Beyond Distributed Subgraph Detection: Induced Subgraphs, Multicolored Problems and Graph Parameters

Amir Nikabadi
ENS de Lyon, France

Janne H. Korhonen
IST Austria, Klosterneuburg, Austria

Abstract

Subgraph detection has recently been one of the most studied problems in the CONGEST model of distributed computing. In this work, we study the distributed complexity of problems closely related to subgraph detection, mainly focusing on *induced subgraph detection*. The main line of this work presents lower bounds and parameterized algorithms w.r.t *structural parameters* of the input graph:

- On general graphs, we give unconditional lower bounds for induced detection of cycles and patterns of treewidth 2 in CONGEST. Moreover, by adapting reductions from centralized parameterized complexity, we prove lower bounds in CONGEST for detecting patterns with a 4-clique, and for induced path detection conditional on the hardness of triangle detection in the congested clique.
- On graphs of bounded degeneracy, we show that induced paths can be detected fast in CONGEST using techniques from parameterized algorithms, while detecting cycles and patterns of treewidth 2 is hard.
- On graphs of bounded vertex cover number, we show that induced subgraph detection is easy in CONGEST for any pattern graph. More specifically, we adapt a centralized parameterized algorithm for a more general *maximum common induced subgraph detection* problem to the distributed setting.

In addition to these induced subgraph detection results, we study various related problems in the CONGEST and congested clique models, including for *multicolored* versions of subgraph-detection-like problems.

2012 ACM Subject Classification Theory of computation → Distributed algorithms; Theory of computation → Computational complexity and cryptography

Keywords and phrases distributed algorithms, parameterized distributed complexity, CONGEST model, induced subgraph detection, graph parameters, lower bounds

Digital Object Identifier 10.4230/LIPIcs.OPODIS.2021.15

Related Version *Full Version*: https://arxiv.org/abs/2109.06561 [32]

Funding *Amir Nikabadi*: Supported by the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).
Janne H. Korhonen: Supported by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 805223 ScaleML).

Acknowledgements We thank François Le Gall and Masayuki Miyamoto for sharing their work on lower bounds for induced subgraph detection [36].

25th International Conference on Principles of Distributed Systems (OPODIS 2021).
Editors: Quentin Bramas, Vincent Gramoli, and Alessia Milani; Article No. 15; pp. 15:1–15:18
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Subgraph detection is one of the most studied problems in the CONGEST and congested clique models of distributed computing [23, 16, 33, 25, 26, 29, 28, 15, 14]. The complexity of distributed subgraph detection is understood for many pattern graphs – for example, in the CONGEST model, tight bounds are known for *path* [33, 26] and *odd cycle* detection [33, 25], and it is known that pattern graphs requiring almost quadratic time exist [28]. However, unresolved questions remain about the exact complexity of, e.g., *triangle* detection in either CONGEST or congested clique, and *even cycle* detection in CONGEST.

In this work, we look at the closely related *induced subgraph detection* problem, which has so far not received any attention in the distributed setting. In particular, we aim to understand the complexity of induced subgraph detection for common pattern graphs, such as paths and cycles, as well as how the situation contrasts with the non-induced case. It is well known that in the centralized setting, induced subgraph detection is generally more difficult than non-induced subgraph detection, so one would expect that situation is the same also in the distributed setting.

1.1 Background and setting

Before presenting our results, we start by discussing the wider context of distributed subgraph detection problems. As mentioned above, we work in the CONGEST and congested clique models of distributed computing, and use G and n to denote the input graph and the number number of nodes in the input graph, respectively.

In the paper, we mostly consider subgraph detection and induced subgraph detection problems; we are given a pattern graph H with k nodes, known to all nodes in G, and the task is to decide if the input graph G contains H as a subgraph or an induced subgraph; more precisely, any node v that is part of an admissible copy of H should report that the input is a yes-instance.

Fixed-parameter tractability. Subgraph and induced subgraph detection problems can be viewed as *parameterized problems*; such problems are studied in centralized setting under the field of *parameterized complexity* [20]. A parameterized problem is defined by the input and a problem parameter k – formally, a *(complexity) parameter* k is a mapping from the input instance to natural numbers. The basic question of centralized parameterized complexity is to understand which problems are *fixed-parameter tractable*, i.e. have algorithms with running time $f(k)|x|^{O(1)}$, where f is an arbitrary function and x is the binary encoding of the input instance. For example, k-cycle detection can be viewed as a parameterized problem.

Similarly, one can consider fixed-parameter tractability in the distributed setting. The strictest definition is to ask which problems have distributed algorithm where the running time depends only on the parameter k [43, 10]. However, this arguably does not capture all fixed-parameter tractability phenomena in distributed models – e.g. k-cycle detection cannot be solved in $f(k)$ rounds for any function f in the CONGEST model.

A more general perspective is to ask what is the smallest function T such that a parameterized problem can be solved in $f(k) \cdot T(n)$ rounds, for some function $f \colon \mathbb{N} \to \mathbb{N}$. Several results of this type are known for subgraph detection problems; for example, k-cycle detection can be solved in $O(k2^k n)$ rounds in the CONGEST model [33, 26], and in $2^{O(k)}n^{0.158}$ rounds in the congested clique model [16], though these bounds are not tight for *even-length* cycles [15, 28].

Parameters and graph structure. For subgraph and induced subgraph detection problems, the natural complexity parameter is the number of nodes k in the pattern graph. However, parameterized complexity frequently studies other complexity parameters – for our purposes, the most relevant are structural graph parameters, in particular degeneracy $d(G)$, treewidth $\mathrm{tw}(G)$, and vertex cover number $\tau(G)$ (see Section 2 for the precise definitions). While bounded degeneracy (equivalently, bounded arboricity) [7, 33, 8] has been studied in the distributed setting, bounded treewidth and bounded vertex cover number less so.

Given a structural parameter p, we can consider the complexity of subgraph or induced subgraph detection parameterized either by the structural parameter $p(G)$ of the input graph, or by the structural parameter $p(H)$ of the pattern graph. Note that we have

$$d(G) \leq \mathrm{tw}(G) \leq \tau(G)\,.$$

For parameters p_1 and p_2 with $p_1(G) \leq p_2(G)$, upper bounds w.r.t. parameter p_2 imply upper bounds w.r.t. parameter p_1, and lower bounds w.r.t. parameter p_1 imply lower bounds w.r.t. parameter p_2.

Lower bounds and reductions. The standard technique for proving unconditional CONGEST lower bounds is by reduction from communication complexity problems, most often using *families of lower bound graphs* [42, 25, 17, 1, 30, 21] (see Section 2). By contrast, reductions between problems are less useful in the CONGEST model, as the model can implement only very limited reductions efficiently.

However, there are still uses for reductions in distributed complexity theory, which we will apply in this work. First, in the congested clique, sub-polynomial round reductions can be used to establish relative complexities of problems [34]. Second, as noted by Bacrach et al. [6], centralized reductions can be used to transform families of lower bound graphs for one problem into families of lower bound graphs for a second problem.

1.2 Results: induced subgraph detection on general graphs

First, we consider the hardness of induced subgraph detection on general graphs. We show that for common pattern graphs, the induced version of the problem is at least as hard as the non-induced version, and in many cases harder.

Unconditional lower bounds. We start with unconditional lower bounds for induced subgraph detection in CONGEST; see Table 1 for a summary of these results.

