

Low-Congestion Shortcut and Graph Parameters

Naoki Kitamura

Nagoya Institute of Technology, Japan
31514002@stn.nitech.ac.jp

Hiroataka Kitagawa

Nagoya Institute of Technology, Japan

Yota Otachi

Kumamoto University, Japan
otachi@cs.kumamoto-u.ac.jp

Taisuke Izumi

Nagoya Institute of Technology, Japan
t-izumi@nitech.ac.jp

Abstract

Distributed graph algorithms in the standard CONGEST model often exhibit the time-complexity lower bound of $\tilde{\Omega}(\sqrt{n} + D)$ rounds for many global problems, where n is the number of nodes and D is the diameter of the input graph. Since such a lower bound is derived from special “hard-core” instances, it does not necessarily apply to specific popular graph classes such as planar graphs. The concept of *low-congestion shortcuts* is initiated by Ghaffari and Haeupler [SODA2016] for addressing the design of CONGEST algorithms running fast in restricted network topologies. Specifically, given a specific graph class X , an f -round algorithm of constructing shortcuts of quality q for any instance in X results in $\tilde{O}(q + f)$ -round algorithms of solving several fundamental graph problems such as minimum spanning tree and minimum cut, for X . The main interest on this line is to identify the graph classes allowing the shortcuts which are efficient in the sense of breaking $\tilde{O}(\sqrt{n} + D)$ -round general lower bounds.

In this paper, we consider the relationship between the quality of low-congestion shortcuts and three major graph parameters, chordality, diameter, and clique-width. The main contribution of the paper is threefold: (1) We show an $O(1)$ -round algorithm which constructs a low-congestion shortcut with quality $O(kD)$ for any k -chordal graph, and prove that the quality and running time of this construction is nearly optimal up to polylogarithmic factors. (2) We present two algorithms, each of which constructs a low-congestion shortcut with quality $\tilde{O}(n^{1/4})$ in $\tilde{O}(n^{1/4})$ rounds for graphs of $D = 3$, and that with quality $\tilde{O}(n^{1/3})$ in $\tilde{O}(n^{1/3})$ rounds for graphs of $D = 4$ respectively. These results obviously deduce two MST algorithms running in $\tilde{O}(n^{1/4})$ and $\tilde{O}(n^{1/3})$ rounds for $D = 3$ and 4 respectively, which almost close the long-standing complexity gap of the MST construction in small-diameter graphs originally posed by Lotker et al. [Distributed Computing 2006]. (3) We show that bounding clique-width does not help the construction of good shortcuts by presenting a network topology of clique-width six where the construction of MST is as expensive as the general case.

2012 ACM Subject Classification Theory of computation → Distributed algorithms

Keywords and phrases distributed graph algorithms, low-congestion shortcut, k -chordal graph, clique width, minimum spanning tree

Digital Object Identifier 10.4230/LIPIcs.DISC.2019.25

Related Version <https://arxiv.org/abs/1908.09473>

Funding *Naoki Kitamura*: This work was supported by JSPS KAKENHI Grant Number JP19J22696, Japan.

Yota Otachi: This work was supported by JSPS KAKENHI Grant Numbers JP18H04091, JP18K11168, and JP18K11169, Japan.

Taisuke Izumi: This work was supported by JST SICORP Grant Number JPMJSC1606 and JSPS KAKENHI Grant Number JP19K11824, Japan.



© Naoki Kitamura, Hiroataka Kitagawa, Yota Otachi, and Taisuke Izumi;
licensed under Creative Commons License CC-BY

33rd International Symposium on Distributed Computing (DISC 2019).

Editor: Jukka Suomela; Article No. 25; pp. 25:1–25:17



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

1.1 Background

The *CONGEST* is one of the standard message-passing models in the development of distributed graph algorithms, especially for global problems such as shortest paths and minimum spanning tree. It is a round-based synchronous system where each link can transfer $O(\log n)$ -bit information per one round (n is the number of nodes in the system). Since most of global distributed tasks as mentioned above inherently require each node to access the information far apart from itself, it is not possible to “localize” the communication assessed for solving those tasks. That is, the $\Omega(D)$ -round complexity often becomes an *universal* lower bound applied to any network topology, where D is the diameter of the input topology. While D -round computation is sufficiently long to make some information reach all the nodes in the network, the constraint of limited bandwidth precludes the centralized solution that one node collects the information of whole network topology because it results in expensive $\Omega(n)$ -round time complexity. The round complexity of CONGEST algorithms solving global tasks is typically represented in the form of $\tilde{O}(n^c + D)$ or $\tilde{O}(n^c D)$ for some constant $0 \leq c \leq 2^1$, and thus the main complexity-theoretic question is how much we can make c small (ideally $c = 0$, which matches the universal lower bound). Unfortunately, achieving such an universal bound is an impossible goal for many problems, e.g., minimum spanning tree (MST), shortest paths, minimum cut, and so on. They exhibit the lower bound of $\tilde{\Omega}(\sqrt{n} + D)$ rounds for general graphs.

Most of $\tilde{\Omega}(n^c + D)$ -round lower bounds for some $c > 0$ are derived from special “hard-core” instances, and does not necessarily apply to popular graph classes such as planar graphs, which evokes the interest of developing efficient distributed graph algorithms for specific graph classes. In the last few years, the study along this line rapidly made progress, where the concepts of *partwise aggregation* and *low-congestion shortcuts* play an important role. In the partwise aggregation problem, all the nodes in the network is initially partitioned into a number of disjoint connected subgraphs, which we call a *part*. The goal of this problem is to perform a certain kind of distributed tasks independently within all the parts in parallel. The executable tasks cover several standard operations such as broadcast, convergecast, leader election, finding minimum, and so on. The *low-congestion shortcut* is a framework of solving the partwise aggregation problem, which is initiated by Ghaffari and Haeupler [11]. The key difficulty of the partwise aggregation problem appears when the diameter of a part is much larger than the diameter D of the original graph. Since the diameter can become $\Omega(n)$ in the worst case, the naive solution which performs the aggregation task only by in-part communication can cause the expensive $\Omega(n)$ -round running time. A low-congestion shortcut is defined as the sets of links augmented to each part for accelerating the aggregation task there. Its efficiency is characterized by two quality parameters: The *dilation* is the maximum diameter of all the parts after the augmentation, and the *congestion* is the maximum edge congestion of all edges e , where the edge congestion of e is defined as the number of the parts augmenting e . In the application of low-congestion shortcuts, the performance of an algorithm typically relies on the sum of the dilation and congestion. Hence we simply call the value of dilation plus congestion the *quality* of the shortcut. It is known that any low-congestion shortcut with quality q and $O(f)$ -round construction time yields an $\tilde{O}(f + q)$ -round solution for the partwise aggregation problem, and $\tilde{O}(f + q)$ -round partwise aggregation yields the efficient solutions for several fundamental graph problems. Precisely, the following meta-theorem holds.

¹ $\tilde{O}(\cdot)$ is a notation which ignores $\text{polylog}(n)$ factors from $O(\cdot)$.

► **Theorem 1** (Ghaffari and Haeupler [11], Haeupler and Li [19]). *Let \mathcal{G} be a graph class allowing the low-congestion shortcut with quality $O(q)$ that can be constructed in $O(f)$ rounds in the CONGEST model. Then there exist three algorithms solving (1) the MST problem in $\tilde{O}(f+q)$ rounds, (2) the $(1+\epsilon)$ -approximate minimum cut problem in $\tilde{O}(f+q)$ rounds for any $\epsilon = \Omega(1)$, and (3) $O(n^{O(\log \log n)/\log \beta})$ -approximate weighted single-source shortest path problem in $\tilde{\Omega}((f+q)\beta)$ rounds for any $\beta = \Omega(\text{polylog}(n))^2$.*

Conversely, if we get a time-complexity lower bound for any problem stated above, then it also applies to the partwise aggregation and low-congestion shortcuts (with respect to quality plus construction time). In fact, the $\tilde{O}(\sqrt{n}+D)$ -round lower bound of shortcuts for general graphs is deduced from the lower bound of MST. On the other hand, the existence of efficient (in the sense of breaking the general lower bound) low-congestion shortcuts is known for several major graph classes, as well as its construction algorithms [11, 13, 14, 17, 18, 20].

