Quadratic and Near-Quadratic Lower Bounds for the CONGEST Model*†

Keren Censor-Hillel[1], Seri Khoury[2], and Ami Paz[3]

1 **Department of Computer Science, Technion, Haifa, Israel**
ckeren@cs.technion.ac.il

2 **Department of Computer Science, Technion, Haifa, Israel**
serikhoury@cs.technion.ac.il

3 **Department of Computer Science, Technion, Haifa, Israel**
amipaz@cs.technion.ac.il

Abstract

We present the first super-linear lower bounds for natural graph problems in the CONGEST model, answering a long-standing open question.

Specifically, we show that any exact computation of a minimum vertex cover or a maximum independent set requires a near-quadratic number of rounds in the CONGEST model, as well as any algorithm for computing the chromatic number of the graph. We further show that such strong lower bounds are not limited to NP-hard problems, by showing two simple graph problems in P which require a quadratic and near-quadratic number of rounds.

Finally, we address the problem of computing an exact solution to weighted all-pairs-shortest-paths (APSP), which arguably may be considered as a candidate for having a super-linear lower bound. We show a simple linear lower bound for this problem, which implies a separation between the weighted and unweighted cases, since the latter is known to have a sub-linear complexity. We also formally prove that the standard Alice-Bob framework is incapable of providing a super-linear lower bound for exact weighted APSP, whose complexity remains an intriguing open question.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems, G.2.2 Graph Theory

Keywords and phrases CONGEST, Lower Bounds, Minimum Vertex Cover, Chromatic Number, Weighted APSP

Digital Object Identifier 10.4230/LIPIcs.DISC.2017.10

1 Introduction

It is well-known and easily proven that many graph problems are *global* for distributed computing, in the sense that solving them necessitates communication throughout the network. This implies tight $\Theta(D)$ complexities, where D is the diameter of the network, for global problems in the LOCAL model. In this model, a message of unbounded size can be sent over each edge in each round, which allows to learn the entire topology in D rounds. Global problems are widely studied in the CONGEST model, in which the size of each message is restricted to $O(\log n)$ bits, where n is the size of the network. The trivial complexity of learning the entire topology of an m-edges graph in the CONGEST model is

* Supported by ISF Individual Research Grant 1696/14.

† A full version of the paper is available at `https://arxiv.org/abs/1705.05646`.

$O(m)$, and since m can be as large as $\Theta(n^2)$, one of the most basic questions for a global problem is how fast in terms of n it can be solved in the CONGEST model.

Some global problems admit fast $O(D)$-round solutions in the CONGEST model, such as constructing a breadth-first search tree [60]. Some others have complexities of $\tilde{\Theta}(D + \sqrt{n})$, such as constructing a minimum spanning tree, and various approximation and verification problems [33, 40, 46, 61, 62, 65]. Some problems are yet harder, with complexities that are near-linear in n [1, 33, 42, 52, 61]. For some problems, no $O(n)$ solutions are known and they are candidates to being even harder that the ones with linear-in-n complexities.

A major open question about global graph problems in the CONGEST model is whether natural graph problems for which a super-linear number of rounds is required indeed exist. In this paper, we answer this question in the affirmative. That is, our conceptual contribution is that **there exist super-linearly hard problems in the CONGEST model**. In fact, the lower bounds that we prove in this paper are as high as quadratic in n, or quadratic up to logarithmic factors, and hold even for networks of a constant diameter. Our lower bounds also imply linear and near-linear lower bounds for the CLIQUE-BROADCAST model.

We note that high lower bounds for the CONGEST model may be obtained rather artificially, by forcing large inputs and outputs that must be exchanged. However, we emphasize that all the problems for which we show our lower bounds can be reduced to simple decision problems, where each node needs to output a single bit. All inputs to the nodes, if any, consist of edge weights that can be represented by polylog$n$ bits.

Technically, we prove a lower bound of $\Omega(n^2/\log^2 n)$ on the number of rounds required for computing an exact minimum vertex cover, which also extends to computing an exact maximum independent set. This is in stark contrast to the recent $O(\log \Delta/\log\log \Delta)$-round algorithm of [8] for obtaining a $(2+\epsilon)$-approximation to the minimum vertex cover. Similarly, we give an $\Omega(n^2/\log^2 n)$ lower bound for 3-coloring a 3-colorable graph, which extends also for deciding whether a graph is 3-colorable, and also implies the same hardness for computing the chromatic number χ or computing a χ-coloring. These lower bounds hold even for randomized algorithms which succeed with high probability.[1]

An immediate question that arises is whether only NP-hard problems are super-linearly hard in the CONGEST model. We provide a negative answer to such a postulate, by showing two simple problems that admit polynomial-time sequential algorithms, but in the CONGEST model require $\Omega(n^2)$ rounds (identical subgraph detection) or $\Omega(n^2/\log n)$ rounds (weighted cycle detection). The latter also holds for randomized algorithms, while for the former we show a randomized algorithm that completes in $O(D)$ rounds, providing the strongest possible separation between deterministic and randomized complexities for global problems in the CONGEST model.

Finally, we address the intriguing open question of the complexity of computing exact weighted all-pairs-shortest-paths (APSP) in the CONGEST model. While the complexity of the unweighted version of APSP is $\Theta(n/\log n)$, as follows from [33, 43], the complexity of weighted APSP remains largely open, and only recently the first sub-quadratic algorithm was given in [29]. With the current state-of-the-art, this problem could be considered as a suspect for having a super-linear complexity in the CONGEST model. While we do not pin-down the complexity of weighted APSP in the CONGEST model, we provide a truly linear lower bound of $\Omega(n)$ rounds for it, which separates its complexity from that of the unweighted case. Moreover, we argue that it is not a coincidence that we are currently unable to show

[1] We say that an event occurs with high probability (w.h.p) if it occurs with probability at least $1 - \frac{1}{n^c}$, for some constant $c > 0$.

super-linear lower bound for weighted APSP, by formally proving that the commonly used framework of reducing a 2-party communication problem to a problem in the CONGEST model cannot provide a super-linear lower bound for weighted APSP, regardless of the function and the graph construction used. This implies that **obtaining any super-linear lower bound for weighted APSP provably requires a new technique**.

1.1 The Challenge

Many lower bounds for the CONGEST model rely on reductions from 2-party communication problems (see, e.g., [1, 17, 26, 28, 33, 42, 57, 58, 62, 65]). In this setting, two players, Alice and Bob, are given inputs of K bits and need to a single output a bit according to some given function of their inputs. One of the most common problem for reduction is Set Disjointness, in which the players need to decide whether there is an index for which both inputs are 1. That is, if the inputs represent subsets of $\{0, \ldots, K-1\}$, the output bit of the players needs to indicate whether their input sets are disjoint. The communication complexity of 2-party Set Disjointness is known to be $\Theta(K)$ [50].