For cycles of length at least 6, we show that the induced cycle detection problem requires at least $O(n/\log n)$ rounds in the CONGEST model. The result follows from a combination of the existing lower bound construction for odd-length cycles, and a new construction for induced even cycles. By comparison, the existing lower bounds for non-induced subgraph detection in CONGEST are $\Omega(n^{1/2}/\log n)$ for even cycle detection [33], and $\Omega(n/\log n)$ for odd cycle detection excluding triangles [25]; it is also known that even cycles can be detected in $O(n^{\delta})$ time, for $\delta < 1$ that depends on the length of the cycle [28].

We also prove that there are pattern graphs for which induced subgraph detection (and also non-induced detection) requires near-quadratic time in CONGEST, in similar spirit at the hard pattern graphs for non-induced subgraph detection presented by Fischer et al. [28]. Moreover, we show that these pattern graphs can be constructed to have treewidth 2; contrast this with the centralized setting, where low-treewidth patterns are easy to detect [5].

Table 1 Lower bounds on general graphs. Improved lower bounds of Le Gall and Miyamoto [36] are independent and concurrent work (see main text.)

Problem	Bound	
Induced $2k$-cycle ($k \geq 3$)	$\Omega(n/\log n)$	Section 3.3
Induced H-detection		
· any H with 4-clique	$\Omega(n^{1/2}/\log n)$	Section 3.2
· some H with $\mathrm{tw}(H) = 2^{\dagger}$	$\Omega(n^{2-\varepsilon})$	Section 3.4
Multicolored k-cycle ($k \geq 4$)	$\Omega(n/\log n)$	Section 4.2
Multicolored induced path of length k ($k \geq 6$)	$\Omega(n/\log n)$	Section 4.2
Induced k-cycle ($k \geq 4$)	$\tilde{\Omega}(n)$	[36]
Induced k-cycle ($k \geq 8$)	$\tilde{\Omega}(n^{2-\Theta(1/k)})$	[36]

†holds for any $\varepsilon > 0$, for some H that is chosen depending ε

Table 2 Bounds w.r.t. structural graph parameters. Results attributed to [33] follow directly from the proofs in that work, but are not stated in that work for induced subgraphs.

Problem	Bound		
Induced k-tree†	$2^{O(kd(G))}k^k + O(\log n)$		Section 5
(Induced) H-detection, some H with $\mathrm{tw}(H) = 2^{\ddagger}$	$\Omega(n^{1-\varepsilon})$	holds for $d(G) = 2$	Section 5.2
(Induced) k-cycle ($k \geq 6$)	$\Omega(n^{1/2}/\log n)$	holds for $d(G) = 2$	[33]
Induced 4-cycle	$O(d(G) + \log n)$		[33]
Induced 5-cycle	$O(d(G)^2 + \log n)$		[33]
MCIS	$2^{O(\tau^2)}$	$\tau = \tau(G) + \tau(H)$	Section 6
Induced subgraph	$2^{O((\tau(G)+k)^2)}$		Section 6

†randomized algorithm, can be derandomized with extra assumptions and worse running time

‡holds for any $\varepsilon > 0$, for some H that is chosen depending ε

Unconditional lower bounds: recent independent work. After submitting this paper, we learned about the independent and concurrent work of Le Gall and Miyamoto [36], which gives lower bounds for induced cycle detection and diamond listing. In particular, they show that detecting induced k-cycles requires $\tilde{\Omega}(n)$ rounds for any $k \geq 4$, and $\tilde{\Omega}(n^{2-\Theta(1/k)})$ rounds for any $k \geq 8$. These results subsume our lower bounds for induced cycle and treewidth-2 subgraph detection.

Reductions. Next, we turn our attention to conditional lower bounds for problems where standard CONGEST lower bound techniques do not immediately yield unconditional lower bounds. See Figure 1 for a summary of these results.

We adapt a centralized reduction of Dalirrooyfard et al. [22] between clique and independent set detection and induced subgraph detection. Specifically, they show that detecting an induced subgraph H that contains a k-clique (k-independent set) is as hard detecting k-clique (k-independent set, resp.). We show that this reduction can also be implemented in the congested clique model.

It follows that detecting induced paths of length at least 5 in either the CONGEST or congested clique model is at least as hard *triangle detection* in the congested clique model, and more generally, detecting paths of length at least $2k-1$ in CONGEST or congested clique is as hard as detecting k-cliques in the congested clique. By comparison, the best known upper bounds in the congested clique are $O(n^{0.158})$ for triangle detection [16], and $O(n^{1-1/k})$ for k-clique detection [23]; while no lower bounds for the congested clique model are known, improving over the $O(n^{0.158})$-round matrix multiplication based triangle detection would have major implications for distributed algorithms. However, it is worth noting that induced paths of length 2 can be detected in $O(1)$ rounds in CONGEST, in contrast to triangles (see Appendix A).

Moreover, the reduction allows us to lift the $\Omega(n^{1/2}/\log n)$ CONGEST lower bound of Czumaj and Konrad [21] for 4-clique detection to induced and non-induced detection of any pattern graph H that contains a 4-clique.

Multicolored problems. Finally, we consider *multicolored* versions of subgraph detection tasks. In multicolored (induced) H-detection, we are given a labelling of the input graph G with k colors, and the task is to find a (induced) copy of H that contains exactly one node of each color. Multicolored versions of problems have proven to be useful starting points for reductions in fixed-parameter complexity, and algorithms for a multicolored version of a problem can often be turned into an algorithm for the standard version via color-coding [5].

We observe that multicolored versions of k-clique and k-independent set are closely related to their standard versions in the distributed setting, by adapting the simple centralized reductions to distributed setting (see Figure 1). We then prove unconditional lower bounds of $\Omega(n/\log n)$ in CONGEST for multicolored versions of k-cycle detection, for $k \geq 4$, and for detection of induced paths of length k, for $k \geq 6$. These results imply that color-coding algorithms cannot be used directly to improve the state of the art for these problems – for comparison, note that k-cycle detection can be solved in CONGEST in $o(n/\log n)$ rounds for even k, non-induced multicolored paths can be detected in $O(1)$ round in CONGEST, and we have no unconditional lower bounds for induced path detection.

1.3 Results: induced subgraph detection with structural parameters

Next, we consider subgraph and induced subgraph detection tasks w.r.t. structural graph parameters. We focus on the degeneracy $d(G)$ and the vertex cover number $\tau(G)$ of the input graph as the parameters in this section. See Table 2 for a summary of the results.

Bounded degeneracy. We show that induced subgraph detection for any *tree* on k nodes can be solved in time $2^{O(kd(G))}k^k + O(\log n)$ rounds in CONGEST. As with the prior results on non-induced path, tree and cycle detection algorithms in CONGEST, this upper bound is based on centralized fixed-parameter algorithms, in this case using color-coding and random separation techniques [4, 13].

On the lower bounds side, we show that there are treewidth 2 pattern graphs that require near-linear time to detect as induced and non-induced subgraphs in CONGEST on input graphs of degeneracy $d(G) = 2$, via a slight modification of the proof for the general case discussed above. Note that any fixed pattern graph can be detected in $O(n)$ rounds when degeneracy is bounded, by having all nodes gather their distance-k neighborhood.

