Filling the Bandwidth Gap in Distributed Complexity for Global Problems*

Hiroaki Ookawa[1] and Taisuke Izumi[1]

Graduate School of Engineering, Nagoya Institute of Technology
cht15031@nitech.jp, t-izumi@nitech.ac.jp

Abstract. Communication complexity theory is a powerful tool to bound time complexity lower bounds of distributed algorithms for global problems such as minimum spanning tree (MST) and shortest path. While it often leads the nearly-tight lower bounds for many problems, polylogarithmic complexity gaps still lies between the currently best upper and lower bounds. In this paper, we propose a new approach for filling the gaps. Using this approach, we achieve tighter deterministic lower bounds for MST and shortest path. Specifically, for those problems, we show the deterministice $\Omega(\sqrt{n})$-round lower bound for graphs with $O(n^\epsilon)$ hop-count diameter, and the deterministice $\Omega(\sqrt{n/\log n})$ lower bound for graphs with $O(\log n)$ hop-count diameter. The main idea of our approach is to introduce a new function we call *permutation identity* and utilize its two-party communication complexity lower bound.

1 Introduction

In distributed computing theory, many graph problems are naturally treated as problems in networks, where each vertex represents a computing entity and each edge does a communication link between two nodes. The theory of *distributed graph algorithms* has been developed so far for efficient in-network computation of graph problems. A crucial factor of distributed graph algorithms is *locality*. Local algorithms require each node to compute its output only by the interaction to the nodes within a bounded distance smaller than the diameter of the network. In other words, local algorithms must terminate within $o(D)$ rounds, where D is the hop-count diameter of the network. There are a number of problems allowing local solutions: Maximal matchings, colorings, independent sets, and so on. On the other hand, some of other graph problems (e.g., minimum spanning tree shortest path, minimum cut) are known to have no local solution. They are called *global problems*. By the definition, the (worst-case) run of any algorithm for global problems inherently takes $\Omega(D)$ rounds.

For both local and global problems, the time complexity analysis for distributed algorithms (i.e., *distributed complexity theory*) are one of the important topics in distributed algorithms. In this paper, we focus on the distributed complexity of two well-known global problems: Minimum spanning tree (MST) and

* This work is supported in part by ——

shortest s-t path. As we stated above, these problems have trivial $\Omega(D)$-round lower bounds. If the communication bandwidth of each link is not bounded, every global problem has an optimal-time algorithm with $O(D)$ rounds: A process aggregates all the information of the network, and computes the result locally. However the assumption of so rich bandwidth is far from real systems, and thus the challenge of global problems is to solve them in the environment with limited bandwidth. Theoretically, such environments are called as the *CONGEST model*, where processes work under the round-based synchrony, and each link can transfer $O(\log n)$-bit messages per one round.

A seminal results about the lower bounds for global problems is the one by Das Sarma et al. [1], which exhibits that many problems, including MST and shortest s-t path, are more expensive tasks. Precisely, it shows that $\Omega(\sqrt{n}/\log n + D)$-round lower bounds hold for many global problems even D is small (i.e., $D = O(\log n)$). The core of this result is a general framework to obtain the lower bounds based on the reduction from two-party communication complexity by Yao [14]. Two-party communication complexity is a theory to reveal the amount of communication to compute a global function whose inputs are distributed among two players. The reduction framework in [1] induces the hardness of MST and shortest s-t path from the two-party communication complexity of *set-disjointness* function. While the framework is a powerful tool to bound the time compelxity of global problems, all the bounds led by that approach have a form of $\Omega(f(n)/(m \log n))$, where $f(n)$ is the amount of information inherently exchanged among the networks to solve the target problem, and m is the number of links where the information must be transferred, and $\log n$ factor is the bandwidth of each link (that is, $m \log n$ is the amount of information transmittable within a round). On the other hand, these lower bounds does not strictly match the known corresponding upper bounds, which typically has the form of $O(f(n)\mathit{polylog}(n)/m)$. That is, for many global problems, the currently best bounds still have (poly)logarithmic gaps.

