Distributed Computing (2021) 34:349–365
https://doi.org/10.1007/s00446-021-00401-x

Low-congestion shortcut and graph parameters

Naoki Kitamura[1] · Hirotaka Kitagawa[1] · Yota Otachi[2] · Taisuke Izumi[3]

Received: 22 October 2020 / Accepted: 8 August 2021 / Published online: 28 August 2021

Abstract

Distributed graph algorithms in the standard CONGEST model often exhibit the time-complexity lower bound of $\tilde{\Omega}(\sqrt{n}+D)$ rounds for several global problems, where n denotes the number of nodes and D the diameter of the input graph. Because 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* was initiated by Ghaffari and Haeupler [SODA2016] for addressing the design of CONGEST algorithms running fast in restricted network topologies. In particular, given a graph class $\mathcal{C}$, an f-round algorithm for constructing shortcuts of quality q for any instance in $\mathcal{C}$ results in $\tilde{O}(q+f)$-round algorithms for solving several fundamental graph problems such as minimum spanning tree and minimum cut, for $\mathcal{C}$. The main interest on this line is to identify the graph classes allowing the shortcuts that are efficient in the sense of breaking $\tilde{O}(\sqrt{n}+D)$-round general lower bounds. In this study, we consider the relationship between the quality of low-congestion shortcuts and the following four major graph parameters: doubling dimension, chordality, diameter, and clique-width. The key ingredient of the upper-bound side is a novel shortcut construction technique known as *short-hop extension*, which might be of independent interest.

Keywords Distributed graph algorithms · Low-congestion shortcut · k-chordal graph · Clique width · Minimum spanning tree

1 Introduction

1.1 Background

The *CONGEST* is one of the standard message-passing models when considering distributed graph algorithms. It is defined as a round-based synchronous system with limited bandwidth, where each link can transfer $O(\log n)$-bit information per round (n is the number of nodes in the system). Because global distributed tasks such as minimum spanning tree (MST) inherently require each node to access the information far apart from itself, the $\Omega(D)$-round complexity becomes an *universal* lower bound applied to any network topology, where D represents the diameter of the input topology. Although 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 the entire network topology. Thus, the round complexity of CONGEST algorithms for 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]. The main complexity-theoretic question is the extent to which we can make c small (ideally $c = 0$, which matches the universal lower bound). Unfortunately, achieving such a universal bound is impossible for several fundamental problems including MST, which typically exhibits the lower bound of $\tilde{\Omega}(\sqrt{n} + D)$ rounds for general graphs.

Most of the $\tilde{\Omega}(n^c + D)$-round lower bounds for some $c > 0$ are derived from special "hard-core" instances, and do 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 has rapidly made progress, where the concepts of *partwise aggregation* and *low-congestion shortcuts* play an important role. In

✉ Naoki Kitamura
ktmr522@yahoo.co.jp

[1] Nagoya Institute of Technology, Gokiso-cho, Syowa-ku, Nagoya, Aichi, Japan

[2] Nagoya University, Huro-cho, Chikusa-ku, Nagoya, Aichi, Japan

[3] Osaka University, 1-5 Yamadaoka, Osaka, Suita-shi, Japan

[1] $\tilde{O}(\cdot)$ is a notation that ignores polylog(n) factors from $O(\cdot)$.

the partwise aggregation problem, all nodes in the network are initially partitioned into a number of disjoint-connected subgraphs known as a *part*. The goal of this problem is to perform a certain type of distributed task independently within all the parts in parallel. The executable tasks cover several standard operations, such as broadcast, convergecast, leader election, and finding minimum. The *low-congestion shortcut* is a framework for solving the partwise aggregation problem, which is initiated by Ghaffari and Haeupler [15]. 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. Because the diameter of a part can become $\Omega(n)$ in the worst case scenario, the naive solution that performs the aggregation task only by in-part communication causes the expensive $\Omega(n)$-round running time. A low-congestion shortcut is defined as the sets of links augmented to each part to accelerate the aggregation task there. Its efficiency is characterized by the following two quality parameters: The *dilation* is the maximum diameter of all the parts after the augmentation, whereas the *congestion* is the maximum edge congestion of all edges e, where the edge congestion of e is defined as the number of 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 refer to the value of dilation plus congestion as 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:

Theorem 1 *(Ghaffari and Haeupler [15], Haeupler and Li [23]) 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 $\beta = \Omega(\mathrm{polylog}(n))$*[2].

Conversely, if we obtain 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. Meanwhile, 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 [15,17,18,21,22,24].

1.2 Our result

Herein, we study the relationship between several major graph parameters and the quality of low-congestion shortcuts. In particular, we focus on the following four parameters, that is: (1) doubling dimension, (2) chordality, (3) diameter, and (4) clique width. The precise statement of our results is as follows:

- There is an $O(1)$-round algorithm that constructs a low congestion shortcut with quality $\tilde{O}(D^x)$ for any doubling dimension-x graph.
- There is an $O(1)$-round algorithm that 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, implying that the quality plus construction time of our algorithm is nearly optimal up to polylogarithmic factors.
- There exists an algorithm for 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 for 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 are similar to the long-standing complexity gap of the MST construction in graphs with small diameters, which was originally proposed by Lotker et al. [32].
- We present a negative instance certifying that bounded clique-width does not help in the construction of good-quality shortcuts. More precisely, we provide 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. Notably, all the parameters considered in this study are independent of the other (known) parameters admitting good shortcuts (e.g., treewidth and genus) because the graphs of bounded doubling dimension, chordality, or diameter can contain the clique of an arbitrary size and thus, are not a subclass of any minor-excluded graphs. Therefore, any result presented in this paper is not a corollary of past results.

For proving our upper bounds, we propose a novel scheme for shortcut construction, known as *short-hop extension*. The simplest form of this scheme is *1-hop extension*, where each node in a part assumes all the incident edges to be the shortcut of its own part. Surprisingly, this very simple construction admits nearly optimal shortcuts for graphs of bounded

[2] The statement of the weighted single-source shortest path problem is slightly simplified. See [23] for details.

chordality or doubling dimension. For graphs of diameters of three or four, the 2-hop extension (that is, each node in a part takes all the two-length paths starting from itself as the shortcut) clearly yield $O(1)$ dilation, but the second edge in each path suffers from high congestion. Our algorithm circumvents this matter through random choice of the second edges based on hash functions, which is simple though far from triviality to bound the quality of constructed shortcuts. The analytic part includes several new ideas and may be of independent interest.

1.3 Related work

The MST problem is one of the most fundamental problems in distributed graph algorithms. As well as its own importance, MST has several applications for solving other graph problems in distributed settings (e.g., detecting connected components and minimum cut). Several studies have addressed the design of efficient MST algorithms in the CONGEST model [11,12,16,19,20,25,29,35,36], and the round-complexity of MST construction is a central topic in distributed complexity theory [8,9,32,33,37,38]. The inherent difficulty of MST construction is solving the partwise aggregation (minimum) problem efficiently. This viewpoint was first identified by Ghaffari and Haeupler [15] explicitly, as well as an efficient algorithm for solving MST construction in planar graphs. The concept of low-congestion shortcuts is newly invented herein for encapsulating the difficulty of partwise aggregation. Recently, several follow-up papers have been published to extend the applicability of low-congestion shortcuts, which break the known general lower bounds of MST and its applications in specific graph classes: This line covers bounded-genus graphs [15,22], bounded-treewidth graphs [22], graphs with excluded minors [24], and expander graphs [17,18] (see Table 1). All the shortcuts stated here belong to the class of *tree-restricted* shortcuts, where the shortcut edges augmented to each part are a subgraph of a precomputed spanning tree (typically a breadth first search (BFS) tree). It is shown that there exists a universal algorithm for computing tree-restricted shortcuts [21]. To the best of our knowledge, the upper bounds presented in this paper are the first to exhibit non-trivial shortcuts not belonging to the tree-restricted class. The application of low-congestion shortcuts is not limited to MST. As stated in Theorem 1, low-congestion shortcuts also admits efficient solutions for an approximate minimum cut, and a single-source shortest path. In addition, a few algorithms utilize low-congestion shortcuts as an important building block, e.g., the depth first search in planar graphs [23], approximate tree decomposition [30], along with diameter and distance labeling scheme in planar graphs [31]. Haeupler et al. [20] show a message-reduction scheme of shortcut-based algorithms, which drops the total number of messages exchanged by the algorithm into $\tilde{O}(m)$, where m denotes the number of links in the network. On the negative side, it is known that the hardness of (approximate) a diameter cannot be encapsulated by low-congestion shortcuts. Abboud et al. [1] showed 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. Because 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.

Although 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. [38]. For any $\log n \geq D \geq 3$, it is proved that there exists a network topology that incurs the $\tilde{\Omega}\left(n^{(D-2)/(2D-2)}\right)$-round time complexity for any MST algorithm. In more restricted cases of $D = 1$ and $D = 2$, Jurdzinski et al. [25] and Lotker et al. [32] showed $O(1)$- and $O(\log n)$-round MST algorithms, respectively.

1.4 Outline of the paper

The paper is organized as follows. In Sect. 2, we introduce the formal definitions of the CONGEST model, partwise aggregation, and low-congestion shortcuts, and other miscellaneous terminologies and notations. In Sect. 3, we present a shortcut construction for graphs of bounded doubling dimensions. In Sect. 4, we show the upper and lower bounds for shortcuts and MST in k-chordal graphs. Section 5 provides the shortcut algorithms for graphs of diameters of three or four. In Sect. 6, we prove the hardness result for bounded clique-width graphs. The paper is concluded in Sect. 7.

