Sumcheck-Based Delegation of Quantum Computing to Rational Server

Yuki Takeuchi[1](✉), Tomoyuki Morimae[2,3], and Seiichiro Tani[1]

[1] NTT Communication Science Laboratories, NTT Corporation,
3-1 Morinosato Wakamiya, Atsugi, Kanagawa 243-0198, Japan
{yuki.takeuchi.yt,seiichiro.tani.cs}@hco.ntt.co.jp

[2] Yukawa Institute for Theoretical Physics, Kyoto University,
Kitashirakawa Oiwakecho, Sakyoku, Kyoto 606-8502, Japan
tomoyuki.morimae@yukawa.kyoto-u.ac.jp

[3] JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama 332-0012, Japan

Abstract. Recently, a new model of delegated quantum computing has been proposed, namely, rational delegated quantum computing. In this model, after a client delegates quantum computing to a server, the client pays a reward to the server. In this paper, we propose novel one-round rational delegated quantum computing protocols. The construction of the previous rational protocols depends on gate sets, while our sumcheck technique can be easily realized with any local gate set. We also show that a constant reward gap can be achieved if two non-communicating but entangled rational servers are allowed. Furthermore, we show, under a certain condition, the equivalence between *rational* and *ordinary* delegated quantum computing protocols.

Keywords: Quantum computing · Rational interactive proof · Game theory

1 Introduction

1.1 Background

Delegated quantum computing enables a client with weak computational power to delegate quantum computing to a remote (potentially malicious) server in such a way that the client can efficiently verify whether the server faithfully computes the delegated problem (i.e., can verify the server's integrity). Due to the size of a universal quantum computer and the difficulty of maintaining it, it is expected that first generation full-fledged quantum computers will be used in the delegated-quantum-computing style. Furthermore, since quantum operations

This work is partially supported by MEXT Quantum Leap Flagship Program (MEXT Q-LEAP) Grant Number JPMXS0118067394, JST PRESTO No. JPMJPR176A, and JSPS Grant-in-Aid for Young Scientists (B) No. JP17K12637.

J. Chen et al. (Eds.): TAMC 2020, LNCS 12337, pp. 69–81, 2020.
https://doi.org/10.1007/978-3-030-59267-7_7

and communication are too demanding (for current technologies), the client's operations and their communication should be made classical.

One of the most important open problems in the field of quantum computing is whether a classical client can efficiently delegate universal quantum computing to a quantum server while efficiently verifying the server's integrity. In delegated quantum computing, the honest server's computational power should be bounded by polynomial-time quantum computing, because delegated quantum computing with a server having unbounded computational power is unrealistic. This limitation is the large difference between delegated quantum computing and interactive proof systems for BQP. In interactive proof systems, the computational power of the prover (i.e., the server) is unbounded. Therefore, this open problem cannot be straightforwardly solved from the well-known containment $\mathsf{BQP} \subseteq \mathsf{PSPACE}{=}\mathsf{IP}$ [1].

In this paper, we take a different approach to construct protocols for classical-client delegated quantum computing. We consider delegating quantum computing to a rational server. This model was first proposed by Morimae and Nishimura [2] based on the concept of rational interactive proof systems [3]. We note again that the computational power of the server is bounded by BQP[1] in rational delegated quantum computing, while it is unbounded in the rational interactive proof systems. In rational delegated quantum computing, after the client interacts with the server, the client pays a reward to the server depending on the server's messages and the client's random bits. In *ordinary* delegated quantum computing, the server may be malicious. On the other hand, in *rational* one, the server is always rational, i.e., he/she tries to maximize the expected value of the reward. In the real world, there are several situations where service providers want to maximize their profits. Since rational delegated quantum computing reflects such situations, this model can be considered as another possible situation for delegated quantum computing. In Ref. [2], it was shown that the classical client can delegate universal quantum computing to the rational quantum server in one round.