In a nutshell, there are roughly two standard frameworks for reducing the 2-party communication problem of computing a function f to a problem P in the CONGEST model. One of these frameworks works as follows. A graph construction is given, which consists of some fixed edges and some edges whose existence depends on the inputs of Alice and Bob. This graph should have the property that a solution to P over it determines the solution to f. Then, given an algorithm ALG for solving P in the CONGEST model, the vertices of the graph are split into two disjoint sets, V_A and V_B, and Alice simulates ALG over V_A while Bob simulates ALG over V_B. The only communication required between Alice and Bob in order to carry out this simulation is the content of messages sent in each direction over the edges of the cut $C = E(V_A, V_B)$. Therefore, given a graph construction with a cut of size $|C|$ and inputs of size K for a function f whose communication complexity on K bits is at least $CC(f)$, the round complexity of ALG is at least $\Omega(CC(f)/|C| \log n)$.

The challenge in obtaining super-linear lower bounds was previously that the cuts in the graph constructions were large compared with the input size K. For example, the graph construction for the lower bound for computing the diameter in [33] has $K = \Theta(n^2)$ and $|C| = \Theta(n)$, which gives an almost linear lower bound. The graph construction in [33] for the lower bound for computing a $(3/2 - \epsilon)$-approximation to the diameter has a smaller cut of $|C| = \Theta(\sqrt{n})$, but this comes at the price of supporting a smaller input size $K = \Theta(n)$, which gives a lower bound that is roughly a square-root of n.

To overcome this difficulty, we leverage the recent framework of [1], which provides a bit-gadget whose power is in allowing a logarithmic-size cut. We manage to provide a graph construction that supports inputs of size $K = \Theta(n^2)$ in order to obtain our lower bounds for minimum vertex cover, maximum independent set and 3-coloring[2]. The latter is also inspired by, and is a simplification of, a lower bound construction for the size of proof labelling schemes [34].

Further, for the problems in P that we address, the cut is as small as $|C| = O(1)$. For one of the problems, the size of the input is such that it allows us to obtain the highest possible lower bound of $\Omega(n^2)$ rounds.

[2] It can also be shown, by simple modifications to our constructions, that these problems require $\Omega(m)$ rounds, for graphs with m edges.

With respect to the complexity of the weighted APSP problem, we show an embarrassingly simple graph construction that extends a construction of [57], which leads to an $\Omega(n)$ lower bound. However, we argue that a new technique must be developed in order to obtain any super-linear lower bound for weighted APSP. Roughly speaking, this is because given a construction with a set S of nodes that touch the cut, Alice and Bob can exchange $O(|S|n \log n)$ bits which encode the weights of all lightest paths from any node in their set to a node in S. Since the cut has $\Omega(|S|)$ edges, and the bandwidth is $\Theta(\log n)$, this cannot give a lower bound of more than $\Omega(n)$ rounds. With some additional work, our proof can be carried over to a larger number of players at the price of a small logarithmic factor, as well as to the second Alice-Bob framework used in previous work (e.g. [65]), in which Alice and Bob do not simulate nodes in a fixed partition, but rather in decreasing sets that partially overlap. Thus, determining the complexity of weighted APSP requires new tools, which we leave as a major open problem.

Roadmap. Section 3 contains our lower bound for computing exact minimum vertex cover or exact maximum independent set. In Section 4 we show our lower bound for computing exact weighted-all-pairs-shortest-paths, and prove that the Alice-Bob framework cannot give a super-linear lower bound for this task. Due to space limitations, our lower bounds for 3-coloring a 3-colorable graph, identical subgraphs detection, and weighted cycle detection appear only in the full version of the paper [18].

1.2 Additional Related Work

Vertex Coloring, Minimum Vertex Cover, and Maximum Independent Set: One of the most central problems in graph theory is vertex coloring, which has been extensively studied in the context of distributed computing (see, e.g., [9, 10, 11, 12, 13, 14, 19, 21, 22, 30, 31, 32, 38, 54, 56, 63, 66] and references therein). The special case of finding a $(\Delta+1)$-coloring, where Δ is the maximum degree of a node in the network, has been the focus of many of these studies, but is a *local* problem, which can be solved in much less than a sublinear number of rounds.

Another classical problem in graph theory is finding a minimum vertex cover (MVC). In distributed computing, the time complexity of approximating MVC has been addressed in several cornerstone studies [5, 6, 8, 14, 35, 36, 37, 45, 47, 48, 49, 59, 64].

Observe that finding a minimum size vertex cover is equivalent to finding a maximum size independent set. However, these problems are not equivalent in an approximation-preserving way. Distributed approximations for maximum independent set has been studied in [7, 15, 23, 53].

Distance Computations: Distance computation problems have been widely studied in the CONGEST model for both weighted and unweighted networks [1, 33, 39, 40, 41, 42, 43, 51, 52, 57, 61]. One of the most fundamental problems of distance computations is computing all pairs shortest paths. For unweighted networks, an upper bound of $O(n/\log n)$ was recently shown by [43], matching the lower bound of [33]. Moreover, the possibility of bypassing this near-linear barrier for any constant approximation factor was ruled out by [57]. For the weighted case, however, we are still very far from understanding the complexity of APSP, as there is still a huge gap between the upper and lower bounds. Recently, Elkin [29] showed an $O(n^{\frac{5}{3}} \cdot \log^{\frac{2}{3}}(n))$ upper bound for weighted APSP, while the previously highest lower bound was the near-linear lower bound of [57] (which holds also for any $(\text{poly}\, n)$-approximation factor in the weighted case).

Distance computation problems have also been considered in the CONGESTED-CLIQUE model [16, 39, 41], in which the underlying communication network forms a clique. In this model [16] showed that unweighted APSP, and a $(1 + o(1))$-approximation for weighted APSP, can be computed in $O(n^{0.158})$ rounds.

Subgraph Detection: The problem of finding subgraphs of a certain topology has received a lot of attention in both the sequential and the distributed settings (see, e.g., [2, 3, 4, 16, 24, 25, 26, 44, 55, 67] and references therein).

The problems of finding paths of length 4 or 5 with zero weight are also related to other fundamental problems, notable in our context is APSP [2].

2 Preliminaries

2.1 Communication Complexity

In a two-party communication complexity problem [50], there is a function $f : \{0,1\}^K \times \{0,1\}^K \to \{\text{TRUE}, \text{FALSE}\}$, and two players, Alice and Bob, who are given two input strings, $x, y \in \{0,1\}^K$, respectively, that need to compute $f(x, y)$. The *communication complexity* of a protocol π for computing f, denoted $CC(\pi)$, is the maximal number of bits Alice and Bob exchange in π, taken over all values of the pair (x, y). The *deterministic communication complexity* of f, denoted $CC(f)$, is the minimum over $CC(\pi)$, taken over all deterministic protocols π that compute f.

In a *randomized protocol* π, Alice and Bob may each use a random bit string. A randomized protocol π computes f if the probability, over all possible bit strings, that π outputs $f(x, y)$ is at least 2/3. The *randomized communication complexity* of f, $CC^R(f)$, is the minimum over $CC(\pi)$, taken over all randomized protocols π that compute f.