2 Preliminaries

2.1 CONGEST model

Throughout this paper, we denote $[a, b]$ as 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 denotes the set of nodes and E 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 by $O(\log n)$ bits. In the CONGEST model, the computation follows a round-based synchrony. In one round, each node sends messages to its neighbors, receives messages from its neighbors, and executes the local computation. It is guaranteed that every message sent in a round is delivered to the destination within the same round. Each link can transfer $O(\log n)$-bit information (bidirectionally) per round, and each node can inject different messages to its incident links. Each node has no prior knowledge of the network topology.

Table 1 Quality bounds of low-congestion shortcuts for specific graph classes

Graph family	Quality	Construction time	Lower bound
General	$\tilde{O}(\sqrt{n}+D)$ [29]	$\tilde{O}(\sqrt{n}+D)$ [29]	$\Omega(\sqrt{n}+D)$ [37]
Planar	$\tilde{O}(D)$ [15]	$\tilde{O}(D)$ [15]	$\tilde{\Omega}(D)$ [15]
Genus-g	$\tilde{O}(\sqrt{g}D)$ [22]	$\tilde{O}(\sqrt{g}D)$ [22]	$\tilde{\Omega}(\sqrt{g}D)$ [22]
Treewidth-k	$\tilde{O}(kD)$ [22]	$\tilde{O}(kD)$ [22]	$\Omega(kD)$ [22]
Clique-width-6	–	–	$\tilde{\Omega}(\sqrt{n}+D)$ (this paper)
Expander	$\tilde{O}\left(\tau 2^{O(\sqrt{\log n})}\right)$ [18]*	$\tilde{O}\left(\tau 2^{O(\sqrt{\log n})}\right)$ [18]	–
Doubling dimension-x	$\tilde{O}(D^x)$ (this study)	$O(1)$ (this paper)	–
k-Chordal	$O(kD)$ (this paper)	$O(1)$ (this study)	$\tilde{\Omega}(kD)$ (this paper)
Minor	$\tilde{O}(D^2)$ [24]	$\tilde{O}(D^2)$ [24]	–
$D = 3$	$\tilde{O}(n^{1/4})$ (this paper)	$\tilde{O}(n^{1/4})$ (this paper)	$\Omega(n^{1/4})$ [32,38]
$D = 4$	$\tilde{O}(n^{1/3})$ (this paper)	$\tilde{O}(n^{1/3})$ (this paper)	$\Omega(n^{1/3})$ [32,38]
$5 \leq D \leq \log n$	–	–	$\tilde{\Omega}\left(n^{(D-2)/(2D-2)}\right)$ [38]

$*$ τ is the mixing time of network graph G

except for its neighbor 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 in G, and let $N^+(v) = N(v) \cup \{v\}$. 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* (that is, the number of edges in the shortest path) between two nodes u and v in G is denoted by $\mathrm{dist}_G(u, v)$, and we let U be a path in G. With a small abuse of notations, we treat U 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. A path $U = \{s_0, s_1, \ldots, s_\ell\}$ is referred to as *chordless* if and only if for any two nodes $s_i, s_j \in U$ and $|i - j| \geq 2$, it holds that $(s_i, s_j) \notin E_G$

2.2 Partwise aggregation

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

Definition 1 (Partwise Aggregation (PA)) Let $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$ 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 $\oplus_{w \in P_i} x_w^i$ 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 in the CONGEST model is to perform the convergecast and broadcast in each part P_i independently. In particular, 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). Formal definition of (d, c)-shortcuts is provided as follows:

Definition 2 (Ghaffari and Haeupler [15]) Given a graph $G = (V, E)$ and partition $\mathcal{P} = \{P_1, P_2, \ldots, P_N\}$, of G into node-disjoint and connected subgraphs, we define a (d, c)-shortcut of G and $\mathcal{P}$ as a set of subgraphs $\mathcal{H} = \{H_1, H_2, \ldots, H_N\}$ 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 $\mathcal{H}$ are known as the *dilation* and *congestion* of $\mathcal{H}$. As a general statement, a (d, c)-shortcut that is constructed in f rounds admits the solution of the partwise aggregation problem in $\tilde{O}(d + c + f)$ rounds [14,15]. Because the parameter $d + c$ asymptotically affects the performance of the application, we refer to the value of

$d + c$ as the *quality* of (d, c)-shortcuts. A low-congestion shortcut with quality q is simply known as a q*-shortcut*.

2.4 Lower-bound framework

To prove the lower bound of MST, we introduce a simplified version of the framework by Das Sarma et al. [38]. In this framework, we consider the graph class $\mathcal{G}(n, b, \ell, c)$ that is defined below. A vertex set $X \subseteq V$ is known as *connected* if the subgraph induced by X is connected.

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

- (**C1**) The vertex set V is partitioned into ℓ disjoint vertex sets $\mathcal{X} = \{X_1, X_2, \ldots, X_\ell\}$, such that X_1 and X_ℓ, are singletons (let $X_1 = \{s\}$ and $X_\ell = \{r\}$).
- (**C2**) The vertex set $V \backslash \{s, r\}$ is partitioned into b disjoint connected sets $\mathcal{Q} = \{Q_1, \ldots, Q_b\}$ such that $|E(X_1, Q_i)| \geq 1$ and $|E(X_l, Q_i)| \geq 1$ hold for any $1 \leq i \leq b$.
- (**C3**) Let $R_i = \bigcup_{i+1 \leq j \leq \ell} X_j$ and $L_i = \bigcup_{0 \leq j \leq \ell - i} X_j$. 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 vertex partition $\mathcal{X}$ and $\mathcal{Q}$ for the hard-core instances presented in the original proof by Das Sarma et al. [38]. This graph belongs to $\mathcal{G}(O(\ell b), b, \ell, O(\log n))$. For class $\mathcal{G}(n, b, \ell, c)$, the following theorem holds, which is just a corollary of the result by Das Sarma et al. [38]:

Theorem 2 *(Das Sarma et al. [38]) For any graph $G \in \mathcal{G}(n, b, \ell, c)$ 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 $\tilde{\Omega}(\min\{b/c, \ell/2 - 1\})$ rounds. This bound holds with high probability even if A is a randomized algorithm.*

2.5 1-hop extension scheme

Throughout this study we utilize the *1-hop extension* scheme for shortcut construction, which is stated as follows:

> 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 $(v, u) \in H_i$.

It is trivial to implement this scheme using only one round in the CONGEST model. Because each node belongs to one part, the congestion of each edge is at most two. Hence, the technical challenge of this scheme is bound dilation. For the proof, we introduce the concept of (a, b)*-path dominating set*, which characterizes the graphs allowing good shortcuts through 1-hop extension.

Definition 4 Given a path $U \subseteq G$, a (a, b)-path dominating set $S \subseteq V_G$ of U is a node subset satisfying the following two conditions:

- For any $u \in V_U$, there exists $s \in N^+(S)$ such that $\mathrm{dist}_U(u, s) \leq a$ holds.
- $|S| \leq b$.

It is easy to check that if S is a (a, b)-path dominating set of U, $S \cap N^+(U)$ is also a (a, b)-path dominating set of U. Thus, in the following argument, we assume that any (a, b)-path dominating set for U is a subset of $N^+(U)$ without loss of generality. We say that G is (a, b)*-path dominating* if and only if any chordless path $U \subseteq G$ has a (a, b)-path dominating set. By definition, any graph having a dominating set of size b is $(0, b)$-path dominating.

Lemma 1 *The 1-hop extension constructs an $O((a + 1)b)$-shortcut for any (a, b)-path dominating graph.*