1.2 Our Contribution

As our main contribution, we propose a novel one-round delegated quantum computing protocol with a classical client and a rational quantum server. More precisely, we construct protocols where the classical client can efficiently delegate to the rational quantum server the estimation of output probabilities of n-qubit quantum circuits. Their estimation has many applications such as estimating the expected values of observables, which are quantities interested especially by physicists, and solving decision problems in BQP. Specifically, we consider any n-qubit polynomial-size quantum circuit with k-qubit output measurements, where $k = O(\log n)$. Since the goal of our rational protocol is to delegate the estimation of the output probabilities, we, for clarity, refer to our protocol as delegated

[1] For simplicity, we sometimes use complexity classes to represent computational powers. For example, we say that a server (a client) is a BQP server (a BPP client) when he/she performs polynomial-time quantum (probabilistic classical) computing.

quantum estimating protocol. As shown in the full paper [4], our argument can also be used to construct a one-round rational delegated quantum computing protocol for any BQP problem. Intuitively, using a certain BQP-complete problem [5], any BQP problem can be reduced to the estimation of the probability of the first qubit being projected onto $|1\rangle$. Therefore, our argument works. Furthermore, if a delegated quantum circuit is approximately sparse, our result can be generalized to the estimation of output probabilities with n-qubit output measurements. For general quantum circuits, such generalization is still open.

Our protocols can be applied to a broader class of universal gate sets than the previous protocols [2]. They work for any universal gate set each of whose elementary gates acts on at most $O(\log n)$ qubits, while the previous protocols are tailored for Clifford gates plus $T \equiv |0\rangle\langle 0| + e^{i\pi/4}|1\rangle\langle 1|$ or classical gates plus the Hadamard gate. Note that we only consider gate sets whose elementary gates can be specified with a polynomial number of bits.

Four conditions should be satisfied by practical rational delegated quantum computing protocols:

1. The reward is upper-bounded by a constant.
2. The reward is always non-negative if the BQP server takes an optimal strategy that maximizes its expected value.[2]
3. The maximum of the expected value of the reward is lower-bounded by a constant.
4. The reward gap [6] is larger than a constant. Here, simply speaking, the reward gap is a minimum loss on the expected value of the server's reward incurred by the server's behavior that makes the client accept an incorrect answer. Note that such behavior may require computational power beyond BQP, while we limit the optimal strategy maximizing the expected value to one that can be executed in quantum polynomial time.

The protocols of Ref. [2] and our protocol satisfy only conditions 1–3. Whether the above four conditions can be satisfied simultaneously is an open problem. In Ref. [2], it is shown that if the reward gap is larger than $1/f(n)$ with a polynomial $f(n)$, a super-polynomial increase of the reward (i.e., the violation of the first condition) is unavoidable in one-round protocols with a single server unless $\mathsf{BQP} \subseteq \Sigma_3^{\mathsf{P}}$. Since this inclusion is considered unlikely given the oracle separation between BQP and PH [7], this implies that it may be impossible to satisfy the above four conditions simultaneously in one-round protocols with a single server.

As the second contribution, for BQP problems, we construct a multi-rational-server delegated quantum computing protocol that satisfies all four conditions simultaneously. In the full paper [4], we also discuss whether a single server is sufficient under the (widely believed) assumption that the learning with errors (LWE) problem is hard for polynomial-time quantum computation.

[2] More precisely, the server takes an optimal strategy that can be executed in quantum polynomial time, because we assume that the computational power of the server is bounded by BQP. Throughout this paper, the server's optimization is limited to one that can be performed in quantum polynomial time unless explicitly noted otherwise.

Finally, apart from these results, we show that under the certain condition introduced in Ref. [12], *rational* and *ordinary* delegated quantum computing protocols can be converted from one to the other and vice versa. This equivalence may provide a new approach to tackle the open problem of whether a classical client can efficiently delegate universal quantum computing to a (non-rational) quantum server while efficiently verifying the server's integrity. Based on this equivalence, we give an amplification method for the reward gap.

2 Preliminaries

2.1 Rational Delegated Quantum Computing

In this subsection, we define rational delegated quantum computing. Following the original definition of rational interactive proof systems [3], we first define the transcript $\mathcal{T}$, the server's view $\mathcal{S}$, and the client's view $\mathcal{C}$ as follows:

Definition 1. *We assume that k is odd. Given an instance x and a round i, we define the ith transcript $\mathcal{T}_i$, the ith server's view $\mathcal{S}_i$, and the ith client's view $\mathcal{C}_i$ as follows* $(0 \leq i \leq k)$*:*

