PAPER Special Section on Foundations of Computer Science

Multi-Party Quantum Communication Complexity with Routed Messages

Seiichiro TANI[†a)], **Masaki NAKANISHI**[††b)], *and* **Shigeru YAMASHITA**[††c)], *Members*

SUMMARY This paper describes a general quantum lower bounding technique for the communication complexity of a function that depends on the inputs given to two parties connected via paths, which may be shared with other parties, on a network of any topology. The technique can also be employed to obtain a lower-bound of the quantum communication complexity of some functions that depend on the inputs distributed over all parties on the network. As a typical application, we apply our technique to the *distinctness* problem of deciding whether there are a pair of parties with identical inputs, on a k-party ring; almost matching upper bounds are also given.

key words: *quantum communication complexity, network topology, distributed computing*

1. Introduction

Studying *communication complexity* has been one of the central issues in computer science since its introduction by Yao [19]. Not only it is interesting in its own right, but it also has many applications such as analyzing VLSI circuit designs, data structures and networks (See the book [13] for more details).

In the simplest case where there are two parties connected to each other by a communication channel, two parties, say, Alice and Bob, get inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, respectively, and compute $f(x,y) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ cooperatively by exchanging messages. For example, Alice first performs local computation depending on her input and sends a message to Bob. He then does some local computation depending on his input and the received message, and sends a message back to Alice. This message exchange is repeated until Alice or Bob outputs the value of f. For any protocol $\mathcal{P}$ that computes f, the cost of $\mathcal{P}$ is the number of communication bits on the worst-case input (x,y). The communication complexity of f, $D(f)$, is the minimum cost of $\mathcal{P}$, over all deterministic protocols $\mathcal{P}$ that compute f. Protocol $\mathcal{P}$ may be randomized, i.e., Alice and Bob can access random strings r_A and r_B, respectively, in addition to the inputs they receive. The communication complexity of a randomized protocol that computes f is the number of communication bits in the worst-case over all inputs and all random strings. The communication complexity $R_\epsilon(f)$ of f for error probability ϵ is the minimum communication complexity over all randomized protocols that compute f with error probability at most ϵ for every input. If ϵ is bounded by a certain constant that is less than 1/2, we call it *bounded error*. Without loss of generality, ϵ is assumed to be 1/3 in the bounded error setting unless it is explicitly set to a different value. There is another randomized setting: a randomized protocol that never outputs an incorrect answer, but may give up with probability at most ϵ. We call such a protocol a Las Vegas protocol or a zero-error protocol. The communication complexity of f in the zero-error setting is denoted by $R_{0,\epsilon}(f)$. Furthermore, there is another way of giving random strings to Alice and Bob: they are allowed to access public coins (*or* a common random string). Formally, the output of protocol $\mathcal{P}$ depends on the inputs and common random string r. The public-coin versions of $R_\epsilon(f)$ and $R_{0,\epsilon}(f)$ are denoted by $R^{pub}_\epsilon(f)$ and $R^{pub}_{0,\epsilon}(f)$, respectively.

Quantum communication complexity, introduced by Yao [20], is the quantum counterpart of (classical) communication complexity. Parties are allowed to perform quantum computation and send/receive quantum bits (*or* qubits). The communication complexities, $Q_E(f)$, $Q_\epsilon(f)$ and $Q_{0,\epsilon}(f)$ are defined as the quantum counterparts of $D(f)$, $R_\epsilon(f)$ and $R_{0,\epsilon}(f)$, respectively. In particular, the quantum counterpart of deterministic computation (protocol, algorithm, etc.) is called exact computation (protocol, algorithm, etc.); it runs in bounded time and always outputs the correct answer.

It is known that there are functions for which non-constant gaps exist between quantum and classical communication complexity. For exact computation, Buhrman et al. [5] proved that for a certain promise version of the equality function EQ'_n, $Q_E(\mathsf{EQ}'_n) = O(\log n)$ while $D(\mathsf{EQ}'_n) \in \Omega(n)$ [7]. In the bounded-error case, Raz [16] showed a promise problem that has an exponential gap between quantum and classical settings, i.e., $Q_{1/3}(f) = O(\log n)$ and $R_{1/3}(f) = \Omega(n^{1/4}/\log n)$. As for total functions, the largest known gap is quadratic: $Q_{1/3}(\mathsf{DISJ}_n) = \Theta(\sqrt{n})$ [1], [17] and $R_{1/3}(\mathsf{DISJ}_n) = \Omega(n)$ [10], where DISJ_n is the $2n$-bit disjoint function, i.e., $\bigwedge_{i=1}^n (\overline{x_i y_i})$. Exponential gaps have been demonstrated for restricted or other models; examples include the one-way bounded-error model [2], [8] and the bounded-error simultaneous message-passing model [6].

As mentioned above, there have been a lot of researches on the standard two-party communication model

Manuscript received March 28, 2008.
Manuscript revised July 8, 2008.

[†]The author is with NTT Communication Science Laboratories, NTT Corporation, Atsugi-shi, 243–0198 Japan.
[††]The authors are with the Graduate School of Information Science, Nara Institute of Science and Technology, Ikoma-shi, 630–0192 Japan.
a) E-mail: tani@theory.brl.ntt.co.jp
b) E-mail: m-naka@is.naist.jp
c) E-mail: ger@is.naist.jp

DOI: 10.1587/transinf.E92.D.191

for quantum communication complexity. On the other hand, unlike the classical case, there is almost no research that considers more general (and more natural when we consider the Internet) model, i.e., distributed quantum computing over multiple parties on a network whose underlying graph is not necessarily complete. In this setting, a certain pair of parties may have to communicate with each other via some other parties; it seems difficult to directly apply known techniques for the standard two-party communication model.

Our contribution. We first show that the quantum communication complexity $Q_{1/3}(f)$ of $f(x, y)$ in the standard two party case implies a non-trivial lower bound for the total quantum communication complexity over all links in a network G (consisting of many parties) for computing $f(x, y)$ (denoted by $Q^G_{1/3}(f)$ hereafter) when x and y are given as n-bit string input to two parties, P_A and P_B, on G. For any protocol with which P_A and P_B compute $f(x, y)$ on a network G, we divide all parties on G into s disjoint *layers* (with some properties which will be described in our proof), where w is the maximum number of links between two adjacent layers. Then, we show that

$$Q^G_{1/3}(f) = \Omega(s(Q_{1/3}(f) - \log \min\{n, s\})/\log w).$$

Note that s and w may be chosen appropriately depending on a problem in order to get a good lower bound.