1.2 Our Result

In this paper, we study the relationship between several major graph parameters and the quality of low-congestion shortcuts. Specifically, we focus on three parameters, that is, (1) chordality, (2) diameter, and (3) clique-width. The precise statement of our result is as follows:

- There is an $O(1)$ -round algorithm which constructs a low-congestion shortcut with quality $O(kD)$ for any k -chordal graph. When $k = O(1)$, its quality matches the $\Omega(D)$ -universal lower bound.
- For $k \leq D$ and $kD \leq \sqrt{n}$, there exists a k -chordal graph where the construction of MST requires $\tilde{\Omega}(kD)$ rounds. It implies that the quality plus construction time of our algorithm is nearly optimal up to polylogarithmic factors.
- There exists an algorithm of constructing a low-congestion shortcut with quality $\tilde{O}(n^{1/4})$ in $\tilde{O}(n^{1/4})$ rounds for any graph of diameter three. In addition, there exists an algorithm of constructing a low-congestion shortcut with quality $\tilde{O}(n^{1/3})$ in $\tilde{O}(n^{1/3})$ rounds for any graph of diameter four. These results almost close the long-standing complexity gap of the MST construction in graphs with small diameters, which is originally posed by Lotker et al. [24].
- We present a negative instance certifying that bounded clique-width does not help the construction of good-quality shortcuts. Precisely, we give an instance of clique-width six where the construction of MST is as expensive as the general case, i.e., $\tilde{\Omega}(\sqrt{n}+D)$ rounds.

Table 1 summarizes the state-of-the-art upper and lower bounds for low-congestion shortcuts. It should be noted that all the parameters considered in this paper is independent of the other parameters such that bounding it admits good shortcuts (e.g., treewidth and genus), and thus any result above is not a corollary of the past results.

For proving our upper bounds, we propose a new scheme of shortcut construction, called *1-hop extension*, where each node in a part only takes all the incident edges as the shortcut edges of its own part. Surprisingly, this very simple construction admits an optimal shortcut for any k -chordal graph. For graphs of diameter three or four, our algorithm is obtained by combining the 1-hop extension scheme with yet another algorithm of finding short low-congestion paths (i.e., paths of length one or two) connecting two moderately-large

² The statement of the weighted single-source shortest path problem is slightly simplified. See [19] for the details.

subgraphs. These algorithms are still simple but it is far from triviality to bound the quality of constructed shortcuts. The analytic part includes several (seemingly) new ideas and may be of independent interest.

■ **Table 1** The quality bounds of Low-Congestion Shortcuts for Specific Graph Classes.

Graph Family	Quality	Construction	Lower bound
General	$\tilde{O}(\sqrt{n} + D)$ [22]	$\tilde{O}(\sqrt{n} + D)$ [22]	$\Omega(\sqrt{n} + D)$ [29]
Planar	$\tilde{O}(D)$ [11]	$\tilde{O}(D)$ [11]	$\tilde{\Omega}(D)$ [11]
Genus- g	$\tilde{O}(\sqrt{g}D)$ [18]	$\tilde{O}(\sqrt{g}D)$ [18]	$\tilde{\Omega}(\sqrt{g}D)$ [18]
Treewidth- k	$\tilde{O}(kD)$ [18]	$\tilde{O}(kD)$ [18]	$\Omega(kD)$ [18]
Clique-width-6	–	–	$\tilde{\Omega}(\sqrt{n} + D)$ (this paper)
Expander	$\tilde{O}\left(\tau 2^{O(\sqrt{\log n})}\right)$ [14] [*]	$\tilde{O}\left(\tau 2^{O(\sqrt{\log n})}\right)$ [14]	–
k -Chordal	$O(kD)$ (this paper)	$O(1)$ (this paper)	$\tilde{\Omega}(kD)$ (this paper)
Excluded Minor	$\tilde{O}(D^2)$ [20]	$\tilde{O}(D^2)$ [20]	–
$D = 3$	$\tilde{O}(n^{1/4})$ (this paper)	$\tilde{O}(n^{1/4})$ (this paper)	$\Omega(n^{1/4})$ [24, 30]
$D = 4$	$\tilde{O}(n^{1/3})$ (this paper)	$\tilde{O}(n^{1/3})$ (this paper)	$\Omega(n^{1/3})$ [24, 30]
$5 \leq D \leq \log n$	–	–	$\tilde{\Omega}\left(n^{(D-2)/(2D-2)}\right)$ [30]

^{*}) τ is the mixing time of the network graph G .

1.3 Related Work

The MST problem is one of the most fundamental problems in distributed graph algorithms. It is not only important by itself, but also has many applications for solving other distributed tasks (e.g., detecting connected components, minimum cut, and so on). Hence many researches have tackled the design of efficient MST algorithms in the CONGEST model so far [7, 8, 12, 15, 16, 21, 22, 27, 28]. The round-complexity lower bound of MST construction is also a central topic in distributed complexity theory [5, 6, 24, 25, 29, 30]. The inherent difficulty of MST construction is of solving the partwise aggregation (minimum) problem efficiently. This viewpoint is first identified by Ghaffari and Haeupler [11] explicitly, as well as an efficient algorithm for solving it in planar graphs. The concept of low-congestion shortcuts is newly invented there for encapsulating the difficulty of partwise aggregation. Recently, several follow-up papers are published to extend the applicability of low-congestion shortcuts, which break the known general lower bounds of several fundamental graph problems in several specific graph classes: This line includes bounded-genus graphs [11, 17], bounded-treewidth graphs [17], graphs with excluded minors [20], expander graphs [13, 14], and so on (see Table 1).

The application of low-congestion shortcuts is not limited only to MST. As stated in Theorem 1, it also admits efficient solutions for approximate minimum cut and single-source shortest path. A few algorithms recently proposed utilize low-congestion shortcuts as an important building block, e.g., the depth first search in planar graphs [19] and approximate treewidth (with decomposition) [23]. Haeupler et al. [16] shows a message-reduction scheme of shortcut-based algorithms, which drop the total number of messages exchanged by the algorithm into $\tilde{O}(m)$, where m is the number of links. On the negative side, it is known that the hardness of (approximate) diameter cannot be encapsulated by low-congestion shortcuts. Abboud et al. [1] shows a hard-core family of unweighted graphs with $O(\log n)$ treewidth where any diameter computation in the CONGEST model requires $\tilde{\Omega}(n)$ rounds. Since any

graph with $O(\log n)$ treewidth admits a low-congestion shortcut of quality $\tilde{O}(D)$, this result implies that it is not possible to compute the diameter of graphs efficiently by using only the property of low-congestion shortcuts.

While our results exhibit a tight upper bound for graphs of diameter three or four, a more generalized lower bound is known for small-diameter graphs. [30]. For any $\log n \geq D \geq 3$, it is proved that there exists a network topology which incurs the $\tilde{\Omega}(n^{(D-2)/(2D-2)})$ -round time complexity for any MST algorithm. In more restricted cases of $D = 1$ and $D = 2$, Jurdzinski et al. [21] and Lotker et al. [24] respectively show $O(1)$ -round and $O(\log n)$ -round MST algorithms.

1.4 Outline of the Paper

The paper is organized as follows: In Section 2, we introduce the formal definitions of the CONGEST model, partwise aggregation, and low-congestion shortcuts, and other miscellaneous terminologies and notations. In Section 3, we show the upper and lower bounds for shortcuts and MST in k -chordal graphs. In Section 4, we present our shortcut algorithms for graphs of diameter three or four. In Section 5, we prove the hardness result for bounded clique-width graphs. The paper is concluded in Section 6.

2 Preliminaries

2.1 CONGEST model

Throughout this paper, we denote by $[a, b]$ the set of integers at least a and at most b . A distributed system is represented by a simple undirected connected graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Let n and m be the numbers of nodes and edges respectively, and D be the diameter of G . Each node has an *ID* from \mathbb{N} (which is represented with $O(\log n)$ bits). In the CONGEST model, the computation follows the round-based synchrony. In one round, each node sends messages to its neighbors, receives messages from its neighbors, and executes local computation. It is guaranteed that every message sent at a round is delivered to the destination within the same round. Each link can transfer $O(\log n)$ -information (bidirectionally) per one round, and each node can inject different messages to its incident links. Each node has no prior knowledge on the network topology except for its neighbor's IDs. Given a graph H for which the node and link sets are not explicitly specified, we denote them by V_H and E_H respectively. Let $N(v)$ be the set of nodes that are adjacent to v , and $N^+(v) = N(v) \cup \{v\}$. We define $N(S) = \cup_{s \in S} N(s)$ and $N^+(S) = \cup_{s \in S} N^+(s)$ for any $S \subseteq V$. For two node subsets $X, Y \subseteq V$, we also define $E(X, Y) = \{(u, v) \in E \mid u \in X, v \in Y\}$. If X (resp. Y) is a singleton $X = \{w\}$, (resp. $Y = \{w\}$), we describe $E(X, Y)$ as $E(w, Y)$ (resp. $E(X, w)$). The *distance* (i.e., the number of edges in the shortest path) between two nodes u and v in G is denoted by $\text{dist}_G(u, v)$. Let S be a path in G . With a small abuse of notations, we often treat S as the sequence of nodes or edges representing the path, as the set of nodes or edges in the path, or the subgraph of G forming the path.