The primary objective of this paper is to fill those gaps. For that goal, we propose a new two-party function whose deterministic communication complexity is slightly more expensive than set-disjointness, called *permutation identity*, and new reductions using it on the top of the framework by Das Sarma et al. [1]. Our contribution is to give tighter deterministic lower bounds for MST and shortest s-t path. Specifically, for those problems, we show the deterministice $\Omega(\sqrt{n})$-round lower bound for graphs with $O(n^\epsilon)$ hop-count diameter, and the deterministice $\Omega(\sqrt{n/\log n})$ lower bound for graphs with $O(\log n)$ hop-count diameter. The comparision with the prior work are shown in Table 1. As far as we consider the complexity of *deterministic* and *exact* computation, our bound beats the currently best ones. It also should be noted that the MST problem is almost closing the gap because the currently best upper bound is $O(\sqrt{n}\log^* n + D)$ rounds [3].

paper	bound	problem	comments
Garay et al. [3]	$O(\sqrt{n}\log^* n + D)$	MST	deterministic
Nanongkai [10]	$O(\sqrt{n}D^{1/4} + D)$	SP	$(1+o(1))$-approximation single-source SP
Das Sarma et al.[1]	$\Omega(\sqrt{\frac{n}{\log n}})$	SP,MST	randomized $\alpha(n)$-approximation $D = O(n^\epsilon)$ $(\epsilon < 1/2)$
Das Sarma et al.[1]	$\Omega(\frac{\sqrt{n}}{\log n})$	SP, MST	randomized $\alpha(n)$-approximation $(D = \Theta(\log n))$
This paper	$\Omega(\sqrt{n})$	SP, MST	deterministic $D = O(n^\epsilon)$ $(\epsilon < 1/2)$
This paper	$\Omega(\sqrt{\frac{n}{\log n}})$	SP, MST	deterministic $D = O(\log n))$

Table 1: Comparison with the prior work. SP (resp. MST) means shortest s-t path (res. minimum spanning tree).

2 Related Work

The paper by Das Sarma et al. [1] is the first one explicitly considering the distributed verification problem, which has given a general framework to lead lower bounds and approximation hardness for a vast class of problems. It is used in several following papers to obtain the complexity for a number of graph problems: Weighted/unweighted diameter and all-pair shortest paths [5, 7, 8, 12], minimum cuts [4, 10], distance sketches [7], weighted single-source shortest paths [7, 10], fast random walks [11], and so on.

While the framework by Das Sarma et al. [1] pointed out a general relationship interconnecting the communication complexity theory and distributed complexity theory, the construction of worst-case instances used in the framework is much inspired by the earlier papers leading the time lower bound for the distributed MST construction [2, 9, 13].

3 Preliminaries

3.1 Round-Based Distributed Systems

A distributed system consists of n nodes interconnected with communication links. We model it by a weighted graph $G = (V, E, w)$, where is the set of nodes, $E \subseteq V \times V$ is the set of links (edges), and $w : E \to \mathbb{R}$ is a weight function. The hop-count diameter of G (i.e., the diameter of the unweighted graph (V, E)) is

denoted by D. Executions of the system proceed with a sequence of consecutive rounds. In each round, each process sends a (possibly different) message to each neighbor, and within the round, all messages are received. After receiving the messages, the process performs local computation. Throughout this paper, we restrict the number of bits transmittable through any communication link per one round to $O(\log n)$ bits. This is known as the CONGEST model.

3.2 Distributed MST and Single-source shortest paths

In this paper we consider two popular graph problems: Minimum spanning tree (MST) and shortest s-t path. The distributed minimum spanning tree problem requires the system to find the MST of the (weighted) network. After the computation by distributed MST algorithms, each node must identify the incident edges constituting the MST. In the shortest s-t path problem, the algorithm takes two input nodes s and t, and computes a shortest path between them. After the computation, each node on the computed path must identify the incident edge toward s and the distance from s.

3.3 Two-Party Communication Complexity

Communication complexity, which is first introduced by Yao [14], reveals the amount of communication to compute a global function whose inputs are distributed in the network. The most successful scenario in communication complexity is *two-party* communication complexity, where two players, called Alice and Bob, respectively have their inputs $x, y \in U$ (where U is the domain of inputs), and compute a global function $f : U \times U \to \{0, 1\}$. The communication complexity of a two-party protocol is the number of one-bit messages exchanged by the protocol for the worst case input (if the protocol is randomized, it is defined as the expected number of bits exchanged for the worst-case input). One of the most popular functions in two-party communication complexity is *set-disjointness*, which is the function over two k-bit 0-1 vectors $x, y \in 0, 1^k$ and return value one if and only if there exists a common position $i \in [0, k-1]$ such that i-th bits of x and y are one.