Our main idea to derive the above lower bound is to extend the classical deterministic lower bound technique in [18] to the quantum case. To do so, we introduce a new notion "*quantum protocol with classical public coins*," and then we modify the classical lower bound technique in a careful combination with the quantum version of the *public-to-private randomness conversion technique.* We also prove similar results in the zero-error setting.

We then apply the lower bound technique to lower-bound the quantum communication complexity of computing the distinctness problem on a k-party ring with bounded-error probability: the problem is deciding whether there are a pair of parties who get identical inputs in $\{0, \ldots, L-1\}$ on a k-party ring, for which we derive lower bound $\Omega(k(\sqrt{k} + \log\log L))$. We also give two quantum protocols for the problem. The first algorithm gives almost the matching upper bound: $O(k(\sqrt{k}\log k + \log\log L))$. The second algorithm has better upper bound than the first if $L < k(\log k)^2$: $O(k\sqrt{L})$, which is optimal in the case of $L = O(k)$. As far as we know, this is the first non-trivial result of almost tight bounds of multi-party quantum communication complexity on a network whose underlying graph is not complete.

2. Basic Tools

2.1 Converting Public Coins into Private Coins

In what follows, we assume that communication is quantum, but parties share no prior-entanglement. If a quantum protocol allows parties to access an arbitrary number of classical public coins, it is called a *quantum protocol with classical public coins*. $Q^{pub}_\epsilon(f)$ is defined as the minimum communication complexity over all *quantum protocols with classical public coins* that compute f with error probability at most ϵ.

As in the classical case [15], we would like to be able to replace many public coins with a small number of communication bits in the case of quantum protocols with classical public coins. Although it looks very similar to the classical case (also mentioned in [11]), the proof needs to be modified to handle quantum errors. The next proposition is used in the proof.

Proposition 1 (Hoeffding inequality (e.g., [13])): Suppose that $X_1, \ldots, X_t$ are t independent random variables with identical probability distribution over the real interval $[a, b]$ that have expected value p. Then

$$\Pr\left[\left|\frac{\sum_{i=1}^t X_i}{t} - p\right| \geq \delta\right] \leq 2e^{-\frac{2t\delta^2}{b-a}}.$$

Lemma 1: Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. For every positive real δ and ϵ ($\delta + \epsilon < 1/2$) , any ϵ-error quantum protocol with classical public coins can be transformed into an $(\epsilon + \delta)$-error quantum protocol without classical public coins by using additional $\lceil \log n + 2\log 1/\delta \rceil$-bit communication.

Proof Suppose that we have any ϵ-error quantum protocol with classical public coins, $\mathcal{P}$, that computes f, and assume that $\mathcal{P}$ chooses a random string according to probability distribution Π over all possible random strings. Let $P(x, y, r)$ be the event that $\mathcal{P}$ is given input (x, y) and chooses particular string r as the random string. The error probability of $\mathcal{P}$ under event $P(x, y, r)$, i.e., the probability that the output of $\mathcal{P}$ under $P(x, y, r)$ is not equal to $f(x, y)$, is denoted by $\mathbf{Er}[P(x, y, r)]$.

We will show that there exist t strings $r_1, \ldots, r_t$ such that, for every input (x, y), the expected value of $\mathbf{Er}[P(x, y, r)]$ for random r chosen uniformly from the t strings is at most $\epsilon+\delta$. Therefore, if Alice randomly chooses one of the t strings and sends the $\lceil \log t \rceil$ bits specifying the chosen string to Bob, then they can compute f with error probability at most $\epsilon + \delta$. The lemma follows.

Choose $t = \lceil n/\delta^2 \rceil$ strings $r_1, \ldots, r_t$ according to the probability distribution Π of common random strings. Since $0 \leq \mathbf{Er}[P(x, y, r_i)] \leq 1$, we can show by the Hoeffding inequality for fixed input (x, y) that

$$\mathbf{Pr}_{r_1,\ldots,r_t}\left[\left(\frac{1}{t}\sum_{i=1}^t \mathbf{Er}[P(x, y, r_i)] - \epsilon\right) > \delta\right] \leq 2e^{-2\delta^2 t}.$$

If we set t to $\lceil n/\delta^2 \rceil$, $2e^{-2\delta^2 t}$ is smaller than 2^{-2n}. Therefore, the probability that, for some input (x, y), $\frac{1}{t}\sum_{i=1}^t \mathbf{Er}[P(x, y, r_i)] > \epsilon + \delta$ is smaller than $2^{-2n} \cdot 2^{2n} = 1$ when $r_1, \ldots, r_t$ is randomly chosen. This implies that there exist $r_1, \ldots, r_t$ such that for every input (x, y), $\frac{1}{t}\sum_{i=1}^t \mathbf{Er}[P(x, y, r_i)] \leq \epsilon + \delta$. □

This lemma can be easily generalized to the case of k

parties, in which every party i gets $x_i \in \{0,1\}^n$ as input and they have to compute function f depending on x_i's.

Lemma 2: Let $f : \{0,1\}^{nk} \to \{0,1\}$ be a function. For every positive real δ and ϵ ($\delta + \epsilon < 1/2$) , any ϵ-error quantum protocol with classical public coins that computes f on k parties can be transformed into an $(\epsilon + \delta)$-error quantum protocol without classical public coins, by using additional communication to broadcast a $\lceil \log(kn) + 2\log 1/\delta \rceil$-bit message.

Proof Follow the same argument with $t = \lceil kn/(2\delta^2) \rceil$. □

In the case of a ring, the additional communication is just $k\lceil \log(kn) + 2\log 1/\delta \rceil$-bits, since broadcasting involves passing the message around the ring.

For zero-error quantum protocols, we can obtain a similar result.

Lemma 3: Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. For every positive real δ and ϵ ($\delta + \epsilon < 1$), any zero-error quantum protocol with classical public coins for computing f that may give up with probability at most ϵ can be transformed into a zero-error quantum protocol without public coins that may give up with probability at most $(\epsilon + \delta)$, by using additional $\lceil \log n + 2\log 1/\delta \rceil$-bit communication.

The proof is realized by considering the deviation of the average give-up probability over t random strings from the average give-up probability over all random strings.

2.2 Quantum Amplitude Amplification

We quote the quantum amplitude amplification theorem, which we will use in our proofs.

Theorem 4 ([4]): Let $\mathcal{A}$ be any quantum algorithm that uses no measurements, and let $\chi : \mathbb{Z} \to \{0,1\}$ be any Boolean function. Given the initial success probability $a > 0$ of $\mathcal{A}$, $Q^m(\mathcal{A},\chi)\mathcal{A}|0\rangle$ gives a good solution with probability $\sin^2((2m+1)\mathrm{Arcsin}\sqrt{a})$, where $Q(\mathcal{A},\chi) = -\mathcal{A}F_0\mathcal{A}^{-1}F_\chi$. Operator F_χ transforms x into $-|x\rangle$ if $\chi(x) = 1$, and leaves x unchanged otherwise; F_0 transforms $|x\rangle$ into $-|x\rangle$ if $x = 0\ldots0$, and leaves $|x\rangle$ unchanged otherwise.

