

# The Impact of Locality on the Detection of Cycles in the Broadcast Congested Clique Model\*

F. Becker<sup>1</sup>, P. Montealegre<sup>2</sup>, I. Rapaport<sup>3</sup>, and I. Todinca<sup>1</sup>

<sup>1</sup> Univ. Orléans, INSA Centre Val de Loire, LIFO EA 4022, Orléans, France

<sup>2</sup> Facultad de Ingeniería y Ciencias, Universidad Adolfo Ibáñez, Santiago, Chile

<sup>3</sup> DIM-CMM (UMI 2807 CNRS), Universidad de Chile, Santiago, Chile

**Abstract.** The *broadcast congested clique* model is a message-passing model of distributed computation where  $n$  nodes communicate with each other in synchronous rounds. The joint input to the  $n$  nodes is an undirected graph  $G$  on the same set of nodes, with each node receiving the list of its *immediate neighbors in  $G$* . In each round each node sends *the same message* to all other nodes, depending on its own input, on the messages it has received so far, and on a public sequence of random bits. One parameter of this model is an upper bound  $b$  on the size of the messages, also known as bandwidth. In this paper we introduce another parameter to the model. We study the situation where the nodes, initially, instead of knowing their immediate neighbors, know their neighborhood up to a fixed radius  $r$ .

In this new framework we study one of the hardest problems in distributed graph algorithms, this is, the problem of detecting, in  $G$ , an induced cycle of length at most  $k$  ( $\text{CYCLE}_{\leq k}$ ) and the problem of detecting in  $G$  an induced cycle of length at least  $k + 1$  ( $\text{CYCLE}_{> k}$ ). For  $r = 1$ , we exhibit a deterministic, one-round algorithm for solving  $\text{CYCLE}_{\leq k}$  with  $b = \mathcal{O}(n^{2/k} \log n)$  if  $k$  is even, and  $b = \mathcal{O}(n^{2/(k-1)} \log n)$  if  $k$  is odd. We also prove, assuming the Erdős Girth Conjecture, that this result is tight for  $k \geq 4$ , up to logarithmic factors. More precisely, if each node, instead of being able to see only its immediate neighbors, is allowed to see up to distance  $r = \lfloor k/4 \rfloor$ , and if we also allowed randomization and multiple rounds, then the number of rounds  $R$  needed to solve  $\text{CYCLE}_{\leq k}$  must be such that  $R \cdot b = \Omega(n^{2/k})$  if  $k$  is even, and  $R \cdot b = \Omega(n^{2/(k-1)})$  if  $k$  is odd.

On the other hand, we show that, if each node is allowed to see up to distance  $r = \lfloor k/2 \rfloor + 1$ , then a polylogarithmic bandwidth is sufficient for solving  $\text{CYCLE}_{> k}$  with only two rounds. Nevertheless, if nodes were allowed to see up to distance  $r = \lfloor k/3 \rfloor$ , then any one-round algorithm that solves  $\text{CYCLE}_{> k}$  needs the bandwidth  $b$  to be at least  $\Omega(n/\log n)$ .

**Keywords:** broadcast congested clique; induced cycles; graph degeneracy

---

\* Additional support from CONICYT via Basal in Applied Mathematics (P.M. and I.R.) and Fondecyt 1170021 (I.R.).

## 1 Introduction

The *broadcast congested clique* model is a message-passing model of distributed computation where  $n$  nodes communicate with each other in synchronous rounds over a *complete network* [1, 2, 4, 6–8, 14, 16, 18, 21]. The joint input to the  $n$  nodes is an undirected graph  $G$  on the same set of nodes, with node  $u$  receiving the list of its neighbors in  $G$ . Nodes have pairwise distinct identities, which are numbers upper bounded by some polynomial in  $n$ . The identity of node  $u$  is denoted by  $\text{id}(u)$ . All nodes know  $n$ , the size of the network.

Each node broadcasts, in each round, a single  $b$ -bit message along each of its  $n - 1$  communication links. The size of the messages is known as the bandwidth of the system, and it is a parameter of the model (which could grow with  $n$ ). Broadcasting is equivalent to writing the messages on a whiteboard, visible to every node. In each round every node produces its message using its input, the contents of the whiteboard, and a sequence of public random bits.

Typically, the goal of an algorithm is to decide whether the input graph  $G$  belongs to some graph class  $\mathcal{C}$ . An algorithm is correct if it terminates with every node knowing the correct answer (that is, whether  $G \in \mathcal{C}$ ) with high probability. The round complexity of an algorithm is the maximum number of rounds over all possible input graphs (of size  $n$ ).

Few fast algorithms are known in the broadcast congested clique model. In fact, if the bandwidth  $b = \mathcal{O}(\log n)$ , then there exist one-round algorithms for deciding whether the input graph  $G$  has bounded degeneracy [6], contains a fixed forest [14], is a cograph [21]. Also, if  $b = \mathcal{O}(\text{polylog } n)$ , then there is a one-round algorithm for deciding whether  $G$  is connected [1, 2].