2.2 Partwise Aggregation

The *partwise aggregation* is a communication abstraction defined over a set $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ of mutually-disjoint and connected subgraphs called *parts*, and provides simultaneous fast communication among the nodes in each P_i . It is formally defined as follows:

► **Definition 2** (Partwise Aggregation (PA)). Let $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ be the set of connected mutually-disjoint subgraphs of G , and each node $v \in V_{P_i}$ maintains variable b_v^i storing an input value $x_v^i \in X$. The output of the partwise aggregation problem is to assign $\bigoplus_{w \in P_i} x_w^i$ with b_v^i for any $v \in V_{P_i}$, where \oplus is an arbitrary associative and commutative binary operation over X .

The straightforward solution of the partwise aggregation problem is to perform the convergecast and broadcast in each part P_i independently. Specifically, we construct a BFS tree for each part P_i (after the selection of the root by any leader election algorithm). The time complexity is proportional to the diameter of each part P_i , which can be large ($\Omega(n)$ in the worst case) independently of the diameter of G .

2.3 (d, c) -Shortcut

As we stated in the introduction, the notion of low-congestion shortcuts is introduced for quickly solving the partwise aggregation problem (for some specific graph classes). The formal definition of (d, c) -shortcuts is given as follows.

► **Definition 3** (Ghaffari and Haeupler [11]). Given a graph $G = (V, E)$ and a partition $\mathcal{P} = \{P_1, P_2, \dots, P_N\}$ of G into node-disjoint and connected subgraphs, we define a (d, c) -shortcut of G and \mathcal{P} as a set of subgraphs $\mathcal{H} = \{H_1, H_2, \dots, H_N\}$ of G such that:

1. For each i , the diameter of $P_i + H_i$ is at most d (d -dilation).
2. For each edge $e \in E$, the number of subgraphs $P_i + H_i$ containing e is at most c (c -congestion).

The values of d and c for a (d, c) -shortcut \mathcal{H} is called the *dilation* and *congestion* of \mathcal{H} . As a general statement, a (d, c) -shortcut which is constructed in f rounds admits the solution of the partwise aggregation problem in $\tilde{O}(d + c + f)$ rounds [10, 11]. Since the parameter $d + c$ asymptotically affects the performance of the application, we call the value of $d + c$ the *quality* of (d, c) -shortcuts. A low-congestion shortcut with quality q is simply called a q -shortcut.

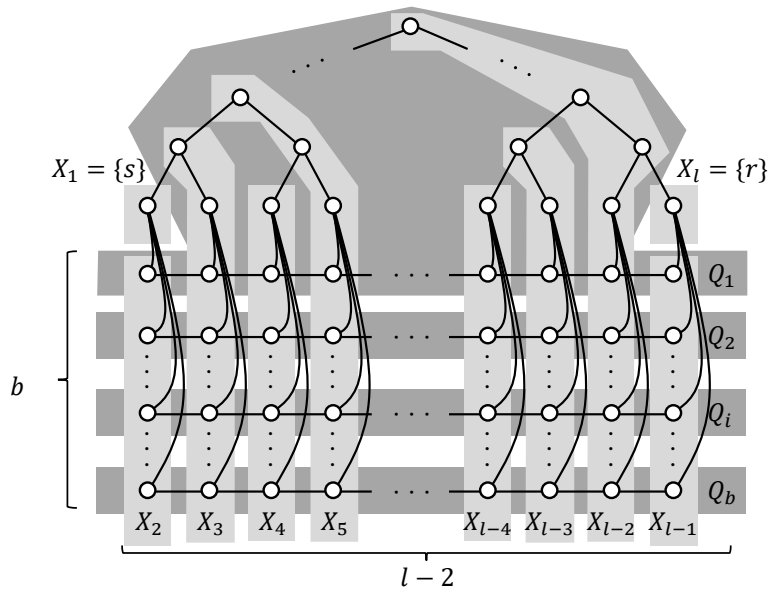
2.4 The framework of the Lower Bound

To prove the lower bound of MST, we introduce a simplified version of the framework by Das Sarma et al. [30]. In this framework, we consider the graph class $\mathcal{G}(n, b, l, c)$ that is defined below. A vertex set $X \subseteq V$ is called *connected* if the subgraph induced by X is connected.

► **Definition 4.** For $n, b, c \geq 0$ and $l \geq 3$, the graph class $\mathcal{G}(n, b, l, c)$ is defined as the set of n -vertex graph $G = (V, E)$ satisfying the following conditions:

- **(C1)** The vertex set V is partitioned into ℓ disjoint vertex sets $\mathcal{X} = \{X_1, X_2, \dots, X_\ell\}$ such that X_1 and X_ℓ are singletons (let $X_1 = \{s\}$ and $X_\ell = \{r\}$).
- **(C2)** The vertex set $V \setminus \{s, r\}$ is partitioned into b disjoint connected sets $\mathcal{Q} = \{Q_1, \dots, Q_b\}$ such that $|E(X_1, Q_i)| \geq 1$ and $|E(X_\ell, Q_i)| \geq 1$ hold for any $1 \leq i \leq b$.
- **(C3)** Let $R_i = \bigcup_{i+1 \leq j \leq \ell} X_j$ and $L_i = \bigcup_{0 \leq j \leq \ell-1-i} X_j$. For $2 \leq i \leq \ell/2 - 1$, $|E(R_i, N(R_i) \setminus R_{i-1})| \leq c$ and $|E(L_i, N(L_i) \setminus L_{i-1})| \leq c$.

Figure 1 shows the graph that is defined vertex partition \mathcal{X} and \mathcal{Q} for the hard-core instances presented in the original proof by Das Sarma et al. [30]. This graph belongs to $\mathcal{G}(O(lb), b, l, O(\log n))$. For class $\mathcal{G}(n, b, l, c)$, the following theorem holds, which is just a corollary of the result by Das Sarma et al. [30].



■ **Figure 1** Example of $\mathcal{G}(O(lb), b, l, O(\log n))$.

► **Theorem 5** (Das Sarma et al. [30]). *For any graph $G \in \mathcal{G}(n, b, l, c)$ and any MST algorithm A , there exists an edge-weight function $w_{A,G} : E \rightarrow \mathbb{N}$ such that the execution of A in G requires $\tilde{\Omega}(\min\{b/c, l/2 - 1\})$ rounds. This bound holds with high probability even if A is a randomized algorithm.*

3 Low-Congestion Shortcut for k -Chordal Graphs

3.1 k -Chordal Graph

A graph G is k -chordal if and only if every cycle of length larger than k has a chord (equivalently, G contains no induced cycle of length larger than k). In particular, 3-chordal graphs are simply called *chordal* graphs, which is known to be much related to various intersection graph families such as interval graphs [9, 26]. Since k -chordal graphs can contain the clique of an arbitrary size for any $k \geq 3$, it is never a subclass of any minor-excluded graphs. Thus no known shortcut algorithm works correctly for k -chordal graphs. The main results of this section are the following two theorems:

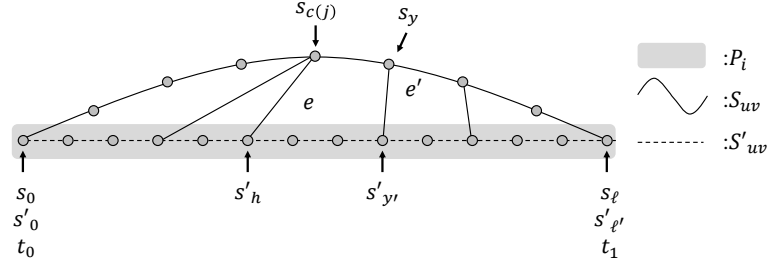
► **Theorem 6.** *There is an $O(1)$ -round algorithm which constructs a $O(kD)$ -shortcut for any k -chordal graph.*

► **Theorem 7.** *For $k \leq D$ and $kD \leq \sqrt{n}$, there exists an unweighted k -chordal graph $G = (V, E)$ where for any MST algorithm A , there exists an edge-weight function $w_A : E \rightarrow \mathbb{N}$ such that the running time of A becomes $\tilde{\Omega}(kD)$ rounds.*

3.2 Proof of Theorem 6

We provide the proof of Theorem 6. The construction algorithm is very simple. It follows the *1-hop extension* scheme stated below:

For any $V_{P_i} \subseteq V$, node $v \in V_{P_i}$ adds each incident edge (v, u) to H_i , and informs u of the fact of $(v, u) \in H_i$.



■ **Figure 2** Proof of Lemma 8.