3. General Lower Bound

Now we describe our key theorem which lower-bounds the total quantum communication complexity over all links of a network of any topology by using the ordinary quantum communication complexity of the two party case.

Theorem 5: Suppose that n-bit strings x and y are given to two parties P_A and P_B, respectively, on network G of any topology. Then suppose any partitioning of the network G into $(s+1)$ layers as in Fig. 1 such that the following conditions are satisfied:

1. every layer is a disjoint subset of the set of all parties in G,

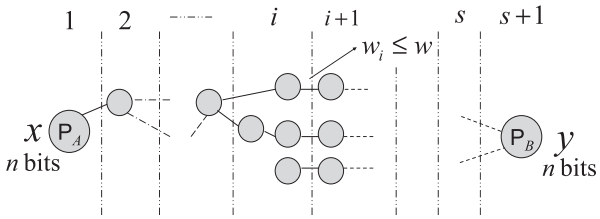

Fig. 1 Network G that is partitioned into $(s+1)$-layers.

2. the finrst layer has a unique member P_A,
3. the $(s+1)$st layer has a unique member P_B,
4. no edge jumps over a layer, i.e., there is no link between a party in the i-th layer and a party in the $(i+j)-$th layer for any $1 \le i < s$ and $i+1 < i+j \le s+1$,
5. w is the maximum number of links between two adjacent layers.

Let $Q_\epsilon^G(f)$ be the total quantum communication complexity over all links in G of computing a Boolean function $f(x,y)$ with error probability at most ϵ ($0 \le \epsilon < 1/2$). Then, for any s and w that satisfy the above conditions of the partitioning of G, $Q_\epsilon^G(f)$ is at least $\frac{s}{\lceil \log w \rceil}$ times

$$\max\{\delta_1(Q_{\epsilon+\delta_1/2+\delta_2}(f) - \lceil \log(n/\delta_2^2) \rceil), \delta_3(Q_{\epsilon+\delta_3/2}(f) - \lceil \log s \rceil)\},$$

where $0 < \delta_1, \delta_2, \delta_3 < 1$ such that $\epsilon + \delta_1/2 + \delta_2$ and $\epsilon + \delta_3/2$ are smaller than 1/2, and $Q_\epsilon(f)$ denotes the quantum communication complexity of $f(x,y)$ for error probability at most ϵ in the ordinary two party case (where the two parties are directly connected by a quantum communication link).

Proof The proof is similar to that given for the classical deterministic setting in [18], but we need to make a slight modification to it in order to handle bounded error setting.

Suppose that $\mathcal{P}$ be the best quantum protocol between P_A and P_B that computes $f(x,y)$ on network G with error probability at most ϵ. We then construct a quantum protocol with classical public coins between two parties, P'_A and P'_B, that are directly connected to each other by simulating protocol $\mathcal{P}$ as follows: if the value of the public coins is $i \in \{1,\ldots,s\}$, the two parties, P'_A and P'_B, simulate the left and the right parts, respectively, of the i-th and the $(i+1)$-st layers.

Let q_i be the number of communicated qubits between the i-th and the $(i+1)$-st layers during the execution of protocol $\mathcal{P}$, and let w_i be the number of links between the i-th and the $(i+1)$-st layers. It follows that the necessary communication bits for P'_A and P'_B in the above simulation is at most $q_i\lceil \log w_i \rceil$, since at most $\lceil \log w_i \rceil$ bits are needed, when simulating each message exchanged between the i-th and the $(i+1)$-st layers, to specify on which link among w_i links the message is sent. The obtained protocol between P'_A and P'_B computes f with error probability at most ϵ and with expected communication complexity $(1/s\sum_i q_i\lceil \log w_i \rceil)$.

To guarantee the worst case communication complexity, we modify this protocol so that if the amount of communication exceeds $1/\delta_1$ times $(1/s\sum_i q_i\lceil \log w_i \rceil)$ for $0 < \delta_1 < 1$, it stops and randomly outputs 0 or 1. The probability of

this event is at most δ_1 by Markov's inequality. On the condition that this event occurs, the probability of outputting the wrong value is exactly 1/2. Thus, the modified protocol has error probability at most $\epsilon + \delta_1/2$. Hence, we have obtained an $(\epsilon + \delta_1/2)$-error quantum protocol with classical public coins, whose complexity is $1/(s\delta_1)\sum_i q_i \lceil \log w_i \rceil \leq \lceil \log w \rceil/(s\delta_1) \cdot \sum_i q_i$. The last equality is due to $w_i \leq w$. This implies that $Q^{pub}_{\epsilon+\delta_1/2}(f) \leq \lceil \log w \rceil/(s\delta_1) \cdot \sum_i q_i$. Amount $\sum_i q_i$ is the total number of qubits communicated by protocol $\mathcal{P}$, $Q^G_\epsilon(f)$, which is lower-bounded by $s\delta_1 Q^{pub}_{\epsilon+\delta_1/2}(f)/\lceil \log w \rceil$.

By applying Lemma 1 to $Q^{pub}_{\epsilon+\delta_1/2}(f)$, we have

$$Q^G_\epsilon(f) \geq s\delta_1(Q_{\epsilon+\delta_1/2+\delta_2}(f) - \lceil \log n + 2\log 1/\delta_2 \rceil)/\lceil \log w \rceil.$$

There is a simpler way of deciding the boundary between P'_A's part and P'_B's part: P'_A randomly chooses one layer-boundary out of the s layer-boundaries and informs P'_B of the chosen layer-boundary by sending a $\lceil \log s \rceil$-bit message. By an argument similar to the one stated above,

$$\begin{aligned} Q_{\epsilon+\delta_3/2}(f) &\leq 1/(s\delta_3)\sum_i q_i \lceil \log w_i \rceil + \lceil \log s \rceil \\ &\leq \lceil \log w \rceil/(s\delta_3) \cdot \sum_i q_i + \lceil \log s \rceil. \end{aligned}$$

This implies that $Q^G_\epsilon(f)$ is lower-bounded by $s\delta_3(Q_{\epsilon+\delta_3/2}(f) - \lceil \log s \rceil)/\lceil \log w \rceil$. □

Remark: To derive a good lower bound, we should be careful to partition a given network into layers so that the values of w and s become appropriately small and large, respectively. However, for some graphs, w may become exponentially large to s by any choice of layers. In such cases, our lower bound is trivial, i.e., essentially the same as the ordinary two party case. Since our lower bound can be applied to any network, this may be inevitable; to find a good application (i.e., network structure) and find a good choice of layers (i.e., w and s) are very important to utilize our lower bound. Indeed, we can derive a nice lower bound in the following section.