While the known best lower bounds for MST and shortest s-t path is obtained by using the communication complexity of set-disjointness, it does not suffice to have a stronger bound we will prove. Thus in this paper, we introduce a new function called *permutation identity*, which is defined as follows:

Definition 1. *Let $\pi_A, \pi_B : [1, N] \to [1, N]$ be permutations over $[1, N]$. the permutation identity function $ident_N$ is defined as follows:*

$$ident_N(\pi_A, \pi_B) = \begin{cases} 1 \text{ if } \forall i \in [1, N] : \pi_A \circ \pi_B(i) = i, \\ 0 \text{ otherwise}, \end{cases}$$

where $\pi_A \circ \pi_B$ means the composition of π_A and π_B, that is, $\pi_A \circ \pi_B(i) = \pi_A(\pi_B(i))$.

Theorem 1. *The deterministic communication complexity of two-party permutation identity over* $[1, N]$ *is* $\Omega(n \log N)$ *bits.*

We also show a fundamental lemma for the permutation identity function, which is used in the following sections.

Lemma 1. *Let* π_A *and* π_B *be permutations over* $[1, N]$*. If* $\pi_A \circ \pi_B$ *is not identical, there exists* $i \in [1, N]$ *such that* $\pi_A \circ \pi_B(i) < i$ *holds.*

For lack of space, the proof for the theorem and lemma above are presented in the appendix.

4 General Framework for the Reduction

The proof of our lower bounds basically follows the framework by Das Sarma et al. [1]. The core of this framework is the reduction from two-party computation via a hard instance for distributed computation. In this section, we introduce the framework which is sligtly modified for our proof.

4.1 Graph Construction

The graph we construct is denoted by $G(N, M)$, where N and M are design parameters of the graph. For simplicity of the argument, throughout the paper, we assume that $M + 1$ is a power of 2, i.e., $M = 2^p - 1$ for some nonnegative integer p. Note that the assumption is not essential and it is not difficult to remove it. The graph is built by the following steps:

1. Prepare N paths of length M, each of which is denoted by P_i $(1 \leq i \leq N)$. The nodes constituting P_i are identified by $v_i^0, v_i^1, \cdots, v_i^M$ from left to right.
2. Add edges (v_i^0, v_j^1) and $(v_i^{(M-1)}, v_j^M)$ for any $i, j \in [1, N]$.
3. Add edges $(v_i^0, v_{(i+1)}^0)$ and $(v_i^M, v_{(i+1)}^M)$ for any $i \in [1, N-1]$.
4. Construct a complete binary tree $T(M)$ with $M + 1$ leaf. where each leave is labeled by $u^0, u^1, \cdots, u^M$ from left to right.
5. Add edges (u^i, v_j^i) for any $i \in [0, M]$ and $j \in [1, N]$.

The weight of each edge depends on concrete reductions, which is determined later. Note that the number n of nodes in $G(N, M)$ is $\Theta(NM)$, and its diameter is $D = O(\log n)$. We also define the sets of nodes $A = \{u^0\} \cup \{v_i^0, v_i^1 | i \in [1, N]\}$ and $B = \{u^M\} \cup \{v_i^{(M-1)}, v_i^M | i \in [1, N]\}$. The whole construction is illustrated in Figure 1. For this graph, we can show the following theorem.

Theorem 2 (Das Sarma et al. [1]). *Let* $\mathcal{A}$ *be any algorithm running on the graph* $G(N, M)$ *with an arbitrary edge-weight function. Then there exists a two-party protocol satisfying the following three properties:*

- *At the beginning of the protocol, Alice (resp. Bob) knows the whole topological information of* $G(N, M)$ *except for the subgraph induced by* B *(resp.* A*),*

- *after the run of the protocol, Alice and Bob output the internal states of the processes in A and B at round $(M-3)/2$ in the execution of $\mathcal{A}$ on $G(N, M)$, respectively, and*
- *the protocol consumes at most $O(M(\log MN)^2)$-bit communication.*

While the graph used in this paper is a slightly modified version of the original construction in [1], the theorem above is proved In the almost same way. So we just quote it without the proof.

4.2 Networked Two-Party Computation

To obtain the lower bounds for distributed algorithms, we uses a variation of the two-party computation problem in distributed settings. We assume that Alice and Bob are placed at two nodes in a network of n nodes, and have inputs $x \in U$ and $y \in U$ for two-party function $f : U \times U \to \{0, 1\}$, respectively. It is also assumed that each node in the network (including ones other than Alice and Bob) knows everything (i.e., the complete knowledge of the network topology) except for the inputs held by Alice and Bob. Then all nodes must work cooperatively for outputting the value of $f(x, y)$ as fast as possible. In what follows, we call this problem setting the *networked two-party computation* (and the networked permutation identity problem if $f = ident_N$). Note that the measurement of the networked two-party computation is not the amount of communication, but the number of rounds.

