Vol.2009-AL-126 No.6
2009/9/15

Quantum Communication Protocols with Public Coins

Seiichiro Tani,[†1] Masaki Nakanishi [†2]
and Shigeru Yamashita[†3]

This paper studies the model of *quantum protocols with classical public coins*, and shows its application to quantum communication complexity of some functions. The paper first proves, by carefully combining quantum Grover search with the concept of quantum protocols with classical public coins, that the quantum communication complexity for $\mathbf{LNE}(l,k)$ is $O(\sqrt{l}\log l + \log k)$, where $\mathbf{LNE}(l,k)$ is an lk-bit total Boolean function called the *list-nonequality function.* The function is a generalization of the equality function and the disjoint function. The above bound gives some separation between quantum and classical communication complexity for a total function, since the classical randomized communication complexity for the same function is $\Theta(l + \log k)$. As a multi-party version of the list-nonequality function, the distinctness problem is considered. The goal of this problem is to decide whether or not there are two parties that have the same inputs. The sub-optimal bound of the communication complexity of the problem is given via the model of quantum protocols with classical public coins.

1. Introduction

Studying *communication complexity* has been one of the central issues in computer science since it was introduced by Yao[21]. Not only it is interesting in its own right, but it also has many applications such as analyzing VLSI circuit design, data structures and networks (See the book[16]for more details).

Informally, the communication complexity of function $f(x,y) : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ is the minimum amount of communication bits sent between two parties, say, Alice and Bob, who get inputs $x \in \{0,1\}^n$ and $y \in \{0,1\}^n$, respectively, and compute f cooperatively by using an optimal protocol. Here the local computation time required for Alice and Bob may be large. For example, Alice first performs local computation depending on her input and sends a message to Bob. Bob then does some local computation depending on his input and the received message, and sends a message back to Alice. This sequence 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 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. Randomized protocol $\mathcal{P}$ computes f with (two-sided) bounded error if for every input (x,y) the probability that the output of $\mathcal{P}$ is not equal to $f(x,y)$ is at most $1/3$ over the random choices of r_A and r_B. The cost of a bounded-error randomized protocol is the worst-case cost over all inputs and all random strings. $R_2(f)$ denotes the minimum worst-case cost over all randomized protocols that compute f with bounded error. In another randomized setting, Alice and Bob are allowed to access public coins (*or* a common random string). Formally, the output of protocol $\mathcal{P}$ depends on inputs and common random string r. $R_2^{pub}(f)$ is the minimum worst-case cost over all randomized protocols that compute f by using public coins with bounded error. Note that, if Alice and Bob have public coins, they can use disjoint subsets of the coins as their local coins: $R_2^{pub}(f) \leq R_2(f)$.

Quantum communication complexity, introduced by Yao[22], 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, $Q_E(f)$ and $Q_2(f)$ are defined as the quantum counterparts of $D(f)$ and $R_2(f)$, respectively. In particular, the quantum counter part 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 that have gaps between quantum and classical communication complexity. For exact computation, Buhrman et al.[6] proved that for a specific promise version of the equality function $\mathbf{EQ}'_n$, $Q_E(\mathbf{EQ}'_n) = O(\log n)$ while $D(\mathbf{EQ}'_n) \in \Omega(n)$[10]. In the bounded-error case, Raz[18] showed a promise problem that has an exponential gap between quantum and classi-

[†1] NTT Communication Science Laboratories, NTT Corporation.
[†2] Faculty of Education, Art and Science, Yamagata University.
[†3] College of Information Science and Engineering, Ritsumeikan University.

cal settings, i.e., $Q_2(f) = O(\log n)$ and $R_2(f) = \Omega(n^{1/4}/\log n)$. As for total functions, the largest known gap is quadratic: $Q_2(\mathbf{DISJ}_n) = O(\sqrt{n})$[1] and $R_2(\mathbf{DISJ}_n) = \Omega(n)$[13], where $\mathbf{DISJ}_n$ is the $2n$-bit disjoint function, $\bigwedge_{i=1}^{n}(\overline{x_i y_i})$. In a restricted model or other models, some exponential gaps have been proved: the one-way bounded-error model[2] and bounded-error simultaneous message-passing model[5],[23].

