On the tightness of the Buhrman-Cleve-Wigderson simulation

Shengyu Zhang

Department of Computer Science and Engineering,
The Chinese University of Hong Kong.
syzhang@cse.cuhk.edu.hk

Abstract. Buhrman, Cleve and Wigderson gave a general communication protocol for block-composed functions $f(g_1(x^1, y^1), \cdots, g_n(x^n, y^n))$ by simulating a decision tree computation for f [3]. It is also well-known that this simulation can be very inefficient for some functions f and $(g_1, \cdots, g_n)$. In this paper we show that the simulation is actually polynomially tight up to the choice of $(g_1, \cdots, g_n)$. This also implies that the classical and quantum communication complexities of certain block-composed functions are polynomially related.

1 Introduction

Decision tree complexity [4] and communication complexity [7] are two concrete models for studying computational complexity. In [3], Buhrman, Cleve and Wigderson gave a general method to design communication protocol for block-composed functions $f(g_1(x^1, y^1), \cdots, g_n(x^n, y^n))$. The basic idea is to simulate the decision tree computation for f, with queries of the i-th variable answered by a communication protocol for $g_i(x^i, y^i)$. In the language of complexity measures, this result gives

$$\mathsf{CC}(f(g_1, \cdots, g_n)) = \tilde{O}(\mathsf{DT}(f) \cdot \max_i \mathsf{CC}(g_i)) \tag{1}$$

where CC and DT are the communication complexity and the decision tree complexity, respectively. This simulation holds for all the models: deterministic, randomized and quantum[1].

It is also well known that the communication protocol by the above simulation can be very inefficient. For example, if f is the n-bit AND function and all g_i's are 2-bit AND function, then even the deterministic communication complexity of $f(g_1, \cdots, g_n)$ is just 1 bit, since Alice can compute and send the AND function of her bits. This is in sharp contrast to the decision tree complexity of the n-bit AND function f, which is $\Theta(n)$ in the randomized case and $\Theta(\sqrt{n})$ in the quantum case.

Turning the simulation around, one can also get a lower bound method for the decision tree complexity by communication complexity. To lower bound $\mathsf{DT}(f)$, we can pick g_i's, then the communication complexity of

[1] For the deterministic model, no error correction for each g_i is needed, so the $\tilde{O}$ can be changed to O.

$\mathsf{CC}(f(g_1,\cdots,g_n))/\max_i \mathsf{CC}(g_i)$ is a lower bound of $\mathsf{DT}(f)$. Actually in the same paper [3], they obtained the almost tight lower bound of $\tilde{\Omega}(n)$ for the quantum decision tree complexity of the Parity and $\mathsf{Majority}$ functions this way. But because of the counterexamples as shown in the last paragraph, it is not clear how tight this lower bound method can be in general.

In this paper, we will show that the simulation is polynomially tight, and actually this can be achieved by each g_i chosen only from $\{\wedge,\vee\}$, *i.e.* 2-bit AND and OR.

Theorem 1. *For Boolean functions f,*

$$\max_{g_i\in\{\wedge,\vee\}} \mathsf{R}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = \Omega(\mathsf{D}^{\mathsf{DT}}(f)^{1/3}), \tag{2}$$

$$\max_{g_i\in\{\wedge,\vee\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = \Omega(\mathsf{D}^{\mathsf{DT}}(f)^{1/6}). \tag{3}$$

For monotone functions f, the bounds can be improved to the following. For two n-bit strings x and y, use $x\wedge_n y$ and $x\vee_n y$ to denote the bit-wise AND and OR of x and y, respectively. We drop the subscript when $n=1$.

Theorem 2. *For monotone Boolean functions f,*

$$\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{R}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = \Omega(\mathsf{D}^{\mathsf{DT}}(f)^{1/2}), \tag{4}$$

$$\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = \Omega(\mathsf{D}^{\mathsf{DT}}(f)^{1/4}). \tag{5}$$

Note that the improvement is two-fold: Besides the better bounds themselves, the range of inner function g is also restricted to $\{\wedge_n,\vee_n\}$. That is, we require all g_i's be the same; they are either all AND or all OR functions.

The bounds give the following corollary about the polynomial relation between quantum and classical communication complexity for composed functions.

