcasts.

To get around this, we instead split the planted clique distributions into a sum of many distributions, such that for each of these distributions, all processors' inputs are independent. Specifically, we can write the planted clique distribution as a sum over all possible cliques C of a random graph with a clique planted at C . Notice that after fixing C , each vertex's input is independent of all other vertices' inputs. Now, when considering whether an algorithm distinguishes between a random graph with a clique at C and a truly random graph, we can consider each node's output on its own, instead of trying to deal with all nodes at the same time. This

is one of our main high level ideas: splitting the distribution into many distributions where in each one, different vertices' inputs are independent. This greatly simplifies the analysis by letting us avoid having to deal with many messages at once, and is what makes proving the lower bounds possible.

Statistical Inequalities. The idea of partitioning a distribution into distributions with independent vertex inputs makes proving the lower bounds possible, but there is still lots of technical work to do. Specifically, we now have a bunch of distributions, and we need to show that any algorithm can distinguish only few of those distributions from uniform. At the high level, the idea to do this is to show that for any algorithm, for almost all cliques, when a vertex broadcasts a bit, the probability of that bit being broadcast with the clique vs without the clique is similar. This is basically a problem about Boolean functions: if we let f be the function which takes in the node's input, and outputs what the node will broadcast, we wish to show that for almost all cliques C , when f 's input is uniform, the output distribution is similar to the distribution when f 's input is chosen with a planted clique at C . The inequalities we show are at the high level similar to this, but many more technical issues arise. For example, since we have to prove a multi-round lower bound, we have to condition on what a node broadcasted in previous rounds, so instead of proving the statistical inequalities for total functions, we instead have to prove the inequalities for functions definitionned on only part of $\{0, 1\}^n$. These inequalities are not true for all partial functions, but we manage to prove that they are true for all functions which are definitionned on a large enough subset. So then, there is another challenge of proving that with high probability, after conditioning on a transcript, the set of possible inputs to a node is large.

1.2 Our Results

Lower bounds for the planted clique problem. One of the problems we consider is the (directed) planted clique problem. In this problem, the input is a directed graph which is either a random graph (each directed edge is included with probability $1/2$), or a random graph where k of the vertices are chosen and all edges within this set of k vertices are added. The goal is to distinguish between these two distributions (or, one can consider the search version of the problem, in which a graph with a planted clique is given, and the goal is to find the clique). Because a random graph contains cliques of size $\Theta(\log n)$, the problem makes sense for larger values of k . Once k goes substantially above $\sqrt{n}$, it is possible to find the clique by considering the vertices with highest degree. Hence, the interesting values of k are between approximately $\log n$ and $\sqrt{n}$.

In the classical (non-distributed) setting, the planted clique problem is very well studied. There exists a spectral algorithm solving the problem when $k = O(\sqrt{n})$, and it remains a major open problem in complexity theory to understand whether the problem is hard when k is smaller.

It has been shown that proving lower bounds in the unicast CONGESTED-CLIQUE⁴ would imply some strong circuit lower

³Note that in the broadcast congested clique model, as opposed to the CONGEST or unicast congested clique model, it is not possible to reduce from directed to undirected in one round.

⁴In the unicast model, in each round each processor can send one bit to each other processor without the requirement that the same bit is broadcast to all other processors; that is, within a single round a processor may choose to send the message "1" to some processors, and "0" to others.

bounds [16], so finding lower bounds for the problem in this setting is quite a challenge. Finding upper bounds for planted clique in the unicast CONGESTED-CLIQUE model is an interesting problem, but it seems difficult – maybe impossible – to improve upon simple sampling-based algorithms for the problem. The next model one can look at is the broadcast CONGESTED-CLIQUE model, which is the model we consider. Specifically, we prove that for cliques of size $O(n^{1/4-\varepsilon})$, the planted clique problem requires polynomially many rounds:

THEOREM 1.1 (PLANTED CLIQUE LOWER BOUND). *When $k = n^{1/4-\varepsilon}$ for a constant ε , no $n^{o(1)}$ round BCAST(1) protocol Π can distinguish between $\mathcal{A}_{\text{rand}}$ and $\mathcal{A}_k$ with advantage⁵ $\Omega(1)$.*

Pseudo-random generators for distributed computation: Historically, pseudo-random generators were used to fool adversaries modeled as Turing Machines or Circuits. One of our main contributions is constructing the first pseudo-random generator which fools a distributed message passing setting. That is, we show how within the BCAST(1) Congested Clique model, the processors can each sample a small random seed, and end with each processor having a longer string than it started with, such that these longer strings *look random* to the system. That is, no low-round BCAST(1) protocol can distinguish between these pseudo-random strings and truly uniformly random strings within few rounds (except with some low probability). Thus, any algorithm can use a pseudo-random string instead of true randomness, thereby saving random bits.

Pseudo-random generators have been used to derandomize specific problems in message passing models, for example in [39]. Our construction is the first pseudo-random generator which fools *all* low-round algorithms in the BCAST model (as opposed to just fooling a specific algorithm).

Definition 1.2 (BCAST(1) pseudo-random generator). A (k, m, n, ℓ) BCAST(1) pseudo-random generator (PRG) is an n -processor BCAST(1) protocol Π such that:

- At the beginning every processor independently gets k private uniform and independent random bits as input.
- After participating in protocol Π , each processor outputs m bits (these bits are not broadcasted: they are the node's pseudo-random bits).
- The joint distribution of all processors' output bits cannot be distinguished (with better than $\frac{1}{n}$ probability⁶) from a truly uniform random distribution by any ℓ round BCAST(1) protocol. Specifically, the statistical distance between the distribution of the protocol's transcript when using a pseudo-random generator and the distribution of the transcript when using true randomness is small.

THEOREM 1.3. *For all $m = O(n)$ and $k = \Omega(\log n)$, there exists an $(O(k), m, n, \Theta(k))$ BCAST(1) PRG that can be constructed within $O(k)$ rounds. In particular, the PRG works as follows*

- Each processor gets $k + k \cdot \frac{(m-k)}{n} = O(k)$ private random bits.