If we set $\epsilon, \delta_1, \delta_2, \delta_3$ to constants such that $\epsilon + \delta_1/2 + \delta_2$ and $\epsilon + \delta_3$ are at most some constant less than 1/2, then $Q_\epsilon(f)$, $Q_{\epsilon+\delta_1/2+\delta_2}(f)$ and $Q_{\epsilon+\delta_3/2}(f)$ differ by at most constant multiplicative factors.

Corollary 6: Suppose that f, G, s and w are defined as above. Then, for constant $0 < \epsilon < 1/2$,

$$Q^G_\epsilon(f) = \Omega(s(Q_\epsilon(f) - \log \min\{n, s\})/\log w).$$

If function f is derived from some symmetric function g, we can obtain a more concrete lower bound.

Corollary 7: Suppose that G, s and w are defined as above, $f(x, y)$ $(x, y \in \{0,1\}^n)$ is of the form $f(x, y) = g(|x \wedge y|)$ for any predicate $g : \{0, \ldots, n\} \to \{0, 1\}$. If $l_0(g) \in \{0, 1, \ldots, \lfloor n/2 \rfloor\}$ and $l_1(g) \in \{0, 1, \ldots, \lfloor n/2 \rfloor\}$ are the smallest integers such that $g(h)$ is constant for any integer $h \in \{l_0(g), l_0(g)+1, \ldots, n - l_1(g)\}$. Then, the total quantum communication complexity over all links of computing $f(x, y)$ in the bounded error setting is

$$\Omega(s(\sqrt{nl_0(g)} + l_1(g) - \log \min\{n, s\})/\log w).$$

Proof By Razborov's lower bound $\Omega(\sqrt{nl_0(g)} + l_1(g))$ for such a predicate as shown in [17]. □

For zero-error quantum protocols, we have a similar but slightly different result: $\delta_1/2$ and $\delta_3/2$ are replaced by δ_1 and δ_3, since the protocol must give up if the amount of communication exceeds $1/\delta_1$ $(1/\delta_3)$ times $1/s\sum_i q_i \lceil \log w_i \rceil$ in order to preserve the zero-error property.

Theorem 8: Define f, G, s and w as above. Let $Q^G_{0,\epsilon}(f)$ be the total quantum communication complexity over all links in G of computing Boolean function $f(x, y)$ with error probability 0 and give-up probability at most ϵ $(0 \leq \epsilon < 1)$. Then, $Q^G_{0,\epsilon}(f)$ is at least $\frac{s}{\lceil \log w \rceil}$ times

$$\max\{\delta_1(Q_{\epsilon+\delta_1+\delta_2}(f) - \lceil \log(n/\delta_2^2) \rceil), \delta_3(Q_{\epsilon+\delta_3}(f) - \lceil \log s \rceil)\},$$

where $0 < \delta_1, \delta_2, \delta_3 < 1$ such that $\epsilon + \delta_1 + \delta_2$ and $\epsilon + \delta_3$ are smaller than 1, and $Q_{0,\epsilon}(f)$ denotes the quantum communication complexity of $f(x, y)$ for error probability 0 but give-up probability at most ϵ in the ordinary two party case.

The proof is given in Appendix B.

4. Application: Almost Tight Bound of Distinctness on a Ring

This section applies the lower bound theorem of the previous section to a distributed computing problem, the distinctness problem, which emerges when checking whether the priorities of processors are totally ordered. The distinctness problem $\mathsf{DISTINCT}^G_{k,L}$ was first introduced by Tiwari [18] and is defined as follows.

Definition 1 ($\mathsf{DISTINCT}^G_{k,L}$): Let k parties be placed on a network G. Let each party P_i $(0 \leq i \leq k-1)$ have an integer $x_i \in \{0, \ldots, L-1\}$ $(k \leq L)$. The goal is to decide whether x_i is not equal to x_j for $i \neq j$. At termination, each party knows a one-bit result.

The main theorem of this section gives almost tight bounds of the bounded-error quantum communication complexity for the distinctness problem on a ring-shaped network.

Theorem 9: The quantum communication complexity of $\mathsf{DISTINCT}^{\text{ring}}_{k,L}$ for $L = k + \Omega(k)$ in the bounded error setting is summarized as follows:

- if $L \leq k(\log k)^2$, $O(k\sqrt{L})$ and $\Omega(k\sqrt{k}) \subseteq \Omega(k\frac{\sqrt{L}}{\log k})$.
- if $L > k(\log k)^2$, $O(k(\sqrt{k}\log k + \log\log L))$ and $\Omega(k(\sqrt{k} + \log\log L))$

The theorem implies that our bounds are tight up to a log factor. In particular, they are optimal $\Theta(k\sqrt{k})$ up to a constant factor for $L \in O(k)$. The theorem is directly obtained from the lemmas in the next subsections. Hereafter, we deal with only bounded-error computation.

4.1 Lower Bound

To get a lower bound, we will prove a lower bound of the quantum communication complexity for a certain distributed computing problem by applying Corollary 6, and then reduce the problem to $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$.

Lemma 10: The quantum communication complexity of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ is $\Omega(k(\sqrt{k} + \log\log L))$, for $L = k + \Omega(k)$.

Proof We will reduce the following problem $\mathsf{DISJ}^{\mathrm{ring}}_{k-2\lceil ck\rceil+2,\lceil ck\rceil}$ to $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$: when party P_A is diametrically opposite P_C on a $(k - 2(\lceil ck\rceil - 1))$-party ring for any positive constant c ($\leq 1/4$) (we assume here $2|(k-2\lceil ck\rceil)$ for simplicity, but this assumption is not essential), and $\lceil ck\rceil$-bit strings x and y are given to P_A and P_C, respectively, the goal is to compute function $\mathsf{DISJ}_{\lceil ck\rceil}(x, y) = \bigwedge_{i=1}^{\lceil ck\rceil} \overline{x_i y_i}$ (see Fig. 2). Problem $\mathsf{DISJ}^{\mathrm{ring}}_{k-2\lceil ck\rceil+2,\lceil ck\rceil}$ has the total communication complexity over all links of $\Omega(k\sqrt{k})$ by Corollary 7 with $n = \lceil ck\rceil$, $w = 2$, $s = (k - 2(\lceil ck\rceil - 1))/2 = O(k)$, $l_0(g) = 1$, and $l_1(g) = 0$.