For cycles, we note that results of Korhonen and Rybicki [33] can be easily seen to imply that detecting induced k-cycles for $k \geq 6$ requires at least $\Omega(n^{1/2}/\log n)$ rounds to detect in CONGEST on graphs of degeneracy $d(G) = 2$, as well as that induced 4-cycles can be detected in $O(d(G) + \log n)$ rounds, and induced 5-cycles in $O(d(G)^2 + \log n)$ rounds.

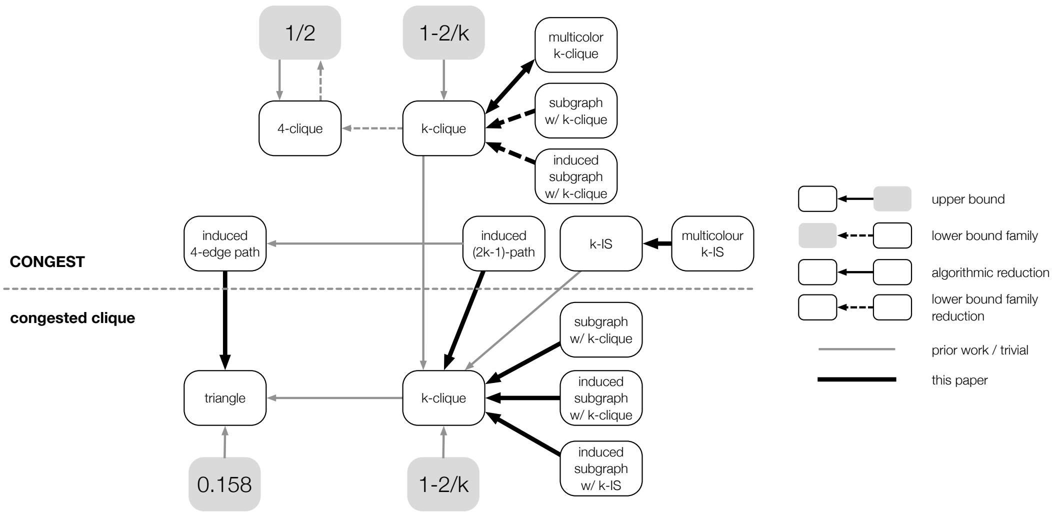


Figure 1 Relationships between problems in CONGEST and congested clique. Results hold for any sufficiently large constant k. *Upper bound* indicates an $\tilde{O}(n^\delta)$ round algorithm for the problem for specified δ; *lower bound family* indicates that there is a lower bound family giving $\tilde{\Omega}(n^\delta)$ lower bound for the problem for specified δ; *algorithmic reduction* from P_1 to P_2 indicates that an algorithm solving P_2 in $O(n^\delta)$ rounds implies the existence of an algorithm solving P_1 in $\tilde{O}(n^\delta)$ rounds, for any $\delta > 0$, and *lower bound reduction* from P_1 to P_2 indicates that a lower bound family giving $\Omega(n^\delta)$ lower bound for P_1 implies the existence of a lower bound family giving $\tilde{\Omega}(n^\delta)$ lower bound for P_2 for any $\delta > 0$. Notation $\tilde{O}$ and $\tilde{\Omega}$ hides polylogarithmic factors in n, as well as factors only depending on k, as we assume k to be constant.

Bounded vertex cover number. For a more restrictive parameter than degeneracy, we consider induced subgraph detection parameterized by the vertex cover number $\tau(G)$ of the input graph. More precisely, we show a more general problem of *maximum common induced subgraph* (*MCIS*) can be solved fast; in this problem, we are given two graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ as input, and the task is to find the maximum-size graph G^* such that G^* appears as induced subgraph of both G and H. In the distributed setting, we assume that G is the input graph, and the second graph H is known to every node.

In more detail, we show that a centralized branching algorithm from MCIS of Abu-Khzam et al. [3] can be implemented in $2^{O((\tau(G)+\tau H)^2)}$ rounds, i.e. without dependence on n, in the CONGEST model. This immediately implies that induced subgraph detection for any pattern graph H on k nodes can also be solved in $2^{O((\tau(G)+k)^2)}$ rounds.

1.4 Additional related work

Centralized subgraph and induced subgraph detection. Subgraph detection has been widely studied in the centralized parameterized setting. Fixed-parameter algorithms, parameterized by the number of nodes k of the pattern graph, are known for example for paths [38, 5, 45, 11], trees [5], even cycles [46], odd cycles [5], and patterns of constant treewidth [5]. By contrast, k-clique detection is known to be W[1]-hard, suggesting that it does not have a fixed-parameter algorithm [24].

Induced subgraph detection, on the other hand, is W[1]-hard even for paths of length k [19]. Any induced or non-induced subgraph on k nodes can be detected in $n^{\omega k/3+O(1)}$ time, where $\omega < 2.3729$ is the matrix multiplication exponent, due to a classical result of Nešetřil and Poljak [39].

Distributed subgraph detection. As mentioned above, distributed subgraph detection has also received attention in the distributed setting recently. In CONGEST, non-trivial upper bounds are known e.g. for path and tree detection [33, 26], cycle detection [29, 33, 28] and clique detection [14]. Likewise, lower bounds have been studied for cycle detection [25, 33] and cliques [21], and pattern graphs requiring near-quadratic time are known to exist [28]. Triangle detection remains a particularly interesting open question—the best known upper bound is $n^{1/3+o(1)}$ rounds [18], but no lower bounds are known.

In the congested clique, triangles can be detected in $O(n^{0.158})$ rounds and odd k-cycles in $2^{O(k)}n^{0.158}$ rounds using fast matrix multiplication [16]. Even cycles can be detected even faster, in $O(k^2)$ rounds for $k = O(\log n)$ [15]. Moreover, any induced or non-induced subgraph detection for k-node patterns can be solved in $O(n^{1-2/k})$ rounds in congested clique [23].

Distributed parameterized complexity. Parameterized distributed algorithms have appeared implicitly in many of the above-mentioned subgraph detection works, and recently Ben-Basat et al. [10] and Siebertz and Vigny [43] have explicitly studied aspects of distributed parameterized complexity. In terms of structural parameters, *maximum degree* is a standard parameter in distributed setting, and algorithms parameterized degeneracy has been studied for various problems and models [9, 7, 31]. Recently, Li [37] has show that the treewidth of the input graph can be approximated in $\tilde{O}(D)$ rounds in CONGEST, and many classical optimization problems that are fixed-parameter tractable w.r.t. treewidth can be solved in $\tilde{O}\big(\mathrm{tw}(G)^{O(\mathrm{tw}(G))}D\big)$ rounds in CONGEST, where $\tilde{O}$ hides polylogarithmic factors in n.

2 Preliminaries

Degeneracy. A graph G is called *d-degenerate* if every induced subgraph of G has a vertex of degree at most d. The minimum number d for which G is *d-generate* is called *degeneracy* of G, denoted by $d(G)$. It is easy to see that every d-degenerate graph admits an acyclic orientation such that the out-degree of each vertex is at most d.

Vertex cover number. A *vertex cover* of G is a subset of vertices $S \subseteq V(G)$ such that every edge in $E(G)$ is incident with at least one vertex in S. The *vertex cover number* $\tau(G)$ of G is the minimum size of a vertex cover of G.