In the *Set Disjointness* problem (DISJ_K), the function f is $\text{DISJ}_K(x, y)$, whose output is FALSE if there is an index $i \in \{0, \ldots, K-1\}$ such that $x_i = y_i = 1$, and TRUE otherwise. In the *Equality* problem (EQ_K), the function f is $\text{EQ}_K(x, y)$, whose output is TRUE if $x = y$, and FALSE otherwise.

Both the deterministic and randomized communication complexities of the DISJ_K problem are known to be $\Omega(K)$ [50, Example 3.22]. The deterministic communication complexity of EQ_K is in $\Omega(K)$ [50, Example 1.21], while its randomized communication complexity is in $\Theta(\log K)$ [50, Example 3.9].

2.2 Lower Bound Graphs

To prove lower bounds on the number of rounds necessary in order to solve a distributed problem in the CONGEST model, we use reductions from two-party communication complexity problems. To formalize them we use the following definition. Let $\mathcal{G}$ be the set of all graphs.

▶ **Definition 1** (Family of Lower Bound Graphs). Fix an integer K, a function $f : \{0,1\}^K \times \{0,1\}^K \to \{\text{TRUE}, \text{FALSE}\}$ and a predicate $P : \mathcal{G} \to \{\text{TRUE}, \text{FALSE}\}$. The family of graphs $\{G_{x,y} = (V, E_{x,y}) \mid x, y \in \{0,1\}^K\}$, is said to be a family of *lower bound graphs w.r.t. f and P* if the following properties hold:

(1) The set of nodes V is the same for all graphs, and we denote by $V = V_A \dot\cup V_B$ a fixed partition of it;
(2) The existence or the weight of edges in $V_A \times V_A$ may depend on x;
(3) The existence or the weight of edges in $V_B \times V_B$ may depend on y;
(4) $P(G_{x,y}) = f(x, y)$.

We use the following theorem, which is standard in the context of communication complexity-based lower bounds for the CONGEST model (see, e.g. [1, 26, 33, 41]) Its proof is by a standard simulation argument.

▶ **Theorem 2.** *Fix a function $f : \{0,1\}^K \times \{0,1\}^K \to \{\text{TRUE}, \text{FALSE}\}$ and a predicate P. If there is a family $\{G_{x,y}\}$ of lower bound graphs with $C = E(V_A, V_B)$ then any deterministic algorithm for deciding P in the CONGEST model requires $\Omega(CC(f)/|C|\log n)$ rounds, and any randomized algorithm for deciding P in the CONGEST model requires $\Omega(CC^R(f)/|C|\log n)$ rounds.*

Proof. Let ALG be a distributed algorithm in the CONGEST model that decides P in T rounds. Given inputs $x, y \in \{0,1\}^K$ to Alice and Bob, respectively, Alice constructs the part of $G_{x,y}$ for the nodes in V_A and Bob does so for the nodes in V_B. This can be done by items (1),(2) and (3) in Definition 1, and since $\{G_{x,y}\}$ satisfies this definition. Alice and Bob simulate ALG by exchanging the messages that are sent during the algorithm between nodes of V_A and nodes of V_B in either direction. (The messages within each set of nodes are simulated locally by the corresponding player without any communication). Since item (4) in Definition 1 also holds, we have that Alice and Bob correctly output $f(x, y)$ based on the output of ALG. For each edge in the cut, Alice and Bob exchange $O(\log n)$ bits per round. Since there are T rounds and $|C|$ edges in the cut, the number of bits exchanged in this protocol for computing f is $O(T|C|\log n)$. The lower bounds for T now follows directly from the lower bounds for $CC(f)$ and $CC^R(f)$. ◀

In what follows, for each decision problem addressed, we describe a fixed graph construction $G = (V, E)$, which we then generalize to a family of graphs $\{G_{x,y} = (V, E_{x,y}) \mid x, y \in \{0,1\}^K\}$, which we show to be a family lower bound graphs w.r.t. to some function f and the required predicate P. By Theorem 2 and the known lower bounds for the 2-party communication problem, we deduce a lower bound for any algorithm for deciding P in the CONGEST model.

▶ Remark. For our constructions that use the Set Disjointness function as f, we need to exclude the possibilities of all-1 input vectors, as otherwise the communication graph is not connected. However, this restriction does not change the asymptotic bounds for Set Disjointness, since computing this function while excluding all-1 input vectors can be reduced to computing this function for inputs that are shorter by one bit (by having the last bit be fixed to 0).

3 Minimum Vertex Cover and Maximum Independent Set

The first near-quadratic lower bound we present is for computing a minimum vertex cover, as stated in the following theorem.

▶ **Theorem 3.** *Any distributed algorithm in the CONGEST model for computing a minimum vertex cover or for deciding whether there is a vertex cover of a given size M requires $\Omega(n^2/\log^2 n)$ rounds.*

Finding the minimum size of a vertex cover is equivalent to finding the maximum size of a maximum independent set, because a set of nodes is a vertex cover if and only if its complement is an independent set. Thus, Theorem 4 is a direct corollary of Theorem 3.

▶ **Theorem 4.** *Any distributed algorithm in the CONGEST model for computing a maximum independent set or for deciding whether there is an independent set of a given size requires $\Omega(n^2/\log^2 n)$ rounds.*

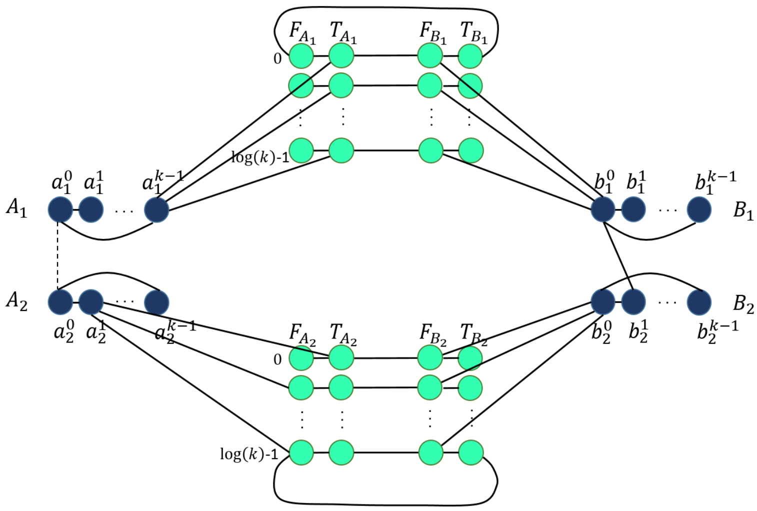

Figure 1 The family of lower bound graphs for deciding the size of a vertex cover, with many edges omitted for clarity. The node a_1^{k-1} is connected to all the nodes in T_{A_1}, and a_2^1 is connected to $t_{A_2}^0$ and to all the nodes in $F_{A_2} \setminus \{f_{A_2}^0\}$. Examples of edges from b_1^0 and b_2^0 to the bit-gadgets are also given. An additional edge, which is among the edges corresponding to the strings x and y, is $\{b_1^0, b_2^1\}$, while the edge $\{a_1^0, a_2^0\}$ does not exist. Here, $x_{0,0} = 1$ and $y_{0,1} = 0$.