We now reduce $\mathsf{DISJ}^{\mathrm{ring}}_{k-2\lceil ck\rceil+2,\lceil ck\rceil}$ to $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ for any $L \geq k + \lceil ck\rceil$. We first partition the k-party ring of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ into four connected segments A, B, C and D of size $\lceil ck\rceil$, $(k - 2\lceil ck\rceil)/2$, $\lceil ck\rceil$ and $(k - 2\lceil ck\rceil)/2$, respectively, where segment A is diametrically opposite C. Let $I_1 = \{0, 1, \ldots, \lceil ck\rceil - 1\}$, $I_2 = \{\lceil ck\rceil, \ldots, 3\lceil ck\rceil - 1\}$ and $I_3 = \{3\lceil ck\rceil, \ldots, L - 1\}$. Next we construct an instance of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ from any instance of $\mathsf{DISJ}^{\mathrm{ring}}_{k-2\lceil ck\rceil+2,\lceil ck\rceil}$ as follows: (1) the ith party of A (C) has $(i - 1) \in I_1$ as input if the ith bit of input to P_A (resp. P_C) of $\mathsf{DISJ}^{\mathrm{ring}}_{k-2\lceil ck\rceil+2,\lceil ck\rceil}$ is 1 for $i = 1, \ldots, \lceil ck\rceil$, otherwise every party in A and C is given any distinct value in I_2, (2) every party in B and D is given any distinct value in I_3. It is not hard to see that $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ is true if and only if there is no i such that the ith party of A has the same input as the ith party of C. Thus, $\mathsf{DISJ}^{\mathrm{ring}}_{k-2\lceil ck\rceil+2,ck}$ can be solved if P_A and P_C simulate segments A and C, respectively, for the above instance of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$. By setting c to an arbitrary small positive constant, the lemma holds for all $L = k + \Omega(k)$.

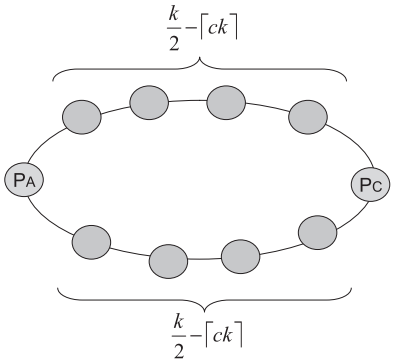

Fig. 2 Problem $\mathrm{DISJ}_{\mathrm{ring}}$.

In the case where $L = 2^{k^{\omega(1)}}$, we will reduce the following problem $\mathsf{NEQ}^{\mathrm{ring}}_{k,\lceil\log L\rceil-1}$ to $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$: when party P_A is diametrically opposite P_C on a k-party ring (we assume again $2|k$ for simplicity, but this assumption is not essential), and $(\lceil\log L\rceil - 1)$-bit strings x and y are given to P_A and P_C, respectively, the goal is to decide whether x does not equal y, i.e., to compute $\mathsf{NEQ}_{\lceil\log L\rceil-1} = \bigvee_{i=1}^{\lceil\log L\rceil-1}(x_i \oplus y_i)$. We apply Corollary 6 to $\mathsf{NEQ}^{\mathrm{ring}}_{k,\lceil\log L\rceil-1}$ with $n = \lceil\log L\rceil - 1$, $w = 2$, $s = O(k)$ and $Q_{1/3}(\mathsf{NEQ}_{\lceil\log L\rceil-1}) = \Omega(\log\log L)$. (This is because NEQ has the same complexity as EQ in the bounded error case, and $Q_{1/3}(\mathsf{EQ}_n) = \Omega(\log n)$ can be derived by the combination of the following two facts: (1) $D(\mathsf{EQ}_n) = \Omega(n)$ by the rank lower bound technique [13], and (2) for any f, $Q_{1/3}(f) > \Omega(\log(D(f))$ [12].) $\mathsf{NEQ}^{\mathrm{ring}}_{k,\lceil\log L\rceil-1}$ has the total communication complexity over all links of $\Omega(k(\log\log L - \log\min\{O(k), O(\log L)\})) = \Omega(k\log\log L)$ when $L = 2^{k^{\omega(1)}}$. Thus, $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ has quantum communication complexity $\Omega(k\log\log L)$.

For the reduction, we construct an instance of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ from any instance of $\mathsf{NEQ}^{\mathrm{ring}}_{k,\lceil\log L\rceil-1}$ as follows. We first partition the k-party ring of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ into four connected segments A, B, C and D of size 1, $(k-2)/2$, 1 and $(k-2)/2$, respectively, where segment A is diametrically opposite C. We then set the most significant bit (MSB) of the input of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ given to each party in segments A and C to 1, while we set the MSBs of the inputs to the other parties to 0. The remaining $(\lceil\log L\rceil - 1)$ bits of the input to the party in segment A (segment C) are set to the input values of $\mathsf{NEQ}^{\mathrm{ring}}_{k,\lceil\log L\rceil-1}$ given to P_A (resp. P_C). The remaining $(\lceil\log L\rceil - 1)$ bits of the input to the other parties are set to distinct values. □

4.2 Upper Bounds

To show the optimality of our lower bound, we show almost matching upper bounds.

Lemma 11: The quantum communication complexity of $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$ is $O(k(\sqrt{k}\log k + \log\log L))$.

Proof

We consider the following search problem: is there any party P_i such that, for some j ($\neq i$), party P_j has the same input as party P_i? Given an oracle that, for input i, answers 1 if there is a party $P_j(\neq P_i)$ that has the same input as party P_i and otherwise answers 0, we can solve the search problem with $O(\sqrt{k})$ queries to the oracle by Grover's quantum search algorithm in [9]. Let party P_0 be distinguished, and she executes the search algorithm on behalf of all the parties. The oracle is simulated in a distributed way by the k parties as follows.