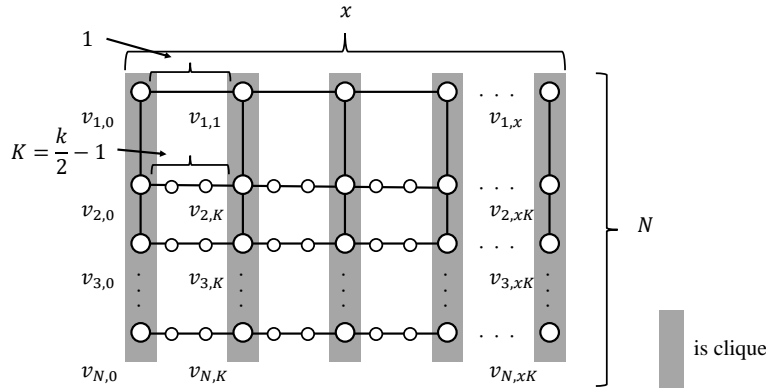
Obviously, this algorithm terminates in one round. Since each node belongs to one part, the congestion of each edge is at most two. Therefore, the technical challenge in proving Theorem 6 is to show that the diameter of $P_i + H_i$ is $O(kD)$ for any $i \in [1, N]$. In other words, the following lemma trivially deduces Theorem 6.

► **Lemma 8.** *Letting $G_i = P_i + H_i$, $\text{dist}_{G_i}(u, v) \leq kD + 2$ holds for any $u, v \in V_{G_i}$.*

Proof. We show that $\text{dist}_{G_i}(u, v) \leq kD$ holds for any $u, v \in V_{P_i}$. Since any node in $v \in V_{G_i} \setminus V_{P_i}$ is a neighbor of a node in V_{P_i} , it obviously follows the lemma.

Let A be the shortest path from u to v in G , and B be that in P_i . We define $T = (t_0, t_1, \dots, t_{z-1})$ as the sequence of nodes in $A \cap B$ which are sorted in the order of A . By definition, $u = t_0$ and $v = t_{z-1}$ holds. The core of the proof is to show that $\text{dist}_{G_i}(t_x, t_{x+1}) \leq k \cdot \text{dist}_G(t_x, t_{x+1})$ for $0 \leq x \leq z - 1$. Summing up this inequality for all x , we obtain $\text{dist}_{G_i}(t_0, t_{z-1}) \leq \sum_{1 \leq j \leq z} k \text{dist}_G(t_{j-1}, t_j) = kD$. By symmetry, we only consider the case of $x = 0$. The case of $x > 0$ is proved similarly. Let $S = (t_0 = s_0, s_1, \dots, s_\ell = t_1)$ be the sub-path of A , and $S' = (t_0 = s'_0, s'_1, \dots, s'_{\ell'} = t_1)$ be the sub-path of B . Given a sequence X , we denote by $X[i, j]$ its consecutive subsequence from the i -th element to the j -th one in X .

We prove that for any $0 \leq j \leq \ell$, there exists a node $s_{c(j)} \in S$ such that $c(j) \geq j$, $\text{dist}_{G_i}(t_0, s_{c(j)}) \leq kj$ and $N^+(s_{c(j)}) \cup S' \neq \emptyset$ hold. The lemma is obtained by setting $j = \ell$ because then $s_{c(j)} = s_\ell = t_1$ holds. The proof follows the induction on j . (Basis) If $j = 0$, then it holds for $s_{c(j)} = s_0$. (Inductive step) Suppose as the induction hypothesis that there exists a node $s_{c(j)}$ satisfying $c(j) \geq j$ and $\text{dist}_{G_i}(t_0, s_{c(j)}) \leq kj$. If $c(j) > j$, obviously $s_{c(j+1)} = s_{c(j)}$ satisfies the case of $j + 1$. Thus, it suffices to consider the case of $c(j) = j$. Let s'_h be the neighbor of $s_{c(j)}$ in S' maximizing h , and $e = (s_{c(j)}, s'_h)$. We consider the cycle C consisting of $S[c(j), \ell]$, $S'[h, \ell']$, and e . If the length of C is at most k , obviously we have $\ell' - h \leq k - 1$. Since $\text{dist}_{G_i}(t_0, s_{c(j)}) \leq kj$ holds by the induction hypothesis, $s_{c(j+1)} = s_\ell$ satisfies the condition. If the length of C is larger than k , C has a chord, which connects two nodes respectively in S and S' because both S and S' are shortest paths. Let $e' = (s_y, s'_{y'})$ be such a chord making the cycle C' consisting of e , e' , $S[s_{c(j)}, s_y]$, and $S'[s_h, s'_{y'}]$ chordless (see Figure 2). Since h is the maximum, we have $y > c(j)$ because if $y = c(j)$ the edge $e' (\neq e)$ is taken as e . Due to the property of k -chordality, the length of C' is at most k , and thus the length of path $S'[h, y'] + \{e, e'\}$ from $s_{c(j)}$ to $s_{c(j)+y}$ is at most $k - 1$, that is, $\text{dist}_{G_i}(s_{c(j)}, s_y) \leq k$. By the induction hypothesis, we obtain $\text{dist}_{G_i}(t_0, s_y) \leq \text{dist}_{G_i}(t_0, s_{c(j)}) + \text{dist}_{G_i}(s_{c(j)}, s_y) \leq k(j + 1)$. Since $s'_{y'}$ is the neighbor of s_y , we have $N^+(s_y) \cup S' \neq \emptyset$. Letting $c(j + 1) = y$, we obtain the proof for $j + 1$. The lemma holds. ◀



■ **Figure 3** Example of k -chordal graph $G(k, x, N)$.

3.3 Proof of Theorem 7

We first introduce the instance mentioned in Theorem 7. Since it has two additional parameters $x \geq 0$ and $N \geq 2$ as well as k , we refer to that instance as $G(k, x, N) = (V(k, x, N), E(k, x, N))$ in the following argument. The parameters x and N are adjusted later for obtaining the claimed lower bound. Let $K = k/2 - 1$ for short. The vertex set and edge set of $G(k, x, N)$ is defined as follows:

- $V(k, x, N) = \{v_{1,j} \mid 0 \leq j \leq x\} \cup \{v_{i,j} \mid 2 \leq i \leq N, 0 \leq j \leq xK\}$.
- $E(k, x, N) = E_1 \cup E_2 \cup E_3 \cup E_4$ such that $E_1 = \{\{v_{1,j}, v_{1,j+1}\} \mid 0 \leq j \leq x - 1\}$, $E_2 = \{\{v_{i,j}, v_{i,j+1}\} \mid 2 \leq i \leq N, 0 \leq j \leq xK - 1\}$, $E_3 = \{\{v_{1,j}, v_{i,h}\} \mid 2 \leq i \leq N, 0 \leq j \leq x, h = jK\}$, and $E_4 = \{\{v_{i,h}, v_{j,h}\} \mid 2 \leq i, j \leq N, i \neq j, h \bmod K = 0\}$.

Figure 3 illustrates the graph $G(k, x, N)$. It is cumbersome to check this graph is k -chordal, but straightforward. One can show the following lemma.

► **Lemma 9.** *For $x \geq 0$ and $N \geq 2$, $G(k, x, N)$ is k -chordal.*

The proof of Theorem 7 follows the framework by Das Sarma et al. [30]. It suffices to show that the following lemma. Theorem 7 is obtained by combining this lemma with Theorem 5.

► **Lemma 10.** *For any $D > 2K$ and $N \geq 2kD$, $G(k, D - K, N) \in \mathcal{G}(n, N, (D - K)K + 3, 1)$ holds.*

4 Low-Congestion Shortcut for Small diameter Graphs

Let $\kappa_D = n^{(D-2)/(2D-2)}$ for short. Note that $\kappa_3 = n^{1/4}$ and $\kappa_4 = n^{1/3}$ hold. The main result in this section is the theorem below.

► **Theorem 11.** *For any graph of diameter $D \in \{3, 4\}$, there exists an algorithm of constructing low-congestion shortcuts with quality $\tilde{O}(\kappa_D)$ in $\tilde{O}(\kappa_D)$ rounds.*

4.1 Centralized Construction

In the following argument, we use term “whp. (with high probability)” to mean that the event considered occurs with probability $1 - n^{-\omega(1)}$ (or equivalently $1 - e^{-\omega(\log n)}$). For simplicity of the proof, we treat any whp. event as if it necessarily occurs (i.e. with probability one). Since the analysis below handles only a polynomially-bounded number of whp. events, the

standard union-bound argument guarantees that everything simultaneously occurs whp. That is, any consequence yielded by the analysis also occurs whp. Since the proof is constructive, we first present the algorithms for $D = 3$ and 4. They are described as a (unified) centralized algorithm, and the distributed implementation is explained later. Let N' be the number of parts whose diameter is more than $12\kappa_D \log^3 n$ (say *large* part). Assume that $P_1, P_2, \dots, P_{N'}$ are large without loss of generality. Since each part P_i ($1 \leq i \leq N'$) contains at least κ_D nodes, $N' \leq n/\kappa_D$ holds obviously. The proposed algorithm constructs the shortcut edges H_i for each large part P_i following the procedure below:

1. Each node $v \in V_{P_i}$ adds its incident edges to H_i (i.e., compute the 1-hop extension).
2. This step adopts two different strategies according to the value of D . ($D = 3$) Each node $u \in N^+(V_{P_i})$ adds each incident edge (u, v) to H_i with probability $1/n^{1/2}$. ($D = 4$) Let $\mathcal{Y} = [1, n^{1/3}/\log n]$. We first prepare an $(n^{1/3} \log^3 n)$ -wise independent hash function $h : [0, N-1] \times V \rightarrow \mathcal{Y}^3$. Each node $u \in V$ adds each incident edge (u, v) to H_i with probability $1/h(u, i)$ if $v \in N^+(V_{P_i})$.