Obviously the time complexity of networked two-party computation problems relies on the target function f and the topology of the network. An useful consequence from Theorem 2 is that we can transform the communication lower bound for any two-party computation into the time lower bound for its networked version. In the original version by Das Sarma et al. [1], the transformation from two-party set-disjointness is considered. Here we problem the similar fact from two-party permutation identify function (the proof are in the appendix):

Theorem 3. *Let $M = N/\log N$. For any deterministic algorithm $\mathcal{A}$ solving the networked permutation identity over $[1, N]$ in $G(N, M)$, its worst-case running time is $\Omega(\sqrt{n/\log n})$ rounds.*

4.3 Lower bound for MST

We show the reduction from the networked permutation identity to MST. In this reduction we construct an instance of the MST problem by virtually assigning some weight to each edge in $G(N, M)$ for $M = N/\log N$ to encode an instance (π_A, π_B) of permutation identity over $[1, N]$. After the construction of the MST, Alice and Bob can determine the identity of $\pi_A \circ \pi_B$ from the computed MST. Let $L(\pi_A, \pi_B)$ be the instance of the MST problem corresponding to the permutation identity instance (π_A, π_B), which is constructed by defining edge-weight function w as follows:

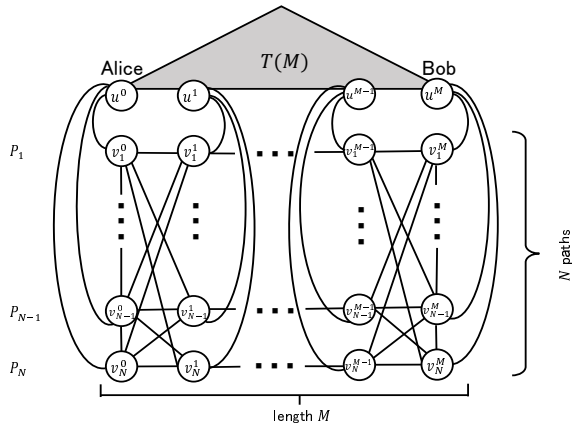

Fig. 1: Construction of $G(N, M)$

1. For any $i \in [1, N]$ and $j \in [1, M-1]$, $w(u^j, v_i^j) = 100NM$.
2. For any $i \in [1, N-1]$, $w(v_i^0, v_{i+1}^0) = 100NM$ and $w(v_i^M, v_{i+1}^M) = 100NM$.
3. For any $i \in [1, N]$, $w(u^0, v_i^0) = 2i$ and $w(u^M, v_i^M) = 2i - 1$.
4. For any $i, j \in [1, N]$, $w(v_i^0, v_j^1) = 1$ if $\pi_A(j) = i$. Otherwise $w(v_i^0, v_j^1) = 100NM$. Similarly, For any $i, j \in [1, N]$, $w(v_i^{M-1}, v_j^M) = 1$ if $\pi_B(j) = i$. Otherwise $w(v_i^{M-1}, v_j^M) = 100NM$.
5. All other edges have weight one.

The construction of $L(\pi_A, \pi_B)$ is illustrated in Figure 2. Let $E_A = \{(u^0, v_i^0) | i \in [1, N]\}$ and $E_B = \{(u^M, v_i^M) | i \in [1, N]\}$. The following lemma is the core of the reduction.

Lemma 2. *The MST of $L(\pi_A, \pi_B)$ contains no edge in E_A if and only if $\pi_A \circ \pi_B$ is identical.*

Proof. Let P'_i be the path consisting of the nodes $v^0_{\pi_A(\pi_B(i))}, v^1_{\pi_B(i)}, v^2_{\pi_B(i)}, \cdots, v^{M-1}_{\pi_B(i)}, v^M_i$. Following the standard greedy algorithm for constructing the MST, every edge with weight one is contained in the MST. Thus, the components $P'_1, P'_2, \cdots, P'_N$ and $T(M)$ are MST fragments. A component P'_i is merged with $T(M)$ by choosing either $(u^0, v^0_{\pi_A(\pi_B(i))})$ or (u^M, v_i^M) (all other edges merging them are too heavy (i.e., $100NM$) and never chosen as a MST edge). If $\pi_A \circ \pi_B$ is identical, $\pi_A(\pi_B(i)) = i$ holds. Thus we have $w(u^0, v^0_{\pi_A(\pi_B(i))}) = 2i$ and $w(u^M, v_i^M) = 2i - 1$ for any $i \in [1, N]$. This implies that P'_i is merged with $T(M)$ by edge $(u^0, v^0_{\pi_A(\pi_B(i))})$ (Figure 3). On the other hand, if $\pi_A \circ \pi_B$ is not identical, from Lemma 1, there exists at least one i satisfying $\pi_A \circ \pi_B(i) < i$. Then for such i we have $w(u^0, v^0_{\pi_A(\pi_B(i))}) \leq 2(i-1)$ and $w(u^M, v_i^M) = 2i - 1$. Thus P'_i and $T(N)$ is merged with edge $w(u^0, v^0_{\pi_A(\pi_B(i))}) \in E_A$ (Figure 4). The lemma is proved □