⁵An algorithm distinguishing between two distributions D_1 and D_2 with advantage ε is an algorithm A which, when given a random sample s which with probability $1/2$ is drawn from D_1 and probability $1/2$ is drawn from D_2 , then A can successfully guess from which distribution s was drawn with probability $1/2 + \varepsilon$.

⁶All of our constructions in the paper can achieve arbitrarily low inverse polynomial probability.

- Then in $O\left(\frac{m-k}{n} \cdot k\right) = O(k)$ rounds, all processors broadcast their last $k \cdot \frac{(m-k)}{n}$ random bits. And they use that to construct a random matrix $M \in \{0, 1\}^{k \times (m-k)}$.
- Each processor's output is simply the concatenation of its first k random bits x and $x^T M$.

That is, with k random bits per processor as a seed, for every constant c , within $O(k)$ rounds we can turn them into cn pseudo-random bits per processor which require $\Omega(k)$ rounds to be distinguished from random.

So, for example, within $O(\log^2(n))$ round (In the BCAST($\log n$) model, $O(\log n)$ rounds would suffice), one can construct a pseudo-random generator which is indistinguishable from random for any $\log^2 n$ -round algorithm, except with low probability (the seed size for each processor would be $O(\log^2 n)$, while the size of the pseudo-random string is $\Theta(n)$).

The PRG is very simple to describe (we describe it here with seed size $2k$): first each processor shares k random bits to create k public random elements from $\mathbb{F}_2^n$ (i.e., k random n -bit vectors). Then, each processor uses its remaining k random bits to pick a random linear combination of those vectors (which requires k bits to sample), and the result of this linear combination is the node's pseudo-random bits. So, essentially the pseudo-random generator is a distribution of low-rank matrices (which is very close to the uniform distribution matrices of rank up to k). Although the description and construction of the PRG are simple, and the seed size is tight up to a constant factor, the analysis is quite technical and involves new techniques. We remark that the pseudo-random generator has the additional nice property that it is computationally cheap; the only operations done by the processors is computing dot products of vectors over $\mathbb{F}_2$.

Efficiently saving random bits. It is possible to show that in the broadcast congested clique there is a randomized-deterministic separation (by reductions from two-player communication complexity for equality). That is, there are certain problems with faster randomized algorithms than the best possible deterministic algorithms. So, there is no hope for a general derandomization theorem. However, one can ask: in general, what is the fewest number of random bits needed to efficiently solve problems in the Broadcast Congested Clique model? Using a technique of Newman [38] from communication complexity, it is possible to show that for any protocol with output size k per processor, $O(k)$ random bits per processor is enough (see the full version of this paper). The main downside of Newman's technique is that it is computationally inefficient: it holds in the case where all the processors have unbounded computational power, and is not a practical tool for saving random bits.

In this work, we ask the following question: In the Broadcast Congested Clique with *computationally bounded (polynomial time)* processors, how many random bits are needed to perform general randomized computation? We can use our pseudo-random generator to show that every k -round algorithm where $k = \Omega(\log n)$ in which every processor uses up to $O(n)$ random bits can be transformed into an $O(k)$ -round algorithm in which every processor uses up to k random bits (that is, we can show that each processor needs to use at most 1 random bit per round, while only increasing

the run-time by a constant factor). Furthermore, this transformation is efficient. That is, if in the original algorithm all processors work in polynomial time, they also work in polynomial time in the new algorithm (in fact, there is only an *additive* overhead of $O(kn)$ computation time for each processor. This overhead is the time required to compute the pseudo-random bits from the seed).

Stated in the setting of BCAST($\log n$), where messages are of size $O(\log n)$ instead of 1, we show that every k round randomized algorithm using up to n random bits per processor can be transformed into an $O(k)$ round algorithm using up to $O(k \log n)$ random bits per processor.

First BCAST(1) Average Case Lower Bound and Hierarchy: As a simple corollary of our PRG construction, we prove a BCAST(1) average case lower bound, which is the first average case lower bound proven in the model. Specifically, we show that when each processor receives a row of a sample from a certain close-to-uniform distribution of $n \times n$ matrices of rank $n - 1$, this cannot be distinguished from each processor receiving n uniformly random bits. We show that this implies that determining whether an input has rank n or not is hard (it takes $\Omega(n)$ rounds), even when the input is chosen uniformly at random:

THEOREM 1.4. *Let n be a large enough integer and $F_{\text{full-rank}} : \{0, 1\}^{n \times n} \rightarrow \{0, 1\}$ be the indicator function which indicates whether the given matrix has full rank. Suppose there are n processors, and the i -th processor is given the i -th row of the input matrix. For all $n/20$ -round BCAST(1) protocols and all processors i in it, i cannot compute F correctly with probability better than 0.99 when the input is a uniformly chosen random matrix from $\{0, 1\}^{n \times n}$.*

We also obtain an average-case time hierarchy theorem for the model:

THEOREM 1.5. *For any $\omega(\log n) \leq k \leq n$, there is a function F such that a k -round BCAST(1) protocol can compute exactly, while any $k/20$ -round BCAST(1) protocols cannot compute F correctly with probability 0.99 over the uniform distribution.*

1.3 Toy Example: One Round Lower Bound for Planted Clique

As a toy example to illustrate our proof framework, in the following we prove that the planted clique problem is hard for one-round BCAST(1) protocols when $k = o(n^{1/4})$. We begin with some notations.

Notations. Let $\mathcal{U}_m$ denote the uniform distribution on $\{0, 1\}^m$. For a function $f : \{0, 1\}^* \rightarrow \{0, 1\}^*$ and a distribution $\mathcal{D}$ on $\{0, 1\}^*$, we use $f(\mathcal{D})$ to denote the distribution of the output of f when the input is drawn from $\mathcal{D}$. For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we use $\|\mathcal{D}_1 - \mathcal{D}_2\| = \frac{1}{2} \sum_{x \in \{0, 1\}^*} |\mathcal{D}_1(x) - \mathcal{D}_2(x)|$ to denote their statistical distance (where $\mathcal{D}(x)$ is the probability that a sample from $\mathcal{D}$ equals x).