One way to increase the computing power of the model is to lift the broadcast restriction and to allow the nodes the possibility of sending *different* messages through different links. This general model, known as *unicast congested clique* [14], gives the possibility to perform a load balancing procedure efficiently. Such enormous intrinsic power has allowed some authors to provide fast algorithms for solving natural problems: an  $\mathcal{O}(\log \log \log n)$ -round algorithm for finding a 3-ruling set [17],  $\mathcal{O}(n^{0.158})$ -round algorithms for counting triangles, for counting 4-cycles and for computing the girth [12], an  $\mathcal{O}(1)$ -round algorithm for detecting a 4-cycle [12], an  $\mathcal{O}(1)$ -round algorithm constructing a minimum spanning tree [20].

Another very natural, much more limited and less dramatic way to increase the computing power of the broadcast congested clique model, *is to expand the local knowledge the nodes initially have about  $G$* . The idea of a constant-radius neighborhood independent of the size of the network is present in the research on local algorithms pioneered by Angluin [3], Linial [23] and Naor and Stockmeyer [25].

We therefore use the  $KT_r$  notion, introduced by Awebuch et. al. [5], which means *Knowledge of Topology* up to distance  $r$ , excluding edges with both endpoints at distance  $r$ . More precisely, we call  $\text{BCLIQUE}[r]$  the extension of the broadcast congested clique model where each node  $u$  “sees” (receives as input) the set of **all edges** lying on a path of length at most  $r$ , starting in  $u$ . Hence,

BCLIQUE[1] corresponds to the classical broadcast congested clique model, and is simply denoted BCLIQUE.

One of the most studied problems in the BCLIQUE model is related to the existence of cycles in the input graph  $G$ . The first natural question one can formulate, that is, deciding whether  $G$  contains a cycle has been, until now, the only question amenable to a simple algorithm. In fact, Becker et al. [6] show that a simple set of logarithmic size messages is sufficient to recognize, deterministically and in one round, whether the input graph  $G$  is acyclic.

Any other natural question concerning cycles has given strong negative results. Drucker et al. [14] showed that, if  $\ell \geq 4$ , then any algorithm that decides whether the  $\ell$ -node cycle  $C_\ell$  is a subgraph (or an induced subgraph) of the input graph  $G$  needs  $\Omega(ex(n, C_\ell)/nb)$  rounds, where  $ex(n, H)$  is the Turán number of  $H$ , i.e., the maximal number of edges of an  $n$ -node graph which does not contain a subgraph isomorphic to  $H$ . Remark that  $ex(n, C_\ell)$  is  $\Theta(n^2)$  for odd values  $\ell$ , and  $\Theta(n^{1+1/\ell})$  for even values (assuming the Erdős Girth Conjecture<sup>4</sup> [15]).

Moreover, *even in the very powerful unicast congested clique model*, the algorithms for cycle detection are rather slow. In fact, the best algorithm for detecting  $C_\ell$  uses  $\mathcal{O}(n^\rho \log n)$  rounds, for every  $\ell \geq 3$ , where  $\rho < 0.15715$  [12]. The only exception being the detection of squares  $C_4$ , for which an extremely elegant  $\mathcal{O}(1)$ -round algorithm has been devised [12].

In this paper, we mainly study two problems:  $\text{CYCLE}_{\leq k}$  and  $\text{CYCLE}_{> k}$ . The first one consists in deciding whether the graph contains an induced cycle of length at most  $k$  (i.e., deciding whether the *girth* of the graph is at most  $k$ ). The second problem, complementary to the first one, consists in detecting the existence of an induced cycle of length at least  $k + 1$ . This difficulty to find fast algorithms for problems related to the existence of cycles is what makes the positive results of this paper surprising.

Note that the existence of an induced cycle of length at most  $k$  is equivalent to the existence of a cycle (not necessarily induced) of length at most  $k$ . On the other hand, as we are going to explain later, finding an algorithm for detecting induced cycles of length at least  $k + 1$  requires much more involved arguments than finding algorithms for detecting cycles (not necessarily induced) of length at least  $k + 1$ .

## Our Results

In Section 3 we show that there is a deterministic, one-round BCLIQUE algorithm for solving problem  $\text{CYCLE}_{\leq k}$  with bandwidth  $\mathcal{O}(n^{2/k} \log n)$  if  $k$  is even, and bandwidth  $\mathcal{O}(n^{2/(k-1)} \log n)$  if  $k$  is odd. The main ingredient for proving this is a deterministic, one-round algorithm given in [24] that reconstructs a graph of degeneracy at most  $d$  in the BCLIQUE model using bandwidth  $\mathcal{O}(d \log n)$  (reconstruction means that every node knows all the edges of the input graph).

---

<sup>4</sup> This conjecture states that there exist graphs with  $n$  vertices and  $\Omega(n^{1+1/k})$  edges not containing cycles of length less than or equal to  $2k$ .

Recall that the degeneracy of  $G$  is the minimum  $d$  such that, by iteratively removing vertices of degree at most  $d$ , we obtain the empty graph.

We also show that previous upper bounds match the lower bounds up to logarithmic factors, even in the  $\text{BCLIQUE}[\lfloor k/4 \rfloor]$  model allowing randomization and multiple rounds. More precisely, if we allowed the nodes to see up to distance  $\lfloor k/4 \rfloor$ , to use public coins and multiple rounds, then the number of rounds  $R$  and bandwidth  $b$  needed to solve  $\text{CYCLE}_{\leq k}$  is such that  $R \cdot b = \Omega(n^{2/k})$  if  $k$  is even, and  $R \cdot b = \Omega(n^{2/(k-1)})$  if  $k$  is odd (in both cases  $k \geq 4$ ), for every  $\epsilon$ -error algorithm. (For these lower bounds we assume the Erdős Girth Conjecture).