The simulation for input i consists of two phases. The purpose of the first phase is for P_0 to get the information of x_i. If $i \neq 0$, party P_0 first prepares a $(\lceil\log k\rceil + \lceil\log L\rceil)$-qubit message $|i\rangle|0^{\lceil\log L\rceil}\rangle$; otherwise P_0 prepares message

$|i\rangle|x_0\rangle$. Party P_0 then sends it to adjacent party P_1. Every party P_j ($j > 0$) except P_i simply passes the received message to adjacent party $P_{j+1 \pmod k}$; party P_i changes message $|i\rangle|0^{\lceil \log L \rceil}\rangle$ to $|i\rangle|x_i\rangle$ before sending it to adjacent party $P_{i+1 \pmod k}$. The purpose of the second phase is to check whether string x_i is identical to one of the $k-1$ strings $\{x_0, x_2, \ldots, x_{k-1}\} \setminus \{x_i\}$. If $i \neq 0$, party P_0 prepares $(\lceil \log L \rceil + \lceil \log k \rceil)$-qubit message $|x_i\rangle|0^{\lceil \log k \rceil}\rangle$; otherwise it prepares message $|x_i\rangle|0^{\lceil \log k \rceil - 1}1\rangle$. Notice that the second register is used to count the number of parties that have values identical to x_i. Party P_0 then sends it to adjacent party P_1. For $j > 0$, P_j just passes the received message to adjacent party $P_{j+1 \pmod k}$ if $x_j \neq x_i$; otherwise party P_j increments the counter, i.e., the contents of the last $\lceil \log k \rceil$ qubits, in the message, and sends it to adjacent party $P_{j+1 \pmod k}$. When the message arrives at P_0, the counter has value of at least two if and only if there are at least two parties that have values identical to x_i. Party P_0 then sets the content of a fresh qubit to 1 if the value of the counter is at least two; otherwise, P_0 sets it to 0. The content of the fresh qubit is the answer of the oracle. Finally, every computation (except the last step) and communication performed in the first and second phases is inverted to disentangle all work qubits including the message qubits.

The first and the second phases including their inversions have the communication complexity of $O(k \log(kL))$, implying that one query needs $O(k \log(kL))$.

By combining Grover's search algorithm with this distributed oracle, $O(k\sqrt{k}\log(kL))$-qubit communication is sufficient to find any party P_i such that there exists party P_j ($j \neq i$) that has the same input as party P_i. If such a party is found, the answer to $\mathsf{DISTINCT}_{k,L}^{\text{ring}}$ is false; otherwise the answer is true. (To inform every party of the answer, a one-bit message needs to be passed around the ring, which does not change the order of complexity.) This complexity is tight up to a log multiplicative factor when L is polynomial in k. For larger L, however, it is not tight. In what follows, we will show more efficient algorithm for larger L by adding a preprocess before running the above algorithm.

The idea is to map $\lceil \log L \rceil$-bit inputs to $3 \log k$-bit strings by using universal hashing (see [14] or Appendix A) and classical public randomness, and then apply the above algorithm of complexity $O(k\sqrt{k}\log(kL))$ with the $3 \log k$-bit strings as input.

Suppose that every party shares classical public coins, By using the public coins, every party selects a common hash function $f : \{0, \ldots, L-1\} \mapsto \{0, \ldots, k^3 - 1\}$ from the family of $O(L^2)$ hash functions. Every party sets its new input to the $3 \log k$-bit string, and runs the $O(k\sqrt{k}\log(kL))$ algorithm for the new input, yielding complexity $O(k\sqrt{k}\log k)$. By Lemma 2 with input size $k\lceil \log L \rceil$, $O(k \log(k \log L))$-bit classical communication is sufficient to realize public coins. Thus, the total communication complexity is $O(k \log(k \log L) + k\sqrt{k}\log k) = O(k(\sqrt{k}\log k + \log\log L)$.

The correctness of this algorithm is proved as follows. When there are a pair of parties that share a common value, the step of applying Grover's search obviously finds one of the parties with bounded error. If there is no such pair, the probability that a certain pair of parties share a common value of the hash function is at most $1/k^3 \times k(k-1)/2 \leq 1/(2k)$. Thus, in this case, the algorithm also guarantees bounded error.

□

In the case of $L < k(\log k)^2$, we can obtain a better bound, which is optimal for $L = O(k)$.

Lemma 12: The quantum communication complexity of $\mathsf{DISTINCT}_{k,L}^{\text{ring}}$ is $O(k\sqrt{L})$.

Proof We consider the following search problem:

Let k parties be placed on a ring, and let each party P_i ($0 \leq i \leq k-1$) have an integer $x_i \in \{0, \ldots, L-1\}$ ($k \leq L$). Is there any element $x \in \{0, \ldots, L-1\}$ such that at least two parties have x?

We employ the protocol due to Aaronson and Ambainis [1] for the two-party disjointness function, which has $O(\sqrt{L})$ communication complexity for $2L$-bit input. In their protocol, Alice and Bob collaborate to recursively execute the amplitude amplification algorithm in a distributed way. By using their technique, we can solve the search problem as follows.

Let party P_0, say Alice, be distinguished, and she executes the recursive amplitude amplification algorithm due to Aaronson and Ambainis on behalf of all parties. Suppose that there is at most one solution, i.e., there is at most one element $x \in \{0, \ldots, L-1\}$ such that at least two parties have x as inputs.

We consider the following algorithm $\mathcal{A}$, which takes $S \subseteq \{0, \ldots, L-1\}$ as an input and finds an element $x \in S$ such that at least two parties have x. Alice divides the search space S into $|S|^{1/5}$ subspaces. She picks up a subspace S_i ($0 \leq i \leq |S|^{1/5} - 1$) uniformly at random, and announces the index of the chosen subspace to all other parties. She then recursively searches the subspace by calling $\mathcal{A}(S_i)$. She applies Theorem 4 to this algorithm m times for the smallest integer m such that $2m + 1 \geq |S|^{1/11}$. At the bottom level of the recursion, where the size of the search space is a constant, Alice solves the problem classically. As we will show in the next paragraph, this solves the problem with probability $\Omega(|S|^{-1/11})$. In other words, $\mathcal{A}(\{0, \ldots, L-1\})$ solves $\mathsf{DISTINCT}_{k,L}^{\text{ring}}$ with probability $\Omega(L^{-1/11})$. Thus, by applying Theorem 4 to $\mathcal{A}(\{0, \ldots, L-1\})$ $O(L^{1/22})$ times, the success probability is boosted to $\Omega(1)$. The pseudo-code is shown in Fig. 3, which consists of a main algorithm and several sub-algorithms.