Treewidth. A *tree decomposition* of a graph $G = (V, E)$ is a pair $(\mathcal{X}, T)$, where $\mathcal{X} = \{X_1, X_2, \dots, X_m\}$ is a collection of subsets of V and T is a tree on $\{1, 2, \dots, m\}$, such that

1. $\bigcup_{i=1}^{m} X_i = V$,
2. for all edges $e \in E$ there exist i with $e \subseteq X_i$
3. for all i, j and k, if j is on the (unique) path from i to k in T, then $X_i \cap X_k \subseteq X_j$.

The *width* of a tree-decomposition $(\mathcal{X}, T)$ is defined as $\max_i |X_i| - 1$. The *treewidth* of a graph G is the minimum width over all possible tree decompositions of G. Connected graphs of treewidth 1 are trees, and connected graphs of treewidth 2 are *series-parallel graphs* (see e.g. [12].)

Lower bound families. For unconditional lower bounds in the CONGEST model, we use the standard framework of reducing from two-party communication complexity. Let $f\colon \{0,1\}^{2k} \to \{0,1\}$ be a Boolean function. In the two-party communication game on f, there are two players who receive a private k-bit string x_0 and x_1 as input, and the task is to have at least one of the players compute $f(x) = f(x_0, x_1)$.

The template for these reductions is captured by *families of lower bound graphs*:

▶ **Definition 1** (e.g. [25, 30, 1]). *Let $f_n\colon \{0,1\}^{2k(n)} \to \{0,1\}$ and $C\colon \mathbb{N} \to \mathbb{N}$ be functions and Π a graph predicate. Suppose there is n_0 such that for any $n \geq n_0$ and all $x_0, x_1 \in \{0,1\}^{k(n)}$ there exists a (weighted) graph $G(n, x_0, x_1)$ satisfying the following properties:*

1. *$G(n, x_0, x_1)$ satisfies Π if and only if $f_n(x_0, x_1) = 1$,*
2. *$G(n, x_0, x_1) = (V_0 \cup V_1, E_0 \cup E_1 \cup S)$, where*
   a. *V_0 and V_1 are disjoint and $|V_0 \cup V_1| = n$,*
   b. *$E_i \subseteq V_i \times V_i$ for $i \in \{0,1\}$,*
   c. *$S \subseteq V_0 \times V_1$ is a cut and has size at least $C(n)$, and*
   d. *subgraph $G_i = (V_i, E_i)$ only depends on i, n and x_i, i.e., $G_i = G_i(n, x_i)$.*

We then say that $\mathcal{F} = (\mathcal{G}(n))_{n \in I}$ is a family of lower bound graphs, *where*

$$\mathcal{G}(n) = \{G(n, x_0, x_1)\colon x_0, x_1 \in \{0,1\}^{k(n)}\}\,.$$

Deterministic communication complexity $\mathsf{CC}(f)$ of a function f is the maximum number of bits the two players need to exchange in the worst case, over all deterministic protocols and input strings, in order to compute $f(x_0, x_1)$. *Randomized communication complexity* $\mathsf{RCC}(f)$ is the worst-case complexity of protocols which compute f with probability at least $2/3$ on all inputs.

▶ **Theorem 2** (e.g. [25, 30, 1]). *Let $\mathcal{F}$ be a family of lower bound graphs. Any algorithm deciding Π on a graph family $\mathcal{H}$ containing $\bigcup \mathcal{G}(n)$ for all $n \geq n_0$ in the* CONGEST *model with bandwidth $b(n)$ needs $\Omega\left(\mathsf{CC}(f_n)/C(n)b(n)\right)$ and $\Omega\left(\mathsf{RCC}(f_n)/C(n)b(n)\right)$ deterministic and randomized rounds, respectively.*

We reduce from the two-player *set disjointness* function $\mathsf{DISJ}_n\colon \{0,1\}^{2n} \to \{0,1\}$, defined as $\mathsf{DISJ}_n(x_0, x_1) = 0$ if and only there is $i \in [n]$ such that $x_0(i) = x_1(i) = 1$. The communication complexity of set disjointness is $\mathsf{CC}(\mathsf{DISJ}_n) = \Omega(n)$ and $\mathsf{RCC}(\mathsf{DISJ}_n) = \Omega(n)$ [35, 41].

3 Induced subgraph detection on general graphs

3.1 Patterns with cliques and independent sets: framework

For the complexity results on detecting pattern graphs that contain large independent sets or clique, we borrow the centralized reduction of of Dalirrooyfard et al. [22]. We present the reduction here in full, as we will need to analyze its implementation in distributed setting.

We will start from instance G of s-clique detection. The reduction will transform G into an instance of (induced) H-detection, where the pattern graph H contains a clique of size s, while increasing the number of nodes by a small factor. We first need the following definition:

▶ **Definition 3** ([22]). *Let $G = (V, E)$ be a graph. A family $\mathcal{C} \subseteq 2^V$ is an* s-clique cover *if*

1. *for each s-clique K in G, there is a $C \in \mathcal{C}$ that contains the nodes of K, and*
2. *the induced subgraph $G[C]$ is s-colorable for each $C \in \mathcal{C}$.*

We say that $\mathcal{C}$ is a minimum s-clique cover *if all s-clique covers of G have at least $|\mathcal{C}|$ sets.*

Note that if $\mathcal{C}$ is a minimum s-clique cover, all induced subgraphs $G[C]$ for $C \in \mathcal{C}$ contain an s-clique, and thus require exactly s colors to color.

Reduction overview. Let $G = (V_G, E_G)$ be the original graph and let $H = (V_H, E_H)$ be the pattern graph. Let $\mathcal{C} = \{C_1, C_2, \dots, C_t\}$ be a minimum s-clique cover of H. We construct a graph G^* as from the input graph G follows:

1. The node set V_{G^*} of G^* consists of the following nodes:
   a. For each $i \in C_1$, there is a copy $V_{G,i} = V_G \times \{i\}$ of the node set of G.
   b. For each $j \in V_H \setminus C_1$, there is a copy j^* of the node j in G^*.
2. The edge set of G^* is defined by the following rules:
   a. Each $V_{G,i}$ is an independent set.
   b. For each $i, j \in C_1$ and $v, u \in V_G$, we add edge between (v, i) and (u, j) if both $\{i, j\} \in E_H$ and $\{v, u\} \in E_G$.
   c. For each $i \in C_1$ and $j \in V_H \setminus C_1$ with $\{i, j\} \in E_H$, we add edges between j^* and all nodes (v, i) for $v \in V_G$.
   d. For each $i, j \in V_H \setminus C_1$ with $\{i, j\} \in E_H$, we add edge between i^* and j^*.

Note that the graph G^* has $sn + |V_H|$ nodes.

▶ **Lemma 4** ([22]). *If G has an s-clique, then G^* has H as an induced subgraph, and if G^* has H as a subgraph, then G has an s-clique.* (▷ See full version.)

3.2 Patterns with cliques and independent sets: implications

Implementing the reduction in the congested clique. Let H be a pattern graph on k nodes containing an s-clique. We now show that the reduction we gave above can be implemented efficiently in the congested clique model.