We start Section 4 by giving a useful, “local” characterization of graphs which do not have long induced cycles. Using this, together with a technique inspired by the linear sketches of [1, 19], we show that, if each node is allowed to see at distance  $\lfloor k/2 \rfloor + 1$ , then a polylogarithmic number of bits is sufficient for detecting in two rounds an induced cycle of length strictly larger than  $k$ . More precisely, we prove that for every  $k \geq 3$ , there exists a two-round algorithm in the  $\text{BCLIQUE}[\lfloor k/2 \rfloor + 1]$  model that solves  $\text{CYCLE}_{>k}$  with high probability using bandwidth  $\mathcal{O}(\log^4 n)$ . The approach is based on the randomized algorithm of Ahn *et al.* [1] for computing a spanning forest in the  $\text{BCLIQUE}$  model with bandwidth  $\mathcal{O}(\log^3 n)$ . With respect to lower bounds, we prove that any one-round, public-coin  $\text{BCLIQUE}[\lfloor k/3 \rfloor]$  algorithm that solves  $\text{CYCLE}_{>k}$  needs the bandwidth to be at least  $\Omega(n/\log n)$ . Note that the case  $k = 3$  corresponds to decide whether the input graph  $G$  is chordal, i.e., whether the only induced cycles in  $G$  are triangles.

The results of this article are summarized in Tables 1 and 2.

	$\text{BCLIQUE}[r]$	# Rounds	Bandwidth	Randomized?
Upper bound Theorem	$r = 1$	1	$O(n^{2/k} \log n)$ , $k$ even $O(n^{2/(k-1)} \log n)$ , $k$ odd	Deterministic
Lower bound Theorem	$r \leq k/3$	1	$\Omega(n^{2/k} / \log n)$ , $k$ even $\Omega(n^{2/(k-1)} / \log n)$ , $k$ odd	Randomized $\epsilon$ -error
Lower bound Theorem	$r \leq k/4$	R	$\Omega(n^{2/k} / R)$ , $k$ even $\Omega(n^{2/(k-1)} / R)$ , $k$ odd	Randomized $\epsilon$ -error

**Table 1.** Results concerning problem  $\text{CYCLE}_{\leq k}$ . The lower bounds assume the Erdős Girth Conjecture.

	$\text{BCLIQUE}[r]$	# Rounds	Bandwidth	Randomized?
Upper bound Theorem	$r \geq \frac{k}{2} + 1$	2	$O(\log^4 n)$	Randomized (w.h.p.)
Lower bound Theorem	$r \leq k/3$	1	$\Omega(n/\log n)$	Randomized $\epsilon$ -error

**Table 2.** Results concerning problem  $\text{CYCLE}_{>k}$

## 2 Basic Definitions and Notations

Let  $G = (V, E)$  be an undirected graph, and let  $u \in V$ . We call  $N_G(u) = \{v \in V \mid uv \in E\}$  and  $N_G[u] = N_G(u) \cup \{u\}$ , the *open and closed neighborhoods* of  $u$ , respectively. Similarly, for  $U \subseteq V$ ,  $N_G(U) = \cup_{u \in U} N_G(u) - U$  and  $N_G[U] = N_G(U) \cup \{U\}$  are the *open and closed neighborhoods of  $U$* , respectively. When no ambiguity is possible, we will omit the subindices. By extension, we denote  $N^r[u]$  the set of vertices at distance at most  $r$  from  $u$ , and we call it *closed  $r$ -neighborhood* of  $u$ . Analogously,  $N^r(u) = N^r[u] \setminus \{u\}$  is the *open  $r$ -neighborhood* of  $u$ .

Graph  $H = (V', E')$  is a *subgraph* of  $G = (V, E)$  if  $V' \subseteq V$  and  $E' \subseteq E$ . If, for any edge  $uv \in E$  with  $u, v \in V'$  we also have  $uv \in E'$ , we say that  $H$  is an *induced* subgraph of  $G$ , or that  $H$  is the subgraph of  $G$  induced by  $V'$ . Given a vertex subset  $S$ , the subgraph induced by  $S$  is denoted  $G[S]$ . We simply write  $G - S$  for  $G[V \setminus S]$ . Also, if  $F$  is a subset of edges, we denote by  $G - F$  the graph obtained from  $G$  by removing the edges of  $F$ . The degeneracy of a graph  $G$  is the minimum  $d$  such that, by iteratively removing vertices of degree at most  $d$ , we obtain the empty graph

If  $S$  is a vertex subset of  $G = (V, E)$ , the *contraction* of  $S$  consists in replacing the whole subset  $S$  by a unique vertex  $v_S$ , such that the neighborhood of  $v_S$  in the new graph is  $N_G(S)$  while  $G - S$  remains unchanged. A *connected component* of  $G$  is the inclusion-maximal set of vertices inducing a connected graph. An induced path (resp. cycle) of graph  $G$  is called a *chordless* path (cycle). A graph is called  *$k$ -chordal* if it does not contain any induced cycle of length greater than  $k$ . The 3-chordal graphs are known as *chordal graphs*.