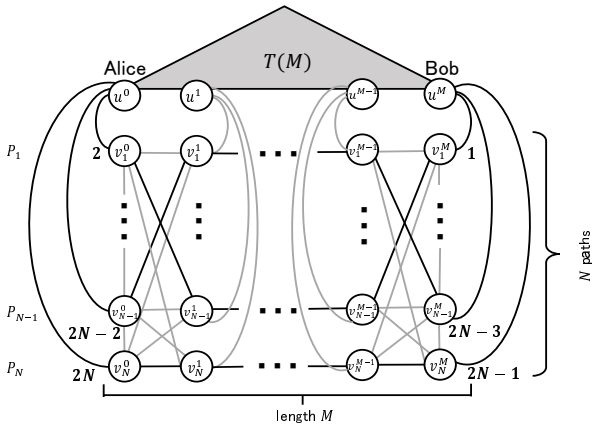

Fig. 2: An example of $L(\pi_A, \pi_B)$. Every unlabeled edge has weight one. All the edges with weight $100NM$ are grayed out.

Lemma 3. *If an algorithm $\mathcal{A}$ solves the MST problem in $L(\pi_A, \pi_B)$ within r rounds, there exists an algorithm solving the networked permutation identity over $[1, N]$ in $G(N, M)$ within $O(r)$ rounds.*

Proof. At the round one and two, each node sets up the instance $L(\pi_A, \pi_B)$ of the MST problem according to the input (π_A, π_B). Then the system runs the MST algorithm $\mathcal{A}$. From lemma 2, no edge in E_A is not included in the constructed MST if $\pi_A \circ \pi_B$ is identical. Then, after the construction of the MST, each node v_i^0 $(i \in [1, N])$ sends to u^0 the information that no incident edge is contained in the MST. By this information, u^0 can determine whether $\pi_A \circ \pi_B$ is identical or not. That is, the networked permutation identity is solved in $G(N, M)$ within $O(r)$ rounds. □

Combining Theorem 3 and Lemma 3, we have the main theorem below.

Theorem 4. *Any deterministic algorithm solving the MST problem, its worst-case running time is $\Omega(\sqrt{n/\log n})$ rounds.*

4.4 Lower Bound for Shortest *s*-*t* Path

The argument in this section is almost the same as Section 4. We construct a graph $L'(\pi_A, \pi_B)$ by fixing a weight function w for the network $G(N, N \log N)$. The weight function w is defined as follows:

1. For any $i \in [1, N]$ and $j \in [0, M]$, $w(u^j, v_i^j) = 100NM$.
2. For any $i \in [1, N-1]$, $w(v_i^0, v_{i+1}^0) = 1$ and $w(v_i^M, v_{i+1}^M) = 1$.

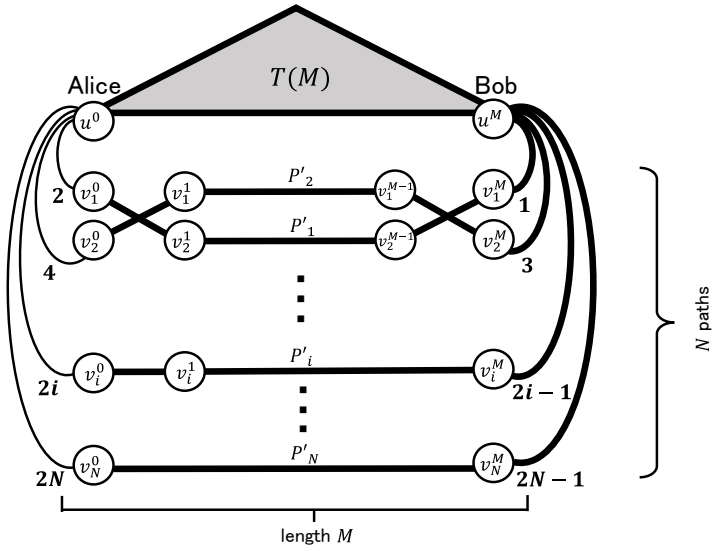

Fig. 3: Graph $L(\pi_A, \pi_B)$ when $\pi_A \circ \pi_B$ is identical.