We now show the reason why the success probability of $\mathcal{A}(S)$ is $\Omega(|S|^{-1/11})$. Let $\mathbf{Pr}(t)$ be the success probability of this algorithm for search space of size t. Note that the success probability before applying quantum amplitude amplification in the algorithm is $t^{-1/5}\mathbf{Pr}(t^{4/5})$. By the property of quantum amplitude amplification, $\mathbf{Pr}(t) = (\sin((2m+1)\theta_a))^2$ for $m = \Theta(t^{1/11})$ and $(\sin\theta_a)^2 = a = t^{-1/5}\mathbf{Pr}(t^{4/5})$. This

```
main()
{
   return(Amplify (O(L^{1/22}), A({0, ..., L − 1}))
}

A(S)
{
   if (|S| is a constant)
   {
      Find x ∈ S such that at least two parties have x by classically communicating with each other, and return the result.
   }
   else
   {
      for (i = 0 to |S|^{1/5} − 1)
      {
         S_i := {S_{i·|S|^{4/5}}, ..., S_{(i+1)·|S|^{4/5}−1}} // Divide S into |S|^{1/5} subsets.
      }
      m := (the smallest integer such that 2m + 1 ≥ |S|^{1/11}.)
      return (Amplify (m, A_sub(S)))
   }
}

A_sub(S)
{
   Pick up i (0 ≤ i < |S|^{1/5}) at random, and announce it to all other parties.
   return(A(S_i))
}

Amplify(k, A)
{
   Amplify algorithm A k times, and return the result.
}
```

Fig. 3 Pseudo-code of the algorithm for $\mathsf{DISTINCT}^{\mathrm{ring}}_{k,L}$.

yields $\mathbf{Pr}(t) = \Omega(t^{-1/11})$ by using $(\sin((2m+1)\theta_a))^2 > ((2m+1)\sqrt{a})^2(1 - ((2m+1)\sqrt{a})^2/3)$ (see also [1]).

Our algorithm performs these operations in a superposition, and maintains the state of the whole system

$$\sum_I \alpha_I |I\rangle^{\otimes k} |z_I\rangle |garbage\rangle,$$

where I means the index of the chosen subspace, which is possessed by every party (thus, this part is a k-tensor product), and z_I is the content of register $\mathbf{Z}$ possessed by Alice, in which the search result for the corresponding space is stored at the bottom level.

To realize F_χ, no communication is needed, since Alice has register $\mathbf{Z}$ and already knows the search result for each subspace. To realize F_0, $O(k)$ bit communication is sufficient; Alice sends one bit message $|0\rangle$, and every party including Alice, upon receiving the message, computes the OR of the contents of the received message and all his quantum registers at the current or deeper recursion levels, and sends the one-bit result to the next party (if he/she is not Alice). At each level of the recursion, Alice announces the index of the chosen subspace to all other parties. For this, $O(k \cdot \log |S|^{1/5})$-qubit communication is sufficient by sending a message of $\lceil \log |S|^{1/5} \rceil$ bits around the ring. At the bottom level of the recursion, they communicate with each other to solve the search problem over a constant-sized search space $\{i, i+1, \ldots, i+j\}$ for some constant j; Alice finally sets the content of register $\mathbf{Z}$ to the search result. This can be solved as follows. For each value in $\{i, i+1, \ldots, i+j\}$, a message consisting of two parts of $\log j$ bits and 2 bits, respectively, is sent around the ring. The first part specifies the value in $\{i, i+1, \ldots, i+j\}$ to be checked. The second part is a counter that is incremented up to two by the parties having the value specified by the first part (if the counter already has value of two, it is never incremented). If this computation is done in a superposition over all constant-sized subspaces, it may leave some garbage at some parties; but this garbage need not be removed, since amplitude amplification $(-\mathcal{A}F_0\mathcal{A}^{-1}F_\chi)^m\mathcal{A}|0\rangle$ already includes the inverting operation of $\mathcal{A}$. Hence, this needs the communication of $O(k)$ qubits. Since amplitude amplification performs $\mathcal{A}$ and $\mathcal{A}^{-1}$ $(2m+1)$ times, and F_0 and F_χ m times for each, we obtain a recurrence with respect to the number of qubits ($C(L)$) to be communicated:

$$C(L) \le (2m_L + 1)\left(C\left(L^{4/5}\right) + k\left\lceil \log\left(L^{1/5}\right)\right\rceil\right) + m_L \cdot O(k),$$

where m_L is the smallest integer such that $2m_L + 1 \ge L^{1/11}$. This resolves to $C(L) = O(k \cdot L^{5/11})$. Since the success probability is $\Omega(L^{-1/11})$, amplifying $O(L^{1/22})$ times boosts it to $\Omega(1)$. Thus the total communication complexity is $O(k\sqrt{L})$.

If there can be two or more solutions, we can reduce this case to the unique solution case as in [1]. The complexity is still $O(k\sqrt{L})$. □

Acknowledgements

This work was supported in part by KAKENHI (16092218), (18700011), (19700010), (19700019).

References

[1] S. Aaronson and A. Ambainis, "Quantum search of spatial regions," Proc. 44th Annual IEEE Symposium on Foundations of Computer Science, pp.200–209, 2003. (Journal version Theory of Computing, vol.1, pp.47–79, 2005.)

[2] Z. Bar-Yossef, T.S. Jayram, and I. Kerenidis, "Exponential separation of quantum and classical one-way communication complexity," Proc. 36th Annual ACM Symposium on Theory of Computing, pp.128–137, 2004.

[3] M. Boyer, G. Brassard, P. Høyer, and A. Tapp, "Tight bounds on quantum searching," Fortschritte Der Physik, vol.46, no.4-5, pp.493–505, 1998.

[4] G. Brassard, P. Høyer, M. Mosca, and A. Tapp, "Quantum amplitude amplification and estimation," in Quantum Computation and Quantum Information: A Millenium Volume, ed. S.J. Lomonaco, Jr. and H.E. Brandt, vol.305 of AMS Contemporary Math. Series, pp.53–74, American Mathematical Society, 2003.

[5] H. Buhrman, R. Cleve, and A. Wigderson, "Quantum vs. classical communication and computation," Proc. 30th Annual ACM Symposium on Theory of Computing, pp.63–68, 1998.

[6] H. Buhrman, R. Cleve, J. Watrous, and R. de Wolf, "Quantum fingerprinting," Phys. Rev. Lett., vol.87, no.16, 167902, 2001.

[7] P. Frankl and V. Rödl, "Forbidden intersections," Trans. Amer. Math. Soc., vol.300, no.1, pp.259–286, 1987.