Proof Because the congestion bound is trivial, we focus on bounding dilation. Let G be any (a, b)-path dominating graph, P_i be any part of G, and H_i is the shortcut through 1-hop extension for part P_i. Let $U = (s_0, s_1, \ldots, s_\ell)$ be any shortest path in P_i. Because U is the shortest, it is chordless, and thus it has a (a, b)-path dominating set S_U of size $b' \leq b$. Let $Z_U = (V_{Z_U}, E_{Z_U})$ be the subgraph of G such that $V_{Z_U} = U \cup S_U$ and $E_{Z_U} = E(U, U) \cup E(U, S_U)$ holds. Because $S_U \subseteq N^+(U) \subseteq N^+(P_i)$, every edge in $E(U, S_U)$ is a shortcut for H_i. Thus, to prove the lemma, it suffices to show that $\mathrm{dist}_{Z_U}(s_0, s_\ell) = O((a + 1)b')$, for any U. The proof is by the induction on b', that is, we show that every chordless path in P_i having (a, b')-path dominating set S_U of size $b' \leq b$ satisfying $\mathrm{dist}_{Z_U}(s_i, s_j)$ is at most $(2a + 3)b'$ for all $b' \leq b$. (Basis) The case of $b' = 1$: Let w be the unique node in S_U, i be the minimum index such that $s_i \in N^+(w)$ holds, and j be the maximum index such that $s_j \in N^+(w)$ holds. Because $S_U = \{w\}$ is a $(a, 1)$-path dominating set, we obtain $\mathrm{dist}_{Z_U}(s_0, s_\ell) \leq \mathrm{dist}_{Z_U}(s_0, s_i) + \mathrm{dist}_{Z_U}(s_j, s_\ell) + 2 \leq 2a + 2$. (Inductive Step) Suppose as the induction hypothesis that any chordless path $U' = (s'_0, s'_1, \ldots s'_{\ell'})$ in P_i having (a, b'')-path dominating set of size $b'' < b'$ satisfies $\mathrm{dist}_{Z_{U'}}(s'_0, s'_{\ell'}) \leq (2a + 3)b''$. Let i be the minimum index such that $s_i \in N^+(S_U)$ holds, w is any node in $S_U \cap N^+(s_i)$, and j be the maximum index such that $s_j \in N^+(w)$ (see Fig. 2). If $\ell - j \leq a$, we obtain $\mathrm{dist}_{Z_U}(s_0, s_\ell) \leq \mathrm{dist}_{Z_U}(s_0, s_i) + \mathrm{dist}_{Z_U}(s_j, s_\ell) + 2 \leq 2a + 2$. In the case of $\ell - j > a$, any node s_h for $j + a + 1 \leq h \leq \ell$ has no node $s_{h'}$ such that $s_{h'} \in N^+(w)$ and $|h' - h| \leq a$ hold. Hence, the vertex set $S_U \backslash \{w\}$ is a (a, b^*)-path dominating set of (chordless) subpath $U^* = (s_{j+a+1}, \ldots, s_\ell)$. Because $b^* < b'$ holds, by the induction hypothesis, we obtain $\mathrm{dist}_{Z_U}(s_{j+a+1}, s_\ell) \leq \mathrm{dist}_{Z_{U^*}}(s_{j+a+1}, s_\ell) \leq (2a + 3)b^*$. It follows that $\mathrm{dist}_{Z_U}(s_0, s_\ell) \leq \mathrm{dist}_{Z_U}(u, s_i) +$

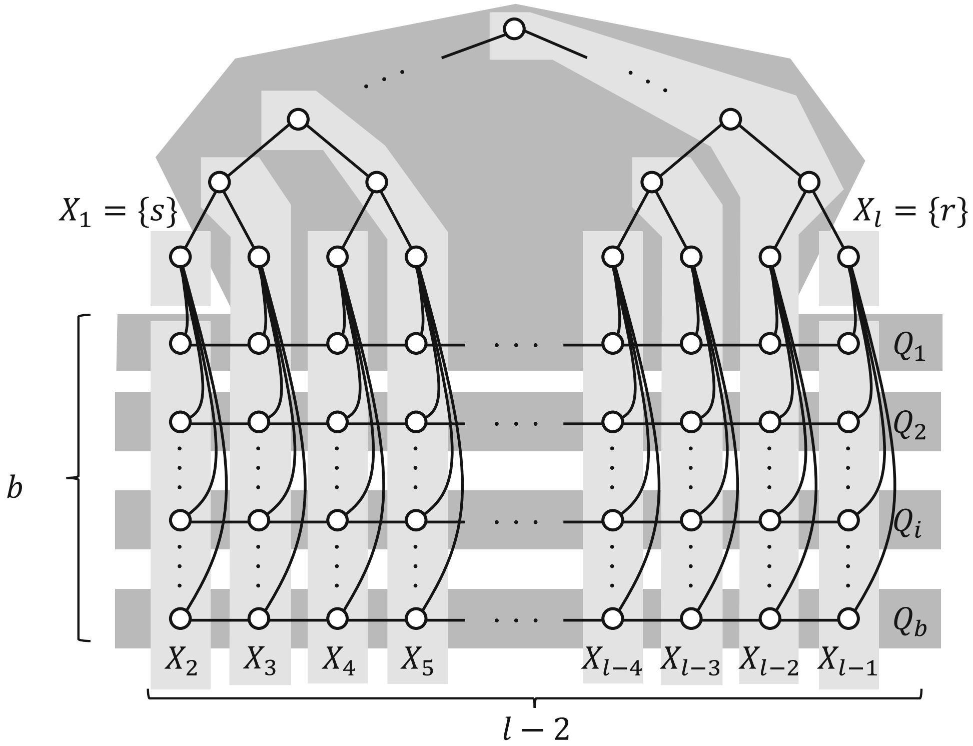

Fig. 1 Example of $\mathcal{G}(O(lb), b, l, O(\log n))$

$\mathrm{dist}_{Z_U}(s_j, s_{j+a+1}) + \mathrm{dist}_{Z_U}(s_{j+a+1}, s_\ell) + 2 \leq a + (a+1) + (2a+3)b^* + 2 \leq (2a+3)b'$. The lemma holds. □

3 Low-congestion shortcut for constant doubling dimension graphs

A pair of a set V and the associated function $\mathrm{dist} : V \times V \rightarrow \mathbb{R}$ is known as a metric space if and only if the following three conditions hold: (1) $\mathrm{dist}(u, v) = 0$ if and only if $v = u$, (2) $\mathrm{dist}(u, v) = \mathrm{dist}(v, u)$ for all $u, v \in V$, and (3) $\mathrm{dist}(u, v) \leq \mathrm{dist}(u, w) + \mathrm{dist}(w, v)$ for all $u, v, w \in V$. The doubling dimension of a metric space V is the smallest positive integer x such that it is possible to cover the ball $B(v, r) = \{u \mid \mathrm{dist}(v, u) < r\}$ of radius r with the union of at most 2^x balls of radius $r/2$ for any $v \in V$ and $r > 0$. A graph $G = (V, E)$ has a doubling dimension x if (V, dist_G) is a metric space of the doubling dimension x. The graphs of bounded doubling dimensions can be assumed to be a generalization of unit disk graphs, and are often considered in the context of distributed computing [7,10,27,28]. The main results of this section are that the graphs of bounded doubling dimensions allow a good shortcut. We have the following theorem:

Theorem 3 *Let x be the doubling dimension of the graph G. Then, there is an $O(1)$-round algorithm that constructs low-congestion shortcuts with quality $\tilde{O}(D^x)$.*

The theorem is obtained by combining the following lemma with Lemma 1. Recall that any graph having a dominating set of size $\tilde{O}(D^x)$ is $(0, \tilde{O}(D^x))$-path dominating.

Lemma 2 *Let G be any graph of the doubling dimension x. There is a dominating set of size $\tilde{O}(D^x)$ in graph G.*

Proof We show that G is covered by at most 2^{ix} balls with radius $(D/2^i)$ for any $1 \leq i \leq \log D$. The lemma is obtained by setting $i = \log D$. The proof follows the induction on i. (Basis) The case of $i = 1$ is obtained from the definition of the doubling dimension. (Inductive step) Suppose as the induction hypothesis that there exists at most 2^{ix} balls with radius $D/2^i$ that cover the graph G. By the definition of the doubling dimension, each ball with radius $D/2^i$ can be covered at most by 2^x balls with radius $D/2^{i+1}$. Therefore, there exist at most $2^{(i+1)x}$ balls with radius $D/2^{i+1}$ that cover the graph G. The lemma is proved. □

4 Low-congestion shortcut for *k*-Chordal graphs

4.1 *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 known as *chordal* graphs, which are known to be related to various intersection graph families such as inter-

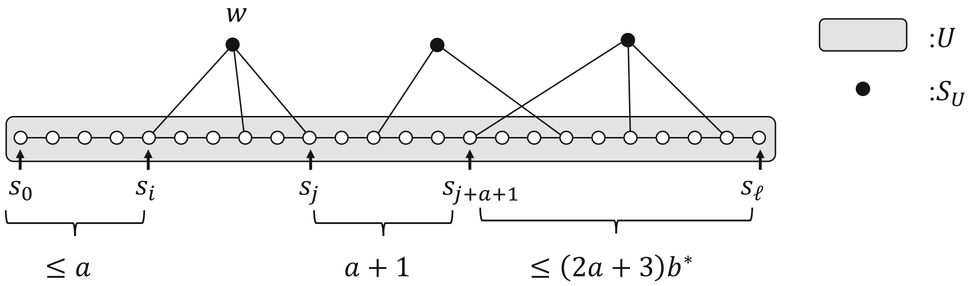

Fig. 2 Proof of Lemma 1

val graphs [13,34]. The main results of this section are the following two theorems:

Theorem 4 *There is an $O(1)$-round algorithm that constructs an $O(kD)$-shortcut for any k-chordal graph.*

Theorem 5 *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.*

4.2 Proof of Theorem 4

The Theorem 4 is deduced from the following lemma and Lemma 1.

Lemma 3 *Any k-chordal graph is $(k, D + 1)$-path dominating.*

Proof Let $U = \{u = s'_0, s'_1, \ldots, s'_{\ell'} = v\}$ be any chordless path, and let $S_U = \{u = s'_0, s'_1, \ldots, s'_{\ell'} = v\}$ be the shortest path from u to v. We show that S_U is a (k, D)-path dominating set for U. Because the diameter of graph G is D, $|S_U| \leq D$ holds. Thus, it suffices to show that for any $v \in V_U$, there exists $v' \in N^+(U)$ such that $\mathrm{dist}_U(v, v') \leq k$ holds. Suppose that there exists a vertex s_i that satisfies $|i - j| > k$ for any $s_j \in N^+(S_U) \cap U$. Let l_i be the maximum index satisfying $s_{l_i} \in N^+(S_U) \cap U$ and $l_i \leq i$, and r_i be the minimum index that satisfies $s_{r_i} \in N^+(S_U) \cap U$ and $i \leq r_i$. Because s_0 and s_ℓ are included in $N^+(S_U)$, s_{l_i} and s_{r_i} always exist.