Corollary 1. *For Boolean functions f,*

$$\max_{g_i\in\{\wedge,\vee\}} \mathsf{D}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = O\left(\max_{g_i\in\{\wedge,\vee\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1,\cdots,g_n))^6\right). \tag{6}$$

If f is monotone, then

$$\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{D}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = O\left(\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1,\cdots,g_n))^4\right). \tag{7}$$

Related work After the results in the current paper being circulated at Institute for Quantum Computing at University of Waterloo and Centre for Quantum Technologies at National University of Singapore in May 2009, Sherstov posted a related paper [10], which does not have our Theorem 1 and Theorem 2, but shows the following result:

Theorem 3. *(Sherstov, [10]) For Boolean functions f,*

$$\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{D}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = O\left(\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1,\cdots,g_n))^{12}\right). \tag{8}$$

The technical ingredient to achieve the above theorem is to observe that the function contains a subfunction g with size-2 block sensitivity $bs^2(g) \geq bs(f)$, and then use a theorem by Kenyon and Kutin [6] that $s(g) = \Omega(\sqrt{bs^2(g)})$.

2 Preliminaries

For an n-bit Boolean function $f : \{0,1\}^n \to \{0,1\}$, a deterministic query algorithm for f accesses the input x only by making queries in the form of "$x_i =$?". Each query has cost 1, and all the other computation between queries are free. A randomized query algorithm is the same except that the algorithm can toss coins to decide the next variable x_i to ask. The quantum query model, formally introduced in [2], has a working state in the form of $\sum_{i,a,z} \alpha_{i,a,z}|i,a,z\rangle$, where i ranges over $[n]$, a ranges over $\{0,1\}$ and z is the content in the working space. A quantum query on the input x corresponds to an oracle O_x, a unitary operation defined by

$$O_x\Big(\sum_{i,a,z} \alpha_{i,a,z}|i,a,z\rangle\Big) = \sum_{i,a,z} \alpha_{i,a,z}|i,a\oplus x_i,z\rangle \tag{9}$$

A T-query quantum query algorithm works as a sequence of operations

$$U_0 \to O_x \to U_1 \to O_x \to \cdots \to U_{T-1} \to O_x \to U_T \tag{10}$$

Here O_x is as defined above, and each U_t does not depend on the input x. In both randomized and quantum query models, we can allow a double-sided error probability of 1/3. The deterministic, randomized and quantum query complexities, denoted by $\mathsf{D}^{\mathsf{DT}}(f)$, $\mathsf{R}^{\mathsf{DT}}(f)$ and $\mathsf{Q}^{\mathsf{DT}}(f)$, are the minimum numbers of queries we need to make in order to compute the function by a deterministic, randomized and quantum query algorithm, respectively.

Communication complexity studies the minimum amount of communication that two or more parties need to compute a given function or a relation of their inputs. Since its inception in the seminal paper by Yao [12], communication complexity has been an important and widely studied research area, both because of the interesting and intriguing mathematics involved in its study, and also because of the fundamental connections it bears with many other areas in theoretical computer science. In the standard *two-party interactive* model, two parties Alice and Bob, each on receiving an input say $x \in \mathcal{X}$ and $y \in \mathcal{Y}$, respectively, sending messages back and forth to jointly compute a function f on input (x,y). Their computation and communication can be deterministic, randomized, and quantum. The deterministic, randomized, and quantum communication complexity of f, denoted by $\mathsf{D}^{\mathsf{CC}}(f)$, $\mathsf{R}^{\mathsf{CC}}(f)$ and $\mathsf{Q}^{\mathsf{CC}}(f)$, is the least

number of bits (or qubits in the quantum case) needed to be transferred in the corresponding model, *s.t.* the protocol gives the correct answer with probability at least 2/3 for all inputs.
For a string $x \in \{0,1\}^n$ and a set $I \subseteq [n]$, the string $x^{(I)}$ is obtained from x by flipping all coordinates in I.

Definition 1 (block sensitivity) *The block sensitivity $bs(f,x)$ of f on x is the maximum number b such that there are disjoint sets $I_1, \cdots, I_b$ for which $f(x) \neq f(x^{I_j})$. The block sensitivity of f is $bs(f) = \max_x bs(f,x)$. For $z \in \{0,1\}$, the z-block sensitivity is $bs_z(f) = \max_{x:f(x)=z} bs(f,x)$.*

The block sensitivity is one of the complexity measures that are polynomially related to each other. In particular, it powered is known to be an upper bound of the deterministic decision tree complexity, as shown in the following theorem by Beals *et. al.* [2]. For many other complexity measures and their relations, we refer the reader to the excellent survey [4].