We show that this algorithm provides a low-congestion shortcut of quality $\tilde{O}(\kappa_D)$. First, we look at the bound for congestion. Let H_i^1 be the set of the edges added to H_i in the first step, and H_i^2 be those in the second step. Since the congestion of 1-hop extension is negligibly small, it suffices to consider the congestion incurred by step 2. Intuitively, we can believe the congestion of $\tilde{O}(\kappa_D)$ from the fact that the expected congestion of each edge is $\tilde{O}(\kappa_D)$: Since the total number of large parts is at most n/κ_D , the expected congestion of each edge incurred in step 2 is $n/\kappa_D \cdot (1/n^{1/2}) = O(n^{1/4})$ for $D = 3$, and $(n/\kappa_D) \sum_{y \in \mathcal{Y}} (1/y) \cdot (1/|\mathcal{Y}|) \leq (n/\kappa_D) \cdot (\log n/|\mathcal{Y}|) = \tilde{O}(n^{1/3})$ for $D = 4$.

► **Lemma 12.** *The congestion of the constructed shortcut is $\tilde{O}(\kappa_D)$ whp.*

Proof. It suffices to show that the congestion of any edge $e = (u, v) \in E$ is $\tilde{O}(\kappa_D)$ whp. For simplicity of the proof, we see an undirected edge $e = (u, v)$ as two (directed) edges (u, v) and (v, u) , and distinguish the events of adding (u, v) to shortcuts by u and that by v . That is, the former is recognized as adding (u, v) , and the latter as adding (v, u) . Obviously, the asymptotic bound holding for directed edge (u, v) also holds for the corresponding undirected edge (u, v) actually existing in G (which is at most twice of the directed bound). Since the first step of the algorithm increases the congestion of each directed edge at most by one, it suffices to show that the congestion incurred by the second step is at most $\tilde{O}(\kappa_D)$.

Let X_i be the indicator random variable for the event $(u, v) \in H_i^2$, and $X = \sum_i X_i$. The goal of the proof is to show that $X = \tilde{O}(\kappa_D)$ holds whp. The cases of $D = 3$ and $D = 4$ are proved separately. ($D = 3$) Since at most n/κ_3 large parts exist, we have $\mathbb{E}[X] \leq (n/\kappa_3) \cdot (1/n^{1/2}) = n^{1/4} = \kappa_3$. The straightforward application of Chernoff bound to X allows us to bound the congestion of e by at most $2\kappa_3$ with probability $1 - e^{-\Omega(n^{1/4})}$. ($D = 4$) Let \mathcal{P}' be the subset of all large parts P_j such that $u \in N^+(P_j)$ holds. Consider an arbitrary partition of \mathcal{P}' into several groups with size at least $(n^{1/3} \log^3 n)/2$ and at most $n^{1/3} \log^3 n$. Let q be the number of groups. Each group is identified by a number $\ell \in [1, q]$. We refer to the ℓ -th group as \mathcal{P}^ℓ . Fixing ℓ , we bound the number of parts in \mathcal{P}^ℓ using $e = (u, v)$ as a shortcut edge. Let Y_i be the value of $h(u, i)$. For $P_i \in \mathcal{P}^\ell$, the probability that $X_i = 1$ is $\Pr[X_i = 1] = \sum_{y \in \mathcal{Y}} \Pr[Y_i = y] 1/y = \text{Har}(|\mathcal{Y}|)/|\mathcal{Y}|$, where $\text{Har}(x)$ is the harmonic number of x , i.e., $\sum_{1 \leq i \leq x} i^{-1}$. Letting $X^\ell = \sum_{j \in \mathcal{P}^\ell} X_j$, we have

³ Let X and Y be two finite sets. For any integer $k \geq 1$, a family of hash functions $\mathcal{H} = \{h_1, h_2, \dots, h_p\}$, where each h_i is a function from X to Y , is called *k-wise independent* if for any distinct $x_1, x_2, \dots, x_k \in X$ and any $y_1, y_2, \dots, y_k \in Y$, a function h sampled from \mathcal{H} uniformly at random satisfies $\Pr[\bigwedge_{1 \leq i \leq k} h(x_i) = y_i] = 1/|Y|^k$.

$\mathbb{E}[X^\ell] = (|P^\ell| \text{Har}(|\mathcal{Y}|))/|\mathcal{Y}|$. Since $\text{Har}(x) \leq \log x$, we have $(|P^\ell| \log n)/|\mathcal{Y}| \geq \mathbb{E}[X^\ell] \geq |P^\ell|/|\mathcal{Y}| = (\log^4 n)/2$. Since the hash function h is $(n^{1/3} \log^3 n)$ -wise independent, it is easy to check that $X_1, X_2, \dots, X_{p^\ell}$ are independent. We apply Chernoff bound to X^ℓ , and obtain $\Pr[X^\ell \leq 2\mathbb{E}[X^\ell]] \geq 1 - e^{-\Omega(\mathbb{E}[X^\ell])} = 1 - e^{-\Omega(\log^4 n)}$. It implies that for any ℓ at most $2\mathbb{E}[X^\ell]$ groups use (u, v) as their shortcut edges. The total congestion of (u, v) is obtained by summing up $2\mathbb{E}[X^\ell]$ for all $\ell \in [1, q]$, which results in $\sum_\ell 2|P^\ell| \log n/|\mathcal{Y}| = 2|\mathcal{P}'| \log n/|\mathcal{Y}| = \tilde{O}(n^{1/3})$. The lemma is proved. \blacktriangleleft

For bounding dilation, we first introduce several preliminary notions and terminologies. Given a graph $G = (V, E)$, a subset $S \subset V$ is called an (α, β) -ruling set if it satisfies that (1) for any $u, v \in S$, $\text{dist}_G(u, v) \geq \alpha$ holds, and (2) for any node $v \in V$, there exists $u \in S$ such that $\text{dist}_G(v, u) \leq \beta$ holds. It is known that there exists an $(\alpha, \alpha + 1)$ -ruling set for any graph G [2]. Let $\hat{P}_i = P_i + H_i^1$ for short. For the analysis of P_i 's dilation, we first consider an $(\alpha, \alpha + 1)$ -ruling set of \hat{P}_i for $\alpha = 12\kappa_D \log^3 n$, which is denoted by $S = \{s_0, s_1, \dots, s_z\}$. Note that this ruling set is introduced only for the analysis, and the algorithm does not construct it actually. The key observation of the proof is that for any s_j ($1 \leq j \leq z$) H_i contains a path of length $\tilde{O}(\kappa_D)$ from s_0 to s_j whp. It follows that any two nodes $u, v \in V_{\hat{P}_i}$ are connected by a path of length $\tilde{O}(\kappa_D)$ in $P_i + H_i$ because any node in $V_{\hat{P}_i}$ has at least one ruling-set node within distance $\alpha + 1$ in $P_i + H_i^1$.

To prove the claim above, we further introduce the notion of *terminal sets*. A terminal set $T_j \subseteq V_{P_i}$ associated with $s_j \in S$ ($0 \leq j \leq z$) is the subset of V_{P_i} satisfying (1) $|T_j| \geq \kappa_D \log^3 n$, (2) $\text{dist}_{P_i + H_i}(s_j, x) \leq 6\kappa_D \log^3 n$ for any $x \in T_j$, and (3) $N^+(x) \cap N^+(y) = \emptyset$ for any $x, y \in T_j$ (notice that $N^+(\cdot)$ is the set of neighbors in G , not in $P_i + H_i^1$). We can show that such a set always exists.