First, we consider the case in which there exists $w \in N^+(s_{l_i}) \cap N^+(s_{r_i}) \cap S_U$. Let C be the cycle consisting of $\{s_{l_i}, \ldots, s_{r_i}, w\}$. Because the length of C is at least k, C has a chord, but the subpath $\{s_{l_i}, \ldots, s_{r_i}\}$ is chordless; thus, there are no edges $(s_j, s_h) \in E_G$ for any $l_i \leq j < h \leq r_j$ and $h - j \geq 2$. By the definition of l_i and r_i, for any $l_i < j < r_i$, the node s_j is not included in $N^+(w)$; that is, C has no chord, but this is a contradiction.

Next, we consider the case in which there are no $w \in N^+(s_{l_i}) \cap N^+(s_{r_i}) \cap S_U$. We choose y and y' satisfying the following conditions (see Fig. 3):

- $s'_y \in N^+(s_{l_i})$ and $s'_{y'} \in N^+(s_{r_i})$.
- Letting $y_{max} = \max(y, y')$ and $y_{min} = \min(y, y')$, for any $y_{min} < j < y_{max}$, there is no s'_j that satisfies $s'_j \in N^+(s_{l_i}) \cup N^+(s_{r_i})$.

We consider the cycle C consisting of $\{s'_{y'_{min}}, \ldots, s'_{y'_{max}}\}$, $(s'_{y'}, s_{r_i})$, $\{s_{l_i}, \ldots, s_{r_i}\}$, and $(s_{l_i}, s'_{y'})$. Because the length of C is at least k, C has a chord. Notably, any shortest paths are chordless paths. Thus, there are no edges $(s'_j, s'_h) \in E_G$ for any $y_{min} \leq j < h \leq y_{max}$ and $h - j \geq 2$. Because the subpath $\{s_{l_i}, \ldots, s_{r_i}\}$ is chordless, there are no edges $(s_j, s_h) \in E_G$ for any $l_i \leq j < h \leq r_j$ and $h - j \geq 2$. By the definition of l_i, r_i, y, and y', we obtain $E(\{s_{l_i}, \ldots, s_{r_i}\}, \{s'_{y'_{min}}, \ldots, s'_{y'_{max}}\}) = \{(s_{l_i}, s'_{y'}), (s'_{y'}, s_{r_i})\}$. Consequently, we can conclude that C has no chord. It is a contradiction; thus, the lemma is proved. □

4.3 Proof of Theorem 5

We first introduce the instance mentioned in Theorem 5. Because it has two additional parameters $x \geq 0$ and $N \geq 2$ as well as k, we refer to this 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 to obtain the claimed lower bound. Let $K = k/2 - 1$ be short. The vertex set and edge set of $G(k, x, N)$ is defined as follows:

- $V(k, x, N) = \{v_{1,j} \mid 0 \leq j \leq x\} \cup \{v_{i,j} | 2 \leq i \leq N, 0 \leq j \leq xK\}$.
- $E(k, x, N) = E_1 \cup E_2 \cup E_3 \cup E_4$ 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 4 illustrates the graph $G(k, x, N)$.

It is cumbersome to check whether this graph is k-chordal, but straightforward. One can show the following lemma.

Lemma 4 *For $x \geq 0$ and $N \geq 2$, $G(k, x, N)$ is the k-chordal.*

Proof For simplicity, we give some of the vertices a name v'_{xy} as follows:

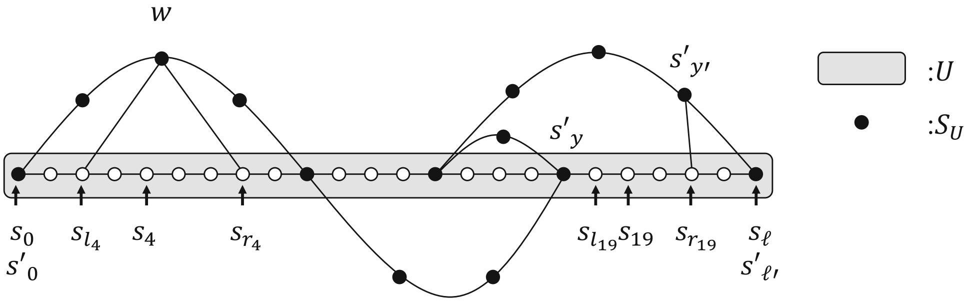

Fig. 3 Proof of Lemma 3

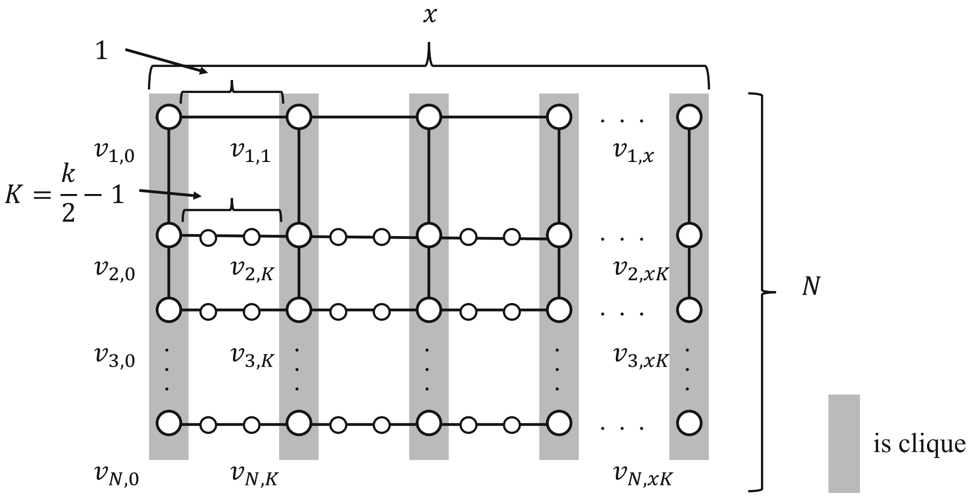

Fig. 4 Example of k-chordal graph $G(k, x, N)$

- $v'_{1,j} = v_{1,j} (0 \leq j \leq x)$
- $v'_{i,j} = v_{i,h} (2 \leq i \leq N, 0 \leq j \leq x, h = jK)$.

We define a subset of vertices known as *row* and *column*. The i-th row R_i is defined as $R_i = \{v'_{i,j} | 0 \leq j \leq x\}$, and the i-th column C_i is defined as $C_i = \{v'_{j,i} | 1 \leq j \leq N\}$. First, we consider the diameter of $G(k, x, N)$. For $2 \leq i \leq N$ and $0 \leq j \leq xK$, we have $\min_{0 \leq k \leq x} \mathrm{dist}(v'_{1,k}, v_{i,j}) = \min_{0 \leq k \leq x} \mathrm{dist}(v'_{i,k}, v_{i,j}) + 1 \leq K/2 + 1$. For $0 \leq i \leq x$ and $0 \leq j \leq x$, $\mathrm{dist}(v'_{1,i}, v_{1,j}) \leq x - 1$, holds; thus, the diameter of $G(k, x, N)$ is at most $K + 1 + x$. We consider a cycle X in $\mathrm{G}(k, x, N)$. Let l and r be the minimum/maximum indices of the rows X intersect, Similarly, let t and b be the minimum/maximum indices of the columns X intersect. Let m be the index such that $|C_m \cap X|$ maximizes, and let $a_m = |C_m \cap X|$ for short. Any cycle X applies to one of the following four cases:

1. $r - l \geq 2$ holds.
2. $a_m \geq 3$ and $r - l \neq 0$ hold.
3. $r - l = 0$ holds.
14. $r - l = 1$ and $a_m = 2$ hold.

We show that Lemma 4 holds for all the cases (Fig. 5 almost states the proof).

1. The case of $r - l \geq 2$: Through the construction of $G(k, x, N)$, l-r path intersects $(l + 1)$-column at least twice. Let u and v be the intersection of the X and $(l+1)$-column. Because C_{l+1} is clique, u and v are adjacent. Thus the edge (u, v) is a chord of X.
2. The case of $a_m \geq 3$ and $r - l \neq 0$: There exists two vertices in C_m, which are not adjacent in X. Because C_m is a clique, there exists an edge between them, and this edge is a chord of X.
3. The case of $r - l = 0$: The cycle X is a clique in graph G and the lemma holds clearly.
4. The case of $r - l = 1$ and $a_m = 2$: The cycle consists of four vertices $v'_{t,l}$,$v'_{t,r}$,$v'_{b,l}$, $v'_{b,r}$ and two paths, that is; the paths connecting $v'_{t,l}$ with $v'_{t,r}$, and $v'_{b,l}$ with $v'_{b,r}$. It follows $\mathrm{dist}(v'_{t,l}, v'_{t,r}) \leq K = k/2 - 1$, $\mathrm{dist}(v'_{b,l}, v'_{b,r}) = K = k/2 - 1$, and $\mathrm{dist}(v'_{t,l}, v'_{b,l}) = \mathrm{dist}(v'_{t,r}, v'_{b,r}) = 1$. Thus, the length of X is at most k.