- $\mathcal{T}_0 = \mathcal{S}_0 = \mathcal{C}_0 = \{x\}$.
- *When i is odd, $\mathcal{T}_i = \{\mathcal{T}_{i-1}, a_i\}$, where a_i is the ith server's message. On the other hand, when $i(>0)$ is even, $\mathcal{T}_i = \{\mathcal{T}_{i-1}, b_i\}$, where b_i is the ith client's message.*
- *For odd i, $\mathcal{S}_i = \{\mathcal{S}_{i-2}, \mathcal{T}_{i-1}, V_i\}$, where V_i is a quantum circuit used to compute a_i. Note that $\mathcal{S}_i$ and V_i are not defined for even i because the even-numbered round is a communication from the client to the server.*
- *For even i, $\mathcal{C}_i = \{\mathcal{C}_{i-2}, \mathcal{T}_{i-1}, r_i\}$, where r_i is a random bit string used to compute b_i. Note that $\mathcal{C}_i$ is not defined for odd i because the odd-numbered round is a communication from the server to the client.*

For all i, messages a_i and b_i are polynomial lengths. Particularly, b_i is generated from $\mathcal{C}_i$ in classical polynomial time. The quantum circuit V_i is decided from $\mathcal{S}_{i-2}$.

Based on Definition 1, we define the following k-round interaction between a BPP client and a server:

Definition 2. *Let k be odd. This means that the protocol begins with the server's step. When k is even, the following definition can be adopted by adding a communication from the server to the client at the beginning of the protocol. Let us consider the following k-round interaction:*

1. *A* BPP *client interacts with a server k times. In the ith round for odd i, the server sends a_i to the client. In the ith round for even i, the client sends b_i to the server.*
2. *The client efficiently calculates a predicate on the instance x and the kth transcript $\mathcal{T}_k$. If the predicate evaluates to $o = 1$, the client answers YES. On the other hand, if $o = 0$, the client answers NO.*

3. *The client efficiently calculates the reward $R \in [0, c]$ and pays it to the server, where c is a positive constant. Note that it is not necessary for the client and server to know the value of c. The reward function $R : \{0,1\}^* \times \{0,1\}^{poly(|x|)} \times \{0,1\}^{poly(|x|)} \to \mathbb{R}_{\geq 0}$ depends on the instance $x \in \{0,1\}^*$, the kth transcript $\mathcal{T}_k \in \{0,1\}^{poly(|x|)}$, and the client's random bits $r_{k+1} \in \{0,1\}^{poly(|x|)}$.*

Rational delegated quantum computing for decision problems is defined as follows:

Definition 3. *The k-round interaction defined in Definition 2 is called a k-round rational delegated quantum computing protocol for decision problems if and only if the following conditions hold: let $\mathbb{E}[f]$ denote the expectation value of a function f. Let $\mathcal{D}_k$ be a distribution that the kth transcript follows. For a language $L \subseteq \{0,1\}^*$ in* BQP, *if $x \in L$, there exists a classical polynomial-time predicate and a distribution $\mathcal{D}_{\mathrm{YES}}$ that can be generated in quantum polynomial time, such that*

$$\Pr[o = 1 \mid \mathcal{D}_k = \mathcal{D}_{\mathrm{YES}}] \geq \frac{2}{3} \tag{1}$$

and

$$\mathbb{E}_{\mathcal{T}_k \sim \mathcal{D}_{\mathrm{YES}}, r_{k+1}}[R(x, \mathcal{T}_k, r_{k+1})] \geq c_{\mathrm{YES}} \tag{2}$$

with some positive constant $c_{\mathrm{YES}} \leq c$.

On the other hand, if $x \notin L$, there exists a classical polynomial-time predicate and a distribution $\mathcal{D}_{\mathrm{NO}}$ that can be generated in quantum polynomial time, such that

$$\Pr[o = 0 \mid \mathcal{D}_k = \mathcal{D}_{\mathrm{NO}}] \geq \frac{2}{3} \tag{3}$$

and

$$\mathbb{E}_{\mathcal{T}_k \sim \mathcal{D}_{\mathrm{NO}}, r_{k+1}}[R(x, \mathcal{T}_k, r_{k+1})] \geq c_{\mathrm{NO}} \tag{4}$$

with some positive constant $c_{\mathrm{NO}} \leq c$.

To generate distributions $\mathcal{D}_{\mathrm{YES}}$ and $\mathcal{D}_{\mathrm{NO}}$, the server decides the ith message a_i following a distribution $\mathcal{D}_i$ that can be generated in quantum polynomial time and satisfies

$$\mathcal{D}_i = \mathrm{argmax}_{\mathcal{D}_i} \mathbb{E}_{\mathcal{D}_k, \mathcal{T}_k \sim \mathcal{D}_k, r_{k+1}}[R(x, \mathcal{T}_k, r_{k+1}) | \mathcal{D}_i, \mathcal{S}_i], \tag{5}$$

where the expectation is taken over all possible distributions $\mathcal{D}_k$ that are compatible with the current server's view $\mathcal{S}_i$. Here, we consider only the maximizations that can be performed in quantum polynomial time.