Observe that a lower bound of L for deciding whether there is a vertex cover of some given size M or not implies a lower bound of $L - O(D)$ for computing a minimum vertex cover. This is because computing the size of a given subset of nodes can be easily done in $O(D)$ rounds using standard tools. Therefore, to prove Theorem 3 it is sufficient to prove its second part. We do so by describing a family of lower bound graphs with respect to the Set Disjointness function and the predicate P that says that the graph has a vertex cover of size M. We begin with describing the fixed graph construction $G = (V, E)$ and then define the family of lower bound graphs and analyze its relevant properties.

The fixed graph construction: Let k be a power of 2. The fixed graph (Figure 1) consists of four cliques of size k: $A_1 = \{a_1^i \mid 0 \leq i \leq k-1\}$, $A_2 = \{a_2^i \mid 0 \leq i \leq k-1\}$, $B_1 = \{b_1^i \mid 0 \leq i \leq k-1\}$ and $B_2 = \{b_2^i \mid 0 \leq i \leq k-1\}$. In addition, for each set $S \in \{A_1, A_2, B_1, B_2\}$, there are two corresponding sets of nodes of size $\log k$, denoted $F_S = \{f_S^h \mid 0 \leq h \leq \log k - 1\}$ and $T_S = \{t_S^h \mid 0 \leq h \leq \log k - 1\}$.

The latter are called *bit-gadgets* and their nodes are *bit-nodes*.

The bit-nodes are partitioned into $2 \log k$ 4-cycles: for each $h \in \{0, \dots, \log k - 1\}$ and $\ell \in \{1, 2\}$, we connect the 4-cycle $(f_{A_\ell}^h, t_{A_\ell}^h, f_{B_\ell}^h, t_{B_\ell}^h)$. Note that there are no edges between pairs of nodes denoted f_S^h, or between pairs of nodes denoted t_S^h.

The nodes of each set $S \in \{A_1, A_2, B_1, B_2\}$ are connected to nodes in the corresponding set of bit-nodes, according to their binary representation, as follows. Let s_ℓ^i be a node in a set $S \in \{A_1, A_2, B_1, B_2\}$, i.e. $s \in \{a, b\}$, $\ell \in \{1, 2\}$ and $i \in \{0, \dots, k-1\}$, and let i_h denote the bit number h in the binary representation of i. For such a node s_ℓ^i define $\text{bin}(s_\ell^i) = \left\{f_S^h \mid i_h = 0\right\} \cup \left\{t_S^h \mid i_h = 1\right\}$, and connect s_ℓ^i by an edge to each of the nodes in $\text{bin}(s_\ell^i)$. The next two claims address the basic properties of vertex covers of G.

▶ **Claim 5.** *Any vertex cover of G must contain at least $k - 1$ nodes from each of the clique A_1, A_2, B_1 and B_2, and at least $4 \log k$ bit-nodes.*

Proof. In order to cover all the edges of each if the cliques on A_1, A_2, B_1 and B_2, any vertex cover must contain at least $k-1$ nodes of the clique. For each $h \in \{0, \ldots, \log k - 1\}$ and $\ell \in \{1, 2\}$, in order to cover the edges of the 4-cycle $(f^h_{A_\ell}, t^h_{A_\ell}, f^h_{B_\ell}, t^h_{B_\ell})$, any vertex cover must contain at least two of the cycle nodes. ◀

▶ **Claim 6.** *If $U \subseteq V$ is a vertex cover of G of size $4(k-1) + 4\log k$, then there are two indices $i, j \in \{0, \ldots, k-1\}$ such that $a^i_1, a^j_2, b^i_1, b^j_2$ are not in U.*

Proof. By Claim 5, U must contain $k-1$ nodes from each clique A_1, A_2, B_1 and B_2, and $4\log k$ bit-nodes, so it must not contain one node from each clique. Let $a^i_1, a^j_2, b^{i'}_1, b^{j'}_2$ be the nodes in A_1, A_2, B_1, B_2 which are not in U, respectively. To cover the edges connecting a^i_1 to $\text{bin}(a^i_1)$, U must contain all the nodes of $\text{bin}(a^i_1)$, and similarly, U must contain all the nodes of $\text{bin}(b^{i'}_1)$.

If $i \neq i'$ then there is an index $h \in \{0, \ldots, \log k - 1\}$ such that $i_h \neq i'_h$. So whether both nodes of the edge $\left(f^h_{A_1}, t^h_{B_1}\right)$ are in U, or both nodes of $\left(t^h_{A_1}, f^h_{B_1}\right)$ are. However, U contains exactly $4\log k$ bit-nodes and at least two nodes from each 4-cycle, and a simple counting argument implies that U also contain at most two nodes from each 4-cycle. So, the other nodes of the 4-cycle $\{f^h_{A_1}, t^h_{B_1}, t^h_{A_1}, f^h_{B_1}\}$ are not in U, and the other edge is not covered. Thus, it must be the case that $i = i'$. A similar argument shows $j = j'$. ◀

Adding edges corresponding to the strings x and y: Given two binary strings $x, y \in \{0,1\}^{k^2}$, we augment the graph G defined above with additional edges, which defines $G_{x,y}$. Assume that x and y are indexed by pairs of the form $(i, j) \in \{0, \ldots, k-1\}^2$. For each such pair (i, j) we add to $G_{x,y}$ the following edges. If $x_{i,j} = 0$, then we add an edge between the nodes a^i_1 and a^j_2, and if $y_{i,j} = 0$ then we add an edge between the nodes b^i_1 and b^j_2. To prove that $\{G_{xy}\}$ is a family of lower bound graphs, it remains to prove the next lemma.

▶ **Lemma 7.** *The graph $G_{x,y}$ has a vertex cover of cardinality $M = 4(k-1) + 4\log k$ iff* $\text{DISJ}(x, y) = \text{FALSE}$.

Proof. For the first implication, assume that $\text{DISJ}(x, y) = \text{FALSE}$ and let $i, j \in \{0, \ldots, k-1\}$ be such that $x_{i,j} = y_{i,j} = 1$. Note that in this case a^i_1 is not connected to a^j_2, and b^i_1 is not connected to b^j_2. We define a set $U \subseteq V$ as the union of the two sets of nodes $(A_1 \setminus \{a^i_1\}) \cup (A_2 \setminus \{a^j_2\}) \cup (B_1 \setminus \{b^i_1\}) \cup (B_2 \setminus \{b^j_2\})$ and $\text{bin}(a^i_1) \cup \text{bin}(a^j_2) \cup \text{bin}(b^i_1) \cup \text{bin}(b^j_2)$, and show that U is a vertex cover of $G_{x,y}$.