Theorem 4. *1.* $\mathsf{D}^{\mathsf{DT}}(f) = O(bs(f)^3)$.
2. For monotone functions f, $\mathsf{D}^{\mathsf{DT}}(f) = O(bs(f)^2)$.

The function $\mathsf{Disj}_n : \{0,1\}^n \times \{0,1\}^n$ is defined as follows: $\mathsf{Disj}_n(x,y) = \mathsf{OR}(x_1 \wedge y_1, \cdots, x_n \wedge y_n)$. The promise version of the problem, $\mathsf{PromiseDisj}_n$, is the same as Disj_n but with the promise that there is at most one i *s.t.* $x_i \wedge y_i = 1$. The randomized and quantum communication complexity for Disj_n and $\mathsf{PromiseDisj}_n$ are known as follows.

Theorem 5.

$$\mathsf{R}^{\mathsf{CC}}(\mathsf{Disj}_n) \geq \mathsf{R}^{\mathsf{CC}}(\mathsf{PromiseDisj}_n) = \Omega(n), \tag{11}$$

$$\mathsf{Q}^{\mathsf{CC}}(\mathsf{Disj}_n) \geq \mathsf{Q}^{\mathsf{CC}}(\mathsf{PromiseDisj}_n) = \Omega(\sqrt{n}). \tag{12}$$

The original randomized lower bound [5, 8, 1] was for Disj instead of $\mathsf{PromiseDisj}$. But the same proof of [1] also carries to the same lower bound for $\mathsf{PromiseDisj}$. The original quantum lower bound [9, 11] was also for Disj, but as mentioned in [10], the same method in [9] also applies to prove the same lower bound for $\mathsf{PromiseDisj}$. [10] also explicitly gives a proof for $\mathsf{Q}^{\mathsf{CC}}(\mathsf{PromiseDisj})$ by adapting the method in [11].

3 Lower bounds for the communication complexity of composed functions

We will actually prove that

Lemma 1. *For Boolean functions f,*

$$\max_{g_i \in \{\wedge, \vee\}} \mathsf{R}^{\mathsf{CC}}(f(g_1, \cdots, g_n)) = \Omega(bs(f)), \tag{13}$$

$$\max_{g_i \in \{\wedge, \vee\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1, \cdots, g_n)) = \Omega(\sqrt{bs(f)}). \tag{14}$$

If f is monotone, then

$$\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{R}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = \Omega(bs(f)), \tag{15}$$

$$\max_{g\in\{\wedge_n,\vee_n\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1,\cdots,g_n)) = \Omega(\sqrt{bs(f)}). \tag{16}$$

Proof. By the definition of block sensitivity, there are an input z and blocks $I_1,\cdots,I_b$, where $b = bs(f)$, *s.t.* f is sensitive on z at those blocks. That is,

$$f(z^{(I_1)}) = \cdots = f(z^{(I_b)}) \neq f(z). \tag{17}$$

We will define g_i's *s.t.* if there is a protocol for $f(g_1,\cdots,g_n)$, then there is a protocol for $\mathsf{PromiseDisj}_b$. The reduction is: on input $(x,y) \in \{0,1\}^b \times \{0,1\}^b$, we define an input $(x',y') \in \{0,1\}^n \times \{0,1\}^n$ for the function $f(g_1,\cdots,g_n)$ as follows.

1. For $i \notin \cup_{j=1}^b I_j$:
$$x'_i = y'_i = z_i, \;\; g_i = \wedge \tag{18}$$
2. For $i \in I_j$:
$$\begin{cases} x'_i = x_j, \;\; y'_i = y_i, \;\; g_i = \wedge, & \text{if } z_i = 0 \\ x'_i = \bar{x}_j, \;\; y'_i = \bar{y}_i, \;\; g_i = \vee, & \text{if } z_i = 1 \end{cases} \tag{19}$$

It is easy to see that for the first case, $g_i(x'_i, y'_i) = z_i \wedge z_i = z_i$. For the second case, if $z_i = 0$, then

$$g_i(x'_i, y'_i) = x_j \wedge y_j = (x_j \wedge y_j) \oplus z_i; \tag{20}$$

if $z_i = 1$, then

$$g_i(x'_i, y'_i) = \bar{x}_j \vee \bar{y}_j = \overline{x_j \wedge y_j} = (x_j \wedge y_j) \oplus z_i. \tag{21}$$