Since the server's computational power is bounded by BQP, it is in general hard for the server to select an optimal message that satisfies Eqs. (1) and (2). Therefore, the server's message a_i should be probabilistically generated. That is why we consider the distribution $\mathcal{D}_{\mathrm{YES}}$. The same argument holds for the NO case.

The value 2/3 in Eqs. (1) and (3) can be amplified to $1 - 2^{-f(|x|)}$, where $f(|x|)$ is any polynomial in $|x|$, using the standard amplification method (i.e., by repeating steps 1 and 2, and then taking the majority vote among outputs in step 2). We here mention that the above rational delegated quantum computing protocol satisfies conditions 1–3 in Sect. 1. This is straightforward from $R \in [0, c]$ and Eqs. (2) and (4).

The server would like to generate the ith message a_i following a distribution that maximizes the expected value of the finally obtained reward. However, at that time, the server cannot predict the future distribution $\mathcal{D}_k$. Therefore, the server also takes the expectation over all possible distributions $\mathcal{D}_k$. The distribution $\mathcal{D}_i$ in Eq. (5) is a distribution that maximizes such expected reward.

All of our rational protocols except for one in Sect. 3 are in accordance with Definition 3. Our rational protocol in Sect. 3 is a rational delegated quantum computing protocol for function problems, which can be defined in a similar way.

2.2 Reward Gap

Guo *et al.* have introduced the reward gap [6]. For convenience, we define a strategy s as a set $\{a_i\}_i$ of the server's messages, which may be adaptively decided according to the previous client's messages. When we focus on the dependence on the server's messages, we write $\mathbb{E}_{\mathcal{T}_k \sim \mathcal{D}, r_{k+1}}[R(x, \mathcal{T}_k, r_{k+1})]$ by $\mathbb{E}_{s \sim \mathcal{D}'}[R(x, s)]$ for short. For decision problems, the reward gap is defined as follows:

Definition 4. *Let $\mathcal{D}'$ be a distribution that the server's strategy s follows. Let $\mathcal{D}'_{\max}$ be the distribution $\mathcal{D}'$, where each message a_i follows the distribution in Eq. (5). We say that a rational delegated quantum computing protocol has a $1/\gamma(|x|)$-reward gap if for any input x,*

$$\mathbb{E}_{s \sim \mathcal{D}'_{\max}}[R(x, s)] - \max_{s \in S_{\text{incorrect}}} \mathbb{E}[R(x, s)] \geq \frac{1}{\gamma(|x|)}, \tag{6}$$

where $\gamma(|x|)$ is any function of $|x|$, and $S_{\text{incorrect}}$ is the set of the server's strategies that make the client output an incorrect answer. Here, the expectation is also taken over the client's random bits. Note that $S_{\text{incorrect}}$ may include strategies that cannot be executed in quantum polynomial time.

From Definition 3, if the server's strategy s follows the distribution $\mathcal{D}'_{\max}$, the client outputs a correct answer with high probability. $\mathbb{E}_{s \sim \mathcal{D}'_{\max}}[R(x, s)]$ is the maximum expected value of the reward paid to the rational BQP server. On the other hand, $\max_{s \in S_{\text{incorrect}}} \mathbb{E}[R(x, s)]$ is the maximum expected value of the reward paid to the *malicious* computationally-unbounded server if the server wants to maximize the expected value as much as possible while deceiving the client. This is because the client outputs an incorrect answer when the server takes the strategy $s \in S_{\text{incorrect}}$. As a result, the reward gap represents how much benefit the rational server can obtain compared with the malicious one. For function problems, we can define the reward gap in a similar way.

3 Sumcheck-Based Rational Delegated Quantum Computing

In this section, we construct a rational delegated quantum computing protocol for estimating output probabilities of n-qubit quantum circuits, which we call the rational delegated quantum estimating protocol. Particularly, we consider any n-qubit polynomial-size quantum circuit with $O(\log n)$-qubit output measurements. We also show that our protocol satisfies conditions 1–3 mentioned in Sect. 1.