First, U covers all the edges inside the cliques A_1, A_2, B_1 and B_2, as it contains $k-1$ nodes from each clique. These nodes also cover all the edges connecting nodes in A_1 to nodes in A_2 and all the edges connecting nodes in B_1 to nodes in B_2. Furthermore, U covers any edge connecting some node $u \in (A_1 \setminus \{a^i_1\}) \cup (A_2 \setminus \{a^j_2\}) \cup (B_1 \setminus \{b^i_1\}) \cup (B_2 \setminus \{b^j_2\})$ with the bit-gadgets. For each node $s \in a^i_1, a^j_2, b^i_1, b^j_2$, the nodes $\text{bin}(s)$ are in U, so U also cover the edges connecting s to the bit gadget. Finally, U covers all the edges inside the bit gadgets, as from each 4-cycle $(f^h_{A_\ell}, t^h_{A_\ell}, f^h_{B_\ell}, t^h_{B_\ell})$ it contains two non-adjacent nodes: if $i_h = 0$ then $f^h_{A_1}, f^h_{B_1} \in U$ and otherwise $t^h_{A_1}, t^h_{B_1} \in U$, and if $j_h = 0$ then $f^h_{A_2}, f^h_{B_2} \in U$ and otherwise $t^h_{A_2}, t^h_{B_2} \in U$. We thus have that U is a vertex cover of size $4(k-1) + 4\log k$, as needed.

For the other implication, let $C \subseteq V$ be a vertex cover of $G_{x,y}$ of size $4(k-1) + 4\log k$. As the set of edges of G is contained in the set of edges of $G_{x,y}$, C is also a cover of G, and by Claim 6 there are indices $i, j \in \{0, \ldots, k-1\}$ such that $a^i_1, a^j_2, b^i_1, b^j_2$ are not in C. Since C is a cover, the graph does not contain the edges (a^i_1, a^j_2) and (b^i_1, b^j_2), so we conclude that $x_{i,j} = y_{i,j} = 1$, which implies that $\text{DISJ}(x, y) = \text{FALSE}$. ◀

Having constructed the family of lower bound graphs, we are now ready to prove Theorem 3.

Proof of Theorem 3. To complete the proof of Theorem 3, we divide the nodes of G (which are also the nodes of $G_{x,y}$) into two sets. Let $V_A = A_1 \cup A_2 \cup F_{A_1} \cup T_{A_1} \cup F_{A_2} \cup T_{A_2}$ and $V_B = V \setminus V_A$. Note that $n \in \Theta(k)$, and thus $K = |x| = |y| = \Theta(n^2)$. Furthermore, note that the only edges in the cut $E(V_A, V_B)$ are the edges between nodes in $\{F_{A_1} \cup T_{A_1} \cup F_{A_2} \cup T_{A_2}\}$ and nodes in $\{F_{B_1} \cup T_{B_1} \cup F_{B_2} \cup T_{B_2}\}$, which are in total $\Theta(\log n)$ edges. Since Lemma 7 shows that $\{G_{x,y}\}$ is a family of lower bound graphs, we can apply Theorem 2 on the above partition to deduce that because of the lower bound for Set Disjointness, any algorithm in the CONGEST model for deciding whether a given graph has a cover of cardinality $M = 4(k-1) + 4\log k$ requires at least $\Omega(K/\log^2(n)) = \Omega(n^2/\log^2(n))$ rounds. ◀

4 Weighted APSP

In this section we use the following natural extension of Definition 1, in order to address more general 2-party functions, as well as distributed problems that are not decision problems.

For a function $f : \{0,1\}^{K_1} \times \{0,1\}^{K_2} \to \{0,1\}^{L_1} \times \{0,1\}^{L_2}$ and a graph problem, we define a family of lower bound graphs in a way similar to Definition 1, replacing item (4) in Definition 1 with a generalized requirement: for $G_{x,y}$, the output values of the nodes in V_A in a solution to the problem uniquely determine the first L_1 bits of $f(x,y)$, and the output values of the of nodes in V_B uniquely determine the last L_2 bits of $f(x,y)$. Next, we argue that theorem similar to Theorem 2 holds for this case.

▶ **Theorem 8.** *Fix a function $f : \{0,1\}^{K_1} \times \{0,1\}^{K_2} \to \{0,1\}^{L_1} \times \{0,1\}^{L_2}$ and a graph problem P. If there is a family $\{G_{x,y}\}$ of lower bound graphs with $C = E(V_A, V_B)$ then any deterministic algorithm for solving P in the CONGEST model requires $\Omega(CC(f)/|C|\log n)$ rounds, and any randomized algorithm for deciding P in the CONGEST model requires $\Omega(CC^R(f)/|C|\log n)$ rounds.*

The proof is similar to that of Theorem 2. Notice that the only difference between the theorems, apart from the sizes of the inputs and outputs of f, are with respect to item (4) in the definition of a family of lower bound graphs. However, the essence of this condition remains the same and is all that is required by the proof: the values that a solution to P assigns to nodes in V_A determines the output of Alice for $f(x,y)$, and the values that a solution to P assigns to nodes in V_B determines the output of Bob for $f(x,y)$.

4.1 A Linear Lower Bound for Weighted APSP

Nanongkai [57] showed that any algorithm in the CONGEST model for computing a poly(n)-approximation for weighted all pairs shortest paths (APSP) requires at least $\Omega(n/\log n)$ rounds. In this section we show that a slight modification to this construction yields an $\Omega(n)$ lower bound for computing exact weighted APSP. As explained in the introduction, this gives a separation between the complexities of the weighted and unweighted versions of APSP. At a high level, while we use the same simple topology for our lower bound as in [57], the reason that we are able to restore the missing logarithmic factor is because our construction uses $O(\log n)$ bits for encoding the weight of each edge out of many optional weights, while in [57] only a single bit is used per edge for encoding one of only two options for its weight.

▶ **Theorem 9.** *Any distributed algorithm in the CONGEST model for computing exact weighted all pairs shortest paths requires at least $\Omega(n)$ rounds.*

The reduction is from the following, perhaps simplest, 2-party communication problem. Alice has an input string x of size K and Bob needs to learn the string of Alice. Any algorithm (possibly randomized) for solving this problem requires at least $\Omega(K)$ bits of communication, by a trivial information theoretic argument.

Notice that the problem of having Bob learn Alice's input is not a binary function as addressed in Section 2. Similarly, computing weighted APSP is not a decision problem, but rather a problem whose solution assigns a value to each node (which is its vector of distances from all other nodes). We therefore use the extended Theorem 8 above.

The fixed graph construction: The fixed graph construction $G = (V, E)$ is defined as follows. It contains a set of $n - 2$ nodes, denoted $A = \{a_0, ..., a_{n-3}\}$, which are all connected to an additional node a. The node a is connected to the last node b, by an edge of weight 0.

Adding edge weights corresponding to the string x: Given the binary string x of size $K = (n - 2)\log n$ we augment the graph G with edge weights, which defines G_x, by having each non-overlapping batch of $\log n$ bits encode a weight of an edge from A to a. It is straightforward to see that G_x is a family of lower bound graphs for a function f where $K_2 = L_1 = 0$, since the weights of the edges determine the right-hand side of the output (while the left-hand side is empty).