Assume we have algorithm $\mathcal{A}$ for (induced) H-detection running in $O(n^\delta)$ rounds in the congested clique. We now show that we can implement the above reduction in the congested clique to obtain an algorithm for detecting an s-clique, as follows:

1. Each node $v \in V_G$ simulates nodes (v, i) for $i \in C_1$, as well as one node from V_H.
2. Since the incident edges of (v, i) for $i \in C_1$ and nodes in $V_H \setminus C_1$ in G^* only depend on the pattern graph H and on the edges incident to v in G, node v can construct the inputs of its simulated nodes locally.
3. Nodes then simulate the execution of $\mathcal{A}$ on a congested clique with $O(sn + k) = O(kn)$ nodes. The running time of $\mathcal{A}$ on the simulated instance is $O\big((kn)^\delta\big)$, and the simulation incurs additional overhead of $O(k^2)$, for a total running time of $O(k^{2\delta} n^\delta)$.

Thus, we obtain the following:

▶ **Theorem 5.** *Let H be a pattern graph with k nodes that has a clique of size s. Then if we can solve H-detection or induced H-detection in the congested clique model in $O(n^\delta)$ rounds, we can find an s-clique in the congested clique in $O(k^{2\delta} n^\delta)$ rounds.*

As an immediate corollary, we obtain a similar hardness result for induced subgraph detection for pattern graphs with large independent set, by observing that we can simply complement the pattern and input graphs. Note that this version only applies for *induced* subgraph detection.

▶ **Corollary 6.** *Let H be a pattern graph with k nodes that has an independent set of size s. Then if we can solve induced H-detection in the congested clique model in $O(n^\delta)$ rounds, we can find an s-clique in the congested clique in $O(k^{2\delta} n^\delta)$ rounds.*

Induced path detection. Corollary 6 immediately implies a conditional lower bound for induced path detection in the CONGEST model, as paths contain large independent sets:

▶ **Corollary 7.** *Let k be fixed. If an induced $2k$-edge path or an induced $(2k+1)$-edge path can be detected in $O(n^\delta)$ rounds in the* CONGEST *model, then a k-clique can be detected in $O(n^\delta)$ rounds in the congested clique model. In particular, if an induced 4-edge path can be detected in $O(n^\delta)$ rounds in the* CONGEST *model, then triangles can be detected $O(n^\delta)$ rounds in the congested clique model.*

Patterns with cliques in CONGEST. As a further application of the reduction of Dalirrooyfard et al. [22], we can transform the unconditional lower bound of Czumaj and Konrad [21] for 4-clique detection in CONGEST into a lower bound for induced subgraph detection for any pattern containing a 4-clique.

▶ **Lemma 8** ([21])**.** *Let Π the graph predicate for existence of a 4-clique. There exists a family of lower bound graphs for Π with $f_n = \mathsf{DISJ}_{\Theta(n^2)}$ and $C(n) = \Theta(n^{3/2})$.*

▶ **Lemma 9.** *Let H be a pattern graph on k nodes that contains a 4-clique, and let Π the graph predicate for existence of either induced or non-induced copy of H. Then there exists a family of lower bound graphs for Π with $f_n = \mathsf{DISJ}_{\Theta(n^2)}$ and $C(n) = \Theta(n^{3/2})$.*

(▷ See full version.)

Theorem 2 and Lemma 9 now immediately imply the following:

▶ **Theorem 10.** *Let H be a pattern graph that contains a 4-clique. Any* CONGEST *algorithm solving either H-detection or induced H-detection needs at least $\Omega(n^{1/2}/\log n)$ rounds.*

3.3 Induced even cycle detection

We next prove an unconditional lower bound for induced even cycle detection in CONGEST. Note that for induced odd cycles, one can easily verify that the construction of Drucker et al. [25] immediately implies a $\Omega(n/\log n)$ lower bound.

▶ **Lemma 11.** *Let $k \geq 3$ be fixed, and let Π the graph predicate for existence of an induced $2k$-cycle. There exists a family of lower bound graphs for Π with $f_n = \mathsf{DISJ}_{\Theta(n^2)}$ and $C(n) = n$.*

(▷ See full version.)

Theorem 2 and Lemma 11 immediately imply the following:

▶ **Theorem 12.** *Any* CONGEST *algorithm solving induced $2k$-cycle detection for $k \geq 3$ needs at least $\Omega(n/\log n)$ rounds.*

3.4 Induced subgraph detection for bounded treewidth patterns

Finally, we consider subgraph and induced subgraph detection for pattern graphs of low treewidth. Recall that in centralized setting, a subgraph H with treewidth t can be detected in time $2^{O(k)}n^{t+1}\log n$ [5], implying that detecting constant-treewidth subgraphs is fixed-parameter tractable. However, in CONGEST model, turns out that pattern of treewidth 2 are already maximally hard.

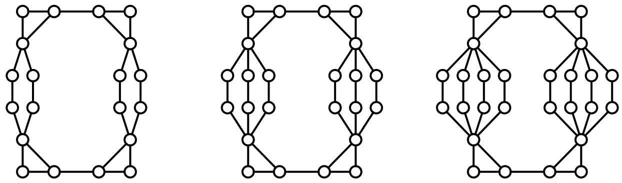
Our construction for the hard pattern graph uses similar ideas as the hard non-induced subgraph detection instances presented by Fischer et al. [28]. However, the pattern graphs they use a fairly dense and have treewidth higher than 2.

▶ **Theorem 13.** *For any $k \geq 2$, there exists a pattern graph H_k of treewidth 2 such that* CONGEST *algorithm solving either H_k-detection or induced H_k-detection needs at least $\Omega(n^{2-1/k})$ rounds.*

Let $k \geq 2$ be fixed. We construct the graph H_k as follows:

1. We start with four triangles A_1, A_2, B_1 and B_2 with nodes labelled by 1, 2 and 3.
2. Nodes 1 of A_1 and A_2 are connected by an edge, and nodes 1 of B_1 and B_2 are connected by an edge.
3. Nodes 2 of A_1 and B_1 are connected with k disjoint paths of length 3. Likewise, Nodes 2 of A_2 and B_2 are connected with k disjoint paths of length 3.

The graph H_k is a *series-parallel* graph, and thus has treewidth 2 [12].



▶ **Lemma 14.** *Let $k \geq 2$ be fixed. There exists a family of lower bound graphs for H_k-detection and induced H_k-detection with $f_n = \mathsf{DISJ}_{\Theta(n^2)}$ and $C(n) = \Theta(n^{1/k})$.* (▷ See full version.)

Theorem 13 now follows immediately by Theorem 2.

4 Multicolored problems

In the *multicolored (induced) subgraph detection*, we are given a pattern graph H on k nodes and an input graph G with a (not necessarily proper) k-coloring, and the task is to find a (induced) copy of H that is multicolored, i.e. a copy where all nodes have different colors.

4.1 Reductions

We first prove that the complexities of multicolored k-clique and k-independent set are closely related to their standard versions also in the distributed setting. These results follow from standard fixed-parameter reductions [40, 27].

▶ **Theorem 15.** *If multicolored k-clique can be solved in $T(n)$ rounds in* CONGEST*, then k-clique can be solved $O(k^2 T(kn))$ rounds in* CONGEST*. If k-clique can be solved in $T(n)$ rounds in* CONGEST*, then multicolored k-clique can be solved $T(n)$ rounds in* CONGEST*.* (▷ See full version.)

In the centralized setting, clique and independent set are equivalent, so the above reductions work also for independent set. However, in the distributed setting, only one direction works immediately, by essentially the same proof.