Let $\{q_z\}_{z\in\{0,1\}^k}$ be the output probability distribution of the quantum circuit U, where $q_z\equiv\langle 0^n|U^\dagger(|z\rangle\langle z|\otimes I^{\otimes n-k})U|0^n\rangle$ and I is the two-dimensional identity operator. We show that if the quantum server is rational, the classical client can efficiently obtain the estimated values $\{p_z\}_{z\in\{0,1\}^k}$ with high probability such that $|p_z-q_z|\leq 1/f(n)$ for any z and any polynomial $f(n)$. Therefore, for example, the classical client can approximately sample with high probability in polynomial time from the output probability distribution $\{q_z\}_{z\in\{0,1\}^k}$ of the quantum circuit U. Before proposing our rational delegated quantum estimating protocol, we calculate q_z using the Feynman path integral. Let $U=u_L\dots u_2u_1\equiv\prod_{i=L}^{1}u_i$, where u_i is an elementary gate in a universal gate set for all i, and L is a polynomial in n. The probability q_z is calculated as follows:

$$q_z=\sum_{s\in\{0,1\}^{(2L-1)n-k}}g(z,s),\tag{7}$$

where

$$g(z,s)\equiv\langle 0^n|u_1^\dagger\left(\prod_{j=L}^{2}u_j|s^{(j-1)}\rangle\langle s^{(j-1)}|\right)^\dagger|zs^{(L)}\rangle\langle zs^{(L)}|\tag{8}$$
$$\left(\prod_{i=L}^{2}u_i|s^{(L+i-1)}\rangle\langle s^{(L+i-1)}|\right)u_1|0^n\rangle,$$

and s is a shorthand notation of the $(2L-1)n-k$ bit string $s^{(1)}s^{(2)}\dots s^{(2L-1)}$. As an important point, given z and s, the function $g(z,s)$ can be calculated in classical polynomial time. This is because each elementary gate acts on at most $O(\log n)$ qubits. Furthermore, from Eq. (8), $0\leq(1+\mathrm{Re}[g(z,s)])/2\leq 1$, where $\mathrm{Re}[g(z,s)]$ is the real part of $g(z,s)$.

To construct our rational delegated quantum estimating protocol, we use the rational sumcheck protocol [8]. The rational sumcheck protocol enables the client to efficiently delegate to the rational server the calculation (or approximation) of $\sum_{i=1}^{l}x_i$, where x_i is an integer for any i. To fit the rational sumcheck protocol to our case, we generalize it for the case of the complex number x_i. As a result, we can set $x_i=g(z,s)$ and z to be a certain fixed value. Our protocol runs as follows:

[**Protocol 1**]

1. For all $z \in \{0,1\}^k$, the rational server and the client perform the following steps:
    (a) The rational server sends to the client a real non-negative number y_z, which is explained later. (Note that y_z is represented by a bit string with logarithmic length; therefore, the message size from the server to the client is logarithmic.)
    (b) The client samples s uniformly at random from $\{0,1\}^{(2L-1)n-k}$.
    (c) The client flips a coin that lands heads with probability $(1+\mathrm{Re}[g(z,s)])/2$. If the coin lands heads, the client sets $b_z = 1$; otherwise, $b_z = 0$.
    (d) Let $Y_z \equiv [y_z + 2^{(2L-1)n-(k+1)}]/2^{(2L-1)n-k}$. The client calculates the reward

$$R(y_z, b_z) \equiv \frac{1}{2^k}\left[2Y_z b_z + 2\left(1-Y_z\right)\left(1-b_z\right) - Y_z^2 - \left(1-Y_z\right)^2 + 1\right], \quad (9)$$

    which is the (slightly modified) Brier's scoring rule [9]. This scoring rule guarantees that the expected value of the reward is maximized when y_z is equal to the probability of $b_z = 1$ up to additive and multiplicative factors. Then, the client pays the reward $R(y_z, b_z)$ to the rational server.
2. The client calculates

$$p_z \equiv \frac{y_z}{\sum_{z\in\{0,1\}^k} y_z} \quad (10)$$

   for all z.

Since the sampling in step (c) can be approximately performed in classical polynomial time, what the client has to do is simply efficient classical computing. Furthermore, since the repetitions in step 1 can be performed in parallel, this is a one-round protocol. Note that except for the communication required to pay the reward to the server, Protocol 1 only requires one-way communication from the server to the client.