Proof of Theorem 9. To prove Theorem 9, we let $V_A = A \cup \{a\}$ and $V_B = \{b\}$. Note that $K = |x| = \Theta(n \log n)$. Furthermore, note that the only edge in the cut $E(V_A, V_B)$ is the edge (a, b). Since we showed that $\{G_x\}$ is a family of lower bound graphs, we apply Theorem 8 on the above partition to deduce that because K bits are required to be communicated in order for Bob to know Alice's K-bit input, any algorithm in the CONGEST model for computing weighted APSP requires at least $\Omega(K/\log n) = \Omega(n)$ rounds. ◀

4.2 The Alice-Bob Framework Cannot Give a Super-Linear Lower Bound for Weighted APSP

In this section we argue that a reduction from any 2-party function with a constant partition of the graph into Alice and Bob's sides is provable incapable of providing a super-linear lower bound for computing weighted all pairs shortest paths in the CONGEST model. A more detailed inspection of our analysis shows a stronger claim: our claim also holds for algorithms for the CONGEST-BROADCAST model, where in each round each node must send the same $(\log n)$-bit message to all of its neighbors. The following theorem states our claim.

▶ **Theorem 10.** *Let $f : \{0,1\}^{K_1} \times \{0,1\}^{K_2} \to \{0,1\}^{L_1} \times \{0,1\}^{L_2}$ be a function and let $G_{x,y}$ be a family of lower bound graphs w.r.t. f and the weighted APSP problem. When applying Theorem 8 to f and $G_{x,y}$, the lower bound obtained for the number of rounds for computing weighted APSP is at most linear in n.*

Roughly speaking, we show that given an input graph $G = (V, E)$ and a partition of the set of vertices into two sets $V = V_A \cup V_B$, such that the graph induced by the nodes in V_A is simulated by Alice and the graph induced by nodes in V_B is simulated by Bob, Alice and Bob can compute weighted all pairs shortest paths by communicating $O(n \log n)$ bits of information for each node touching the cut $C = (V_A, V_B)$ induced by the partition. This means that for any 2-party function f and any family of lower bound graphs w.r.t. f and weighted APSP according to the extended definition of Section 4.1, since Alice and Bob can compute weighted APSP which determines their output for f by exchanging only $O(|V(C)|n \log n)$ bits, where $V(C)$ is the set of nodes touching C, the value $CC(f)$ is at most $O(|V(C)|n \log n)$. But then the lower bound obtained by Theorem 8 cannot be better than $\Omega(n)$, and hence no super-linear lower can be deduced by this framework as is.

Formally, given a graph $G = (V = V_A \dot\cup V_B, E)$ we denote $C = E(V_A, V_B)$. Let $V(C)$ denote the nodes touching the cut C, with $C_A = V(C) \cap V_A$ and $C_B = V(C) \cap V_B$. Let $G_A = (V_A, E_A)$ be the subgraph induced by the nodes in V_A and let $G_B = (V_B, E_B)$ be the subgraph induced by the nodes in V_B. For a graph H, we denote the weighted distance between two nodes u, v by $\mathrm{wdist}_H(u, v)$.

▶ **Lemma 11.** *Let $G = (V = V_A \dot\cup V_B, E, w)$ be a graph with an edge-weight function $w : E \to \{1, \ldots, W\}$, such that $W \in \mathrm{poly}\, n$. Suppose that G_A, C_B, C and the values of w on E_A and C are given as input to Alice, and that G_B, C_A, C and the values of w on E_B and C are given as input to Bob.*

Then, Alice can compute the distances in G from all nodes in V_A to all nodes in V and Bob can compute the distances from all nodes in V_B to all the nodes in V, using $O(|V(C)|\, n \log n)$ bits of communication.

Proof. We describe a protocol for the required computation, as follows. For each node $u \in C_B$, Bob sends to Alice the weighted distances in G_B from u to all nodes in V_B, that is, Bob sends $\{\mathrm{wdist}_{G_B}(u, v) \mid u \in C_B, v \in V_B\}$ (or ∞ for pairs of nodes not connected in G_B). Alice constructs a virtual graph $G'_A = (V'_A, E'_A, w'_A)$ with the nodes $V'_A = V_A \cup C_B$ and edges $E'_A = E_A \cup C \cup (C_B \times C_B)$. The edge-weight function w'_A is defined by $w'_A(e) = w(e)$ for each $e \in E_A \cup C$, and $w'_A(u, v)$ for $u, v \in C_B$ is defined to be the weighted distance between u and v in G_B, as received from Bob. Alice then computes the set of all weighted distances in G'_A, $\{\mathrm{wdist}_{G'_A}(u, v) \mid u, v \in V'_A\}$.

Alice assigns her output for the weighted distances in G as follows. For two nodes $u, v \in V_A \cup C_B$, Alice outputs their weighted distance in G'_A, $\mathrm{wdist}_{G'_A}(u, v)$. For a node $u \in V'_A$ and a node $v \in V_B \setminus C_B$, Alice outputs $\min\{\mathrm{wdist}_{G'_A}(u, x) + \mathrm{wdist}_{G_B}(x, v) \mid x \in C_B\}$, where $\mathrm{wdist}_{G'_A}$ is the distance in G'_A as computed by Alice, and wdist_{G_B} is the distance in G_B that was sent by Bob.

For Bob to compute his required weighted distances, for each node $u \in C_A$, similar information is sent by Alice to Bob, that is, Alice sends to Bob the weighted distances in G_A from u to all nodes in V_A. Bob constructs the analogous graph G'_B and outputs his required distance. The next paragraph formalizes this for completeness, but may be skipped by a convinced reader.

Formally, Alice sends $\{\mathrm{wdist}_{G_A}(u, v) \mid u \in C_A, v \in V_A\}$. Bob constructs $G'_B = (V'_B, E'_B, w'_B)$ with $V'_B = V_B \cup C_A$ and edges $E'_B = E_B \cup C \cup (C_A \times C_A)$. The edge-weight function w'_B is defined by $w'_B(e) = w(e)$ for each $e \in E_B \cup C$, and $w'_B(u, v)$ for $u, v \in C_A$ is defined to be the weighted distance between u and v in G_A, as received from Alice (or ∞ if they are not connected in G_A). Bob then computes the set of all weighted distances in G'_B, $\{\mathrm{wdist}_{G'_B}(u, v) \mid u, v \in V'_B\}$. Bob assigns his output for the weighted distances in G as follows. For two nodes $u, v \in V_B \cup C_A$, Bob outputs their weighted distance in G'_B, $\mathrm{wdist}_{G'_B}(u, v)$. For a node $u \in V'_B$ and a node $v \in V_A \setminus C_A$, Bob outputs $\min\{\mathrm{wdist}_{G'_B}(u, x) + \mathrm{wdist}_{G_A}(x, v) \mid x \in C_A\}$, where $\mathrm{wdist}_{G'_B}$ is the distance in G'_B as computed by Bob, and wdist_{G_A} is the distance in G_A that was sent by Alice. ◀

The proof of Theorem 10 appears in the full version of the paper [18].

▶ Remark. In the full version of the paper [18] we show that generalizing the Alice-Bob framework to a shared-blackboard multi-party setting is still insufficient for providing a super-linear lower bound for weighted APSP. We suspect that a similar argument can be applied for the framework of non-fixed Alice-Bob partitions (e.g., [65]), but this requires precisely defining these frameworks which is not addressed here.