The lemma is proved. □

The proof of Theorem 5 follows the framework of Das Sarma et al. [38]. It suffices to prove the following lemma. Theorem 5 is obtained by combining this lemma with Theorem 2.

Lemma 5 *For any $D > 2K$ and $N \geq 2kD$, $G(k, D - K, N) \in \mathcal{G}(n, N, (D - K)K + 3, 1)$, holds.*

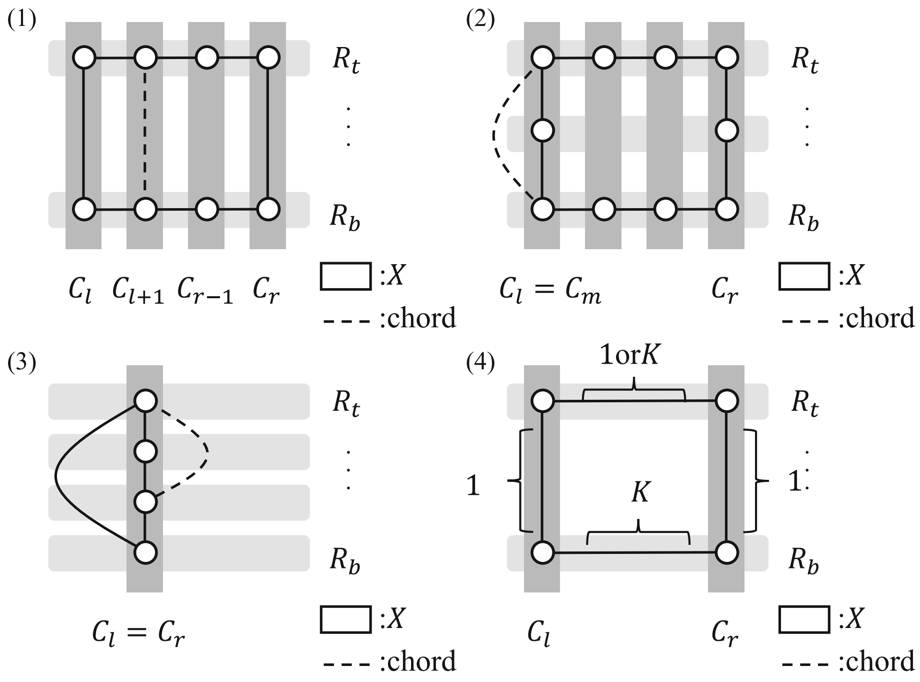

Fig. 5 Proof of Lemma 4

Proof We define $\mathcal{X} = \{X_1, X_2, \ldots, X_{(D-K)K+3}\}$ for $G(k, D-K, N)$ as follows:

- $X_i = \{v_{1,0}\}$ $(i = 1)$.
- $X_i = \{v_{j,0} \mid 2 \le j \le N\}$ $(i = 2)$.
- $X_i = \{v_{j,i-2} \mid 2 \le j \le N\} \cup \{v_{(i-2)/K,1}\}$ $(3 \le i \le (D-K)K, i \bmod K = 2)$.
- $X_i = \{v_{j,i-2} \mid 2 \le j \le N\}$ $(3 \le i \le (D-K)K, i \bmod K \neq 2)$.
- $X_i = \{v_{j,(D-K)K-1} \mid 2 \le j \le N\}$ $(i = (D-K)K+2)$.
- $X_i = \{v_{1,(D-K)}\}$ $(i = (D-K)K+3)$

We define $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_N\}$ for $G(k, D-K, N)$ as follows:

- $Q_i = \{v_{1,j} \mid 1 \le j \le (D-K)-1\}$ $(i = 1)$.
- $Q_i = \{v_{i,j} \mid 0 \le j \le (D-K)K\}$ $(2 \le i \le N)$

It is easy to check that (**C1**) and (**C2**) are satisfied. Thus, we only show that (**C3**) is satisfied. We have $E(R_i, N(R_i)\backslash R_{i-1})$ and $E(L_i, N(L_i)\backslash L_i)$ as follows:

- $E(R_i, N(R_i)\backslash R_{i-1}) = \{v_{1,0}\}$ $(i = 2)$.
- $E(R_i, N(R_i)\backslash R_{i-1}) = \{v_{1,\lfloor(i-1)/K\rfloor}$ $(3 \le i \le ((D-K)K)/2, i \bmod K \neq 2)$.
- $E(R_i, N(R_i)\backslash R_{i-1}) = \emptyset$ $(3 \le i \le ((D-K)K)/2, i \bmod K = 2)$.
- $E(L_i, N(L_i)\backslash L_{i-1}) = \{v_{1,D-K}\}$ $(i = 2)$.
- $E(L_i, N(L_i)\backslash L_{i-1}) = v_{1,D-K-\lfloor(i-2)/K\rfloor}\}$ $(3 \le i \le ((D-K)K)/2, i \bmod K \neq 2)$.
- $E(L_i, N(L_i)\backslash L_{i-1}) = \emptyset$ $(3 \le i \le ((D-K)K)/2, i \bmod K = 2)$.

Thus, we have $|E(R_i, N(R_i)\backslash R_{i-1})| \le 1$, and $|E(L_i, N(L_i)\backslash L_{i-1})| \le 1$. Therefore, we can prove that the graph $G(k, D-K, N)$ is included in $\mathcal{G}(n, N, (D-K)K+3, 1)$. □

5 Low-congestion shortcut for small diameter graphs

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

Theorem 6 *For any graph of diameter $D \in \{3, 4\}$, there exists an algorithm for constructing low-congestion shortcuts with quality $\tilde{O}(\kappa_D)$ in $\tilde{O}(\kappa_D)$ rounds.*

5.1 Centralized construction

In the following argument, we use terminology “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 P=1). Because 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. Because 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 greater than $12\kappa_D \log^3 n$ (say *large* part). Assume that $P_1, P_2, \ldots, P_{N'}$ are large without loss of generality. Because each part P_i ($1 \leq i \leq N'$) contains at least κ_D nodes, $N' \leq n/\kappa_D$ holds clearly. Our technical challenge is to reduce the dilation of the large part. To this end, we separate the large part into the subparts whose diameters are $\tilde{O}(\kappa_D)$, and shows that the shortcut edges establish at least one length-D path between any two subparts. Note that this separation scheme is introduced only for the analysis, and the algorithm does not actually construct it. The detailed explanation of the scheme is explained later. First, each large part computes 1-hop extension. As shown in the previous section, the 1-hop extension only increases the congestion by $O(1)$. Therefore, it suffices to show that at least one shortcut path of length $D-2$ is established between any two extended subparts. For the case of $D=3$, the independent sampling of each edge with probability $1/n^{1/2}$ guarantees the construction of such paths (of length $D-2=1$). For the case of $D=4$, we introduce a new edge sampling scheme based on hash functions of limited independence, which positively correlates two edges incident to a common vertex, and thus amplifies the probability of establishing length-2 shortcut paths without too much increase of congestion. The precise description of the algorithms is stated below. It is applied to each large part P_i for the construction of H_i.

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 $h : [0, N-1] \times V \to \mathcal{Y}$,[3]. At node $u \in V$, each incident edge (u,v) satisfying $v \in N^+(V_{P_i})$ is independently sampled with probability $1/h(u,i)$. All the sampled edges are added to H_i.

We show that this algorithm provides a low-congestion shortcut of quality $\tilde{O}(\kappa_D)$. First, we observe 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 added in the second step. Because the congestion of the 1-hop extension is negligibly small, it suffices to consider the congestion incurred by step 2. Intuitively, we can believe the congestion of $\tilde{O}(\kappa_D)$ as the expected congestion of each edge is $\tilde{O}(\kappa_D)$: Because 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 6 *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 proof, we observe an undirected edge $e=(u,v)$ as two (directed) edges (u,v) and (v,u), and distinguish the events by adding (u,v) to shortcuts by u and that by v; that is, the former is recognized as adding (u,v), whereas the latter is recognized by adding (v,u). Clearly, 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). Because 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 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. ($D=3$) Because at most n/κ_3 large parts exist, we have that $\mathbb{E}[X] \leq (n/\kappa_3)\cdot(1/n^{1/2}) = n^{1/4} = \kappa_3$. The straightforward application of the 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 a size of 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 was identified by a number $\ell \in [1,q]$. We refer to the ℓ-th group as $\mathcal{P}^\ell$. Fixing ℓ, we bound the number of parts in $\mathcal{P}^\ell$ using $e=(u,v)$ as the 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

$$\Pr[X_i = 1] = \sum_{y\in\mathcal{Y}} \Pr[Y_i = y]\frac{1}{y} = \frac{Har\,(|\mathcal{Y}|)}{|\mathcal{Y}|},$$

where $Har(x)$ is the harmonic number of x, i.e., $\sum_{1\leq i\leq x} i^{-1}$. Letting $X^\ell = \sum_{j\in P^\ell} X_j$, we have $\mathbb{E}[X^\ell] = (|P^\ell| Har(|\mathcal{Y}|))/|\mathcal{Y}|$. Because $Har(x) \geq 1$, we have $\mathbb{E}[X^\ell] \geq |P^\ell|/|\mathcal{Y}| = (\log^4 n)/2$. As the hash function h is $(n^{1/3}\log^3 n)$-wise independent, it is easy to check that $X_1, X_2, \ldots, X_{p^\ell}$ 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 $2\mathbb{E}[X^\ell]$