Let $\mathcal{A}_{\text{rand}}^n$ be the distribution on $\{0, 1\}^{n \times n}$ such that for a sample A from $\mathcal{A}_{\text{rand}}^n$, for all $i \neq j$, $A_{i,j}$ is an independent uniform random bit in $\{0, 1\}$, and $A_{i,i}$ is always 0 for all i . Let C be a subset of $[n]$. We use $\mathcal{A}_C^n$ to denote the conditional distribution of $\mathcal{A}_{\text{rand}}^n$ on the event that for all $i, j \in C$ and $i \neq j$, $A_{i,j} = 1$ (that is, C is a clique). We also use $\mathcal{A}_k^n$ to be the mixed distribution of $\mathcal{A}_C^n$'s when C is a

uniform random subset of $[n]$ of size k . When the meaning is clear, we often drop the superscripts of the aforementioned distributions for simplicity.

So, to summarize, $\mathcal{A}_{\text{rand}}^n$ is the uniform distribution over a random directed graph, $\mathcal{A}_C^n$ is the distribution where the vertices of C are in a clique, and the rest of the edges are uniformly random, and $\mathcal{A}_k^n$ is the distribution where a random k vertices are chosen to be a clique, and the rest of the edges are chosen uniformly at random.

By Yao's principle [41], we can assume all processors are deterministic as we are trying to prove a lower bound for distinguishing two input distributions. Processor i can then be definitioned by a function $f_i : \{0, 1\}^n \times \{0, 1\}^* \rightarrow \{0, 1\}$, such that $f_i(z, p)$ is the bit that processor i outputs when it gets the input z and transcript p .⁷ We use $f_i^{|p|}$ to denote the function $f_i(\cdot, p)$ for simplicity. If transcript p is incompatible with processor i having input z , then we set $f_i(z, p)$ arbitrarily.

Given a BCAST(1) protocol Π and an input distribution $\mathcal{D}$, we use $\mathcal{P}(\Pi, \mathcal{D})$ to denote the distribution of the transcripts of the protocol Π running on an input drawn from $\mathcal{D}$ (that is, given a matrix A which is drawn from the distribution $\mathcal{D}$, the processor i gets the i -th row of A , and all processors act according to the protocol Π).

In this section we prove the following theorem.

THEOREM 1.6. *Let n be the number of processors and k be an integer. For any one round BCAST(1) protocol Π , we have*

$$\|\mathcal{P}(\Pi, \mathcal{A}_{\text{rand}}) - \mathcal{P}(\Pi, \mathcal{A}_k)\| \leq O\left(\frac{k^2}{\sqrt{n}}\right).$$

That is, for any one-round protocol, the distribution of transcripts when run on a uniformly random input is statistically close to the distribution on an input with a planted clique. As a simple corollary, we immediately have:

COROLLARY 1.7. *When $k = o(n^{1/4})$, no one-round BCAST(1) protocol Π can distinguish between $\mathcal{A}_{\text{rand}}$ and $\mathcal{A}_k$ with advantage $\Omega(1)$ (that is, any protocol which accepts on $\mathcal{A}_{\text{rand}}$ with probability p , must accept on $\mathcal{A}_k$ with probability $p \pm o(1)$).*

That is, there is no way to solve the planted clique problem for cliques of size $o(n^{1/4})$ within one round of the Broadcast Congested Clique.

Let $\mathcal{S}_k^T$ be the uniform distribution on all size- k subsets of T . To prove Theorem 1.6, we need the following technical lemma, whose proof is deferred to the end of this section. In this lemma $f(x)$ represents the bit broadcasted by a processor, and x represents the input to the processor. The lemma states that when picking a random clique C of size k , then any function f behaves similarly when the input is sampled from the uniform distribution vs. when the input is sampled from the uniform distribution with a clique planted on C . Basically, this means that for almost all possible cliques, f does not substantially help distinguish between the clique existing, or the input being uniform.

⁷In a zero-round protocol, the processor's f_i does not take in an input p , since there is no transcript yet, just an input. However, in our proof, we assume that the processors broadcast their messages sequentially (that is, first the first processor speaks, then the second, and so on). In this stronger model, all but one of the processors do see a transcript before they broadcast their first bit.

LEMMA 1.8. Let n, k be integers such that $k \leq n^{1/4}$, and $\mathcal{U}_n^C$ be the uniform distribution on $\{x : x \in \{0, 1\}^n, x_i = 1 \text{ for all } i \in C\}$. For all function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we have

$$\mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|f(\mathcal{U}_n) - f(\mathcal{U}_n^C)\| \leq O\left(\frac{k}{\sqrt{n}}\right).$$

We also need the following lemma to bound the increase of the statistical distance when a processor speaks.

LEMMA 1.9. Let X and Y be two sets, $\mathcal{D}$ and $\mathcal{D}'$ be two distributions on $X \times Y$. Let $\mathcal{D}_{|X}$ and $\mathcal{D}'_{|X}$ be the respective marginal distribution of $\mathcal{D}$ and $\mathcal{D}'$ on set X . For $a \in X$, we use $\mathcal{D}_{X=a}$ and $\mathcal{D}'_{X=a}$ to denote the respective conditional distribution of $\mathcal{D}$ and $\mathcal{D}'$ on Y conditioning on $X = a$.⁸ We have

$$\|\mathcal{D} - \mathcal{D}'\| \leq \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\| + \mathbb{E}_{a \sim \mathcal{D}_{|X}} \|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|.$$

PROOF. We first define an auxiliary distribution $\mathcal{D}_{\text{aux}}$ as follows: for $(a, b) \in X \times Y$, if $\mathcal{D}'_{|X}(a) > 0$, (we use $\mathcal{D}'_{|X}(a)$ to denote the probability that a sample from $\mathcal{D}'_{|X}$ equal a).

$$\mathcal{D}^{\text{aux}}(a, b) := \mathcal{D}'(a, b) \cdot \frac{\mathcal{D}_{|X}(a)}{\mathcal{D}'_{|X}(a)}.$$

Otherwise, we set $\mathcal{D}^{\text{aux}}(a, b) := \frac{1}{|Y|} \cdot \mathcal{D}_{|X}(a)$. It is easy to verify that $\mathcal{D}_{|X}^{\text{aux}} = \mathcal{D}_{|X}$ and for all a $\mathcal{D}_{X=a}^{\text{aux}} = \mathcal{D}'_{X=a}$, and therefore it is a distribution.