The BCLIQUE[ $r$ ] model is formally defined as follows. There are  $n$  nodes which are given distinct identities (IDs), that we assume for simplicity to be numbers between 1 and  $n$ . In this paper we consider the situation where the joint input to the nodes is a graph  $G$ . More precisely, *each node  $u$  receives as input the subgraph of radius  $r$  around itself* (i.e., all edges lying on a path of length at most  $r$ , starting in  $u$ ). Nodes execute an algorithm, broadcasting  $b$ -bit messages in synchronous rounds. Their goal is to compute some function  $f$  that depends on  $G$ . When an algorithm stops *every node must know  $f(G)$* . Function  $f$  defines the problem to be solved. A 0 – 1 function corresponds to a decision problem.

An algorithm may be deterministic or randomized. We distinguish two sub-cases of randomized algorithms: the private-coin setting, where each node flips its own coin; and the public-coin setting, where the coin is shared between all nodes. (In this work we are going to consider public-coin algorithms only). An  $\varepsilon$ -error algorithm  $\mathcal{A}$  that computes a function  $f$  is a randomized algorithm such that, for every input graph  $G$ ,  $\Pr\{\mathcal{A} \text{ outputs } f(G)\} \geq 1 - \varepsilon$ . In the case where  $\varepsilon \rightarrow 0$  as  $n \rightarrow \infty$ , we say that  $\mathcal{A}$  computes  $f$  with high probability (whp).

We consider several decision problems in this paper: CYCLE $_{=k}$ , CYCLE $_{\leq k}$  and CYCLE $_{>k}$ . These problems consist in deciding, respectively, whether the input graph has an induced cycle of length exactly  $k$ , at most  $k$ , and strictly larger than  $k$ . Problems SUB-CYCLE $_{=k}$ , SUB-CYCLE $_{\leq k}$  and SUB-CYCLE $_{>k}$  are defined

in a similar way, but in this case we ask whether the input graph has a cycle as a subgraph (induced or not) of length  $k$ , at most  $k$ , and strictly larger than  $k$ .

### 3 Detection of Short Cycles

Let us denote by  $ex(n, k)$  the maximum number of edges in an  $n$ -vertex graph not containing a cycle of length at most  $k$ . A very helpful result in the study of graphs without short cycles is the one that relates the nonexistence of short cycles in  $G$  with the degeneracy of  $G$ . More precisely, graphs with no cycles of length at most  $k$  (as subgraphs) have a relatively small degeneracy.

**Proposition 1 ([14]).** *Graphs with no cycles of length at most  $k$  are of degeneracy  $\mathcal{O}(ex(n, k)/n)$ .*

In [24] it is shown that graphs of degeneracy at most  $d$  can be recognized, and even reconstructed, by a one-round algorithm in the BCLIQUE model using bandwidth  $\mathcal{O}(d \cdot \log n)$ . Recall that reconstruction means that at the end of the algorithm, every node knows all the edges of the input graph.

**Theorem 1 ([24]).** *There is a one-round, deterministic algorithm in the model BCLIQUE, that reconstructs the input graph  $G$  if the graph is  $d$ -degenerate, and rejects otherwise, using bandwidth  $\mathcal{O}(d \cdot \log n)$ .*

By Proposition 1, the degeneracy of the NO-instances of  $\text{CYCLE}_{\leq k}$  is upper bounded by  $\mathcal{O}(ex(n, k)/n)$ . Therefore, from Theorem 1, we conclude the existence of a one-round algorithm for  $\text{CYCLE}_{\leq k}$  such that, each node, either (1) fully reconstructs the graph and decides the existence of a cycle of length at most  $k$  or (2) notices that the degeneracy of the input graph is larger than the bound required by the NO instances, and concludes that the input graph must be a YES instance. Therefore, we have the following corollary.

**Corollary 1.** *Problem  $\text{CYCLE}_{\leq k}$  can be solved with a one-round, deterministic algorithm in the BCLIQUE model using bandwidth  $\mathcal{O}((ex(n, k)/n) \log n)$ .*

Previous algorithm is rather restrictive. It is deterministic, it works in one-round and the information each node has about the graph is minimal, consisting in the 1-neighborhood. The question we ask here is the following: is it possible, by lifting previous restrictions, to decrease the *total* number of bits broadcasted by each node? Next results give a negative answer to this question. In other words, the one-round deterministic algorithm based on the degeneracy seems to be the best we can do.

Recall that  $\text{BCLIQUE}[r]$  is the extension of the broadcast congested clique model where each node  $u$  receives as input the set of all edges lying on a path of length at most  $r$ , starting in  $u$ . Our first result tackles the case where  $r \leq \lfloor k/4 \rfloor$ .

**Theorem 2.** *Let  $\epsilon \leq 1/3$  and  $0 < r \leq k/4$ . Then, any  $\epsilon$ -error,  $R$ -round,  $b$ -bandwidth algorithm in the  $\text{BCLIQUE}[r]$  model solving  $\text{CYCLE}_{\leq k}$  satisfies  $R \cdot b = \Omega(ex(n, k)/n)$ .*

In the case where the nodes have more knowledge of the graph, i.e., when  $k/4 \leq r \leq k/3$ , we obtain a tight bound for one-round algorithms.