We show that p_z satisfies $\sum_{z\in\{0,1\}^k} |p_z - q_z| \le 1/f(n)$ for any fixed polynomial $f(n)$ with high probability. This means that p_z is an approximated value of q_z for each z with high probability. More precisely, we show the following theorem:

Theorem 1. *Let $f(n)$ and $h(n)$ be any polynomials in n. Let $q_z = \langle 0^n|U^\dagger(|z\rangle\langle z| \otimes I^{\otimes n-k})U|0^n\rangle$, and p_z be the probability given in Eq. (10). Then, for any $f(n)$ and $h(n)$, there exists Protocol 1 such that $\sum_{z\in\{0,1\}^k} |p_z - q_z| \le 1/f(n)$ with probability of at least $1 - e^{-h(n)}$.*

The proof is given in the full paper [4]. The intuitive idea is that the expected value of our reward function increases as y_z becomes to be close to $q_z/2$ for all z. Therefore, the rational server essentially sends approximated values of $\{q_z\}_{z\in\{0,1\}^k}$ to the client.

From Theorem 1, by approximately sampling from $\{p_z\}_{z\in\{0,1\}^k}$, the client can approximately sample from $\{q_z\}_{z\in\{0,1\}^k}$ with high probability. Given the values of $\{p_z\}_{z\in\{0,1\}^k}$, the approximate sampling from $\{p_z\}_{z\in\{0,1\}^k}$ can be classically performed in polynomial time.

In Protocol 1, we assume that $(1+\mathrm{Re}[g(z,s)])/2$ can be exactly represented using a polynomial number of bits. If this is not the case, the classical client has to approximate $(1+\mathrm{Re}[g(z,s)])/2$. As a result, as shown in the full paper [4], the expected value of the reward is maximized when $y_z = q_z/2+\delta$, where the real number δ satisfies $|\delta| \leq 2^{-f'(n)}$ for a polynomial $f'(n)$. Therefore, even in the approximation case, the classical client can efficiently obtain the estimated values of the output probabilities of quantum circuits.

Next, we show the following theorem:

Theorem 2. *In Protocol 1, the total reward* $\sum_{z\in\{0,1\}^k} R(y_z,b_z)$ *is between* $3/2-O(1/2^{(2L-1)n-k})$ *and* $3/2+O(1/2^{(2L-1)n-k})$ *for* $b_z\in\{0,1\}$ *and any real values* $y_z\in[0,1/2]$. *Furthermore, the maximum expected value of the total reward is lower-bounded by* $3/2+O\left(1/2^{2(2L-1)n-k}\right)$.

The proof is given in the full paper [4]. From this theorem, Protocol 1 satisfies conditions 1–3 in Sect. 1.

4 Multi-Rational-Server Delegated Quantum Computing with a Constant Reward Gap

In this section, we consider the reward gap. Although a large reward gap is desirable to incentivize the server to behave optimally, our sumcheck-based protocol has only an exponentially small gap as in the existing rational delegated quantum computing protocols [2]. It is open as to whether a constant reward gap is possible. However, in this subsection, we show that if non-communicating but entangled multiservers are allowed, we can construct a rational delegated quantum computing protocol with a constant reward gap for BQP problems while keeping three conditions 1–3 in Sect. 1. To this end, we utilize multiprover interactive proof systems for BQP. In some multiprover interactive proof systems proposed for BQP, the computational ability of the honest provers is bounded by BQP but that of the malicious provers is unbounded (e.g., Refs. [10,11]). Simply speaking, these multiprover interactive proof systems satisfy the following: for any language $L\in\mathsf{BQP}$, there exists a $poly(|x|)$-time classical verifier V interacting with a constant number of non-communicating but entangled provers, such that for instances x, if $x\in L$, then there exists a $poly(|x|)$-time quantum provers' strategy in which V accepts with probability of at least $2/3$, and if $x\notin L$, then for any (computationally-unbounded) provers' strategy, V accepts with probability of at most $1/3$. We denote the above interaction between V and provers as π_L for the language $L\in\mathsf{BQP}$.

Using the above multiprover interactive proof systems and the construction used in Ref. [3], we construct the following rational delegated quantum computing protocol:

[**Protocol 2**]

1. For a given BQP language L and an instance x, one of M rational servers sends $b \in \{0,1\}$ to the client. As shown in Theorem 3, if the server is rational, $b = 1(0)$ when x is in L (x is not in L).
2. If $b = 1$, the client and M servers simulate π_L for the language L and instance x; otherwise, the client and M servers simulate $\pi_{\bar{L}}$ for the complement $\bar{L}$ and the instance x.
3. The client pays reward $R = 1/M$ to each of the M servers if the simulated verifier accepts. On other hand, if the simulated verifier rejects, the client pays $R = 0$.
4. The client concludes $x \in L$ if $b = 1$; otherwise, the client concludes $x \notin L$.