Now, it is easy to see that

$$\|\mathcal{D}^{\text{aux}} - \mathcal{D}'\| = \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\|.$$

Moreover, we have

$$\|\mathcal{D}^{\text{aux}} - \mathcal{D}\| = \mathbb{E}_{a \sim \mathcal{D}_{|X}} \|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|.$$

Putting everything together, we have

$$\begin{aligned} \|\mathcal{D} - \mathcal{D}'\| &\leq \|\mathcal{D}^{\text{aux}} - \mathcal{D}'\| + \|\mathcal{D}^{\text{aux}} - \mathcal{D}\| \\ &\leq \|\mathcal{D}_{|X} - \mathcal{D}'_{|X}\| + \mathbb{E}_{a \sim \mathcal{D}_{|X}} \|\mathcal{D}_{X=a} - \mathcal{D}'_{X=a}\|. \end{aligned}$$

□

Now we are ready to prove Theorem 1.6.

PROOF OF THEOREM 1.6. Instead of viewing the algorithm as a single round algorithm, we will prove a slightly stronger lower bound. Consider the model where we have n turns. On the t^{th} turn, processor t gets to send a single bit. This model is stronger than one round of the BCAST(1) model, since it allows the later processors to condition their outputs on earlier the processors' messages. Hence, our lower bound implies a lower bound for the BCAST(1) model as well.

Let $\mathcal{P}_{\text{rand}}^{(t)}$ and $\mathcal{P}_C^{(t)}$ be the distributions of the transcript of the first t turns when the input is drawn from $\mathcal{A}_{\text{rand}}$ or $\mathcal{A}_C$, respectively. Note that to prove the theorem, it suffices to show that $\mathcal{P}_{\text{rand}}^{(n)}$ is close to most $\mathcal{P}_C^{(n)}$. For this purpose, we are going to prove the following inequality holds for any $t \leq n$:

⁸For simplicity, we let $\mathcal{D}_{X=a}$ be the uniform distribution on Y if $\Pr_{(x, y) \sim \mathcal{D}}[x = a] = 0$.

$$\mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|\mathcal{P}_{\text{rand}}^{(t)} - \mathcal{P}_C^{(t)}\| \leq t \cdot c_1 \cdot \frac{k^2}{n} \cdot \frac{1}{\sqrt{n}}, \quad (1)$$

where c_1 is a large enough universal constant. This inequality states that when picking a clique at random, the distribution of transcripts is similar to the distribution of transcripts when the input is chosen uniformly at random. It is easy to see that plugging in $t = n$, (1) implies the theorem.

We prove the inequality above inductively. Clearly, (1) holds when $t = 0$. So it suffices to show that when it holds for $t - 1$, it also holds for t .

Let $\mathcal{D}_t$ and $\mathcal{D}_t^C$ be the input distributions to processor t in $\mathcal{A}_{\text{rand}}$ and $\mathcal{A}_C$, respectively. For a fixed $C \subseteq [n]$, by Lemma 1.9, we have:

$$\|\mathcal{P}_{\text{rand}}^{(t)} - \mathcal{P}_C^{(t)}\| \leq \|\mathcal{P}_{\text{rand}}^{(t-1)} - \mathcal{P}_C^{(t-1)}\| + \mathbb{E}_{p \sim \mathcal{P}_{\text{rand}}^{(t-1)}} \mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|f_t^p(\mathcal{D}_t) - f_t^p(\mathcal{D}_t^C)\|. \quad (2)$$

We think of $\|\mathcal{P}_{\text{rand}}^{(t)} - \mathcal{P}_C^{(t)}\|$ as the amount of evidence the transcript is giving us about whether the distribution is uniform or if it has C as a clique. (If this value were 0, that would mean the transcript gives us no evidence. If the value were 1, that would mean we have "full" evidence and could distinguish between the two with no error given the transcript). So, with this interpretation, the inequality above is basically stating that the amount of evidence we have after round t is the sum of the evidence from all rounds up to $t - 1$, plus the extra evidence we get from the broadcast of the processor in round t .

By definition, $\mathcal{D}_t$ is the uniform distribution on the set $\{x : x \in \{0, 1\}^n, x_t = 0\}$. And $\mathcal{D}_t^C = \mathcal{D}_t$ if $t \notin C$, and is the uniform distribution on the set $\{x : x \in \{0, 1\}^n, x_t = 0, x_j = 1 \text{ for all } j \in C \setminus \{t\}\}$ otherwise.

We care not about the probability of distinguishing a particular clique existing, but about whether any clique exists, so we take the expected value over all possible cliques of both sides of (2) gives:

$$\begin{aligned} &\mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|\mathcal{P}_{\text{rand}}^{(t)} - \mathcal{P}_C^{(t)}\| \\ &\leq \mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|\mathcal{P}_{\text{rand}}^{(t-1)} - \mathcal{P}_C^{(t-1)}\| + \mathbb{E}_{p \sim \mathcal{P}_{\text{rand}}^{(t-1)}} \mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|f_t^p(\mathcal{D}_t) - f_t^p(\mathcal{D}_t^C)\| \end{aligned} \quad (3)$$

We can bound $\mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|\mathcal{P}_{\text{rand}}^{(t-1)} - \mathcal{P}_C^{(t-1)}\|$ by the inductive hypothesis, so it suffices to bound

$\mathbb{E}_{p \sim \mathcal{P}_{\text{rand}}^{(t-1)}} \mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \|f_t^p(\mathcal{D}_t) - f_t^p(\mathcal{D}_t^C)\|$. For $p \sim \mathcal{P}_{\text{rand}}^{(t-1)}$, there are two cases:

- When $t \notin C$, which happens with probability $1 - \frac{k}{n}$, we have

$$\|f_t^p(\mathcal{D}_t) - f_t^p(\mathcal{D}_t^C)\| = 0,$$

as $\mathcal{D}_t^C = \mathcal{D}_t$. That is, since t is not in C , what t says gives us no information about whether C is a clique (whether there is a clique or not does not affect the input of t , and therefore does not affect their message).