5 Discussion

This work provides the first super-linear lower bounds for the CONGEST model, raising a plethora of open questions. First, we showed for some specific problems, namely, computing a minimum vertex cover, a maximum independent set and a χ-coloring, that they are nearly as hard as possible for the CONGEST model. However, we know that approximate solutions for some of these problems can be obtained much faster, in a polylogarithmic number of rounds or even less. A family of specific open questions is then to characterize the exact trade-off between approximation factors and round complexities for these problems.

Another specific open question is the complexity of weighted APSP, which has also been asked in previous work [27, 57]. Our proof that the Alice-Bob framework is incapable of providing super-linear lower bounds for this problem may be viewed as providing evidence that weighted APSP can be solved much faster than is currently known. Together with the recent sub-quadratic algorithm of [29], this brings another angle to the question: can weighted APSP be solved in linear time?

Finally, we propose a more general open question which addresses a possible classification of complexities of global problems in the CONGEST model. Some such problems have complexities of $\Theta(D)$, such as constructing a BFS tree. Others have complexities of $\tilde{\Theta}(D+\sqrt{n})$, such as finding an MST. Some problems have near-linear complexities, such as unweighted APSP. And now we know about the family of hardest problems for the CONGEST model, whose complexities are near-quadratic. Do these complexities capture all possibilities, when natural global graph problems are concerned? Or are there such problems with a complexity of, say, $\Theta(n^{1+\delta})$, for some constant $0 < \delta < 1$? A similar question was recently addressed in [20] for the LOCAL model, and we propose investigating the possibility that such a hierarchy exists for the CONGEST model.

Acknowledgements. We thank Amir Abboud, Ohad Ben Baruch, Michael Elkin, Yuval Filmus and Christoph Lenzen for useful discussions, and the anonymous reviewers.

References

1 Amir Abboud, Keren Censor-Hillel, and Seri Khoury. Near-linear lower bounds for distributed distance computations, even in sparse networks. In *30th International Symposium on Distributed Computing, DISC*, pages 29–42, 2016.

2 Amir Abboud and Kevin Lewi. Exact weight subgraphs and the k-sum conjecture. In *40th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 1–12, 2013.

3 Amir Abboud, Kevin Lewi, and Ryan Williams. Losing weight by gaining edges. In *22th Annual European Symposium on Algorithms, ESA*, pages 1–12, 2014.

4 Noga Alon, Raphael Yuster, and Uri Zwick. Finding and counting given length cycles. *Algorithmica*, 17(3):209–223, 1997.

5 Matti Åstrand, Patrik Floréen, Valentin Polishchuk, Joel Rybicki, Jukka Suomela, and Jara Uitto. A local 2-approximation algorithm for the vertex cover problem. In *23rd International Symposium on Distributed Computing, DISC*, pages 191–205, 2009.

6 Matti Åstrand and Jukka Suomela. Fast distributed approximation algorithms for vertex cover and set cover in anonymous networks. In *22nd Annual ACM Symposium on Parallelism in Algorithms and Architectures, SPAA*, pages 294–302, 2010.

7 Reuven Bar-Yehuda, Keren Censor-Hillel, Mohsen Ghaffari, and Gregory Schwartzman. Distributed approximation of maximum independent set and maximum matching. In *ACM Symposium on Principles of Distributed Computing, PODC*, 2017.

8 Reuven Bar-Yehuda, Keren Censor-Hillel, and Gregory Schwartzman. A distributed $(2+\epsilon)$-approximation for vertex cover in o($\log\delta/\epsilon \log\log \delta$) rounds. In *2016 ACM Symposium on Principles of Distributed Computing, PODC*, pages 3–8, 2016.

9 Leonid Barenboim. On the locality of some NP-complete problems. In *39th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 403–415, 2012.

10 Leonid Barenboim. Deterministic $(\Delta + 1)$-coloring in sublinear (in Δ) time in static, dynamic, and faulty networks. *J. ACM*, 63(5):47:1–47:22, 2016.

11 Leonid Barenboim and Michael Elkin. Deterministic distributed vertex coloring in polylogarithmic time. *J. ACM*, 58(5):23:1–23:25, 2011.

12 Leonid Barenboim and Michael Elkin. Combinatorial algorithms for distributed graph coloring. *Distributed Computing*, 27(2):79–93, 2014.

13 Leonid Barenboim, Michael Elkin, and Fabian Kuhn. Distributed (delta+1)-coloring in linear (in delta) time. *SIAM J. Comput.*, 43(1):72–95, 2014.

14 Leonid Barenboim, Michael Elkin, Seth Pettie, and Johannes Schneider. The locality of distributed symmetry breaking. *J. ACM*, 63(3):20:1–20:45, 2016.

15 Marijke H. L. Bodlaender, Magnús M. Halldórsson, Christian Konrad, and Fabian Kuhn. Brief announcement: Local independent set approximation. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 93–95, 2016.

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

17 Keren Censor-Hillel, Telikepalli Kavitha, Ami Paz, and Amir Yehudayoff. Distributed construction of purely additive spanners. *Distributed Computing*, 2017.

18 Keren Censor-Hillel, Seri Khoury, and Ami Paz. Quadratic and near-quadratic lower bounds for the CONGEST model. *CoRR*, abs/1705.05646, 2017. URL: `https://arxiv.org/abs/1705.05646`.

19 Yi-Jun Chang, Tsvi Kopelowitz, and Seth Pettie. An exponential separation between randomized and deterministic complexity in the LOCAL model. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, pages 615–624, 2016.

20 Yi-Jun Chang and Seth Pettie. A time hierarchy theorem for the LOCAL model. *CoRR*, abs/1704.06297, 2017.

21 Kai-Min Chung, Seth Pettie, and Hsin-Hao Su. Distributed algorithms for the lovász local lemma and graph coloring. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 134–143, 2014.

22 Richard Cole and Uzi Vishkin. Deterministic coin tossing with applications to optimal parallel list ranking. *Information and Control*, 70(1):32–53, 1986.

23 Andrzej Czygrinow, Michal Hanckowiak, and Wojciech Wawrzyniak. Fast distributed approximations in planar graphs. In *22nd International Symposium on Distributed Computing, DISC*, pages 78–92, 2008.

24 Søren Dahlgaard, Mathias Bæk Tejs Knudsen, and Morten Stöckel. Finding even cycles faster via capped k-walks. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 112–120, 2017.

25 Danny Dolev, Christoph Lenzen, and Shir Peled. Tri, Tri Again: Finding triangles and small subgraphs in a distributed setting. In *26th International Symposium on Distributed Computing, DISC*, pages 195–209, 2012.

26 Andrew Drucker, Fabian Kuhn, and Rotem Oshman. On the power of the congested clique model. In *33rd ACM Symposium on Principles of Distributed Computing, PODC*, pages 367–376, 2014.