With regard to classical communication complexity, protocols with public coins are often discussed since they can always be converted to ones without public coins by using *public-private coin conversion*[17]: if there is a bounded error protocol that uses (possibly) many public coins, a bounded error protocol that does not use public coins exists with only $O(\log n)$ additional communication bits, where n is the input length. Thus, in the classical communication protocol, a standard technique is to work out a public-coin protocol first and then convert it to one without public coins. This often provides us with many nice communication protocols.

In the quantum setting, the natural counter part of public coins may be *prior entanglement*. Many papers[4],[7],[8] have shown how to reduce communication bits by using prior entanglement. However, unlike the classical case, it seems to be hopeless to convert (possibly) many prior entanglement resources to only, say, $O(\log n)$ additional communication (qu)bits. Thus, we have no standard technique unlike the classical case mentioned above. Instead, we may have the following technique: first we consider what we call "*quantum protocols with classical public coins*," which is a quantum protocol where Alice and Bob are allowed to access classical correlated randomness (not quantum prior entanglement). This protocol model was studied in the literature[14] in the context of privacy and secure quantum communication.

This paper studies a model of quantum protocols with classical public coins in the context of communication complexity, and shows some interesting applications. More concretely, we first consider the list-nonequality function with $2lk$ variables; the function consists of l instances of the function $\mathbf{EQ}_k$ with $2k$ variables and is false if and only if at least one of the l instances of $\mathbf{EQ}_k$ is true. The functions can be a generalization of two well-studied functions $\mathbf{EQ}_k$ and $\mathbf{DISJ}_l$, which have quite different properties. $\mathbf{EQ}_k$ can efficiently be computed in the classical and quantum settings, implying no gap exists between the classical and quantum communication complexities, whereas there is a quadratic gap between those for $\mathbf{DISJ}_l$, which is the maximum gap known for total functions. Thus, it is interesting to consider the communication complexity of $\mathbf{LNE}(l,k)$, a mixture of $\mathbf{EQ}_k$ and $\mathbf{DISJ}_l$. We show that $Q_2(\mathbf{LNE}(l,k)) = O(\sqrt{l}\log l + \log k)$. Our technique is to carefully combine the concept of quantum protocols with classical public coins, with Grover search. This gives some separation between quantum and classical communication complexity, since $R_2(\mathbf{LNE}(l,k)) = \Theta(l + \log k)$[16]. It should be noted that this may not be achieved without using the notion of quantum protocols with classical public coins.

For the multi-party case, we consider the Distinctness problem: given that every party gets a value drawn from a fixed range, the goal of the problem is to decide whether or not there are two parties who have the same input. For instance, this problem needs to be solved when every party on a network wants to check if the priorities of all parties are totally ordered. From a theoretical point of view, this problem can be considered as a multi-party version of the list non-equality function in the two-party case, in the sense that the latter is reducible to the former. We again apply the concept of quantum protocols with classical public coins to the Distinctness problem on a ring and give a sub-optimal upper bound of its quantum communication complexity. Furthermore, we show that a modification of the algorithm can solve a more general problem, called the Max-Coalition problem, of computing the maximum number of parties that have the same input.

2. Preliminaries

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*. $Q_\epsilon^{pub}(f)$ 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[17], 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, the proof needs to be modified to handle quantum errors. The next proposition is used in the proof.

Proposition 1 (Hoeffding inequality[16]) 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 2 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 3 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.

2.2 Quantum amplitude amplification

We quote the quantum amplitude amplification theorem by Brassard et al.[3], which we will use in our proofs.

Theorem 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.