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

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 the following:

$$\begin{aligned}\sum_\ell 2\mathbb{E}[X^\ell] &\leq \sum_\ell \frac{2|P^\ell|\log n}{|\mathcal{Y}|}\\ &= \frac{2|\mathcal{P}'|\log n}{|\mathcal{Y}|}\\ &= \tilde{O}(n^{1/3}).\end{aligned}$$

The lemma is proved. □

For bounding dilation, we first introduce several preliminary notions and terminologies. Given a graph $G = (V, E)$, a subset $S \subset V$ is known as an (α, β)-*ruling set* if it satisfies the following: (1) for any $u, v \in S$, $\mathrm{dist}_G(u, v) \geq \alpha$ holds, and (2) for any node $v \in V$, there exists $u \in S$ such that $\mathrm{dist}_G(v, u) \leq \beta$ 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, it suffices to consider the case where the diameter of $\hat{P}_i$ is greater than $12\kappa_D \log^3 n$. 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 $S = \{s_0, s_1, \ldots, s_z\}$. Note that this ruling set is introduced only for the analysis, and the algorithm does not actually construct it. 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 above-mentioned claim, 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) $\mathrm{dist}_{P_i+H_i}(s_j, x) \leq 6\kappa_D \log^3 n$ for any $x \in T_j$, and (3) $N^+(x) \cap N^+(y) = \emptyset$ for any $x, y \in T_j$ (note 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 7 *Let $S = \{s_0, s_1, \ldots, s_z\}$ be any $(\alpha, \alpha + 1)$-ruling set of $\hat{P}_i$ for $\alpha = 14\kappa_D \log^3 n$, there always exists a family of the terminal sets $\mathcal{T} = \{T_0, T_1, \ldots, T_z\}$ associated with S.*

Proof The proof is constructive. Let $c = 6\kappa_D \log^3 n$. We take an arbitrary shortest path $Q = (s_j = u_0, u_1, u_2, \ldots, u_c)$ of length c in $P_i + H_i^1$, starting from $s_j \in S$. Because 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 $N^+(V_{P_i}) \setminus V_{P_i}$; implying that at least half of the nodes in Q belong to V_{P_i}. Let $q' = (u'_0, u'_1, \ldots u'_{c'})$ be the subsequence of Q consisting of the nodes in V_{P_i}. Then, we define $T_j = \{u'_0, u'_3, \ldots, u'_{3\lfloor c'/3 \rfloor}\}$, 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 $\mathrm{dist}_G(u'_x, u'_{x+a}) \geq 3$ (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 that $\mathrm{dist}_G(u'_x, u'_{x+a}) \leq 2$ holds for some $a \geq 3$ and $x \in [1, c' - a]$, the distance between u'_x and u'_{x+a} implies $N^+(u'_x) \cap N^+(u'_{x+a}) \neq \emptyset$, and thus $\mathrm{dist}_{\hat{P}_i}(u'_x, u'_{x+a}) \leq 2$ holds. Then, bypassing the subpath from u'_x to u'_{x+a} in Q through a distance-two path, we obtain a path from s_j to u_c shorter than Q. This contradicts the fact that Q is the shortest path. □

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

Lemma 8 *Let $S = \{s_0, s_1, \ldots, s_z\}$ be any $(\alpha, \alpha + 1)$-ruling set of $\hat{P}_i$ for $\alpha = 14\kappa_D \log^3 n$, and $\mathcal{T} = \{T_0, T_1, \ldots, T_z\}$ be a family of terminal sets associated with S. For any $j \in [0, z]$, there exist $u \in T_0$ and $v \in T_j$ such that $\mathrm{dist}_{P_i+H_i}(u, v) = O(1)$ holds.*

Proof Because the distances s_0 and s_j are 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 $D = 4$. ($D = 3$) Under 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 the terminal sets and because $N^+(T_0) \cap N^+(T_j) = \emptyset$, 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 $a_1, a_2 \in T_0$, $b_1, b_2 \in T_j$ and $(a_1, b_1) \neq (a_2, b_2)$. The first property of terminal sets implies that $|F| = |T_0||T_j| \geq (\kappa_D \log^3 n)^2$. Because 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 $\mathrm{dist}_{P_i+H_i}(a, b) \leq 3$ 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 $c_{uv} \in N^+(v)$ hold (if $P_2(u, v)$ is not uniquely determined, an arbitrary length-two sub-path is chosen). We refer to (a_{uv}, b_{uv}) and (b_{uv}, c_{uv}), the *first* and *second edges* of $P_2(u, v)$, respectively. (See Fig. 6.) Let $\mathcal{P}_2 = \{P_2(u, v) \mid u \in T_0, v \in T_j\}$, G' is 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 the first edge of a 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. Because 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. Although some edges may be used as both the first and second edges, the total number of paths using e is bounded by $|T_0| + |T_j| = 2\kappa_D \log^3 n$, implying 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. Owing 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. By the definition of $\Delta(b)$, the probability of adding no first edge with b and the node in T_0 as the endpoints to H_i is at most $(1 - 1/h(b, i))^{\Delta(b)}$. Similarly, the probability of adding no second edge with b and the node in T_j as the endpoints to H_i is at most $(1 - 1/h(b, i))^{\Delta(b)}$. Then, we obtain $\Pr[X_b = 1 \mid Y_b = y] \geq 1 - (1 - 1/y)^{\Delta(b)} - (1 - 1/y)^{\Delta(b)} \geq 1 - 2e^{-\Delta(b)/y}$, and thus, $\Pr[X_b = 1 \mid Y_b \leq \max\{1, \Delta(b)/\log^2 n\}] \geq 1 - e^{-\Omega(\log^2 n)}$ holds. Note that, if $Y_b = 1$, then all incident edges of b included in $\mathcal{P}'_2$ must be added to H_i, so $X_b = 1$ always holds. Therefore, $\Pr[X = 1 \mid Y = 1] \geq 1 - e^{-\Omega(\log^2 n)}$ holds. Because h is $(n^{1/3} \log^3 n)$-wise independent, Y_b for all $b \in B$ are independent. Thus, we obtain the following:

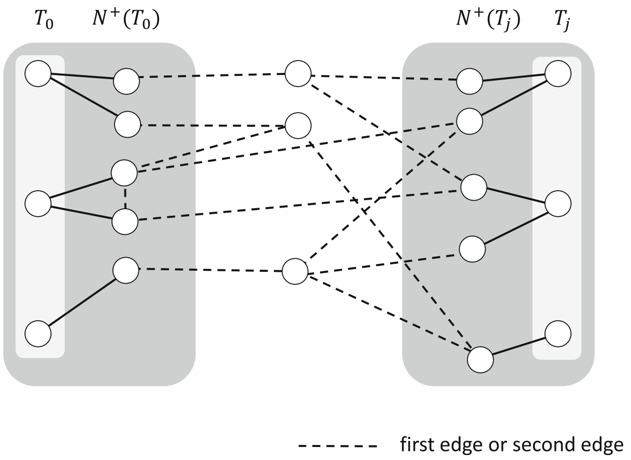

Fig. 6 Proof of Lemma 8

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

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

5.2 Distributed implementation

We explain the above-mentioned implementation details of the algorithm in the CONGEST model as follows:

- (**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. The exact identification of the diameter is usually a hard task;

only 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 $\Theta(\kappa_D)$ are arbitrarily identified. This loose identification is easily implemented through simple distance-bounded aggregation. The algorithm for part P_i is as follows: (1) In the first round, each node in P_i sends its ID to all the neighbors, and (2) in the subsequent rounds, each node forwards the minimum ID received thus 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. Thereafter, 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 matter 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 [39] 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 that u does not know whether P_i is large or not, and/or if v belongs to $N^+(P_i)$. It makes u difficult to determine if (u, v) should be added to H_i. 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 $1/h(u, i)$. Because $v \in N^+(P_i)$, v knows if P_i is large or not (informed in step 1), and v can also compute $h(u, i)$ locally. Thus, the choice of (u, v) is locally decidable at v. Because 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.

6 Low-congestion shortcut for bounded clique-width graphs

Let $G = (V, E)$ be 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 $G = (V, E)$ is the minimum k such that there exists a k-graph $G = (V, E, f)$, which is constructed through 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 [3,6]; that is, the class of bounded clique-width can contain several graphs with high treewidths. In centralized settings, one can often obtain polynomial-time algorithms for several non-deterministic-polynomial complete (NP-complete) problems under the assumption of bounded clique-width. Courcelle et al. [5] showed that for some fixed k, any problem that can be expressed in Monadic second-order logic with quantification over vertices (MSO_1) can be solved in linear time on any class of graphs of clique-width at most k if we obtain k-expressions that define the input graph. Coudert et al. [4] showed several polynomial dependencies in the fixed parameter algorithm (P-FPT) in a bounded clique-width graph. The following negative result, however, states that bounding clique-width does not admit any good solution for the MST problem (and thus, for the low-congestion shortcut).

Theorem 7 *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 with 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 graph obtained through 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) join with labels 5 and 6, (4) relabel 5 and 6 with 4, and then 1 with 5, (5) add a node with label 1 through operation introduce, (6) join with 1 and 5, and (7) relabel 5 with 4. This process is illustrated in Fig. 7.