► **Lemma 13.** *Letting $S = \{s_0, s_1, \dots, s_z\}$ be any $(\alpha, \alpha + 1)$ -ruling set of \hat{P}_i for $\alpha = 14\kappa_D \log^3 n$, there always exists a terminal set $\mathcal{T} = \{T_0, T_1, \dots, T_z\}$ associated with S .*

Proof. The proof is constructive. Let $c = 6\kappa_D \log^3 n$ for short. We take an arbitrary shortest path $Q = (s_j = u_0, u_1, u_2, \dots, u_c)$ of length c in $P_i + H_i^1$ starting from $s_j \in S$. Since no two nodes in $N^+(V_{P_i}) \setminus V_{P_i}$ are adjacent in $P_i + H_i^1$, Q contains no two consecutive nodes which are both in $N^+(V_{P_i}) \setminus V_{P_i}$. It implies that at least half of the nodes in Q belongs to V_{P_i} . Let $q' = (u'_0, u'_1, \dots, u'_{c'})$ be the subsequence of Q consisting of the nodes in V_{P_i} . Then we define $T_j = \{u'_0, u'_3, \dots, u'_{\lfloor c'/3 \rfloor}\}$, which satisfies the three properties of terminal sets: It is easy to check that the first and second properties hold. In addition, one can show that $\text{dist}_G(u'_x, u'_{x+a}) \geq 3$ (which is equivalent to $N^+(u'_x) \cap N^+(u'_{x+a}) = \emptyset$) holds for any $a \geq 3$ and $x \in [1, c' - a]$: Suppose for contradiction that $\text{dist}_G(u'_x, u'_{x+a}) \leq 2$ holds for some $a \geq 3$ and $x \in [1, c' - a]$. The distance two between u'_x and u'_{x+a} implies $N^+(u'_x) \cap N^+(u'_{x+a}) \neq \emptyset$, and thus $\text{dist}_{\hat{P}_i}(u'_x, u'_{x+a}) \leq 2$ holds. Then bypassing the subpath from u'_x to u'_{x+a} in Q through the distance-two path we obtain a path from s_j to u_c shorter than Q . It contradicts the fact that Q is the shortest path. \blacktriangleleft

The second property of terminal sets and the following lemma deduces the fact that $\text{dist}_{P_i + H_i}(s_0, s_j) = \tilde{O}(\kappa_D)$ holds for any $j \in [0, z]$.

► **Lemma 14.** *Letting $S = \{s_0, s_1, \dots, s_z\}$ be any $(\alpha, \alpha + 1)$ -ruling set of \hat{P}_i for $\alpha = 14\kappa_D \log^3 n$, and $\mathcal{T} = \{T_0, T_1, \dots, T_z\}$ be a terminal set associated with S . For any $j \in [0, z]$, there exist $u \in T_0$ and $v \in T_j$ such that $\text{dist}_{P_i + H_i}(u, v) = O(1)$ holds.*

Proof. Since the distance of s_0 and s_j is at least $14\kappa_D \log^3 n$, we have $N^+(T_0) \cap N^+(T_j) = \emptyset$. The proof is divided into the cases of $D = 3$ and $D = 4$. ($D = 3$) By the conditions of $N^+(T_0) \cap N^+(T_j) = \emptyset$ and $D = 3$, there exists a path of length exactly three from

any node $a \in T_0$ to any node $b \in T_j$. Letting $e_{a,b}$ be the second edge in that path, we define $F = \{e_{a,b} \mid a \in T_0, b \in T_j\}$. By the third property of terminal sets and the fact of $N^+(T_0) \cap N^+(T_j) = \emptyset$, for any two edges $(x_1, y_1), (x_2, y_2) \in F$, either $x_1 \neq x_2$ or $y_1 \neq y_2$ holds. That is, $e_{a_1, b_1} \neq e_{a_2, b_2}$ holds for any $a_1, a_2 \in T_0$ and $b_1, b_2 \in T_j$. By the second property of terminal sets, it implies $|F| = |T_0||T_j| \geq (\kappa_D \log^3 n)^2$. Since each edge in F is added to H_i^2 with probability $1/n^{1/2} = 1/\kappa_D^2$, the probability that no edge in F is added to H_i^2 is at most $(1 - 1/\kappa_D^2)^{(\kappa_D \log^3 n)^2} \leq e^{-\Omega(\log^6 n)}$. That is, an edge $e_{a,b}$ is added to H_i whp. and then $\text{dist}_{P_i+H_i}(a, b) \leq 3$ holds. ($D = 4$) For any node $u \in T_0$ and $v \in T_j$, there exists a path from u to v of length three or four in G . That path necessarily contains a length-two sub-path $P_2(u, v) = (a_{uv}, b_{uv}, c_{uv})$ such that $a_{uv} \in N^+(u)$ and $c_{uv} \in N^+(v)$ holds (if $P_2(u, v)$ is not uniquely determined, an arbitrary one is chosen). We call (a_{uv}, b_{uv}) and (b_{uv}, c_{uv}) the *first* and *second edges* of $P_2(u, v)$ respectively. Let $\mathcal{P}_2 = \{P_2(u, v) \mid u \in T_0, v \in T_j\}$, G' be the union of $P_2(u, v)$ for all $u \in T_0$ and $v \in T_j$, and $\mathcal{P}_2^e = \{P_2(u, v) \in \mathcal{P}_2 \mid e \in P_2(u, v)\}$ for any $e \in E_{G'}$. We first bound the size of \mathcal{P}_2^e . Assume that e is a first edge of some path in \mathcal{P}_2^e . Let $e = (a, b)$ and $u \in T_0$ be the (unique) node such that $a \in N^+(u)$ holds. Since at most $|T_j|$ paths in \mathcal{P}_2 can start from a node in $N^+(u)$, the number of paths in \mathcal{P}_2 using e as their first edges is at most $|T_j|$. Similarly, if e is the second edge of some path in \mathcal{P}_2^e , at most $|T_0|$ paths in \mathcal{P}_2 can contain e as their second edges. While some edge may be used as both first and second edges, the total number of paths using e is bounded by $|T_0| + |T_j| = 2\kappa_D \log^3 n$. It implies that any path $P_2(u, v)$ can share edges with at most $4\kappa_D \log^3 n$ edges, and thus \mathcal{P}_2 contains at least $|T_0||T_j|/(4\kappa_D \log^3 n + 1) \geq \kappa_D \log^3 n/5$ edge-disjoint paths. Let $\mathcal{P}'_2 \subseteq \mathcal{P}_2$ be the maximum-cardinality subset of \mathcal{P}_2 such that any $P_2(u_1, v_1), P_2(u_2, v_2) \in \mathcal{P}'_2$ is edge-disjoint. We define $B = \{b \mid (a, b, c) \in \mathcal{P}'_2\}$. Let $\Delta(b)$ be the number of paths in \mathcal{P}'_2 containing $b \in B$ as the center. Due to the edge disjointness of \mathcal{P}'_2 , we have $|E_G(N^+(T_0), b)| \geq \Delta(b)$ and $|E_G(N^+(T_j), b)| \geq \Delta(b)$ for any $b \in B$. Let Y_b be the value of $h(b, i)$, and X_b be the indicator random variable that takes one if a path in \mathcal{P}'_2 which contains b as the center is added to H_i , and zero otherwise. Let X and Y be the indicator random variables corresponding to the events of $\bigvee_{b \in B} X_b = 1$ and $\bigvee_{b \in B} Y_b \leq \Delta(b)/\log^2 n$ respectively. Then we obtain $\Pr[X_b = 1 \mid Y_b = y] \geq 1 - (1 - 1/y)^{\Delta(b)} \geq 1 - 2e^{-\Delta(b)/y}$, and thus $\Pr[X_b = 1 \mid Y_b \leq \Delta(b)/\log^2 n] \geq 1 - e^{-\Omega(\log^2 n)}$ holds. That is, $\Pr[X = 1 \mid Y = 1] \geq 1 - e^{-\Omega(\log^2 n)}$ holds. Since h is $(n^{1/3} \log^3 n)$ -wise independent, Y_b for all $b \in B$ are independent. Thus we obtain

$$\begin{aligned}
 \Pr[Y = 1] &= 1 - \Pr[Y = 0] \\
 &= 1 - \Pr \left[\bigwedge_{b \in B} Y_b > \frac{\Delta(b)}{\log^2 n} \right] \\
 &= 1 - \prod_{b \in B} \Pr \left[Y_b > \frac{\Delta(b)}{\log^2 n} \right] \\
 &= 1 - \prod_{b \in B} \left(1 - \frac{\Delta(b)}{n^{\frac{1}{3}} \log n} \right) \\
 &\geq 1 - e^{-\sum_{b \in B} \frac{\Delta(b)}{n^{\frac{1}{3}} \log n}} \\
 &= 1 - e^{-\frac{|\mathcal{P}'_2|}{n^{\frac{1}{3}} \log n}} \\
 &\geq 1 - e^{-\Omega(\log^2 n)}.
 \end{aligned}$$

Consequently, we have $\Pr[X = 1] \geq \Pr[X = 1 \wedge Y = 1] \Pr[Y = 1] \geq (1 - e^{-\Omega(\log^2 n)})^2$. The lemma is proved. \blacktriangleleft

4.2 Distributed Implementation

We explain below the implementation details of the algorithm stated above in the CONGEST model.