▶ **Theorem 16.** *If multicolored k-independent set can be solved in $T(n)$ rounds in* CONGEST*, then k-independent set can be solved $O(k^2 T(kn))$ rounds in* CONGEST*.*

4.2 Lower bounds

Next, we prove some simple unconditional lower bounds for multicolored (induced) cycle detection and multicolored induced path detection.

▶ **Theorem 17.** *For any $k \geq 4$, any* CONGEST *algorithm solving multicolored (induced) k-cycle detection needs at least $\Omega(n/\log n)$ rounds.* (▷ See full version.)

▶ **Theorem 18.** *For any $k \geq 6$, any* CONGEST *algorithm solving multicolored induced k-edge path detection needs at least $\Omega(n/\log n)$ rounds.* (▷ See full version.)

5 Induced subgraph detection on bounded degeneracy graphs

5.1 Induced tree detection

We start by giving a parameterized distributed algorithm for detecting induced trees, parameterized by the degeneracy $d = d(G)$ of the input graph. This result is based on the *random separation* algorithm of Cai et al. [13], adapted to distributed setting. For this result, we assume for convenience that all nodes are given the parameter d as input; we discuss at the end how to remove this dependence for the randomized version of the algorithm.

Preliminaries. Let $G = (V, E)$ be a graph. We say that an orientation σ of the edges of G is an *α-bounded orientation*, or simply *α-orientation*, if every node $v \in V$ has out-degree at most α in σ, and σ is acyclic. A graph G is d-degenerate if and only if has an d-orientation; moreover, an $O(d)$-orientation can be computed fast in the CONGEST model:

▶ **Lemma 19** ([7])**.** *Let G be a d-degenerate graph, and let $\varepsilon > 0$. We can compute a $(2+\varepsilon)d$-orientation of G in $O(\log n)$ rounds in the* CONGEST *model, assuming d is known to all nodes. If d is not known, we instead can compute a $(4+\varepsilon)d$-orientation of G in $O(\log n)$ rounds.*

Multicolored induced trees with orientation. Let T be a tree on k nodes. We first to show how to solve a specific multicolored version of induced T-detection, given an acyclic orientation of G as input.

More precisely, let the graph G, let σ be an α-bounded orientation of G, and let $\chi \colon V \to \{0, 1, \ldots, k\}$ be a (not necessarily proper) $(k+1)$-coloring of G. Moreover, assume that the tree T is labelled in a bottom-up manner with $1, 2, \ldots, k$ with an arbitrary node as a root – that is, the root has label k, and each node has a smaller label than their parent. We say that an induced copy H of T in G is *proper* w.r.t σ and χ if the node in H corresponding to node i in T has color i, and every node that is an out-neighbor of some node in H has color 0.

▶ **Lemma 20.** *Given a graph $G = (V, E)$, an orientation σ of G, and a coloring χ as input, we can find a proper induced copy of a tree T in $O(k)$ rounds using $O(1)$-bit messages in* CONGEST *model.* (▷ See full version.)

Induced trees. Using Lemma 20 as a subroutine, we now show how to detect induced copies of any tree T. We use random separation [13] and color-coding [5] techniques to reduce the general problem to detection of proper induced copies of T.

▶ **Theorem 21.** *Finding induced copy of a tree T on k nodes in a d-degenerate graph G can be done in $k2^{O(dk)}k^k + O(\log n)$ rounds in the* CONGEST *model using a randomized algorithm.* (▷ See full version.)

Derandomization. Finally, we note that the algorithms can be derandomized using standard derandomization tools from fixed-parameter algorithms. Specifically, we use the derandomization of Alon and Gutner [4] to avoid incurring extra $O(\log n)$ factor that would follow from the original derandomization of Cai et al. [13].

▶ **Theorem 22.** *Finding induced copy of a tree T on k nodes in a d-degenerate graph G can be done in $f(d,k) + O(\log n)$ rounds in the* CONGEST *model using a deterministic algorithm for some function f, assuming d is known to all nodes.* (▷ See full version.)

Unknown degeneracy. The only part where the randomized algorithm uses the knowledge of $d(G)$ is for deciding how many repeats of the random coloring it performs; Lemma 19 can be used without knowing $d(G)$. Without knowledge of $d(G)$, nodes can determine the largest out-degree in orientation σ in their radius-k neighborhood and use that as a proxy for $d(G)$ to determine how many repeats of the random coloring they should participate in; it is easy to verify that this still retains the correctness of the algorithm. The only caveat is that different nodes can terminate at different times, and cannot determine when all nodes have terminated.

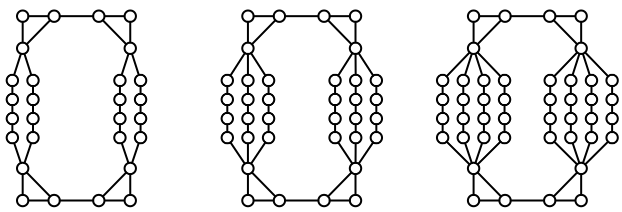
The deterministic algorithm, on the other hand, requires that all nodes know the degeneracy $d(G)$, or the same upper bound for this value. While we can compute an $O(\kappa(G))$-orientation σ for G in $O(\log n)$ rounds, all nodes do not necessarily learn the largest out-degree in σ; indeed, one can trivially see that having all nodes learn $d(G)$ requires $\Omega(D)$ rounds in the worst case.

5.2 Induced subgraph detection for bounded treewidth patterns

We now show that with slight modification, the hard treewidth 2 patterns presented in Section 3.4 can be adapted to bounded degeneracy setting. Recall that as mentioned in the introduction, any pattern graph on k nodes can be detected in $O(kd(G)n)$ rounds by having all nodes gather full information about their distance-k neighborhood; thus, the following lower bound is almost tight.

▶ **Theorem 23.** *For any $k \geq 2$, there exists a pattern graph H_k of treewidth 2 such that* CONGEST *algorithm solving either H-detection or induced H-detection on graphs of degeneracy 2 needs at least $\Omega(n^{1-1/k})$ rounds.*

We use the same construction for $k \geq 2$ for the pattern graph as in Lemma 13, but add paths of length 5 instead of paths of length 3 between triangles A_1 and B_1, and triangles A_2 and B_2. Let us denote the resulting graph by H'_k.



▶ **Lemma 24.** *Let $k \geq 2$ be fixed. There exists a family of lower bound graphs of degeneracy 2 for H'_k-detection and induced H'_k-detection with $f_n = \mathsf{DISJ}_{\Theta(n)}$ and $C(n) = \Theta(n^{1/k})$.*
(▷ See full version.)

6 Bounded vertex cover number and MCIS

Finally, we consider induced subgraph detection parameterized by vertex cover number $\tau(G)$. Specifically, we show that a more general problem of *maximum common induced subgraph (MCIS)* can be solved in constant rounds on graphs of constant vertex cover number, which implies our results for induced subgraph detection.

Maximum common induced subgraph. In the centralized version of maximum common induced subgraph, we are given graphs $G = (V_G, E_G)$ and $H = (V_H, E_H)$ as input, and the task is to find the maximum-size graph G^* such that G^* appears as induced subgraph of both G and H. More precisely, the output should be a function $f\colon V_G \to V_H \cup \{\bot\}$ such that for the set $U_G = \{v \in V_G \colon f(v) \neq \bot\}$, the function f restricted to U_G is an isomorphism between $G[U_G]$ and $H[f(U_G)]$.