**Theorem 3.** *Let  $\epsilon \leq 1/3$  and  $k/4 < r \leq k/3$ . Then, any  $\epsilon$ -error, one-round algorithm in the BCLIQUE[ $r$ ] model that solves CYCLE $_{\leq k}$  requires bandwidth  $b = \Omega(ex(n, k)/(n \log n))$ .*

*Remark 1.* Bondy and Simonovits [10] showed that  $ex(n, k) = \mathcal{O}(n^{1+2/k})$  if  $k$  is even, and  $ex(n, k) = \mathcal{O}(n^{1+2/(k-1)})$  if  $k$  is odd. On the other hand, the Erdős Girth Conjecture states that this bound is tight, implying the results of Table 1. Note that currently, the best constructions provide a lower bound for  $ex(n, k) = \Omega(n^{1+4/(3k-7)})$  if  $k$  is even, and  $ex(n, k) = \Omega(n^{1+4/(3k-9)})$  if  $k$  is odd [22].

## 4 Detection of Long Cycles

Recall that graphs without induced cycles of length greater than  $k$  are called  $k$ -chordal [11]. 3-chordal graphs, i.e., graphs in which every cycle (not necessarily induced) of 4 or more vertices has a chord, are called chordal graphs. It is known that a graph  $G$  is chordal if and only if, for each vertex  $u \in V$ , and each connected component  $C$  in  $G - N[u]$ , the neighborhood  $N(C)$  of this component induces a clique in  $G$ . This “local” characterization has been exploited by Chandrasekharan *et al.* [13] for devising a fast parallel algorithm recognizing chordal graphs. We begin this section by extending previous characterization to arbitrary chordalities  $k > 3$  in order to take advantage of this in our distributed framework.

Let  $G$  be a graph,  $u \in V(G)$  and  $k > 0$ . Let  $D_1, \dots, D_p$  be the  $p$  connected components of  $G - N^{\lfloor k/2 \rfloor}[u]$  (obtained by removing the vertices at distance at most  $\lfloor k/2 \rfloor$  from  $u$ ). Let  $H_u^k$  denote the graph obtained from  $G$  by contracting each component  $D_i$  into a single node  $d_i$ .

**Lemma 1.** *Let  $G$  be a graph.  $G$  is  $k$ -chordal if and only if, for every  $u \in V(G)$ ,  $H_u^k$  is  $k$ -chordal.*

Lemma 1 provides us with a strategy for deciding  $k$ -chordality, i.e., for deciding whether the input graph  $G$  is a NO instance of problem CYCLE $_{>k}$ . For doing this every node  $x$  must compute the graph  $H_x^k$  and then decides whether  $H_x^k$  is  $k$ -chordal. In order to compute  $H_x^k$ , each node  $x$  needs first to find the connected components of  $G - N^{\lfloor k/2 \rfloor}[x]$ . Let  $F_x$  is the set of all edges lying on a path of length at most  $\lfloor k/2 \rfloor + 1$  starting in  $x$ . We need then each node to compute the connected components of  $G - F_x$  outside  $N^{\lfloor k/2 \rfloor}[x]$ .

### 4.1 Computing the Connected Components of $G - F_x$

Ahn *et al.* provide a probabilistic, one-round algorithm for computing a spanning forest of the input graph  $G$ , in the BCLIQUE model using bandwidth  $\mathcal{O}(\log^3 n)$  [1]. In their algorithm, each node constructs a message based on its

neighborhood and on a sequence of public random coins, and broadcasts it to all other nodes. Using all these messages, every node is able to construct a spanning forest of the graph with probability  $1 - \epsilon$ , for a fixed  $\epsilon > 0$ .

We want each node  $x$  to compute the connected components of  $G - F_x$ . Recall that  $F_x$  is the set of all edges lying on a path of length at most  $\lfloor k/2 \rfloor + 1$  starting in  $x$ . We place ourselves in the  $\text{BCLIQUE}[\lfloor k/2 \rfloor + 1]$  model with bandwidth  $\mathcal{O}(\log^4 n)$ . We amplify the bandwidth by a  $\log(n)$  factor, with respect to the spanning tree algorithm of [1], to ensure that it succeeds with high probability. Also, every node needs to know all the set of edges  $F_x$ , that is why we choose the  $\text{BCLIQUE}[\lfloor k/2 \rfloor + 1]$  model. Using the spanning forest algorithm of [1], we prove that each node  $x$  can construct a spanning forest of  $G - F_x$  with high probability.

The key observation is that the messages produced by each vertex is a linear function (w.r.t. to the edges of the graph). Therefore, from the messages of  $G$ , each vertex  $x$  computes the messages that the algorithm *would have constructed* on  $G - F_x$ .

**Definition 1.** Let  $n, k, \delta > 0$ . A  $\delta$ -linear sketch of size  $k$  is a function  $S : \{0, 1\}^{\mathcal{O}(\log n)} \times \{-1, 0, 1\}^n \rightarrow \{0, 1\}^k$ , such that, if we call  $S_r = S(r, \cdot)$ , then

- $S_r$  is linear, for each  $r \in \{0, 1\}^{\mathcal{O}(\log n)}$ ;
- If  $r$  is chosen uniformly at random, then there is an algorithm that on input  $S_r(x)$  returns **ERROR** with probability at most  $\delta$ , and otherwise returns a pair  $(i, x_i)$  such that  $x_i \neq 0$  and coordinate  $i$  is picked uniformly at random between the non-zero coordinates of  $x$ . The probabilities are taken over the random choices of  $r$ .

**Proposition 2 ([19]).** For each  $n, \delta > 0$ , there exists a  $\delta$ -linear sketch of size  $\mathcal{O}(\log^2 n \log \delta^{-1})$ .