[8] D. Gavinsky, J. Kempe, I. Kerenidis, R. Raz, and R. de Wolf, "Exponential separations for one-way quantum communication complexity, with applications to cryptography," Proc. 39th Annual ACM

Symposium on Theory of Computing, pp.516–525, 2007.
[9] L.K. Grover, "A fast quantum mechanical algorithm for database search," Proc. 28th Annual ACM Symposium on Theory of Computing, pp.212–219, 1996.
[10] B. Kalyanasundaram and G. Schnitger, "The probabilistic communication complexity of set intersection," SIAM J. Discrete Math., vol.5, no.4, pp.545–557, 1992.
[11] H. Klauck, "On quantum and approximate privacy," Theory of Computing Systems, vol.37, pp.221–246, 2004.
[12] I. Kremer, Quantum communication, Master's Thesis, Hebrew University, Computer Science Department, 1995.
[13] E. Kushilevitz and N. Nisan, Communication Complexity, Cambridge University Press, 1997.
[14] C.E. Leiserson, R.L. Rivest, C. Stein, and T.H. Cormen, Introduction to Algorithms, MIT Press, 2001.
[15] I. Newman, "Private vs. common random bits in communication complexity," Inf. Process. Lett., vol.39, pp.67–71, 1991.
[16] R. Raz, "Exponential separation of quantum and classical communication complexity," Proc. 31st Annual ACM Symposium on Theory of Computing, pp.358–367, 1999.
[17] A.A. Razborov, "Quantum communication complexity of symmetric predicates," Izvestiya Mathematics, vol.67, no.1, pp.145–159, 2003.
[18] P. Tiwari, "Lower bounds on communication complexity in distributed computer networks," J. ACM, vol.34, no.4, pp.921–938, 1987.
[19] A.C.-C. Yao, "Some complexity questions related to distributed computing," Proc. 11th Annual ACM Symposium on Theory of Computing, pp.209–213, 1979.
[20] A.C.-C. Yao, "Quantum circuit complexity," Proc. 34th Annual IEEE Symposium on Foundations of Computer Science, pp.352–361, 1993.

Appendix A: Universal Hashing

For prime p and positive integer m such that $p > m$, define the hash function $h_{a,b}$ that maps $U = \{0, \ldots, p-1\}$ to $\{0, \ldots, m-1\}$ as follows: For any $a \in Z_p^* := \{1, 2, \ldots, p-1\}$ and any $b \in Z_p := \{0, 1, 2, \ldots, p-1\}$,

$$h_{a,b}(k) := ((ak + b) \bmod p) \bmod m.$$

Then, the class $\mathcal{H}_{p,m} := \{h_{a,b} : a \in Z_p^* \text{ and } b \in Z_p\}$ of hash functions $h_{a,b}$ is universal. In other words, for each pair of distinct keys $k, l \in U$, the number of hash functions $h_{a,b} \in \mathcal{H}_{p,m}$ for which $h_{a,b}(k) = h_{a,b}(l)$ is at most $|\mathcal{H}|/m$.

Appendix B: Proof of Theorem 8

Proof We first partition network G into $(s+1)$ layers as in the case of Lemma 5. Suppose that $\mathcal{P}$, the best quantum protocol between P_A and P_B on network G, computes $f(x, y)$ with give-up probability at most ϵ.

We then construct a zero-error quantum protocol with classical public coins between two parties, P'_A and P'_B, that are directly connected to each other by simulating protocol $\mathcal{P}$ as in the proof of Lemma 5. The obtained protocol between P'_A and P'_B computes f with give-up probability at most ϵ and with expected communication complexity $(1/s \sum_i q_i \lceil \log w_i \rceil)$. To guarantee the worst case communication complexity, we modify this protocol so that if the amount of communication exceeds $1/\delta_1$ times $(1/s \sum_i q_i \lceil \log w_i \rceil)$ for $0 < \delta_1 < 1$, it gives up. The probability of this event is at most δ_1 by Markov's inequality. Thus, the modified protocol gives up with probability at most $\epsilon + \delta_1$. Hence, we have obtained a zero-error quantum protocol with classical public coins, whose complexity is $1/(s\delta_1) \sum_i q_i \lceil \log w_i \rceil \le \lceil \log w \rceil / (s\delta_1) \cdot \sum_i q_i$. The last equality is due to $w_i \le w$. This implies that $Q^{pub}_{0,\epsilon+\delta_1}(f) \le \lceil \log w \rceil / (s\delta_1) \cdot \sum_i q_i$. Amount $\sum_i q_i$ is the total number of qubits communicated by protocol $\mathcal{P}$, $Q^G_{0,\epsilon}(f)$, which is lower-bounded by $s\delta_1 Q^{pub}_{0,\epsilon+\delta_1}(f)/\lceil \log w \rceil$. By applying Lemma 3 to $Q^{pub}_{0,\epsilon+\delta_1}(f)$, we have

$$Q^G_{0,\epsilon}(f) \ge s\delta_1 (Q_{0,\epsilon+\delta_1+\delta_2}(f) - \lceil \log n + 2 \log 1/\delta_2 \rceil)/\lceil \log w \rceil.$$

There is a simpler way of deciding the boundary between P'_A's part and P'_B's part: P'_A randomly chooses one layer-boundary out of the s layer-boundaries and informs P'_B of the chosen layer-boundary by sending a $\lceil \log s \rceil$-bit message. By an argument similar to the one stated above, $Q_{0,\epsilon+\delta_3}(f) \le 1/(s\delta_3) \sum_i q_i \lceil \log w_i \rceil + \lceil \log s \rceil \le \lceil \log w \rceil / (s\delta_3) \cdot \sum_i q_i + \lceil \log s \rceil$. This implies that $Q^G_{0,\epsilon}(f)$ is lower-bounded by $s\delta_3 (Q_{0,\epsilon+\delta_3}(f) - \lceil \log s \rceil)/\lceil \log w \rceil$. □

Seiichiro Tani is a research scientist of NTT Communication Science Laboratories, NTT Corporation. He received B.E. degree in information science from Kyoto University, Kyoto, Japan, in 1993, and M.S. and Ph.D. degrees in computer science from the University of Tokyo, Tokyo, Japan, in 1995 and 2006, respectively. In 1995, he joined NTT corporation. From 2004, he has also been a researcher at Quantum Computation and Information Project, ERATO (ERATO-SORST from 2005), Japan Science and Technology Agency. He received the 2000 IEICE Information and Systems Society Best Paper Award. His current interests include quantum algorithms and quantum complexity theory. He is a member of ACM, IEEE and IPSJ.

Masaki Nakanishi received the B.E., M.E. and Ph.D. degrees from Osaka University, Japan, in 1996, 1998 and 2002, respectively. He is currently with the Graduate School of Information Science, Nara Institute of Science and Technology, as an assistant professor. His current interests include quantum computing and design of combinatorial algorithms.

Shigeru Yamashita is an associate professor of Graduate School of Information Science, Nara Institute of Science and Technology. He received his B.E., M.E. and Ph.D. degrees in information science from Kyoto University, Kyoto, Japan, in 1993, 1995 and 2001, respectively. In 1995, he joined NTT Communication Science Laboratories, where he engaged in research of computer aided design of digital systems and new type of computer architectures. During 2000 to 2005, he was also a researcher at Quantum Computation and Information, ERATO, Japan Science and Technology Agency. He received the 2000 IEEE Circuits and Systems Society Transactions on Computer-Aided Design of Integrated Circuits and Systems Best Paper Award. He is a member of IPSJ.