Note that since BQP is closed under complement, $\pi_{\bar{L}}$ exists for the complement $\bar{L}$. Here, we notice that even if the simulated verifier accepts, each server can obtain only $1/M$ as the reward. However, since the number M of the servers is two in the multiprover interactive proof systems in Refs. [10,11], the reward $1/M$ paid to each server can be made $1/2$. Furthermore, when we use the results in Refs. [10,11], the number of rounds in Protocol 2 becomes a constant.

We clarify the meaning of "rational" in multi-rational-server delegated quantum computing. We can consider at least two possible definitions of "rational". One is that each server wants to maximize each reward, and the other is that all servers want to collaboratively maximize their total reward. Fortunately, in Protocol 2, these two definitions are equivalent. In other words, the total reward is maximized if and only if the reward paid to each server is maximized. Hereafter, we therefore do not distinguish between these two definitions.

Before we show that Protocol 2 has a constant reward gap, we show that if the servers are rational, the client's answer is correct. More formally, we prove the following theorem:

Theorem 3. *In Protocol 2, if the servers are rational, i.e., take the strategy that maximizes the expectation value of the reward, then $b = 1$ if and only if $x \in L$.*

Proof. First, we consider the YES case, i.e., the case where x is in L. If $b = 1$, the client and the servers perform π_L for the language L and the instance x. Therefore, when the servers simulate the honest provers in π_L, the client accepts with probability of at least $2/3$. On the other hand, if $b = 0$, the client accepts with probability less than or equal to $1/3$. This is because x is a NO instance for the complement $\bar{L}$, i.e., $x \notin \bar{L}$. In $\pi_{\bar{L}}$, when the answer is NO, the acceptance probability is at most $1/3$ for any provers' strategy. Since the completeness-soundness gap $1/3$ is a positive constant, one of the rational servers sends $b = 1$ if $x \in L$. By following the same argument, one of them sends $b = 0$ when $x \notin L$.

From this proof, we notice that the reward gap has the same value as the completeness-soundness gap.[3] Protocol 2 has a $1/3$ reward gap, which is constant. Furthermore, it can be straightforwardly shown that Protocol 2 also satisfies conditions 1–3 mentioned in Sect. 1 as follows. Since the total reward $M \times R$

[3] Precisely speaking, since the computational power of the server is bounded by BQP, the server sends $b = 0(1)$ with an exponentially small probability when the correct

paid to M servers is 0 or 1, the first and second conditions are satisfied. When the servers behave rationally, the client accepts with probability at least 2/3. Therefore, the expected value of the total reward paid to the rational servers is at least 2/3, which satisfies the third condition.

5 Relation Between Rational and Ordinary Delegated Quantum Computing Protocols

In Sect. 4, by incorporating *ordinary* delegated quantum computing into *rational* delegated quantum computing, we have shown that the four conditions can be simultaneously satisfied. In this section, we consider the reverse direction, i.e., constructing *ordinary* delegated quantum computing protocols from *rational* delegated quantum computing protocols. By combining this construction with the idea in Sect. 4, we obtain an equivalence (under a certain condition) between these two types of delegated quantum computing. Note that in ordinary ones, the server's ability is unbounded in NO cases (i.e., when $x \notin L$).

To construct ordinary delegated quantum computing protocols from rational ones, we consider the general $poly(|x|)$-round rational delegated quantum computing protocol defined in Definition 3, which we call RDQC for short. By adding two conditions for RDQC, we define constrained RDQC as follows:

Definition 5. *The constrained RDQC protocol is an RDQC protocol defined in Definition 3 such that*

1. *There exists a classically efficiently computable polynomial $f(|x|)$ such that*

$$c_{\rm YES} - \max{}_{s\in S_{\rm incorrect}, x\notin L}\mathbb{E}[R(s,x)] \geq \frac{1}{f(|x|)}, \tag{11}$$

2. *The upper-bound c of the reward is classically efficiently computable.*

The first condition was introduced in Ref. [12]. It is worth mentioning that the second condition is satisfied in our sumcheck-based protocol, while the first condition is not. Note that the left-hand side of Eq. (11) is not the reward gap.