This theorem can be considered as a generalization of Grover's search algorithm[11]. In what follows, we may say Grover search (algorithm) to mean this theorem.

3. List-Nonequality Function

We consider the list-nonequality function $\mathbf{LNE}(l, k)$: the negation of $\mathbf{EQ}_k^{\vee l}$, where $f^{\vee l} = \bigvee_{i=1}^{l} f(x^i, y^i)$ and $\mathbf{EQ}_k : \{0,1\}^k \times \{0,1\}^k \to \{0,1\}$ is $\mathtt{true}$ if and only if two inputs $x, y \in \{0,1\}^k$ are identical. More precisely,

$$\mathbf{LNE}(l,k) = \bigwedge_{i=1}^{l} \bigvee_{j=1}^{k} (x_i[j] \oplus y_i[j]) = \neg(\bigvee_{i=1}^{l} \bigwedge_{j=1}^{k} \neg(x_i[j] \oplus y_i[j])),$$

where $x_i, y_i \in \{0,1\}^k$ and $x_i[j]$ and $y_i[j]$ are the jth bit of x_i and y_i, respectively. When Alice and Bob compute $\mathbf{LNE}(l,k)$, suppose that Alice is given $x_1, \ldots, x_l$ and Bob is given $y_1, \ldots, y_l$ for $x_i, y_i \in \{0,1\}^k$.

We first show the algorithm for initialization.

Algorithm INIT(t)

(1) Alice generates $\frac{1}{\sqrt{t}} \sum_{i=0}^{t-1} |i\rangle|i\rangle$ in $\lceil \log t \rceil$ qubit registers $\mathbf{R}_A$ and $\mathbf{R}_{A'}$.

(2) Alice sends the qubits in $\mathbf{R}_{A'}$ to Bob.
Bob receives the qubits and puts them into a $\lceil \log t \rceil$-qubit register $\mathbf{R}_B$.

(3) Alice and Bob each generate a string $s(i)$, in registers S_A and S_B, for the content of $\mathbf{R}_A$ and $\mathbf{R}_B$, respectively, to make the entire state

$$\frac{1}{\sqrt{t}} \sum_{i=0}^{t-1} |i\rangle|s(i)\rangle|i\rangle|s(i)\rangle.$$

Then we perform the following algorithm that computes $\neg\mathbf{LNE}(l,k)$ provided that a randomized string $r = s(i)$ shared by Alice and Bob. The algorithm can be obtained by converting Grover Search algorithm to a quantum communication protocol with the technique in 6).

Algorithm $\neg\mathbf{LNE}(l,k)$

(1) Alice generates $\frac{1}{\sqrt{l}} \sum_{j=1}^{l} |j\rangle$.

(2) Alice searches j such that $x_j = y_j$ by using Grover Search with $O(\sqrt{l})$ calls of the following subroutine which simulates an oracle.

Subroutine Oracle

(1) Alice performs the transformation $|j\rangle|0\rangle|0\rangle \to |j\rangle|f(x_j)\rangle|0\rangle$, and sends all the qubits to Bob, where $f(x_j)$ is a fingerprinting of x_j generated with r, namely, $f\colon \{0,1\}^k \to \{0,1\}^w$, where $w = O(\log l)$, maps x_j to $(x_j \oplus r_{1,j}) \cdots (x_j \oplus r_{w,j})$, where $r_{i,j}$ is a disjoint substring of r.

(2) Bob performs $|j\rangle|f(x_j)\rangle \to |j\rangle|f(x_j)\rangle|\mathrm{guess}[x_j = y_j]\rangle$, where guess$[x_j = y_j]$ is true if and only if $f(x_j)$ is equal to $f(y_j)$ (notice that guess$[x_j = y_j]$ is always true if $x_j = y_j$, and false with probability at least $1 - 1/l^2$ otherwise). Bob sends all the qubits back to Alice.

(3) Alice performs $|j\rangle|f(x_j)\rangle|\mathrm{guess}[x_j = y_j]\rangle \to |j\rangle|0\rangle|\mathrm{guess}[x_j = y_j]\rangle$.