3. For any $i, j \in [1, N]$, $w(v_i^0, v_j^1) = 100NM$ if $\pi_A(j) = i$. Otherwise $w(v_i^0, v_j^1) = 100NM$. Similarly, For any $i, j \in [1, N]$, $w(v_i^{M-1}, v_j^M) = 1$ if $\pi_B(j) = i$. Otherwise $w(v_i^{M-1}, v_j^M) = 100NM$.
4. For any $i \in [1, N]$ and $j \in [1, M-1]$, $w(v_i^j, v_i^{j+1}) = 1$.
5. Every edge in $T(M)$ has weight $100NM$.

We also define $s = v_1^0$ and $t = v_N^M$. Then, we have the following lemma:

Lemma 4. *In graph $L'(\pi_A, \pi_B)$, the length of the shortest s-t path is $N+M-1$ if and only if $\pi_A \circ \pi_B$ is identical.*

Proof. The path $v_1^0, v_2^0, \cdots v_N^0, v_N^1, v_N^2, \cdots V_N^{M-1}, v_N^M$ is the s-t path of length $N + M - 1$. We first show that this is the shortest path if $\pi_A \circ \pi_B$ is identical. Since the length of the shortest path between s and t is at most $N + M - 1$, it contains no edge with weight $100NM$. Thus we omit those edges. Then, if $\pi_A \circ \pi_B$ is identical, v_i^0 and v_i^M are connected by a path of length M. Thus, the graph (where all isolated nodes in $T(M)$ are removed) becomes a subdivision of a ladder graph. It is not difficult to see that the shortest path between s and t is $N + M - 1$ (Figure 5).

We next consider the case where $\pi_A \circ \pi_B$ is not identical. Then, from Lemma 1, there exists i satisfying $\pi_A \circ \pi_B(i) < i$. Then, we have an s-t path $v_1^0, v_2^0, \cdots v^0_{\pi_A(\pi_B(i))}, v^1_{\pi_B(i))}, v^2_{\pi_B(i))}, \cdots, v^{M-1}_{\pi_B(i))}, v_i^M, v_{i+1}^M, \cdots, v_N^M$ of length less than $N + M - 1$ (Figure 6). The lemma is proved □

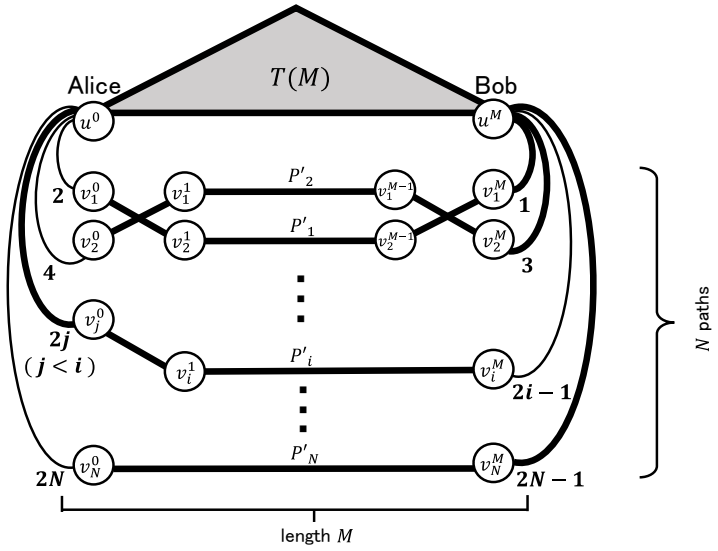

Fig. 4: Graph $L(\pi_A, \pi_B)$ when $\pi_A \circ \pi_B$ is not identical.

Lemma 5. *If an algorithm $\mathcal{A}$ solves the shortest s-t path problem in $L'(\pi_A, \pi_B)$ within r rounds, there exists an algorithm solving the networked permutation identity over $[1, N]$ within $O(r)$ rounds.*

The proof is almost the same as that for Lemma 3, and thus we omit it. Finally we obtain the following theorem.