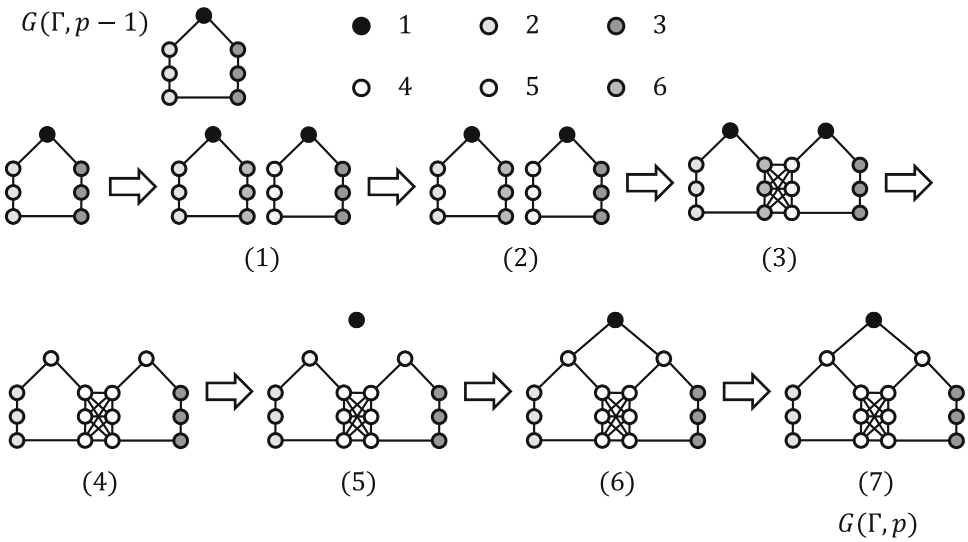

Fig. 7 Graph $G \oplus H$

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. Notably, two labels suffice to construct $K_{\Gamma,\Gamma}$. (2) Add three nodes with labels 1, 5, and 6 through operation introduce. (3) Join with labels 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. Thereafter, we define $G(\Gamma, p) = G(\Gamma, p-1) \oplus G(\Gamma, p-1)$. The instance claimed in Theorem 7 is $G(\sqrt{n}, \log n/2)$, which is illustrated in Fig. 8. This instance is very close to the standard hard-core instance used in prior work (for example, [37,38]. See Fig. 1). Thus it is not difficult to observe that $\tilde{\Omega}(\sqrt{n})$-round lower bound for MST construction also applies to $G(\sqrt{n}, \log n/2)$. It suffices to show the subsequent lemma. Combined with Theorem 2, we obtain Theorem 7.

Lemma 9 $G(\Gamma, p) \in \mathcal{G}(O(\Gamma(2^p+2)), \Gamma, 2^p+2, 3p)$.

Proof First, let us formally specify the graph $G(\Gamma, p)$, which is defined as follows (vertex IDs introduced below are described in Fig. 8):

- $V(\Gamma, p) = T \cup \bigcup_{1 \le l \le \Gamma} V_l$ such that $T = \{u_i^j \mid 0 \le i \le 2^j - 1, 0 \le j \le p\}$, $V_l = \{v_i^l \mid 0 \le i \le 2^p - 1\}$.
- $E(\Gamma, p) = E_1 \cup E_2 \cup E_3$ such that $E_1 = \{(u_i^j, u_{\lfloor \frac{i}{2} \rfloor}^{j-1}) \mid 0 \le i \le 2^j - 1, 1 \le j \le p\}$. $E_2 = \{(u_i^p, v_i^j) \mid 0 \le i \le 2^p - 1, 1 \le j \le \Gamma\}$, $E_3 = \{(v_i^j, v_{i+1}^k) \mid 0 \le i \le 2^p - 2, 1 \le j \le \Gamma, 1 \le k \le \Gamma\}$.

We define $\mathcal{X} = \{X_1, X_2, \ldots, X_{2^p+2}\}$ for graph $G(\Gamma, p)$ as follows:

- $X_i = \{u_0^p\}$ $(i = 1)$.
- $X_i = \{v_0^j \mid 1 \le j \le \Gamma\}$ $(i = 2)$.
- $X_i = \{v_{i-2}^j \mid 2 \le j \le N\} \cup \{u_{(i-1)/2^j-1}^{p-j} \mid 0 \le j \le p, i-1 \bmod 2^j = 0\}$ $(3 \le i \le 2^p - 1)$.
- $X_i = \{v_{2^p-2}^j \mid 1 \le j \le \Gamma\} \cup \{u_{2^p-2}^p\} \cup \{u_{2^j-1}^j \mid 0 \le j \le p-1\}$ $(i = 2^p)$.
- $X_i = \{v_{2^p-1}^j \mid 1 \le j \le \Gamma\}$ $(i = 2^p + 1)$
- $X_i = \{u_{2^p-1}^p\}$ $(i = 2^p + 2)$

We define $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_\Gamma\}$ for graph $G(\Gamma, p)$ as follows:

- $Q_i = V_1 \cup (T \backslash (s \cup r))$ $((i = 1))$.
- $Q_i = V_i$ $((2 \le i \le \Gamma))$.

It is easy to check that (**C1**) and (**C2**) are satisfied. Thus, we only show that (**C3**) is satisfied. Let $V_{R_i} = R_i \cap \bigcup_{j=1}^{\Gamma} V_j$. For $2 \le i \le (2^p+2)/2$, we have $(N(V_{R_i}) \backslash R_{i-1}) = \emptyset$. For any ℓ and $1 \le i \le 2^{p-2}$, if u_i^p is included in R_ℓ, then the neighbors of u_i^p are included in R_ℓ. For any ℓ, $1 \le i \le p$ and $0 \le j \le 2^i - 2$, if u_j^i is included in R_ℓ, then u_{j+1}^i is included in R_ℓ. Let $u^i(R_\ell)$ be the leftmost vertex whose level is i of T and included in R_ℓ. For any ℓ, $1 \le i \le p$ and $0 \le j \le 2^i - 1$, if $u_j^i \neq u^i(R_\ell)$ and u_j^i are included in R_ℓ, then the parent of u_j^i is included in R_ℓ. Thus, $|(N(R_\ell) \backslash R_{\ell-1})|$ only includes neighbors of $u^i(R_\ell)$ for $1 \le i \le p$ and $2 \le \ell \le (2^p+2)/2$. Because the tree T is a binary tree, $u^i(R_\ell)$ has at most 3 neighbors in T. Therefore we have $|E((N(R_i) \backslash R_{i-1}))| \le 3p$. Similarly, we have $|E((N(L_i) \backslash L_{i-1}))| \le 3p$. Therefore, we can prove that the graph $G(\Gamma, p)$ is included in $\mathcal{G}(O(\Gamma(2^p+2)), \Gamma, 2^p+2, 3p)$. By Theorem 2, the lower bound for constructing the MST in $\mathcal{G}(O(\Gamma(2^p+2)), \Gamma, 2^p+2, 3p)$ is $\tilde{\Omega}((\min\{\Gamma/3p, ((2^p+2)/2 - 1\})$. When $\Gamma = \Theta(\sqrt{n})$ and $2^p = \Theta(\sqrt{n})$, we obtain the $\tilde{\Omega}(\sqrt{n})$ lower bound. □

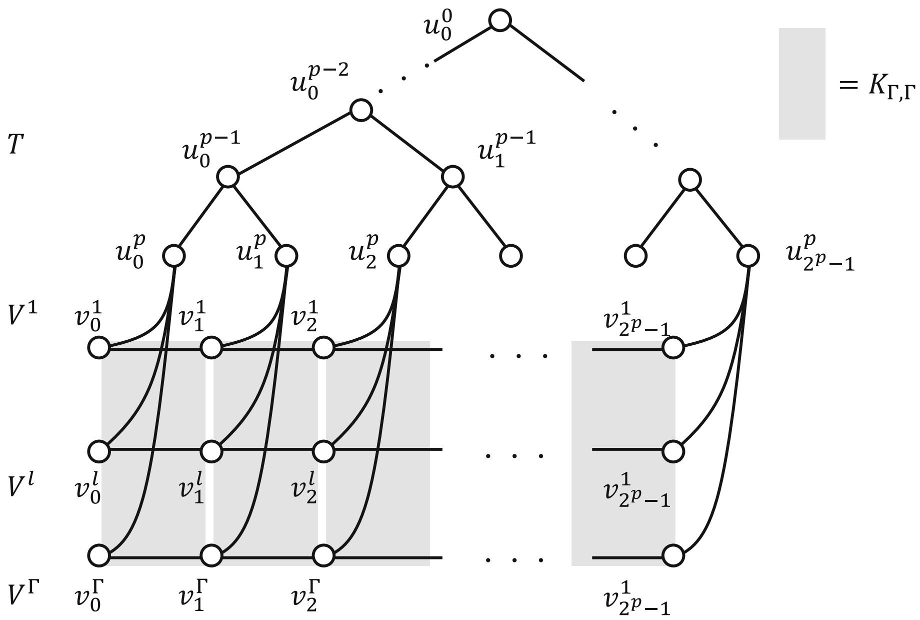

Fig. 8 Example of clique-width 6 graph $G(\Gamma, p)$

7 Conclusion

In this study, 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 for constructing optimal low-congestion shortcuts with quality $\tilde{O}(\kappa_D)$ in $\tilde{O}(\kappa_D)$ rounds for $D = 3$ and 4, which yields the optimal algorithms for MST matching the known lower bounds by Lotker et al. [32]. On the negative side, $O(1)$-clique width does not allow us to have good shortcuts. We conclude this paper by posing three related open problems as follows: (1) Can we have good shortcuts for $D \geq 5$? (2) Can we have good shortcuts for the k-clique width where $k \leq 5$? (3) While the bounded clique width does not contribute to solving MST efficiently, it seems to provide several edge-disjoint paths (not necessarily short). Can we find any problem that can use the benefit of bounded clique width?