Theorem 5 There is a quantum protocol that can compute $\mathbf{LNE}(l,k)$ with probability at least some constant with quantum communication complexity $O(\sqrt{l}\log l + \log k)$.

Proofsketch We first consider the algorithm for computing $\neg\mathbf{LNE}(l,k)$. Notice that for each basis state of each oracle call guess$[x_j = y_j]$ with error probability at most $1/l^2$. Thus, the accumulated error is at most $O(\frac{\sqrt{l}l}{l^2}) = O(1/\sqrt{l})$. Thus, together with Theorem 4, $\neg\mathbf{LNE}(l,k)$ can be computed with bounded error if public randomness is provided.

Algorithm INIT essentially has the two parties share a string sampled from t strings. Lemma 2 says there is a certain set of t strings for $t = O(lk)$ such that the above sampling decreases the success probability by at most some constant.

Therefore, the overall success probability is at least some constant and the total complexity is $O(\sqrt{l}\log l + \log k)$. ■

Remark 6 The protocol in Theorem 5 uses only pure states, although it essentially uses public coins. Hence, a divide-and-conquer approach with respect to l can somewhat improve the upper bound as in 12).

Theorem 7 The quantum query complexity of $\mathbf{LNE}(l,k))$ is $\Omega(\sqrt{l} + \log k)$, while the randomized query complexity is $R_2(\mathbf{LNE}(l,k)) = \Omega(l + \log k)$.

Proof We first reduce the equality function $\mathbf{EQ}_k$ to $\mathbf{LNE}(l,k)$. We associate an $\mathbf{EQ}_k$ instance $x, y \in \{0,1\}^k$ with an $\mathbf{LNE}(l,k)$ instance x_i, y_i for $i = 1, \ldots, l$ such that $x_1 = \cdots = x_l = x$ and $y_1 = \cdots = y_l = y$. It is easy to see that $\mathbf{LNE}(l,k)$ is false if and only if $\mathbf{EQ}_k$ is true.

We then reduce the function $\mathbf{INT}_l$ to $\mathbf{LNE}(l,k)$, where $\mathbf{INT}_l$ is the negation of $\mathbf{DISJ}_l$. We associate an $\mathbf{INT}_l$ instance $x', y' \in \{0,1\}^l$ with an $\mathbf{LNE}(l,k)$ instance x_i, y_i for $i = 1, \ldots, l$ such that

- $x_i = 110^{l-2}$ if $x'_i = 1$
- $x_i = 010^{l-2}$ if $x'_i = 0$
- $y_i = 110^{l-2}$ if $y'_i = 1$
- $y_i = 100^{l-2}$ if $y'_i = 0$.

We can see that $\mathbf{LNE}(l,k)$ is false if and only if $\mathbf{INT}_l$ is true. Notice that $Q_2(\mathbf{EQ}_k) = \Omega(\log k)$ can be derived by combining the following two facts: (1) $D(\mathbf{EQ}_k) = \Omega(k)$ by the rank lower bound technique[16], and (2) for any f, $Q_2(f) > \Omega(\log(D(f))$[15]. This fact and $Q_2(\mathbf{INT}_l) = \Omega(\sqrt{l})$[19] imply the quantum lower bound. Similarly, the randomized lower bound follows from $R_2(\mathbf{EQ}_k) \geq Q_2(\mathbf{EQ}_k) = \Omega(\log k)$ and $R_2(\mathbf{INT}_l) = \Omega(l)$[13]. ■

4. Distinctness and Max-Coalition

The distinctness problem $\mathbf{Distinctness}^{G}_{k,L}$ was first introduced by Tiwari[20] and is defined as follows.

Definition 8 ($\mathbf{Distinctness}^{G}_{k,L}$) 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 any i,j $(i \neq j)$. At termination, each party knows a one-bit result.