Theorem 5. *Any deterministic algorithm solving the shortest s-t path problem, its worst-case running time is $\Omega(\sqrt{n/\log n})$ rounds.*

5 Lower bound for the graphs with $O(n^\epsilon)$ hop-count diameter

For the case of larger diameter graphs, we obtain stronger bounds by slightly modifying the framework graph $G(N, M)$. Since the fundamental idea has been proposed in the prior work [1], we state only the results in this paper. The Theorem 4 and 5 are extended as follows:

Theorem 6. *Any deterministic algorithm solving the MST problem or the shortest s-t path problem, its worst-case running time is $\Omega(\sqrt{n/\log n})$ rounds for graphs with diamter $O(\log n)$. In addition, for graphs with diameter $O(n^\epsilon)$ $(0 < \epsilon < 1/2)$, the worst-case running time is $\Omega(\sqrt{n/\log n})$ rounds.*

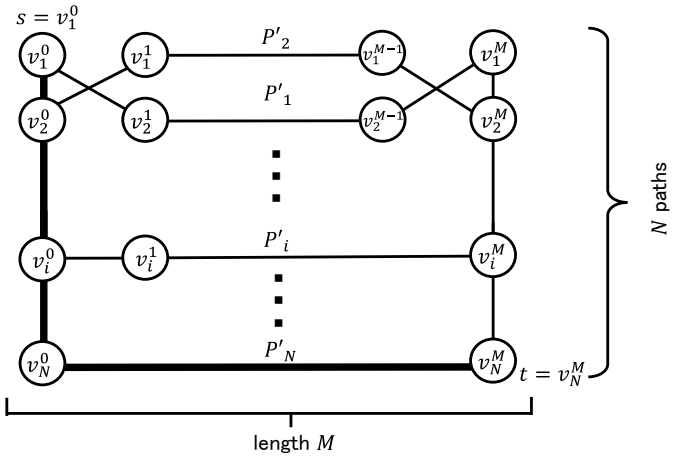

Fig. 5: Example of shortest path (marked with big edges) when $\pi_A \circ \pi_B$ is identical

6 Concluding Remarks

In this paper, we inroduced a new function called *permutation identity*. By using the seminal reduction framework by Das Sarma et al.[1], we show the deterministic $\Omega(\sqrt{\frac{n}{\log n}})$-round lower bounds for MST and shortest s-t path. Furtuermore, for graph with for graphs with $O(n^\epsilon)$ hop-count diameter, we obtained $\Omega(\sqrt{n})$ lower bound, For the MST problem, this lower bound is almost closing the logarithmic gap because the best upper bound is $O(\sqrt{n}\log^* n + D)$.

References

1. 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 *Proc. of the 43rd Annual ACM Symposium on Theory of Computing*, pages 363–372, 2011.
2. Michael Elkin. An unconditional lower bound on the hardness of approximation of distributed minimum spanning tree problem. In *Proc the 30th ACM Symposium on Theory of Computing(STOC)*, pages 331 – 340, 2004.
3. Juan A. Garay, Shay Kutten, and David Peleg. A sublinear time distributed algorithm for minimum-weight spanning trees. *SIAM Journal on Computing*, 27(1):302–316, 1998.
4. Mohsen Ghaffari and Fabian Kuhn. Distributed minimum cut approximation. In *Proc. of 27th International Symposium on Distributed Computing (DISC)*, pages 1 – 15, 2013.
5. Stephan Holzer and Roger Wattenhofer. Optimal distributed all pairs shortest paths and applications. In *Proc. of the 2012 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 355–364, 2012.

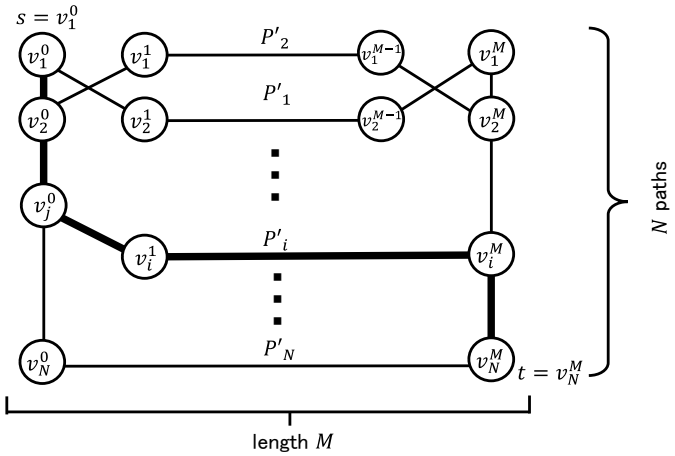

Fig. 6: Example of shortest path (marked with big edges) when $\pi_A \circ \pi_B$ is not identical ($\pi_A \circ \pi_B(i) = j$)