In this section, we consider MCIS parameterized by the sum of the vertex cover numbers $\tau(G) + \tau(H)$. Note that when H is complete graph and $|G| = |H|$, the problem is equivalent to maximum clique and hence NP-hard. It is W[1]-hard when parameterized by the solution size k, and W[1]-hard parameterized by the size of a minimum vertex cover of only one of the input graphs, even when restricted to bipartite graphs (see e.g. [2, 3] for more discussion).

Distributed MCIS. In the distributed version of the MCIS problem, the input graph $G = (V_G, E_G)$ is the communication network, and full information about the second input graph $H = (V_H, E_H)$ is given to every node as local input. Each node v needs to give a local output $f(v) \in V_H \cup \{\bot\}$ such that the global function f satisfies the conditions of MCIS solution.

▶ **Theorem 25.** *Solving the maximum common induced subgraph problem on communication graph G and target graph H can be done in $2^{O(\tau^2)}$ rounds in the* CONGEST *model deterministically, where $\tau = \max(\tau(G), \tau(H))$.* (▷ See full version.)

Induced subgraph detection on bounded vertex cover number graphs. As an immediate consequence of the MCIS algorithm, we obtain a parameterized distributed algorithm for detecting an induced copy of H, for any pattern graph H, as a graph H on k nodes has vertex cover number at most k.

▶ **Theorem 26.** *Let H be a pattern graph on k nodes. Finding induced copy H can be done in $2^{O((\tau(G)+k)^2)}$ rounds in the* CONGEST *model deterministically.*

7 Conclusions and open problems

A central takeaway of this work is that centralized parameterized complexity offers both algorithmic techniques and perspectives for distributed computing. In particular, we believe that the study of structural graph parameters is a valuable paradigm for understanding sparse and structured networks in general. However, we note that there still remain open research directions related to topics studied in this paper:

- In terms of immediate open questions left by our work, we note that we currently do not have any systematic results on separation between the hardness of induced and non-induced subgraph detection for a given pattern H. For example, the induced cycle detection lower bound of Le Gall and Miyamoto [36] gives a near-linear – or super-linear, in case of even cycles – gap between induced and non-induced cycle detection, but it would be interesting to explore similar results for other pattern graphs in systematic fashion.

- More generally, we do not understand the complexity of subgraph detection type problems in distributed setting as well as in the centralized setting. For example, the complexity of k-independent set detection in CONGEST remains open, whereas in the centralized setting, it is equivalent to k-clique – a correspondence that does not hold in CONGEST.
- Besides degeneracy and vertex cover number, there are many other structural graph parameters commonly studied in parameterized complexity – for example, feedback vertex and edge sets, treewidth and pathwidth. Whereas Li [37] provides a framework for using treewidth for global optimization problems, it does not directly imply results for local problems such as subgraph detection; one might expect that considering something akin to *local* treewidth of a graph would be more appropriate for local graph problems. A secondary question is understanding what structural graph parameters are relevant from the perspective of real-world networks.

References

1 Amir Abboud, Keren Censor-Hillel, and Seri Khoury. Near-linear lower bounds for distributed distance computations, even in sparse networks. In *Proc. 30th International Symposium on Distributed Computing (DISC 2016)*, 2016. doi:10.1007/978-3-662-53426-7_3.

2 Faisal N. Abu-Khzam. Maximum common induced subgraph parameterized by vertex cover. *Information Processing Letters*, 114(3):99–103, 2014. doi:10.1016/j.ipl.2013.11.007.

3 Faisal N. Abu-Khzam, Édouard Bonnet, and Florian Sikora. On the complexity of various parameterizations of common induced subgraph isomorphism. *Theoretical Computer Science*, 697:69–78, 2017. doi:10.1016/j.tcs.2017.07.010.

4 Noga Alon and Shai Gutner. Linear time algorithms for finding a dominating set of fixed size in degenerated graphs. In *Proc. 13th International Computing and Combinatorics Conference (COCOON 2007)*, pages 394–405. Springer, 2007.

5 Noga Alon, Raphael Yuster, and Uri Zwick. Color-coding. *Journal of the ACM*, 42(4):844–856, 1995. doi:10.1145/210332.210337.

6 Nir Bacrach, Keren Censor-Hillel, Michal Dory, Yuval Efron, Dean Leitersdorf, and Ami Paz. Hardness of distributed optimization. In *ProcȦCM Symposium on Principles of Distributed Computing (PODC 2019)*, pages 238–247, 2019.

7 Leonid Barenboim and Michael Elkin. Sublogarithmic distributed MIS algorithm for sparse graphs using Nash-Williams decomposition. *Distributed Computing*, 22(5–6):363–379, 2010. doi:10.1007/s00446-009-0088-2.

8 Leonid Barenboim and Michael Elkin. *Distributed Graph Coloring: Fundamentals and Recent Developments*. Synthesis Lectures on Distributed Computing Theory. Morgan & Claypool, 2013. doi:10.2200/S00520ED1V01Y201307DCT011.

9 Leonid Barenboim and Victor Khazanov. Distributed symmetry-breaking algorithms for congested cliques. In *International Computer Science Symposium in Russia*, pages 41–52. Springer, 2018.

10 Ran Ben-Basat, Ken ichi Kawarabayashi, and Gregory Schwartzman. Parameterized Distributed Algorithms. In *Proc. 33rd International Symposium on Distributed Computing (DISC 2019)*, 2019. doi:10.4230/LIPIcs.DISC.2019.6.

11 Andreas Björklund, Thore Husfeldt, Petteri Kaski, and Mikko Koivisto. Narrow sieves for parameterized paths and packings. *Journal of Computer and System Sciences*, 87:119–139, 2017. doi:10.1016/j.jcss.2017.03.003.

12 Andreas Brandstädt, Van Bang Le, and Jeremy P. Spinrad. *Graph classes: a survey*. SIAM Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics, 1999.

13 Leizhen Cai, Siu Man Chan, and Siu On Chan. Random separation: A new method for solving fixed-cardinality optimization problems. In *Proc. 2nd International Workshop on Parameterized and Exact Computation (IWPEC 2006)*, pages 239–250. Springer, 2006.

14 Keren Censor-Hillel, Yi-Jun Chang, François Le Gall, and Dean Leitersdorf. Tight distributed listing of cliques. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA 2021)*, 2021.

15 Keren Censor-Hillel, Orr Fischer, Tzlil Gonen, François Le Gall, Dean Leitersdorf, and Rotem Oshman. Fast Distributed Algorithms for Girth, Cycles and Small Subgraphs. In Hagit Attiya, editor, *Proc. 34th International Symposium on Distributed Computing (DISC 2020)*, volume 179, 2020. doi:10.4230/LIPIcs.DISC.2020.33.

16 Keren Censor-Hillel, Petteri Kaski, Janne H. Korhonen, Christoph Lenzen, Ami Paz, and Jukka Suomela. Algebraic methods in the congested clique. In *Proc. ACM Symposium on Principles of Distributed Computing (PODC 2015)*, pages 143–152, 2015.

17 Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and near-quadratic lower bounds for the CONGEST model. In *Proc. 31st International Symposium on Distributed Computing (DISC 2017)*, 2017. doi:10.4230/LIPIcs.DISC.2017.10.