Theorem 9 (Lower Bound) For $L \geq k$, the quantum communication complexity of $\mathbf{Distinctness}^{\rm ring}_{k,L}$ is $\Omega(k(\sqrt{k} + \log\log L))$.

Proofsketch We will reduce $\neg\mathbf{LNE}(ck, \lceil \log L \rceil - \lceil \log(ck) \rceil)$ with an arbitrary small constant $c < 1$ to $\mathbf{Distinctness}^{\rm ring}_{k,L}$.

We first partition the k-party ring of $\mathbf{Distinctness}^{\rm ring}_{k,L}$ 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. Next, we construct an instance of $\mathbf{Distinctness}^{\rm ring}_{k,L}$ from any instance $(x_1,\ldots,x_l,y_1,\ldots,y_l)$, where $l = \lceil \log L \rceil - \lceil \log(ck) \rceil$, of $\neg\mathbf{LNE}(ck, \lceil \log L \rceil - \lceil \log(ck) \rceil)$ in the following way: (1) The input of the ith party of A is set to $x_i \circ (i)_2$, where $(i)_2$ is the binary expression of i and "$\circ$" denotes concatenation;(2) the input of the ith party of C is set to $y_i \circ (i)_2$. Then, Alice and Bob can compute $\neg\mathbf{LNE}(ck, \lceil \log L \rceil - \lceil \log(ck) \rceil)$ by using any protocol $\mathcal{P}$ that computes $\mathbf{Distinctness}^{\rm ring}_{k,L}$ as follows.

(1) Alice generates a random $\log((k-ck)/2)$-bit string r.
(2) Alice sends r to Bob.
(3) Alice and Bob construct an instance of $\mathbf{Distinctness}^{\rm ring}_{k,L}$ from the given instance of $\neg\mathbf{LNE}(ck, \log L - \log(ck))$ in the way described above.
(4) Alice and Bob simulate $\mathcal{P}$ on the instance in the following way: Alice simulates segment A and all the parties that are away by distance at most r from segment A and Bob simulates the rest.

Thus, the expected quantum communication complexity of $\neg\mathbf{LNE}(ck, \lceil \log L \rceil - \lceil \log(ck) \rceil)$ over all strings r is at most $\frac{1}{(k-2\lceil ck \rceil)/2+1} Q_2(\mathbf{Distinctness}^{\rm ring}_{k,L})$ plus $O(\log k)$. By Markov's inequality, the worst case quantum communication complexity is equal to the expected quantum communication complexity up to a constant factor. Therefore, the theorem follows from Theorem 7. ■

Theorem 10 (Upper Bound) For $L \geq k$, the quantum communication complexity of $\mathbf{Distinctness}^{\rm ring}_{k,L}$ is $O(k(\sqrt{k}\log k + \log\log L))$.

Proofsketch 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 query 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. Let party P_0 be distinguished, and she executes the search algorithm on behalf of all the parties. We assume that all parties share coins, i.e., a random string of sufficient length, and we will remove this shared random string later by using Lemma 3.

The oracle is simulated in a distributed way by the k parties as follows. The simulation for query i consists of two phases. The purpose of the first phase is for P_0 to get a t-bit information on x_i, namely,

$$\tilde{x}_i = (x_i \oplus r_1) \cdots (x_i \oplus r_t),$$

where r_i is a disjoint $\lceil \log L \rceil$-bit substring of the shared random string. If $i \neq 0$, party P_0 first prepares a $(\lceil \log k \rceil + t)$-qubit message $|i\rangle|0^t\rangle$; otherwise P_0 prepares message $|i\rangle|\tilde{x}_0\rangle$. 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 $|i\rangle|0^t\rangle$ to $|i\rangle|\tilde{x}_i\rangle$ 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 $\{x_0, \ldots, x_{k-1}\} \setminus \{x_i\}$ with high probability. If $i \neq 0$, party P_0 prepares $(t + \lceil \log k \rceil)$-qubit message $|\tilde{x}_i\rangle|0^{\lceil \log k \rceil}\rangle$; otherwise it prepares message $|\tilde{x}_i\rangle|0^{\lceil \log k \rceil - 1}1\rangle$. Notice that the second register is used to count the number of parties that have values identical to x_i (with high probability). 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 $\tilde{x}_j \neq \tilde{x}_i$; 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. If we set $t = O(\log k)$, then the oracle gives a correct answer with error probability at most $1/poly(k)$. In this case, the quantum communication complexity of one oracle call is $O(k \log k)$.