Let  $G = (V, E)$  be a graph of size  $n$ , and  $x \in V$ . We call  $a^x$  the *connectivity vector* of  $x$  in  $G$ , defined as the vector of dimension  $\binom{V}{2}$  such that:

$$a_{\{u,v\}}^x = \begin{cases} 1 & \text{if } \{u,v\} \in E, x = u \text{ and } u < v, \\ -1 & \text{if } \{u,v\} \in E, x = v \text{ and } u < v, \\ 0 & \text{otherwise.} \end{cases}$$

For  $r \in \{0, 1\}^{\mathcal{O}(\log n)}$ , we say that  $S_r(G) = \{S_r(a^x)\}_{x \in V(G)}$  is a  $\delta$ -connectivity sketch of  $G$ , where  $S$  is a  $\delta$ -linear sketch. Note that for any  $x \in V$ , each non zero coordinate of  $a^x$  represents an edge of  $N(x)$ , and for any  $U \subseteq V$  the non zero coordinates of  $\sum_{x \in U} a^x$  are exactly the edges in the cut between  $U$  and its complement  $V \setminus U$ .

Let  $G = (V, E)$  be the input graph. The one-round algorithm in the  $\text{BCLIQUE}$  model devised by Ahn et al. for computing a spanning forest of  $G$  works as follows. Let  $t = \lceil \log n \rceil$ . Each node computes and sends  $t$  independent  $\delta$ -linear sketches of its connectivity vector, using  $t$  random strings  $r_1, \dots, r_t$  picked uniformly at random. Using these messages, any node can compute  $t$  independent  $\delta$ -connectivity sketches of  $G$  and therefore it can compute a spanning tree using



the following  $t$  steps procedure. First, let us denote by  $\hat{V}$  the set of *supernodes*, which initially are the  $n$  singletons  $\{\{u\} | u \in V\}$ . At step  $0 \leq i < t$ , each node samples an incident edge to each set  $\hat{v} \in \hat{V}$  using the  $i$ th collection of linear sketches  $\sum_{x \in \hat{v}} S_{r_i}(a^x)$ , and merge the obtained connected components into a single supernode. The procedure finishes before  $t = \lceil \log n \rceil$  steps since the number of supernodes at least halves at each step. This idea is behind the proof of the following proposition.

**Proposition 3 (Ahn et al. [1]).** *Let  $n, \delta > 0$  and  $t = \lceil \log n \rceil$ . There exists an algorithm that receives  $t$  independent  $\delta$ -connectivity sketches of a graph  $G$ , produced with  $r_1, \dots, r_t \in \{0, 1\}^{\mathcal{O}(\log n)}$  random strings picked uniformly at random, and outputs a spanning forest of  $G$  with probability  $1 - \delta$ .*

**Lemma 2.** *There exists a one-round algorithm in the  $\text{BCLIQUE}[\lceil k/2 \rceil + 1]$  model which computes, for every node  $x \in V$ , the connected components of  $G - N^{\lfloor k/2 \rfloor}[x]$ , using bandwidth  $\mathcal{O}(\log^4 n)$  and with high probability.*

*Proof.* The algorithm works as follows. First, each node  $x$  sends  $t = \lceil \log n \rceil$  different  $1/n^2$ -linear sketches of its connectivity vector  $a^x$ , using  $t$  random strings  $r_1, \dots, r_t$ . Note that each node knows  $F_x$ . Observe that the components of  $G - N^{\lfloor k/2 \rfloor}[x]$  are exactly the components of  $G - F_x$  without considering the nodes in  $N^{\lfloor k/2 \rfloor}[x]$ . In the following, we show that after the communication round, each node  $x$  can compute a spanning forest of  $G - F_x$  with probability at least  $1 - 1/n^2$ . Therefore, the whole algorithm succeeds with probability at least  $1 - 1/n$ .

Let  $S_r(G) = (S_r(a^{x_1}), \dots, S_r(a^{x_n}))$  be one of the  $1/n^2$ -connectivity sketches of  $G$ , produced with the random string  $r$ , received in the communication round. Consider, for each  $e \in F_x$  and  $u \in e$ , the vector  $b^{u,e}$  of dimension  $\binom{n}{2}$  where,

$$b_{e'}^{u,e} = \begin{cases} -a_e^u & \text{if } e' = e, \\ 0 & \text{otherwise} \end{cases}, \text{ for each } e' \in \binom{n}{2}.$$

Let us call  $c^u$  be the connectivity vector of node  $u$  in  $G - F_x$ . Note that, for each  $e \in \binom{n}{2}$ ,

$$c_e^u = a_e^u + \sum_{\{e' \in F_x : u \in e'\}} b_e^{u,e'} = \begin{cases} a_e^u & \text{if } e \in E(G) \setminus F_x, \\ 0 & \text{otherwise.} \end{cases}$$

If we define  $S_r^u = S_r(a^u) + \sum_{\{e \in F_x : u \in e\}} S_r(b^{u,e})$ , we obtain, by linearity of  $S_r$ , that  $S_r^u = S_r(c^u)$  and then  $\{S_r(c^u)\}_{u \in V}$  is a  $1/n^2$ -connectivity sketch of  $G - F_x$  produced with  $r$ .