18 Yi-Jun Chang and Thatchaphol Saranurak. Improved distributed expander decomposition and nearly optimal triangle enumeration. In *Proc. ACM Symposium on Principles of Distributed Computing (PODC 2019)*, pages 66–73, 2019.

19 Yijia Chen and Jorg Flum. On parameterized path and chordless path problems. In *Proc. 22nd Annual IEEE Conference on Computational Complexity (CCC 2007)*, pages 250–263, 2007.

20 Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

21 Artur Czumaj and Christian Konrad. Detecting cliques in congest networks. *Distributed Computing*, 2019. doi:10.1007/s00446-019-00368-w.

22 Mina Dalirrooyfard, Thuy Duong Vuong, and Virginia Vassilevska Williams. Graph pattern detection: Hardness for all induced patterns and faster non-induced cycles. In *Proc. 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC 2019)*, pages 1167–1178, 2019. doi:10.1145/3313276.3316329.

23 Danny Dolev, Christoph Lenzen, and Shir Peled. “Tri, tri again”: Finding triangles and small subgraphs in a distributed setting. In *Proc. 26th International Symposium on Distributed Computing (DISC 2012)*, pages 195–209, 2012. doi:10.1007/978-3-642-33651-5_14.

24 Rod G. Downey and Michael R. Fellows. Fixed-parameter tractability and completeness II: On completeness for W[1]. *Theoretical Computer Science*, 141(1):109–131, 1995. doi:10.1016/0304-3975(94)00097-3.

25 Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *Proc. 33rd ACM Symposium on Principles of Distributed Computing (PODC 2014)*, pages 367–376, 2014. doi:10.1145/2611462.2611493.

26 Guy Even, Orr Fischer, Pierre Fraigniaud, Tzlil Gonen, Reut Levi, Moti Medina, Dennis Olivetti Pedro Montealegre, Rotem Oshman, Ivan Rapaport, and Ioan Todinca. Three notes on distributed property testing. In *Proc. 31st International Symposium on Distributed Computing (DISC 2017)*, 2017.

27 Michael R. Fellows, Danny Hermelin, Frances Rosamond, and Stéphane Vialette. On the parameterized complexity of multiple-interval graph problems. *Theoretical Computer Science*, 410(1):53–61, 2009. doi:10.1016/j.tcs.2008.09.065.

28 Orr Fischer, Tzlil Gonen, Fabian Kuhn, and Rotem Oshman. Possibilities and impossibilities for distributed subgraph detection. In *Proc. 30th Symposium on Parallelism in Algorithms and Architectures (SPAA 2018)*, pages 153–162, 2018.

29 Pierre Fraigniaud and Dennis Olivetti. Distributed detection of cycles. In *Proc. 29th ACM Symposium on Parallelism in Algorithms and Architectures (SPAA 2017)*, pages 153–162, 2017. doi:10.1145/3087556.3087571.

30 Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In *Proc. 23rd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, 2012.

31 Mohsen Ghaffari and Ali Sayyadi. Distributed arboricity-dependent graph coloring via all-to-all communication. In *Proc. 46th International Colloquium on Automata, Languages, and Programming (ICALP 2019)*, 2019.

32 Janne H. Korhonen and Amir Nikabadi. Beyond distributed subgraph detection: Induced subgraphs, multicolored problems and graph parameters, 2021. `arXiv:2109.06561`.

33 Janne H. Korhonen and Joel Rybicki. Deterministic subgraph detection in broadcast CONGEST. In *Proc. 21st International Conference on Principles of Distributed Systems (OPODIS 2017)*, 2017. `doi:10.4230/LIPIcs.OPODIS.2017.4`.

34 Janne H. Korhonen and Jukka Suomela. Towards a complexity theory for the congested clique. In *Proc̊30th on Symposium on Parallelism in Algorithms and Architectures (SPAA 2018)*, pages 163–172, 2018.

35 Eyal Kushilevitz and Noam Nisan. *Communication complexity*. Cambridge University Press, 1997.

36 François Le Gall and Masayuki Miyamoto. Lower bounds for induced cycle detection in distributed computing. In *Proc. 32nd International Symposium on Algorithms and Computation (ISAAC 2021)*, 2021. `arXiv:2110.00741`.

37 Jason Li. Distributed treewidth computation, 2018. `arXiv:1805.10708`.

38 Burkhard Monien. How to find long paths efficiently. *North-Holland Mathematics Studies*, 109:239–254, 1985.

39 Jaroslav Nešetřil and Svatopluk Poljak. On the complexity of the subgraph problem. *Commentationes Mathematicae Universitatis Carolinae*, 26(2):415–419, 1985.

40 Krzysztof Pietrzak. On the parameterized complexity of the fixed alphabet shortest common supersequence and longest common subsequence problems. *Journal of Computer and System Sciences*, 67(4):757–771, 2003. `doi:10.1016/S0022-0000(03)00078-3`.

41 Alexander A. Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992. `doi:10.1016/0304-3975(92)90260-M`.

42 Atish Das Sarma, Stephan Holzer, Liah Kor, Amos Korman, Danupon Nanongkai, Gopal Pandurangan, David Peleg, and Roger Wattenhofer. Distributed verification and hardness of distributed approximation. *SIAM Journal on Computing*, 41:1235–1265, 2012. `doi:10.1137/11085178X`.

43 Sebastian Siebertz and Alexandre Vigny. Parameterized distributed complexity theory: A logical approach. *arXiv preprint*, 2019. `arXiv:1903.00505`.

44 Virginia Vassilevska. *Efficient algorithms for path problems in weighted graphs*. PhD thesis, Carnegie Mellon University, 2008.

45 Ryan Williams. Finding paths of length k in $O^*(2^k)$ time. *Information Processing Letters*, 109(6):315–318, 2009.

46 Raphael Yuster and Uri Zwick. Finding even cycles even faster. *SIAM Journal on Discrete Mathematics*, 10(2):209–222, 1997.

A Induced short paths

Induced paths with two edges can be detected in $O(1)$ rounds, in contrast to the situation with e.g. triangle detection. The proof follows the centralized algorithm of Vassilevska [44].

▶ **Theorem 27.** *Given a graph G on n nodes, detecting an induced path of length 2 on G can be done in $O(1)$ rounds in the broadcast* CONGEST *model.*

Proof. As the first step, we assign a label $\ell(v)$ for each node as follows. First, each node $v \in V$ broadcast its identifiers to all its neighbors $N(v)$, and then each node v picks the label $\ell(v)$ to be the smallest identifier from the set that it received, or its own identifier if that is smaller. The nodes then broadcast their label $\ell(v)$ and and their degree $\deg(v)$ to all their neighbors.

Each node v then checks the following conditions, and reports that induced 2-path exists if at least one of them is satisfied:

1. The exists a neighbor $u \in N(v)$ with $\deg(v) \neq \deg(v)$.
2. There exists neighbors $u, w \in N(v)$ with $\ell(u) \neq \ell(w)$.

For the correctness of the algorithm, we first observe that a graph does not contain an induced 2-path if and only if each connected component is a clique. If none of the nodes report an induced 2-path, then by conditions (a) and (b), each connected component is clique. Likewise, if G consists of disjoint cliques, no node will report an induced 2-path. Finally, we note that the algorithm takes 3 rounds in CONGEST. ◀