By combining Grover's search algorithm with this distributed oracle, $O(k\sqrt{k} \log k)$-qubit communication is sufficient to find any party P_i with bounded error 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 $\mathbf{Distinctness}_{k,L}^{\mathrm{ring}}$ 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.)

Now we removed the assumption that all parties share a random string by using Lemma 3: The random string can be substituted for broadcasting an $O(\log(k \log L))$-bit message, for which $O(k \log(k \log L))$-bit communication is needed. Therefore, the total communication complexity is $O(k\sqrt{k} \log k) + O(k \log(k \log L)) = O(k(\sqrt{k} \log k + \log \log L))$. ■

The following problem is a generalization of $\mathbf{Distinctness}_{k,L}^{\mathrm{ring}}$.

Definition 11 ($\mathbf{MAXCOALITION}_{k,L}^{G}$) 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 compute $\max_{j \in \{0,\ldots,L-1\}} |\{i \colon x_i = j\}|$. At termination, each party knows a $\lceil \log k \rceil$-bit result.

It is easy to see that, if every party knows a solution of $\mathbf{MAXCOALITION}_{k,L}^{\mathrm{ring}}$, he/she can know a solution of $\mathbf{Distinctness}_{k,L}^{\mathrm{ring}}$. Therefore, the lower bound of the quantum communication complexity of $\mathbf{MAXCOALITION}_{k,L}^{\mathrm{ring}}$ is $\Omega(k(\sqrt{k} + \log \log L))$. This bound is almost tight due to the following theorem.

Theorem 12 For $L \geq k$, the quantum communication complexity of $\mathbf{MAXCOALITION}_{k,L}^{G}$ is $O(k(\sqrt{k} \log k + \log \log L))$.

Proofsketch The algorithm can be obtained by combining a modification of the algorithm in Theorem 10 with the maximum (minimum) finding algorithm by Dürr et al.[9].

More precisely, suppose that every party shares a random string of sufficient length. P_0 first picks $i \in \{1, \ldots, k\}$ uniformly at random. P_0 then searches j such that the number of parties, n_j, that have the value equal to x_j is larger than that of parties, n_i, that have the value equal to x_i. If such j is found, P_0 sets i to j. Then P_0 repeats the same procedure $c\sqrt{k}$ times for a certain constant c. P_0 computes the number of parties that have the value equal to x_i and outputs it. To perform this search, we need an oracle that, for query i, answers 1 if $n_j > n_i$ for some j. This oracle can be implemented with communication complexity $O(k \log k)$ by slightly modifying the oracle in Theorem 10.

The above algorithm can solve the problem with bounded error in an analysis similar to that of the minimum finding algorithm in 9). The total communication complexity is $O(k\sqrt{k} \log k)$ with a shared random string. The theorem follows from Lemma 3 (Broadcasting the result needs only $O(k \log k)$-bit communication). ■

Acknowledgments This work was supported in part by KAKENHI (19700010)(19700019)(21700016).

References

1) Aaronson, S. and Ambainis, A.: Quantum Search of Spatial Regions, *Theory of Computing*, Vol.1, No.1, pp.47–79 (2005).

2) Bar-Yossef, Z., Jayram, T.S. and Kerenidis, I.: Exponential Separation of Quantum and Classical One-Way Communication Complexity, *Proceedings of the Thirty-Sixth Annual ACM Symposium on Theory of Computing*, pp.128–137 (2004).