- When $t \in C$, which happens with probability $\frac{k}{n}$, by Lemma 1.8, we have

$$\mathbb{E}_{C \sim \mathcal{S}_{k-1}^{[n] \setminus \{t\}}} \left\| f_t^p(\mathcal{D}_t) - f_t^p(\mathcal{D}_t^{C \cup \{t\}}) \right\| \leq O\left(\frac{k}{\sqrt{n}}\right).$$

That is, when t is in C , while t might give information about whether C is a clique, there are many cliques that may include t , and the inequality states that t cannot give too much information about many of cliques (the expected amount of information revealed about a randomly chosen clique of size k containing t is bounded by $O\left(\frac{k}{\sqrt{n}}\right)$).

So, now we have bounded how much evidence the t -th processor reveals when broadcasting. We know that when the clique is chosen randomly, with probability $1 - k/n$ no information is revealed, and with probability k/n at most $O(k/\sqrt{n})$ information is revealed in expectation. Combining these facts gives:

$$\mathbb{E}_{p \sim \mathcal{P}^{(t-1)}_{\text{rand}}} \mathbb{E}_{C \sim \mathcal{S}_k} \left\| f_t^p(\mathcal{D}_t) - f_t^p(\mathcal{D}_t^C) \right\| \leq \frac{k}{n} \cdot O\left(\frac{k}{\sqrt{n}}\right),$$

which, plugging into (3) and using the inductive hypothesis proves inequality (1) for t . $\square$

1.3.1 Proof of Lemma 1.8. We need the following lemma first, whose proof is based on tools from information theory, and is deferred to the end of this subsection.

LEMMA 1.10. *Let n be an integer, and $\mathcal{U}_n^{[i]}$ be the uniform distribution on $\{x : x \in \{0, 1\}^n, x_i = 1\}$. For all function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we have*

$$\mathbb{E}_{i \leftarrow [n]} \left\| f(\mathcal{U}) - f(\mathcal{U}^{[i]}) \right\| \leq O\left(\frac{1}{\sqrt{n}}\right).$$

That is, if we consider a function f , suppose that on a uniform distribution, the probability it outputs 1 is p . Then, if we pick a random index and set it to 1, we still expect that if we randomly pick the rest of the coordinates, the output will be 1 with probability approximately p .

Now we are ready to prove Lemma 1.8 (restated below).

Reminder of Lemma 1.8 *Let n, k be integers such that $k \leq n^{1/4}$, and $\mathcal{U}_n^C$ be the uniform distribution on $\{x : x \in \{0, 1\}^n, x_i = 1 \text{ for all } i \in C\}$. For all function $f : \{0, 1\}^n \rightarrow \{0, 1\}$, we have*

$$\mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \left\| f(\mathcal{U}_n) - f(\mathcal{U}_n^C) \right\| \leq O\left(\frac{k}{\sqrt{n}}\right).$$

The idea of the proof is that each bit we set to 1, by Lemma 1.10, will change the expected output of f by $O\left(\frac{1}{\sqrt{n}}\right)$. Hence, if we set k of those bits to 1, that will change the expected outcome by $O\left(\frac{1}{\sqrt{n}}\right)$ at most k times, for a total of $O\left(\frac{k}{\sqrt{n}}\right)$. A formal proof is included below:

PROOF OF LEMMA 1.8. Instead of choosing C from $\mathcal{S}_k^{[n]}$, we choose an ordered k -tuple of $a = (a_1, a_2, \dots, a_k)$ of k distinct elements in $[n]$ uniformly at random. Let the distribution be $\mathcal{T}_k^{[n]}$.

We have

$$\begin{aligned} & \mathbb{E}_{C \sim \mathcal{S}_k^{[n]}} \left\| f(\mathcal{U}_n) - f(\mathcal{U}_n^C) \right\| \\ &= \mathbb{E}_{a \sim \mathcal{T}_k^{[n]}} \left\| f(\mathcal{U}_n) - f(\mathcal{U}_n^{\{a_i\}_{i=1}^k}) \right\| \\ &\leq \sum_{\ell=1}^k \mathbb{E}_{a \sim \mathcal{T}_\ell^{[n]}} \left\| f(\mathcal{U}_n^{\{a_i\}_{i=1}^{\ell-1}}) - f(\mathcal{U}_n^{\{a_i\}_{i=1}^\ell}) \right\| \\ &\leq \sum_{\ell=0}^{k-1} \mathbb{E}_{a \sim \mathcal{T}_\ell^{[n]}} \mathbb{E}_{j \leftarrow [n] \setminus \{a_i\}_{i=1}^\ell} \left\| f(\mathcal{U}_n^{\{a_i\}_{i=1}^\ell}) - f(\mathcal{U}_n^{\{a_i\}_{i=1}^\ell \cup \{j\}}) \right\|. \end{aligned} \quad (4)$$

Now we are to bound the right side of (5) for each $0 \leq \ell \leq k-1$ separately. Applying Lemma 1.10 on the restriction of f such that all bits in $\{a_i\}$ are set to 1, we have

$$\mathbb{E}_{j \leftarrow [n] \setminus \{a_i\}_{i=1}^\ell} \left\| f(\mathcal{U}_n^{\{a_i\}_{i=1}^\ell}) - f(\mathcal{U}_n^{\{a_i\}_{i=1}^\ell \cup \{j\}}) \right\| \leq O\left(\frac{1}{\sqrt{n-\ell}}\right).$$

Plugging the above in (5) and noting that $k \leq n^{1/4}$ completes the proof. $\square$

The proof of Lemma 1.10 makes use several tools from information theory, we first introduce them here.