6. E. Kushilevitz and N. Nisan. *Communication Complexity*. Cambridge University Press, 1997.
7. Christoph Lenzen and Boaz Patt-Shamir. Fast routing table construction using small messages: Extended abstract. In *Proc. of the 45th Annual ACM Symposium on Symposium on Theory of Computing (STOC)*, pages 381–390, 2013.
8. Christoph Lenzen and David Peleg. Efficient distributed source detection with limited bandwidth. In *Proc. of the 2013 ACM Symposium on Principles of Distributed Computing (PODC)*, pages 375–382, 2013.
9. Zvi Lotker, Boaz Patt-Shamir, and David Peleg. Distributed mst for constant diameter graphs. *Distributed Computing*, 18(6):453–460, 2006.
10. Danupon Nanongkai. Distributed approximation algorithms for weighted shortest paths. In *Proc. of the 46th ACM Symposium on Theory of Computing (STOC)*, 2014.
11. Danupon Nanongkai, Atish Das Sarma, and Gopal Pandurangan. A tight unconditional lower bound on distributed random walk computation. In *Proc. of the 30th Annual ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing (PODC)*, pages 257–266, 2011.
12. David Peleg, Liam Roditty, and Elad Tal. Distributed algorithms for network diameter and girth. In *Proc. of the 39th International Colloquium Conference on Automata, Languages, and Programming (ICALP)*, pages 660–672, 2012.
13. 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*, 30(5):1427–1442, 2000.
14. Andrew Chi-Chih Yao. Some complexity questions related to distributive computing(preliminary report). In *Proc. of the 11th Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.

A Omitted Proofs

A.1 Proof of Theorem 1

Proof. The proof of this theorem is almost the same as the well-known result for the two-party equality function. More precisely, the proof follows the *fooling-set* argument. Let $f : U \times U \to \{0, 1\}$ be a two-party function over input domain U. A subset $S \subset U \times U$ is called a *fooling set* of function f if the following conditions are satisfied for some $z \in \{0, 1\}$: (1) For any $(x, y) \in U \times U$, $f(x, y) = z$, and (2) for any distinct inputs $(x_1, y_1), (x_2, y_2) \in S$, either $f(x_1, y_2) \neq z$ or $f(x_2, y_1) \neq z$. It is well-known that the deterministic communication complexity of function f is bounded by $\Omega(\log |S|)$ (the detailed argument is found in the standard textbook of communication complexity theory [6]).

Thus it suffices to show that function $ident_N$ has a fooling set S of size $2^{\Omega(N \log N)}$. We constitute S by including all pairs (π_A, π_B) such that $\pi_A \circ \pi_B$ becomes the identical mapping. Then for any $(\pi_A, \pi_B) in S$, there is no other mapping π_C such that $\pi_A \circ \pi_C$ or $\pi_C \circ \pi_B$ becomes identical. Thus the set S clearly satisfies the conditions of fooling sets. Since the cardinarity of S is $N!$, we have the communication complexity lower bound of $\Omega(\log N!) = \Omega(N \log N)$ bits. The theorem is proved. □

A.2 Proof of Lemma 1

Proof. Suppose for contradiction that $\pi_A \circ \pi_B$ is not identical but $\pi_A \circ \pi_B(i) \geq i$ holds for any $i \in [1, N]$. Then, clearly we have $\pi_A \circ \pi_B(N) = N$, and thus we have $\pi_A \circ \pi_B(N - 1) = N - 1$, $\pi_A \circ \pi_B(N - 2) = N - 2$, $\cdots$, $\pi_A \circ \pi_B(0) = 0$. Consequently $\pi_A \circ \pi_B$ becomes identical. It is a contradiction. □

A.3 Proof of Theorem 3

Proof. Since $M = N/\log N$, we have $n = \Theta(N \cdot N/\log N) = \Theta(N^2/\log N)$. Thus we also have $\Theta(\log N) = \Theta(\log n)$ and thus, $n \log n = \Theta(N^2)$ holds. It implies $N = \sqrt{n \log n}$ and $M = \sqrt{n/\log n}$. To prove the lemma, it suffices to show that the running time of $\mathcal{A}$ in $G(N, M)$ is $\Omega(M)$ rounds. Suppose for contradiction that $\mathcal{A}$ terminates within $o(M)$ rounds. Consider the network $G(N, M)$ where Alice and Bob are respectively placed at u_0 and u_M. Then, following Theorem 2, we can construct a two-party permutation identity protocol over $[1, N]$ by simulating the execution of $\mathcal{A}$ in $G(N, M)$. That is, Alice and Bob (in the two-party computation) first set up the initial configuration of $\mathcal{A}$ by installing their own inputs x and y and run the simulation. After the simulation, they output the computation result $ident_N(x, y)$ as the result of the two-party computation. This two-party protocol consumes $o(M(\log N)^2) = o(N \log N)$ bits. It contradicts Theorem 1. □