3) Brassard, G., Høyer, P., Mosca, M. and Tapp, A.: Quantum amplitude amplification and estimation, *Quantum Computation and Quantum Information: A Millennium Volume*, AMS Contemporary Mathematics Series, Vol.305, AMS, pp.53–74 (2002).

4) Buhrman, H. M., Cleve, R. E. and van Dam, W.: Quantum entanglement and communication complexity, *SIAM Journal on Computing*, Vol.30, No.6, pp.1829–1841 (2000).

5) Buhrman, H.M., Cleve, R.E., Watrous, J.H. and de Wolf, R.: Quantum Fingerprinting, *Physical Review Letters*, Vol.87, No.16, p.167902 (2001).

6) Buhrman, H.M., Cleve, R.E. and Wigderson, A.: Quantum vs. classical communication and computation, *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing*, pp.63–68 (1998).

7) Cleve, R.E. and Buhrman, H.M.: Substituting quantum entanglement for communication, *Physical Review A*, Vol.56, No.2, pp.1201–1204 (1997).

8) Cleve, R.E., van Dam, W., Nielsen, M. and Tapp, A.: Quantum entaglement and the communication complexity of the inner product function, *Proceedings of the First NASA International Conference Quantum Computing and Quantum Communications*, Lecture Notes in Computer Science, Vol. 1509, Springer, pp. 61–74 (1998).

9) Dürr, C. and Høyer, P.: A Quantum Algorithm for Finding the Minimum, Technical Report quant-ph/9607014, arXiv (1996).

10) Frankl, P. and Rödl, V.: Forbidden Intersection, *Transactions of the American Mathematical Society*, Vol.300, No.1, pp.259–286 (1987).

11) Grover, L.K.: A fast quantum mechanical algorithm for database search, *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, pp. 212–219 (1996).

12) Høyer, P., Mosca, M. and de Wolf, R.: Quantum Search on Bounded-Error Inputs, *Proceedings of Thirtieth International Colloquium on Automata, Languages and Programming*, Lecture Notes in Computer Science, Vol.2719, Eindhoven, The Netherlands, Springer, pp.291–299 (2003).

13) Kalyanasundaram, B. and Schnitger, G.: The probabilistic communication complexity of set intersection, *SIAM Journal on Discrete Mathematics*, Vol.5, No.4, pp. 545–557 (1992).

14) Klauck, H.: Quantum and Approximate Privacy, *Theory of Computing Systems*, Vol.37, No.1, pp.221–246 (2004).

15) Kremer, I.: Quantum Communication, Master's thesis, Computer Science Department, The Hebrew University (1995).

16) Kushilevitz, E. and Nisan, N.: *Communication Complexity*, Cambridge University Press (1997).

17) Newman, I.: Private vs. Common Random Bits in Communication Complexity., *Information Processing Letters*, Vol.39, No.2, pp.67–71 (1991).

18) Raz, R.: Exponential Separation of Quantum and Classical Communication Complexity, *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing*, pp.358–367 (1999).

19) Razborov, A. A.: Quantum communication complexity of symmetric predicates, *Izvestiya Mathematics*, Vol.67, No.1, pp.145–159 (2003).

20) Tiwari, P.: Lower bounds on communication complexity in distributed computer networks, *Journal of the ACM*, Vol.34, No.4, pp.921–938 (1987).

21) Yao, A. C.-C.: Some complexity questions related to distributed computing, *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing*, pp. 209–213 (1979).

22) Yao, A. C.-C.: Quantum circuit complexity, *Proceedings of the Thirty-Fourth Annual IEEE Symposium on Foundations of Computer Science*, pp.352–361 (1993).

23) Yao, A. C.-C.: On the Power of Quantum Fingerprinting, *Proceedings of the Thirty-Fifth Annual ACM Symposium on Theory of Computing*, pp.77–81 (2003).