Then, after the communication round, any node  $x$  can obtain  $t$  different  $1/n^2$ -connectivity sketches of  $G - F_x$  produced with random strings  $r_1, \dots, r_t$  picked uniformly at random. Therefore, by Proposition 3, it can produce a spanning forest of that graph with probability at least  $1 - 1/n^2$ .  $\square$

## 4.2 Deciding $k$ -Chordality

We are now able to express the distributed algorithm recognizing  $k$ -chordal graphs, see Algorithm 1.

**Theorem 4.** *Let  $k \geq 3$ . There exists a two-round randomized algorithm in the  $\text{BCLIQUE}[\lfloor k/2 \rfloor + 1]$  model, that recognizes  $k$ -chordal graphs, and thus solves problem  $\text{CYCLE}_{>k}$ , with bandwidth  $\mathcal{O}(\log^4 n)$  and high probability.*

*Proof.* In the first round, each node  $x \in G$  computes the connected components of  $G - N^{\lfloor k/2 \rfloor}[x]$  using the algorithm of Lemma 2. After the first round, each node  $x$  uses its knowledge of  $G$  to locally reconstruct  $H_x^k$  by identifying the connected components  $D_1, \dots, D_p$  of  $G - N^{\lfloor k/2 \rfloor}[x]$  and contracting each  $D_i$  into a unique vertex  $d_i$ . Note that  $x$  sees the edges between  $D_i$  and  $N^{\lfloor k/2 \rfloor}[x]$ . Finally,  $x$  checks whether  $H_x^k$  is  $k$ -chordal and communicates the answer in the second round. By Lemma 1, the input graph is chordal if and only if each vertex  $x$  communicated a YES answer. We emphasize that the second round is needed only because the nodes must all agree on the output.

The algorithm may fail only when some node  $x$  fails to compute the components of  $G - N^{\lfloor k/2 \rfloor}[x]$ ; this event may occur, from Lemma 2, with probability at most  $1/n$ .  $\square$

---

### Algorithm 1: $k$ -chordality

---

- 1 **Round 1**
  - 2     Run the algorithm of Lemma 2 to compute the components of  $G - N^{\lfloor k/2 \rfloor}[x]$ ;
  - 3 **Round 2**
  - 4     Each node  $x$  builds  $H_x^k$  contracting each component of  $G - N^{\lfloor k/2 \rfloor}[x]$  into a single node;
  - 5     Each node  $x$  checks whether  $H_x^k$  is  $k$ -chordal and communicates the answer to the other nodes;
  - 6     The graph is  $k$ -chordal if all messages communicated during this round are YES messages;
- 

We end this section giving a lower-bound on the bandwidth  $b$  for any one-round algorithm solving  $\text{CYCLE}_{>k}$  in the  $\text{BCLIQUE}_r$  model, when  $0 < r \leq k/3$ .

**Theorem 5.** *Let  $\epsilon \leq 1/3$ , and  $0 < r \leq k/3$ . Any  $\epsilon$ -error, one-round algorithm in the  $\text{BCLIQUE}[r]$  model that solves  $\text{CYCLE}_{>k}$  requires bandwidth  $\Omega(n/\log n)$ .*

## 5 Conclusion

All throughout the paper we considered problems  $\text{CYCLE}_{\leq k}$  and  $\text{CYCLE}_{>k}$ . Let us briefly discuss the similar problems  $\text{SUB-CYCLE}_{\leq k}$ ,  $\text{SUB-CYCLE}_{>k}$  and

SUB-CYCLE<sub>=k</sub>, which consist in deciding whether the input graphs has, as a sub-graph, a cycle of length at most  $k$ , greater than  $k$ , and equal to  $k$ , respectively.

Observe that SUB-CYCLE<sub>≤k</sub> is identical to CYCLE<sub>≤k</sub>, so upper and lower bounds coincide. We emphasize that, for  $k ≥ 3r$ , the lower and upper bounds for these problems are tight up to polylogarithmic factors.

Unlike the case of short cycles, there is a significative difference between detecting long induced cycles and detecting long cycles (induced or not). By a result of Birmelé [9], graphs with no cycles of length greater than  $k$  have treewidth (and hence degeneracy) at most  $k$ . Therefore, they can be recognized by a one-round deterministic algorithm in the BCLIQUE model with bandwidth  $O(k \log n)$ , based on Theorem 1.

Further lower bounds can be obtained for both CYCLE<sub>=k</sub> and SUB-CYCLE<sub>=k</sub> problems in the BCLIQUE[ $r$ ] model, when  $k$  is an odd number between  $3r$  and  $4r$ . These bounds are obtained by a reduction from a 3-party Number-On-the-Forehead version of the disjointness problem DISJ, and show that any deterministic  $R$ -round  $b$ -bandwidth algorithm for this problem, in the BCLIQUE[ $r$ ] model, is such that  $R \cdot b = \Omega(n^{1-o(1)})$ . Under some stronger complexity assumptions, this lower bound can be extended to randomized algorithms.

When  $k$  is even, problem SUB-CYCLE<sub>=k</sub> can be solved by a one-round deterministic algorithm in BCLIQUE with bandwidth  $O(n^{2/k} \log n)$ , thanks to degeneracy arguments.

We leave as open problems the question whether CYCLE<sub>>k</sub> can be solved by a non-trivial one-round algorithm in the BCLIQUE[ $\lceil k/2 \rceil + 1$ ] model, as well as the question of multi-round lower bounds for this problem in the BCLIQUE[ $r$ ] model for  $r < k/2$ .