We show that an *ordinary* delegated quantum computing protocol with a single BQP server and a single BPP client can be constructed from any constrained RDQC protocol. To this end, we show the following theorem:

Theorem 4. *If a language L in BQP has a k-round constrained RDQC protocol, then L has a k-round interactive proof system with the completeness-soundness gap $1/(cf(|x|))$ between an honest BQP prover and a BPP verifier.*

answer is YES (NO). Therefore, the finally obtained reward gap is decreased by the inverse of an exponential from the original completeness-soundness gap. However, this is negligible because the original completeness-soundness gap is a constant.

The proof is essentially the same as that of Theorem 4 in Ref. [12].

As shown in the full paper [4], from Theorem 4, we can show that if there exists a constant-round constrained RDQC protocol for BQP, then $\mathsf{BQP} \subseteq \prod_2^{\mathsf{p}}$, which seems to be unlikely due to the oracle separation between BQP and PH [7].

We show that the reverse conversion is also possible using the idea in Sect. 4.

Theorem 5. *If a language L in* BQP *has an interactive proof system with an honest* BQP *prover and a* BPP *verifier, then L has a constrained RDQC protocol.*

The detail is given in the full paper [4].

Finally, by applying Theorems 4 and 5, we give the following amplification method for the reward gap:

Corollary 1. *The reward gap of the constrained RDQC can be amplified to a constant.*

The proof is given in the full paper [4]. Here, we explain the basic idea of the proof. Using the conversion between rational and ordinary delegated quantum computing protocols, we show that the amplification of the reward gap can be replaced with that of the soundness-completeness gap. This means that the traditional amplification method for the soundness-completeness gap can be used to amplify the reward gap. Remarkably, this amplification method works even if the original constrained RDQC protocol has only an exponentially small reward gap. This is because the original constrained RDQC protocol satisfies Eq. (11).

References

1. Shamir, A.: IP=PSPACE. In: Proceedings of the 31st Annual Symposium on Foundations of Computer Science, pp. 11–15. IEEE, St. Louis (1990)
2. Morimae, T., Nishimura, H.: Rational proofs for quantum computing. arXiv:1804.08868
3. Azar, P.D., Micali, S.: Rational proofs. In: Proceedings of the 44th Symposium on Theory of Computing, pp. 1017–1028. ACM, New York (2012)
4. Takeuchi, Y., Morimae, T., Tani, S.: Sumcheck-based delegation of quantum computing to rational server. arXiv:1911.04734
5. Aharonov, D., Ben-Or, M., Eban, E.: Interactive proofs for quantum computations. In: Proceedings of Innovations in Computer Science 2010, pp. 453–469. Tsinghua Univ. Press, Beijing (2010)
6. Guo, S., Hubáček, P., Rosen, A., Vald, M.: Rational arguments: single round delegation with sublinear verification. In: Proceedings of the 5th Conference on Innovations in Theoretical Computer Science, pp. 523–540. ACM, New Jersey (2014)
7. Raz, R., Tal, A.: Oracle separation of BQP and PH. In: Proceedings of the 51st Annual Symposium on Theory of Computing, pp. 13–23. ACM, New York (2019)
8. Guo, S., Hubáček, P., Rosen, A., Vald, M.: Rational sumchecks. In: Kushilevitz, E., Malkin, T. (eds.) TCC 2016. LNCS, vol. 9563, pp. 319–351. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49099-0_12
9. Brier, G.W.: Verification of forecasts expressed in terms of probability. Mon. Weather. Rev. **78**, 1–3 (1950). https://journals.ametsoc.org/mwr/article/78/1/1/96424/VERIFICATION-OF-FORECASTS-EXPRESSED-IN-TERMS-OF

10. Coladangelo, A., Grilo, A.B., Jeffery, S., Vidick, T.: Verifier-on-a-leash: new schemes for verifiable delegated quantum computation, with quasilinear resources. In: Ishai, Y., Rijmen, V. (eds.) EUROCRYPT 2019. LNCS, vol. 11478, pp. 247–277. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-17659-4_9
11. Grilo, A.B.: A simple protocol for verifiable delegation of quantum computation in one round. In: Proceedings of the 46th International Colloquium on Automata, Languages, and Programming, pp. 28:1–28:13. EATCS, Patras (2019)
12. Chen, J., McCauley, S., Singh, S.: Efficient rational proofs with strong utility-gap guarantees. In: Deng, X. (ed.) SAGT 2018. LNCS, vol. 11059, pp. 150–162. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-99660-8_14