Preliminaries on Information Theory. Let X, Y be random variables in the same probability space Ω . The entropy of X , denoted by $H(X)$, is defined as $H(X) := \Pr_{a \sim X} \log \frac{1}{\Pr[X=a]}$. The conditional entropy of X given Y , denoted as $H(X|Y)$, is defined as $H(X|Y) := \mathbb{E}_{y \sim Y} H(X|Y=y)$.

The mutual information between X and Y , denoted by $I(X; Y)$, is defined as $I(X; Y) := H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X, Y)$.

For two distributions $\mathcal{P}$ and $\mathcal{Q}$ on the same set S , their Kullback-Leibler (KL) divergence is defined as

$$D(\mathcal{P}||\mathcal{Q}) := \sum_{s \in S} \mathcal{P}(s) \cdot \log \frac{\mathcal{P}(s)}{\mathcal{Q}(s)}.$$

KL divergence is related to mutual information in the following way.

FACT 1.

$$I(X; Y) := \mathbb{E}_{x \sim X} D(Y|X=x||Y),$$

LEMMA 1.11 (PINSKER'S INEQUALITY). *For two distributions $\mathcal{D}_1$ and $\mathcal{D}_2$, we have*

$$\|\mathcal{D}_1 - \mathcal{D}_2\| \leq \sqrt{\frac{1}{2} \cdot D(\mathcal{D}_1||\mathcal{D}_2)}.$$

Now we are ready to prove Lemma 1.10.

PROOF OF LEMMA 1.10. Throughout the proof we will assume X is a random variable drawn uniformly from $\{0, 1\}^n$. For $i \in [n]$, let X_i be the random variable of the i -th bit of X .

We have

$$I(X_i; f(X)) = H(X_i) - H(X_i|f(X)) = 1 - H(X_i|f(X)).$$

And by the sub-additivity of conditional entropy, we have

$$\sum_{i=1}^n H(X_i | f(X)) \geq H(X | f(X)) \geq n - 1.$$

Therefore,

$$\sum_{i=1}^n I(X_i; f(X)) \leq n - (n - 1) \leq 1.$$

or equivalently,

$$\mathbb{E}_{i \leftarrow [n]} I(X_i; f(X)) \leq \frac{1}{n}.$$

Note that by Fact 1,

$$I(X_i; f(X)) := \mathbb{E}_{x \sim X_i} D(f(X)_{X_i=x} || f(X)).$$

Taking expected values over i of both sides and using

$$\mathbb{E}_{i \leftarrow [n]} I(X_i; f(X)) \leq \frac{1}{n}$$

gives

$$\mathbb{E}_{i \leftarrow [n]} I(X_i; f(X)) = \mathbb{E}_{i \leftarrow [n]} \mathbb{E}_{x \sim X_i} D(f(X)_{X_i=x} || f(X)) \leq \frac{1}{n}.$$

By Pinsker's inequality (Lemma 1.11) and the fact that $\sqrt{x}$ is a concave function, we have

$$\mathbb{E}_{i \leftarrow [n]} \mathbb{E}_{x \sim X_i} \|f(X)_{X_i=x} - f(X)\| \leq \sqrt{\frac{1}{n}},$$

and

$$\mathbb{E}_{i \leftarrow [n]} \frac{1}{2} \cdot \|f(X)_{X_i=1} - f(X)\| \leq \sqrt{\frac{1}{n}}.$$

Note that by definition, $f(X)_{X_i=1}$ is distributed identically to $f(\mathcal{U}^{[i]})$, which completes the proof. $\square$

1.4 Related Work

BCAST(1) Congested Clique: The specific distributed model we investigate is the Broadcast Congested Clique. In this model, there are n processors, and computation proceeds in rounds. In each round, each processor broadcasts a short message to all other processors. It has recently been studied in [3, 4, 8, 9, 12, 16, 21, 24, 25, 29, 30, 35, 37], among others. It has been used to study other areas in computer science such as streaming algorithms [1] and mechanism design [15].

Complexity Theoretic approaches in Distributed Computing: Recently, more complexity theoretic approaches and results have been made in the congested clique and distributed computation in general, for example in [10, 20, 23, 32].

Pseudo-randomness and Distributed Computing: In [5], the authors construct a pseudo-random generator which creates additional shared random bits in a distributed system. Specifically, the authors work in a setting where every pair of processors can privately communicate with each other (whereas we work in the broadcast model), and some of the processors may be adversarially faulty. They show how to use few shared random bits, and unlimited private random bits to efficiently compute more shared random bits. In our setting, we are saving on private random bits flipped (when all processors are non-faulty, it is easy to turn a private random bit into a public random bit – simply broadcast it).

In [27], the authors construct pseudo-randomness for a different distributed system, in which the network has a topology. Their main application is constructing pseudo-randomness that fools all low-space computation.

In [36], a different setting than ours is considered, in which the processors are computationally bounded, and a cryptographic pseudo-random function is being evaluated.

In [39], a pseudo-random generator that fools DNFs is used to deterministically construct spanners in the congested clique. In that work, the pseudo-random generator is used for the specific problem considered, as opposed to being a pseudo-random generator which fools all algorithms in the model.

Pseudo-randomness in the context of complexity theory has been very widely studied. See Vadhan's survey [40].

In [22], the authors introduce general methods for derandomizing algorithms in the LOCAL model to obtain better deterministic algorithms.