27 Michael Elkin. Distributed approximation: a survey. *SIGACT News*, 35(4):40–57, 2004.

28 Michael Elkin. An unconditional lower bound on the time-approximation trade-off for the distributed minimum spanning tree problem. *SIAM J. Comput.*, 36(2):433–456, 2006.

29 Michael Elkin. Distributed exact shortest paths in sublinear time. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 757–770, 2017.

30 Yuval Emek, Christoph Pfister, Jochen Seidel, and Roger Wattenhofer. Anonymous networks: randomization = 2-hop coloring. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 96–105, 2014.

31 Pierre Fraigniaud, Cyril Gavoille, David Ilcinkas, and Andrzej Pelc. Distributed computing with advice: information sensitivity of graph coloring. *Distributed Computing*, 21(6):395–403, 2009.

32 Pierre Fraigniaud, Marc Heinrich, and Adrian Kosowski. Local conflict coloring. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS*, pages 625–634, 2016.

33 Silvio Frischknecht, Stephan Holzer, and Roger Wattenhofer. Networks cannot compute their diameter in sublinear time. In *ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1150–1162, 2012.

34 Mika Göös and Jukka Suomela. Locally checkable proofs in distributed computing. *Theory of Computing*, 12(1):1–33, 2016.

35 Fabrizio Grandoni, Jochen Könemann, and Alessandro Panconesi. Distributed weighted vertex cover via maximal matchings. *ACM Trans. Algorithms*, 5(1):6:1–6:12, 2008.

36 Fabrizio Grandoni, Jochen Könemann, Alessandro Panconesi, and Mauro Sozio. A primal-dual bicriteria distributed algorithm for capacitated vertex cover. *SIAM J. Comput.*, 38(3):825–840, 2008.

37 Michal Hanckowiak, Michal Karonski, and Alessandro Panconesi. On the distributed complexity of computing maximal matchings. *SIAM J. Discrete Math.*, 15(1):41–57, 2001.

38 David G. Harris, Johannes Schneider, and Hsin-Hao Su. Distributed (Δ+1)-coloring in sublogarithmic rounds. In *48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 465–478, 2016.

39 Monika Henzinger, Sebastian Krinninger, and Danupon Nanongkai. A deterministic almost-tight distributed algorithm for approximating single-source shortest paths. In *48th Annual ACM SIGACT Symposium on Theory of Computing, STOC*, pages 489–498, 2016.

40 Stephan Holzer, David Peleg, Liam Roditty, and Roger Wattenhofer. Distributed 3/2-approximation of the diameter. In *28th International Symposium on Distributed Computing, DISC*, pages 562–564, 2014.

41 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*, pages 6:1–6:16, 2015.

42 Stephan Holzer and Roger Wattenhofer. Optimal distributed all pairs shortest paths and applications. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 355–364, 2012.

43 Qiang-Sheng Hua, Haoqiang Fan, Lixiang Qian, Ming Ai, Yangyang Li, Xuanhua Shi, and Hai Jin. Brief announcement: A tight distributed algorithm for all pairs shortest paths and applications. In *28th ACM Symposium on Parallelism in Algorithms and Architectures, SPAA*, pages 439–441, 2016.

44 Allan Grønlund Jørgensen and Seth Pettie. Threesomes, degenerates, and love triangles. In *55th IEEE Annual Symposium on Foundations of Computer Science, FOCS*, pages 621–630, 2014.

45 Samir Khuller, Uzi Vishkin, and Neal E. Young. A primal-dual parallel approximation technique applied to weighted set and vertex covers. *J. Algorithms*, 17(2):280–289, 1994.

46 Liah Kor, Amos Korman, and David Peleg. Tight bounds for distributed MST verification. In *28th International Symposium on Theoretical Aspects of Computer Science, STACS*, pages 69–80, 2011.

47 Christos Koufogiannakis and Neal E. Young. Distributed and parallel algorithms for weighted vertex cover and other covering problems. In *28th Annual ACM Symposium on Principles of Distributed Computing, PODC*, pages 171–179, 2009.

48 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. The price of being near-sighted. In *17th Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 980–989, 2006.

49 Fabian Kuhn, Thomas Moscibroda, and Roger Wattenhofer. Local computation: Lower and upper bounds. *J. ACM*, 63(2):17:1–17:44, 2016.

50 Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, New York, 1997.

51 Christoph Lenzen and Boaz Patt-Shamir. Fast partial distance estimation and applications. In *2015 ACM Symposium on Principles of Distributed Computing, PODC*, 2015.

52 Christoph Lenzen and David Peleg. Efficient distributed source detection with limited bandwidth. In *ACM Symposium on Principles of Distributed Computing, PODC*, pages 375–382, 2013.

53 Christoph Lenzen and Roger Wattenhofer. Leveraging linial's locality limit. In *22nd International Symposium on Distributed Computing, DISC*, pages 394–407, 2008.

54 Nathan Linial. Locality in distributed graph algorithms. *SIAM J. Comput.*, 21(1):193–201, 1992.

55 Dániel Marx and Michal Pilipczuk. Everything you always wanted to know about the parameterized complexity of subgraph isomorphism (but were afraid to ask). In *31st International Symposium on Theoretical Aspects of Computer Science, STACS*, pages 542–553, 2014.

56 Thomas Moscibroda and Roger Wattenhofer. Coloring unstructured radio networks. *Distributed Computing*, 21(4):271–284, 2008.

57 Danupon Nanongkai. Distributed approximation algorithms for weighted shortest paths. In *Symposium on Theory of Computing, STOC,*, pages 565–573, 2014.

58 Danupon Nanongkai, Atish Das Sarma, and Gopal Pandurangan. A tight unconditional lower bound on distributed randomwalk computation. In *30th Annual ACM Symposium on Principles of Distributed Computing, PODC*, pages 257–266, 2011.

59 Alessandro Panconesi and Romeo Rizzi. Some simple distributed algorithms for sparse networks. *Distributed Computing*, 14(2):97–100, 2001.

60 David Peleg. *Distributed Computing: A Locality-Sensitive Approach*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2000.

61 David Peleg, Liam Roditty, and Elad Tal. Distributed algorithms for network diameter and girth. In *39th International Colloquium on Automata, Languages, and Programming, ICALP*, pages 660–672, 2012.

62 David Peleg and Vitaly Rubinovich. A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. *SIAM J. Comput.*, 30(5):1427–1442, 2000.

63 Seth Pettie and Hsin-Hao Su. Distributed coloring algorithms for triangle-free graphs. *Inf. Comput.*, 243:263–280, 2015.

64 Valentin Polishchuk and Jukka Suomela. A simple local 3-approximation algorithm for vertex cover. *Inf. Process. Lett.*, 109(12):642–645, 2009.

65 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 J. Comput.*, 41(5):1235–1265, 2012.

66 Johannes Schneider and Roger Wattenhofer. Distributed coloring depending on the chromatic number or the neighborhood growth. In *18th International Colloquium on Structural Information and Communication Complexity, SIROCCO*, pages 246–257, 2011.

67 Virginia Vassilevska Williams and Ryan Williams. Finding, minimizing, and counting weighted subgraphs. *SIAM J. Comput.*, 42(3):831–854, 2013.