- **(Preprocessing)** In the algorithm stated above, the shortcut construction is performed only for large parts, which is crucial to bound the congestion of each edge. Thus, as a preprocessing task, each node has to know if its own part is large (i.e. having a diameter larger than κ_D) or not. While the exact identification of the diameter is usually a hard task, just an asymptotic identification is sufficient for achieving the shortcut quality stated above, where the parts of diameter $\omega(\kappa_D)$ and diameter $o(\kappa_D)$ must be identified as large and small ones, but those of diameter $\Theta(\kappa_D)$ is identified arbitrarily. This loose identification is easily implemented by a simple distance-bounded aggregation. The algorithm for part P_i is that: (1) At the first round, each node in P_i sends its ID to all the neighbors, and (2) in the following rounds, each node forwards the minimum ID it received so far. The algorithm executes this message propagation during κ_D rounds. If the diameter is (substantially) larger than κ_D , the minimum ID in P_i does not reach all the nodes in P_i . Then there exists an edge whose endpoints identify different minimum IDs. The one-more-round propagation allows those endpoints to know the part is large. Then they start to broadcast the signal “large” using the following κ_D rounds. If κ_D is large, the signal “large” is invoked at several nodes in P_i , and κ_D -round propagation guarantees that every node receives the signal. That is, any node in P_i identifies that P_i is large. The running time of this task is $O(\kappa_D)$ rounds.
- **(Step 1)** As we stated, the 1-hop extension is implemented in one round. In this step, each node $v \in V_{P_i}$ tells all the neighbors if P_i is large or not. Consequently, if part P_i is identified as a large one, all the nodes in $N^+(P_i)$ know it after this step.
- **(Step 2)** The algorithm for $D = 3$ is trivial. For $D = 4$, there are two non-trivial matters. The first one is the preparation of hash function h . We realize it by sharing a random seed of $O(n^{1/3} \log^3 n \log |\mathcal{Y}|)$ -bit length in advance. A standard construction by Wegman and Carter [31] allows each node to construct the desired h in common. Sharing the random seed is implemented by the broadcast of one $O(n^{1/3} \log^3 n \log |\mathcal{Y}|)$ -bit message, i.e., taking $\tilde{O}(\kappa_D)$ rounds. The second matter is to address the fact that u does not know if P_i is large or not, and/or if v belongs to $N^+(P_i)$ or not. It makes u difficult to determine if (u, v) should be added to H_i or not. Instead, our algorithm simulates the task of u by the nodes in $N(u)$. More precisely, each node $v \in N^+(V_{P_i})$ adds each incident edge (u, v) to H_i with probability $1/h(u, i)$. Due to the fact of $v \in N^+(P_i)$, v knows if P_i is large or not (informed in step 1), and also can compute $h(u, i)$ locally. Thus the choice of (u, v) is locally decidable at v . Since this simulation is completely equivalent to the centralized version, the analysis of the quality also applies.

It is easy to check that the construction time of the distributed implementation above is $\tilde{O}(\kappa_D)$ in total.

5 Low-Congestion Shortcut for Bounded Clique-width Graphs

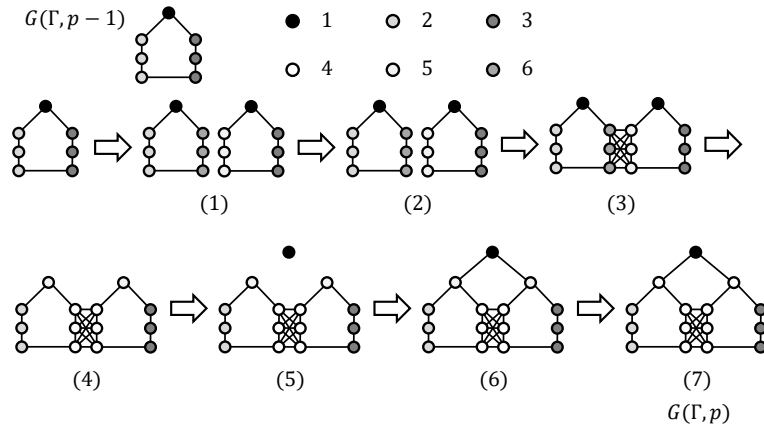
Let $G = (V, E)$ a graph. A k -graph ($k \geq 1$) is a graph whose vertices are labeled by integers in $[1, k]$. A k -graph is naturally defined as a triple (V, E, f) , where f is the labeling function $f : V \rightarrow [1, k]$. The clique-width of $G = (V, E)$ is the minimum k such that there exists a k -graph $G = (V, E, f)$ which is constructed by means of repeated application of the following four operations: (1) introduce: create a graph of a single node v with label $i \in [1, k]$, (2)

disjoint union: take the union $G \cup H$ of two k -graphs G and H , (3) relabel: given $i, j \in [1, k]$, change all the labels i in the graph to j , and (4) join: given $i, j \in [1, k]$, connect all vertices labeled by i with all vertices labeled by j by edges.

The clique-width is invented first as a parameter to capture the tractability for an easy subclass of high treewidth graphs [3, 4]. That is, the class of bounded clique-width can contain many graphs with high treewidth. In centralized settings, one can often obtain polynomial-time algorithms for many NP-complete problems under the assumption of bounded clique-width. The following negative result, however, states that bounding clique-width does not admit any good solution for the MST problem (and thus also for the low-congestion shortcut).

► **Theorem 15.** *There exists an unweighted n -vertex graph $G = (V, E)$ of clique-width six where for any MST algorithm A there exists an edge-weight function $w_A : E \rightarrow \mathbb{N}$ such that the running time of A becomes $\tilde{\Omega}(\sqrt{n} + D)$ rounds.*

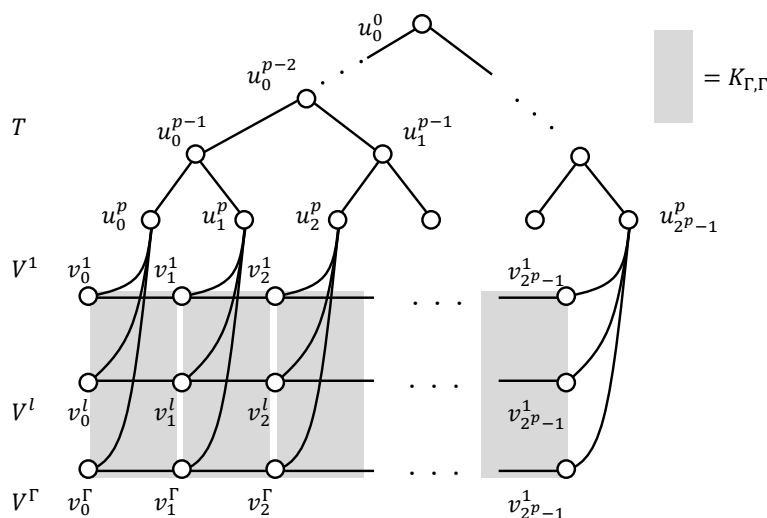
We introduce the instance stated in this theorem, which is denoted by $G(\Gamma, p)$ (Γ and p are the parameters fixed later), using the operations specified in the definition of clique-width. That is, this introduction itself becomes the proof of clique-width six. Let $\mathcal{G}(\Gamma)$ be the set of 6-graphs that contains one node with label 1, Γ nodes with label 2, and Γ nodes label 3, and all other nodes are labeled by 4. Then we define the binary operation \oplus over $\mathcal{G}(\Gamma)$. For any $G, H \in \mathcal{G}(\Gamma)$, the graph $G \oplus H$ is defined as the one obtained by the following operations: (1) Relabel 2 in G with 5 and relabel 3 in H with 6, (2) take the disjoint union $G \cup H$, (3) joins with labels 5 and 6, (4) relabel 5 and 6 with 4, and then 1 with 5, (5) Add a node with label 1 by operation introduce (6) join with 1 and 5, and (7) relabel 5 with 4. This process is illustrated in Figure 4.



■ **Figure 4** Graph $G \oplus H$.

Now we are ready to define $G(\Gamma, p)$. The construction is recursive. First, we define $G(\Gamma, 1)$ as follows: (1) Prepare a (2Γ) -biclique $K_{\Gamma, \Gamma}$ where one side has label 2, and the other side has label 3. Note that two labels suffice to construct $K_{\Gamma, \Gamma}$. (2) Add three nodes with label 1, 5, and 6 by operation introduce. (3) Join with label 2 and 5, and with 3 and 6. (4) Join with label 1 and 5, and with 1 and 6. (5) Relabel 5 and 6 with 4. Then, we define $G(\Gamma, p) = G(\Gamma, p - 1) \oplus G(\Gamma, p - 1)$. The instance claimed in Theorem 15 is $G(\sqrt{n}, \log n/2)$, which is illustrated in Figure 5. This instance is very close to the standard hard-core instance used in the prior work (e.g., [29, 30]. See Figure 1). Thus it is not difficult to see that $\tilde{\Omega}(\sqrt{n})$ -round lower bound for the MST construction also applies to $G(\sqrt{n}, \log n/2)$. It suffices to show that the following lemma. Combined with Theorem 5, we obtain Theorem 15.