Planted Clique: The planted clique problem (or hidden clique problem) was introduced in [28] and [33]. The best known classical algorithm [13, 17] can find the hidden clique when its size is $k = \Omega(\sqrt{n})$ in near linear-time. For $k \ll \sqrt{n}$, the naïve algorithm (looking for a clique of size $10 \log n$ with brute force, and then extending that clique to the whole clique) can solve it in $n^{O(\log n)}$ time, and the problem is conjectured to be not solvable in polynomial time. However, since it is an average-case problem, it is unlikely that the hardness of this problem can be derived from standard complexity assumptions such as $P \neq NP$ [6, 19]. Therefore, much work has been put into trying to show limitations for certain classes of algorithm on this problem [2, 14, 18, 26, 34], or showing tight hardness for closely-related worst-case problems under standard assumptions [7].

Distributed Clique lower bounds: There is some literature on lower bounds for finding cliques in the congested clique model [16], as well as the standard CONGEST model [11]. These lower bounds hold in the worst case, and have no direct implications about the hardness of the planted clique problem.

ACKNOWLEDGMENTS

The first author is supported by NSF CCF-1741615. The second author is supported by the Fannie and John Hertz Foundation Fellowship, an NSF Graduate Research Fellowship, NSF CNS-1413920, DARPA 491512803, Sloan 996698, and MIT/IBM W1771646. This work was done in part while the authors were visiting the Simons Institute for the Theory of Computing.

We would like to thank Shafi Goldwasser, Yang Liu, and Merav Parter for helpful discussions. Many thanks to Sidhanth Mohanty for very fruitful discussions about the planted clique problem.

Thanks to anonymous reviewers for many useful comments.

REFERENCES

- [1] Noga Alon, Yossi Matias, and Mario Szegedy. 1999. The Space Complexity of Approximating the Frequency Moments. *J. Comput. Syst. Sci.* 58, 1 (1999), 137–147. <https://doi.org/10.1006/jcss.1997.1545>
- [2] Boaz Barak, Samuel B. Hopkins, Jonathan A. Kelner, Pravesh Kothari, Ankur Moitra, and Aaron Potechin. 2016. A Nearly Tight Sum-of-Squares Lower Bound for the Planted Clique Problem. In *IEEE 57th Annual Symposium on Foundations*