## References

1. Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Analyzing graph structure via linear measurements. In *Proceedings of the 23rd Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2012, pages 459–467, 2012.
2. Kook Jin Ahn, Sudipto Guha, and Andrew McGregor. Graph sketches: Sparsification, spanners, and subgraphs. In *Proceedings of the 31st Symposium on Principles of Database Systems*, PODS 2012, pages 5–14, 2012.
3. Dana Angluin. Local and global properties in networks of processors. In *Proceedings of the twelfth annual ACM symposium on Theory of computing*, pages 82–93. ACM, 1980.
4. Heger Arfaoui, Pierre Fraigniaud, David Ilcinkas, and Fabien Mathieu. Distributedly Testing Cycle-Freeness. In *Proceedings of the 40th International Workshop on Graph-Theoretic Concepts in Computer Science*, volume 8747 of LNCS, pages 15 – 28, Nouan-le-Fuzelier, France, June 2014.
5. Baruch Awerbuch, Oded Goldreich, Ronen Vainish, and David Peleg. A trade-off between information and communication in broadcast protocols. *J. ACM*, 37(2):238–256, 1990.
6. Florent Becker, Adrian Kosowski, Martín Matamala, Nicolas Nisse, Ivan Raporport, Karol Suchan, and Ioan Todinca. Allowing each node to communicate only

- once in a distributed system: shared whiteboard models. *Distributed Computing*, 28(3):189–200, 2015.
7. Florent Becker, Martín Matamala, Nicolas Nisse, Ivan Rapaport, Karol Suchan, and Ioan Todinca. Adding a referee to an interconnection network: What can(not) be computed in one round. In *Proceedings of the 25th IEEE International Parallel and Distributed Processing Symposium*, IPDPS 2011, pages 508–514, 2011.
  8. Florent Becker, Pedro Montealegre, Ivan Rapaport, and Ioan Todinca. The simultaneous number-in-hand communication model for networks: Private coins, public coins and determinism. In *Proceedings of the 21st International Colloquium, on Structural Information and Communication Complexity*, SIROCCO 2014, pages 83–95, 2014.
  9. Etienne Birmelé. Tree-width and circumference of graphs. *Journal of Graph Theory*, 43(1):24–25, 2003.
  10. John A. Bondy and Miklós Simonovits. Cycles of even length in graphs. *Journal of Combinatorial Theory, Series B*, 16(2):97–105, 1974.
  11. Andreas Brandstädt, Jeremy P Spinrad, et al. *Graph classes: a survey*, volume 3. Siam, 1999.
  12. Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. Algebraic methods in the congested clique. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing, PODC 2015*, pages 143–152, 2015.
  13. N. Chandrasekharan and S. Sitharama Iyengar. NC algorithms for recognizing chordal graphs and  $k$  trees. *IEEE Trans. Computers*, 37(10):1178–1183, 1988.
  14. Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *ACM Symposium on Principles of Distributed Computing, PODC 2014*, pages 367–376, 2014.
  15. Paul Erdős. Extremal problems in graph theory. In *Theory of graphs and its applications*. Proc. Sympos. Smolenice, 1964.
  16. Sudipto Guha, Andrew McGregor, and David Tench. Vertex and hyperedge connectivity in dynamic graph streams. In *Proceedings of the 34th ACM Symposium on Principles of Database Systems*, pages 241–247. ACM, 2015.
  17. James W. Hegeman, Sriram V. Pemmaraju, and Vivek Sardeshmukh. Near-constant-time distributed algorithms on a congested clique. In *Proceedings of the 28th International Symposium on Distributed Computing, DISC 2014*, pages 514–530, 2014.
  18. Stephan Holzer and Nathan Pinsker. Approximation of Distances and Shortest Paths in the Broadcast Congest Clique. In *19th International Conference on Principles of Distributed Systems (OPODIS 2015)*, volume 46 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 1–16, 2016.
  19. Hossein Jowhari, Mert Saglam, and Gábor Tardos. Tight bounds for lp samplers, finding duplicates in streams, and related problems. In *Proceedings of the 30th ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, PODS 2011*, pages 49–58, 2011.
  20. Tomasz Jurdzinski and Krzysztof Nowicki. MST in  $O(1)$  rounds of the congested clique. *Preprint arXiv:1707.08484*, 2017.
  21. Jarkko Kari, Martin Matamala, Ivan Rapaport, and Ville Salo. Solving the induced subgraph problem in the randomized multiparty simultaneous messages model. In *Proceedings of the 21st International Colloquium, on Structural Information and Communication Complexity*, SIROCCO 2015, pages 370–384, 2015.

22. Felix Lazebnik, Vasily A. Ustimenko, and Andrew J. Woldar. A new series of dense graphs of high girth. *Bulletin of the American mathematical society*, 32(1):73–79, 1995.
23. Nathan Linial. Locality in distributed graph algorithms. *SIAM Journal on Computing*, 21(1):193–201, 1992.
24. Pedro Montealegre and Ioan Todinca. Brief announcement: deterministic graph connectivity in the broadcast congested clique. In *Proceedings of the 35th ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC 2016)*, 2016.
25. Moni Naor and Larry Stockmeyer. What can be computed locally? *SIAM Journal on Computing*, 24(6):1259–1277, 1995.