► **Lemma 16.** $G(\Gamma, p) \in \mathcal{G}(O(\Gamma(2^p + 2)), \Gamma, 2^p + 2, 3p)$.



■ **Figure 5** Example of clique-width 6 graph $G(\Gamma, p)$.

6 Conclusion

In this paper, we have shown the upper and lower bounds for the round complexity of shortcut construction and MST in k -chordal graphs, diameter-three or four graphs, and bounded clique-width graphs. We presented an $O(1)$ -round algorithm constructing an optimal $O(kD)$ -quality shortcut for any k -chordal graphs. We also presented the algorithms of constructing optimal low-congestion shortcuts with quality $\tilde{O}(\kappa_D)$ in $\tilde{O}(\kappa_D)$ rounds for $D = 3$ and 4 , which yield the optimal algorithms for MST matching the known lower bounds by Lotker et al. [24]. On the negative side, $O(1)$ -clique-width does not allow us to have good shortcuts. We conclude this paper posing three related open problems. (1) Can we have good shortcuts for $D \geq 5$? (2) Can we have good shortcuts for k -clique width where $k \leq 5$? (3) While bounded clique-width does not contribute to solving MST efficiently, it seems to provide many edge-disjoint paths (not necessarily so short). Can we find any problem that can use the benefit of bounded clique-width?

References

- 1 Amir Abboud, Keren Censor-Hillel, and Seri Khoury. Near-Linear Lower Bounds for Distributed Distance Computations, Even in Sparse Networks. In *Proceedings of 30th International Symposium on Distributed Computing (DISC)*, pages 29–42, 2016. doi:10.1007/978-3-662-53426-7_3.
- 2 Baruch Awerbuch, Andrew V. Goldberg, Michael Luby, and Serge A. Plotkin. Network Decomposition and Locality in Distributed Computation. In *Proceedings of 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 364–369, 1989. doi:10.1109/SFCS.1989.63504.
- 3 Derek G. Corneil and Udi Rotics. On the Relationship Between Clique-Width and Treewidth. *SIAM Journal on Computing*, pages 825–847, 2005. doi:10.1137/S0097539701385351.
- 4 Bruno Courcelle and Stephan Olariu. Upper bounds to the clique width of graphs. *Discrete Applied Mathematics*, pages 77–114, 2000. doi:10.1016/S0166-218X(99)00184-5.

- 5 Michael Elkin. Distributed approximation: a survey. *ACM SIGACT News*, pages 40–57, 2004. doi:10.1145/1054916.1054931.
- 6 Michael Elkin. An Unconditional Lower Bound on the Time-Approximation Trade-off for the Distributed Minimum Spanning Tree Problem. *SIAM Journal on Computing*, pages 433–456, 2006. doi:10.1137/S0097539704441058.
- 7 Robert G. Gallager, Pierre A. Humblet, and Philip M. Spira. A Distributed Algorithm for Minimum-Weight Spanning Trees. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, pages 66–77, 1983. doi:10.1145/357195.357200.
- 8 Juan A. Garay, Shay Kutten, and David Peleg. A Sublinear Time Distributed Algorithm for Minimum-Weight Spanning Trees. *SIAM Journal on Computing*, pages 302–316, 1998. doi:10.1137/S0097539794261118.
- 9 Fănică Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, pages 47–56, 1974. doi:10.1016/0095-8956(74)90094-X.
- 10 Mohsen Ghaffari. Near-Optimal Scheduling of Distributed Algorithms. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 3–12, 2015. doi:10.1145/2767386.2767417.
- 11 Mohsen Ghaffari and Bernhard Haeupler. Distributed Algorithms for Planar Networks II: Low-Congestion Shortcuts, MST, and Min-Cut. In *Proceedings of the twenty-seventh annual ACM-SIAM symposium on Discrete algorithms (SODA)*, pages 202–219, 2016. doi:10.1137/1.9781611974331.ch16.
- 12 Mohsen Ghaffari and Fabian Kuhn. Distributed MST and Broadcast with Fewer Messages, and Faster Gossiping. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pages 30:1–30:12, 2018. doi:10.4230/LIPIcs.DISC.2018.30.
- 13 Mohsen Ghaffari, Fabian Kuhn, and Hsin-Hao Su. Distributed MST and Routing in Almost Mixing Time. In *Proceedings of 31st International Symposium on Distributed Computing (DISC)*, pages 131–140, 2017. doi:10.1145/3087801.3087827.
- 14 Mohsen Ghaffari and Jason Li. New Distributed Algorithms in Almost Mixing Time via Transformations from Parallel Algorithms. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pages 31:1–31:16, 2018. doi:10.4230/LIPIcs.DISC.2018.31.
- 15 Robert Gmyr and Gopal Pandurangan. Time-Message Trade-Offs in Distributed Algorithms. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pages 32:1–32:18, 2018. doi:10.4230/LIPIcs.DISC.2018.32.
- 16 Bernhard Haeupler, D. Ellis Hershkowitz, and David Wajc. Round- and Message-Optimal Distributed Graph Algorithms. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 119–128, 2018. doi:10.1145/3212734.3212737.
- 17 Bernhard Haeupler, Taisuke Izumi, and Goran Zuzic. Low-Congestion Shortcuts without Embedding. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 451–460, 2016. doi:10.1145/2933057.2933112.
- 18 Bernhard Haeupler, Taisuke Izumi, and Goran Zuzic. Near-Optimal Low-Congestion Shortcuts on Bounded Parameter Graphs. In *Proceedings of 30th International Symposium on Distributed Computing (DISC)*, pages 158–172, 2016. doi:10.1007/978-3-662-53426-7_12.
- 19 Bernhard Haeupler and Jason Li. Faster Distributed Shortest Path Approximations via Shortcuts. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pages 33:1–33:14, 2018. doi:10.4230/LIPIcs.DISC.2018.33.
- 20 Bernhard Haeupler, Jason Li, and Goran Zuzic. Minor Excluded Network Families Admit Fast Distributed Algorithms. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 465–474, 2018. doi:10.1145/3212734.3212776.
- 21 Tomasz Jurdzinski and Krzysztof Nowicki. MST in $O(1)$ rounds of congested clique. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2620–2632, 2018. doi:10.1137/1.9781611975031.167.

- 22 Shay Kutten and David Peleg. Fast Distributed Construction of Small k -Dominating Sets and Applications. *Journal of Algorithms*, pages 40–66, 1998. doi:10.1006/jagm.1998.0929.
- 23 Jason Li. Distributed Treewidth Computation. *arXiv*, 2018. arXiv:1805.10708.
- 24 Zvi Lotker, Boaz Patt-Shamir, and David Peleg. Distributed MST for constant diameter graphs. *Distributed Computing*, pages 453–460, 2006. doi:10.1007/s00446-005-0127-6.
- 25 Hiroaki Ookawa and Taisuke Izumi. Filling Logarithmic Gaps in Distributed Complexity for Global Problems. In *Proceedings of 41st International Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, pages 377–388, 2015. doi:10.1007/978-3-662-46078-8_31.
- 26 Madhumangal Pal. Intersection Graphs: An Introduction. *arXiv*, 2014. arXiv:1404.5468.
- 27 Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. A time- and message-optimal distributed algorithm for minimum spanning trees. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 743–756, 2017. doi:10.1145/3055399.3055449.
- 28 Gopal Pandurangan, Peter Robinson, and Michele Scquizzato. The Distributed Minimum Spanning Tree Problem. *Bulletin of the European Association for Theoretical Computer Science (EATCS)*, 2018. URL: <http://eatcs.org/beatcs/index.php/beatcs/article/view/538>.
- 29 David Peleg and Vitaly Rubinovich. A Near-Tight Lower Bound on the Time Complexity of Distributed Minimum-Weight Spanning Tree Construction. *SIAM Journal on Computing*, pages 1427–1442, 2000. doi:10.1137/S0097539700369740.
- 30 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. In *Proceedings of the 43th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pages 363–372, 2011. doi:10.1145/1993636.1993686.
- 31 Mark N. Wegman and Larry Carter. New Hash Functions and Their Use in Authentication and Set Equality. *Journal of Computer and System Sciences*, pages 265–279, 1981. doi:10.1016/0022-0000(81)90033-7.