- of Computer Science, FOCS 2016, 9–11 October 2016, Hyatt Regency, New Brunswick, New Jersey, USA. 428–437. <https://doi.org/10.1109/FOCS.2016.53>
- [3] Florent Becker, Antonio Fernández Anta, Ivan Rapaport, and Eric Rémila. 2015. Brief Announcement: A Hierarchy of Congested Clique Models, from Broadcast to Unicast. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015, Donostia-San Sebastián, Spain, July 21 – 23, 2015*. 167–169. <https://doi.org/10.1145/2767386.2767447>
 - [4] Florent Becker, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. 2018. The Impact of Locality on the Detection of Cycles in the Broadcast Congested Clique Model. In *Latin American Symposium on Theoretical Informatics*. Springer, 134–145.
 - [5] Mihir Bellare, Juan A. Garay, and Tal Rabin. 1996. Distributed Pseudo-Random Bit Generators - A New Way to Speed-Up Shared Coin Tossing. In *Proceedings of the Fifteenth Annual ACM Symposium on Principles of Distributed Computing, Philadelphia, Pennsylvania, USA, May 23–26, 1996*. 191–200. <https://doi.org/10.1145/248052.248090>
 - [6] Andrej Bogdanov and Luca Trevisan. 2006. On Worst-Case to Average-Case Reductions for NP Problems. *SIAM J. Comput.* 36, 4 (2006), 1119–1159. <https://doi.org/10.1137/S0097539705446974>
 - [7] Mark Braverman, Young Kun-Ko, Aviad Rubinfeld, and Omri Weinstein. 2017. ETH Hardness for Densest- k -Subgraph with Perfect Completeness. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16–19*. 1326–1341. <https://doi.org/10.1137/1.9781611974782.86>
 - [8] 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 2015, Donostia-San Sebastián, Spain, July 21 – 23, 2015*. 143–152. <https://doi.org/10.1145/2767386.2767414>
 - [9] Keren Censor-Hillel, Merav Parter, and Gregory Schwartzman. 2016. Derandomizing local distributed algorithms under bandwidth restrictions. *arXiv preprint arXiv:1608.01689* (2016).
 - [10] Yi-Jun Chang and Seth Pettie. 2017. A Time Hierarchy Theorem for the LOCAL Model. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15–17, 2017*. 156–167. <https://doi.org/10.1109/FOCS.2017.23>
 - [11] Artur Czumaj and Christian Konrad. 2018. Detecting cliques in CONGEST networks. *arXiv preprint arXiv:1807.01070* (2018).
 - [12] Pablo Moisset de Espanés, Ivan Rapaport, Daniel Remenik, and Javiera Urrutia. 2016. Robust reconstruction of Barabási-Albert networks in the broadcast congested clique model. *Networks* 67, 1 (2016), 82–91. <https://doi.org/10.1002/net.21662>
 - [13] Yael Dekel, Ori Gurel-Gurevich, and Yuval Peres. 2014. Finding Hidden Cliques in Linear Time with High Probability. *Combinatorics, Probability & Computing* 23, 1 (2014), 29–49. <https://doi.org/10.1017/S096354831300045X>
 - [14] Yash Deshpande and Andrea Montanari. 2015. Improved Sum-of-Squares Lower Bounds for Hidden Clique and Hidden Submatrix Problems. In *Proceedings of The 28th Conference on Learning Theory, COLT 2015, Paris, France, July 3–6, 2015*. 523–562.
 - [15] Shahar Dobzinski, Noam Nisan, and Sigal Oren. 2014. Economic efficiency requires interaction. In *Symposium on Theory of Computing, STOC 2014, New York, NY, USA, May 31 – June 03, 2014*. 233–242. <https://doi.org/10.1145/2591796.2591815>
 - [16] Andrew Drucker, Fabian Kuhn, and Rotem Oshman. 2014. On the power of the congested clique model. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15–18, 2014*. 367–376. <https://doi.org/10.1145/2611462.2611493>
 - [17] Uriel Feige and Robert Krauthgamer. 2000. Finding and certifying a large hidden clique in a semirandom graph. *Random Struct. Algorithms* 16, 2 (2000), 195–208.
 - [18] Uriel Feige and Robert Krauthgamer. 2003. The Probable Value of the Lovász-Schrijver Relaxations for Maximum Independent Set. *SIAM J. Comput.* 32, 2 (2003), 345–370. <https://doi.org/10.1137/S009753970240118X>
 - [19] Joan Feigenbaum and Lance Fortnow. 1993. Random-Self-Reducibility of Complete Sets. *SIAM J. Comput.* 22, 5 (1993), 994–1005. <https://doi.org/10.1137/0222061>
 - [20] Pierre Fraigniaud, Amos Korman, and David Peleg. 2013. Towards a complexity theory for local distributed computing. *J. ACM* 60, 5 (2013), 35:1–35:26. <https://doi.org/10.1145/2499228>
 - [21] François Le Gall. 2016. Further Algebraic Algorithms in the Congested Clique Model and Applications to Graph-Theoretic Problems. In *Distributed Computing - 30th International Symposium, DISC 2016, Paris, France, September 27–29, 2016. Proceedings*. 57–70. https://doi.org/10.1007/978-3-662-53426-7_5
 - [22] Mohsen Ghaffari, David G. Harris, and Fabian Kuhn. 2018. On Derandomizing Local Distributed Algorithms. In *FOCS, to appear*.
 - [23] Mohsen Ghaffari, Fabian Kuhn, and Yannic Maus. 2017. On the complexity of local distributed graph problems. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19–23, 2017*. 784–797. <https://doi.org/10.1145/3055399.3055471>
 - [24] Ofer Grossman, Bernhard Haeupler, and Sidhanth Mohanty. 2018. Algorithms for Noisy Broadcast with Erasures. In *LIPICs-Leibniz International Proceedings in Informatics*, Vol. 107. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
 - [25] Stephan Holzer and Nathan Pinsker. 2015. Approximation of Distances and Shortest Paths in the Broadcast Congest Clique. In *19th International Conference on Principles of Distributed Systems, OPODIS 2015, December 14–17, 2015, Rennes, France*. 6:1–6:16. <https://doi.org/10.4230/LIPICs.OPODIS.2015.6>
 - [26] Samuel B Hopkins, Pravesh Kothari, Aaron Henry Potechin, Prasad Raghavendra, and Tselil Schramm. 2018. On the integrality gap of degree-4 sum of squares for planted clique. *ACM Transactions on Algorithms (TALG)* 14, 3 (2018), 28.
 - [27] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. 1994. Pseudorandomness for network algorithms. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*. ACM, 356–364.
 - [28] Mark Jerrum. 1992. Large Cliques Elude the Metropolis Process. *Random Struct. Algorithms* 3, 4 (1992), 347–360. <https://doi.org/10.1002/rsa.3240030402>
 - [29] Tomasz Jurdzinski and Krzysztof Nowicki. 2017. Brief Announcement: On Connectivity in the Broadcast Congested Clique. In *LIPICs-Leibniz International Proceedings in Informatics*, Vol. 91. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik.
 - [30] Tomasz Jurdzinski and Krzysztof Nowicki. 2017. MSF and connectivity in limited variants of the congested clique. *arXiv preprint arXiv:1703.02743* (2017).
 - [31] Janne H Korhonen and Joel Rybicki. 2017. Deterministic subgraph detection in broadcast CONGEST. *arXiv preprint arXiv:1705.10195* (2017).
 - [32] Janne H Korhonen and Jukka Suomela. 2017. Towards a complexity theory for the congested clique. *arXiv preprint arXiv:1705.03284* (2017).
 - [33] Ludek Kucera. 1995. Expected Complexity of Graph Partitioning Problems. *Discrete Applied Mathematics* 57, 2–3 (1995), 193–212. [https://doi.org/10.1016/0166-218X\(94\)00103-K](https://doi.org/10.1016/0166-218X(94)00103-K)
 - [34] Raghu Meka, Aaron Potechin, and Avi Wigderson. 2015. Sum-of-squares Lower Bounds for Planted Clique. In *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing, STOC 2015, Portland, OR, USA, June 14–17, 2015*. 87–96. <https://doi.org/10.1145/2746539.2746600>
 - [35] Pedro Montealegre and Ioan Todinca. 2016. Deterministic graph connectivity in the broadcast congested clique. *arXiv preprint arXiv:1602.04095* (2016).
 - [36] Moni Naor, Benny Pinkas, and Omer Reingold. 1999. Distributed Pseudo-random Functions and KDCs. In *Advances in Cryptology - EUROCRYPT '99, International Conference on the Theory and Application of Cryptographic Techniques, Prague, Czech Republic, May 2–6, 1999, Proceeding*. 327–346. https://doi.org/10.1007/3-540-48910-X_23
 - [37] Jelani Nelson and Huacheng Yu. 2018. Optimal Lower Bounds for Distributed and Streaming Spanning Forest Computation. *arXiv preprint arXiv:1807.05135* (2018).
 - [38] Ilan Newman. 1991. Private vs. common random bits in communication complexity. *Information processing letters* 39, 2 (1991), 67–71.
 - [39] Merav Parter and Eylon Yogev. 2018. Congested Clique Algorithms for Graph Spanners. *CoRR abs/1805.05404* (2018). [arXiv:1805.05404](https://arxiv.org/abs/1805.05404)
 - [40] Salil P Vadhan. 2012. Pseudorandomness. *Foundations and Trends® in Theoretical Computer Science* 7, 1–3 (2012), 1–336.
 - [41] Andrew Chi-Chih Yao. 1977. Probabilistic Computations: Toward a Unified Measure of Complexity (Extended Abstract). In *18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, 31 October - 1 November 1977*. 222–227. <https://doi.org/10.1109/SFCS.1977.24>