Funding This work was supported by JSPS KAKENHI Grant Numbers JP18H04091, JP18K11168, JP18K11169, JP19K11824, and JP19J22696, and JST SICORP Grant Number JPMJSC1606, Japan.

Availability of data and material A preliminary conference version [26] of this paper is published in the 33rd International Symposium on Distributed Computing.

Declarations

Conflict of interest Not applicable

Code availability. Notapplicable.

References

1. Abboud, A., Censor-Hillel, K., Khoury, S.: Near-linear lower bounds for distributed distance computations, even in sparse networks. In *Proceedings of 30nd International Symposium on Distributed Computing (DISC)*, pp. 29–42 (2016). https://doi.org/10.1007/978-3-662-53426-7_3
2. Awerbuch, B., Goldberg, A.V., Luby, M., Plotkin, S.A.: Network decomposition and locality in distributed computation. In *Proceedings of 30th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 364–369 (1989). https://doi.org/10.1109/SFCS.1989.63504
3. Corneil, D.G., Rotics, U.: On the relationship between clique-width and treewidth. SIAM J. Comput. (2005). https://doi.org/10.1137/S0097539701385351
4. Coudert, D., Ducoffe, G., Popa, A.: Fully polynomial fpt algorithms for some classes of bounded clique-width graphs. In *Proceedings of the 2018 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2765–2784 (2018). https://doi.org/10.1137/1.9781611975031.176
5. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. Theory Comput. Syst. (2000). https://doi.org/10.1007/s002249910009

6. Courcelle, B., Olariu, S.: Upper bounds to the clique width of graphs. Discret. Appl. Math. (2000). https://doi.org/10.1016/S0166-218X(99)00184-5
7. Damian, M., Pandit, S., Pemmaraju, S.: Distributed spanner construction in doubling metric spaces. In *Proceedings of the 10th International Conference on Principles of Distributed Systems (OPODIS)*, pp. 157–171 (2006). https://doi.org/10.1007/11945529_12
8. Elkin, M.: Distributed approximation: a survey. ACM SIGACT News (2004). https://doi.org/10.1145/1054916.1054931
9. Elkin, M.: An unconditional lower bound on the time-approximation trade-off for the distributed minimum spanning tree problem. SIAM J. Comput. (2006). https://doi.org/10.1137/S0097539704441058
10. Elkin, M., Filtser, A., Neiman, O.: Distributed construction of light networks. *arXiv* (2019). arXiv:1905.02592
11. Gallager, R.G., Humblet, P.A., Spira, PM.: A distributed algorithm for minimum-weight spanning trees. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, pp. 66–77, 1983. https://doi.org/10.1145/357195.357200
12. Garay, J.A., Kutten, S., Peleg, D.: A sublinear time distributed algorithm for minimum-weight spanning trees. SIAM J. Comput. (1998). https://doi.org/10.1137/S0097539794261118
13. Gavril, F.: The intersection graphs of subtrees in trees are exactly the chordal graphs. J. Comb. Theory Ser. B (1974). https://doi.org/10.1016/0095-8956(74)90094-X
14. Ghaffari, M.: Near-optimal scheduling of distributed algorithms. In *Proceedings of the 2015 ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 3–12 (2015). https://doi.org/10.1145/2767386.2767417
15. Ghaffari, M., Haeupler, B.: 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)*, pp. 202–219 (2016). https://doi.org/10.1137/1.9781611974331.ch16
16. Ghaffari, M., Kuhn, F.: Distributed MST and broadcast with fewer messages, and faster gossiping. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pp. 30:1–30:12 (2018). https://doi.org/10.4230/LIPIcs.DISC.2018.30
17. Ghaffari, M., Kuhn, F., Su, H-H.: Distributed MST and routing in almost mixing time. In *Proceedings of 31nd International Symposium on Distributed Computing (DISC)*, pp. 131–140 (2017). https://doi.org/10.1145/3087801.3087827
18. Ghaffari, M., Li, J.: New distributed algorithms in almost mixing time via transformations from parallel algorithms. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pp. 31:1–31:16 (2018). https://doi.org/10.4230/LIPIcs.DISC.2018.31
19. Gmyr, R., Pandurangan, G.: Time-message trade-offs in distributed algorithms. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pp. 32:1–32:18 (2018). https://doi.org/10.4230/LIPIcs.DISC.2018.32
20. Haeupler, B., Hershkowitz, D.E., Wajc, D.: Round- and message-optimal distributed graph algorithms. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 119–128 (2018). https://doi.org/10.1145/3212734.3212737
21. Haeupler, B., Izumi, T., Zuzic, G.: Low-congestion shortcuts without embedding. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 451–460 (2016). https://doi.org/10.1145/2933057.2933112
22. Haeupler, B., Izumi, T., Zuzic, G.: Near-optimal low-congestion shortcuts on bounded parameter graphs. In *Proceedings of 30nd International Symposium on Distributed Computing (DISC)*, pp. 158–172 (2016). https://doi.org/10.1007/978-3-662-53426-7_12
23. Haeupler, B., Li, J.: Faster distributed shortest path approximations via shortcuts. In *Proceedings of 32nd International Symposium on Distributed Computing (DISC)*, pp. 33:1–33:14 (2018). https://doi.org/10.4230/LIPIcs.DISC.2018.33
24. Haeupler, B., Li, J., Zuzic, G.: Minor excluded network families admit fast distributed algorithms. In *Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing (PODC)*, pp. 465–474 (2018). https://doi.org/10.1145/3212734.3212776
25. Jurdzinski, T., Nowicki, K.: MST in $O(1)$ rounds of congested clique. In *Proceedings of the Twenty-Ninth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pp. 2620–2632 (2018). https://doi.org/10.1137/1.9781611975031.167
26. Kitamura, N., Kitagawa, H., Otachi, Y., Izumi, T.: Low-congestion shortcut and graph parameters. In *Proccedings of 33rd International Symposium on Distributed Computing (DISC)*, pp. 25:1–25:17, 2019. https://doi.org/10.4230/LIPIcs.DISC.2019.25
27. Kuhn, F., Moscibroda, T., Nieberg, T., Wattenhofer, R.: Fast deterministic distributed maximal independent set computation on growth-bounded graphs. In *Proccedings of 19th International Symposium on Distributed Computing (DISC)*, pp. 273–287 (2005). https://doi.org/10.1007/11561927_21
28. Kuhn, F., Moscibroda, T., Wattenhofer, R.: On the locality of bounded growth. In *Proceedings of the 24th annual ACM symposium on Principles of distributed computing (PODC)*, pp. 60–68 (2005). https://doi.org/10.1145/1073814.1073826
29. Kutten, S., Peleg, D.: Fast distributed construction of small k-dominating sets and applications. J. Algorith. (1998). https://doi.org/10.1006/jagm.1998.0929
30. Li, J.: Distributed treewidth computation. *arXiv*, 2018. arXiv:1805.10708
31. Li, J., Parter, M.: Planar diameter via metric compression. In *Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 152–163 (2019). https://doi.org/10.1145/3313276.3316358
32. Lotker, Z., Patt-Shamir, B., Peleg, D.: Distributed MST for constant diameter graphs. Distrib. Comput. (2006). https://doi.org/10.1007/s00446-005-0127-6
33. Ookawa, H., Izumi, T.: Filling logarithmic gaps in distributed complexity for global problems. In *Proccedings of 41st International Conference on Current Trends in Theory and Practice of Informatics (SOFSEM)*, pp. 377–388 (2015). https://doi.org/10.1007/978-3-662-46078-8_31
34. Pal, Madhumangal.: Intersection graphs: An introduction. *arXiv*, 2014. arXiv:1404.5468
35. Pandurangan, G., Robinson, P., Scquizzato, M.: 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)*, pp. 743–756 (2017). https://doi.org/10.1145/3055399.3055449
36. Pandurangan, G., Robinson, P., Scquizzato, M.: 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
37. Peleg, D., Rubinovich, V.: A near-tight lower bound on the time complexity of distributed minimum-weight spanning tree construction. SIAM J. Comput. (2000). https://doi.org/10.1137/S0097539700369740
38. Sarma, A.D., Holzer, S., Kor, L., Korman, A., Nanongkai, D., Pandurangan, G., Peleg, D., Wattenhofer, R.: Distributed verification and hardness of distributed approximation. In *Proceedings of the 43th Annual ACM SIGACT Symposium on Theory of Computing (STOC)*, pp. 363–372 (2011). https://doi.org/10.1145/1993636.1993686

39. Wegman, M.N., Carter, L.: New hash functions and their use in authentication and set equality. J. Comput. Syst. Sci. (1981). https://doi.org/10.1016/0022-0000(81)90033-7

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.