Thus for each $j = 1, 2, \cdots, b$, it holds that

$$x_j \wedge y_j = 1 \Leftrightarrow g_i(x'_i, y'_i) = \bar{z}_i, \forall i \in I_j \tag{22}$$

Therefore,

$$\text{Distinguishing between } x \wedge y = 0 \text{ and } \exists \text{ unique } j, \;\; s.t. \;\; x_j \wedge y_j = 1 \tag{23}$$

$$\Leftrightarrow \text{Distinguishing between } g(x', y') = z \text{ and } \exists \text{ unique } j, \;\; s.t. \;\; g(x', y') = z^{(I_j)} \tag{24}$$

Now if we have a protocol to compute $f(g_1,\cdots,g_n)$, then we can use it to solve the problem in Eq. (24). By the equivalence, this also solves the problem in Eq. (23), *i.e.* the $\mathsf{PromiseDisj}_b$ problem. Since

$$\mathsf{R}^{\mathsf{CC}}(\mathsf{PromiseDisj}_b) = \Omega(b), \quad \mathsf{Q}^{\mathsf{CC}}(\mathsf{PromiseDisj}_b) = \Omega(\sqrt{b}) \tag{25}$$

we proved the conclusion for general Boolean function f.

If f is monotone, then observe that we can assume that each sensitive block I_j contains all 0's or all 1's. Actually, suppose $f(z) = 0$ and $I_j = I_{j,0} \uplus I_{j,1}$ where z on $I_{j,b}$ contains only bits equal to b. Then we can remove $I_{j,1}$ and let the $I_{j,0}$ be the new block. It is still disjoint with all other blocks, yet by monotonicity, it has $f(z^{(I_{j,0})}) \geq f(z^{(I_j)}) = 1$. In this way we can assume that for $f(z) = 0$, all sensitive blocks contains only 0's. Thus using similar reductions, we have

$$\mathsf{R}^{\mathsf{CC}}(f(\wedge, \cdots, \wedge)) = \Omega(\mathsf{R}^{\mathsf{CC}}(\mathsf{Disj}_{bs_0(f)})) = \Omega(bs_0(f)), \tag{26}$$

$$\mathsf{Q}^{\mathsf{CC}}(f(\wedge, \cdots, \wedge)) = \Omega(\mathsf{Q}^{\mathsf{CC}}(\mathsf{Disj}_{bs_0(f)})) = \Omega(\sqrt{bs_0(f)}). \tag{27}$$

Note that here because of the monotonicity, we can reduce the problem to Disj instead of $\mathsf{PromiseDisj}$, though this does not give us any stronger bound. Similarly, we have

$$\mathsf{R}^{\mathsf{CC}}(f(\vee, \cdots, \vee)) = \Omega(\mathsf{R}^{\mathsf{CC}}(\mathsf{Disj}_{bs_1(f)})) = \Omega(bs_1(f)), \tag{28}$$

$$\mathsf{Q}^{\mathsf{CC}}(f(\vee, \cdots, \vee)) = \Omega(\mathsf{Q}^{\mathsf{CC}}(\mathsf{Disj}_{bs_1(f)})) = \Omega(\sqrt{bs_1(f)}). \tag{29}$$

Since $bs(f) = \max\{bs_0(f), bs_1(f)\}$, this finishes the proof of the lemma.

Theorem 1 and 2 follow from the above lemma and Theorem 4. Corollary 1 is also easy:

$$\max_{g_i \in \{\wedge, \vee\}} \mathsf{D}^{\mathsf{CC}}(f(g_1, \cdots, g_n)) \tag{30}$$

$$= O(\mathsf{D}^{\mathsf{DT}}(f)) \qquad \text{(by [3])} \tag{31}$$

$$= O\Big(\max_{g_i \in \{\wedge, \vee\}} \mathsf{Q}^{\mathsf{CC}}(f(g_1, \cdots, g_n))^6\Big) \qquad \text{(by Theorem 1)} \tag{32}$$

The monotone function case follows similarly.

References

1. Ziv Bar-Yossef, T. S. Jayram, Ravi Kumar, and D. Sivakumar. An information statistics approach to data stream and communication complexity. *Journal of Computer and System Sciences*, 68(4):702–732, 2004.
2. Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum lower bounds by polynomials. *Journal of the ACM*, 48(4):778–797, 2001.
3. Harry Buhrman, Richard Cleve, and Avi Wigderson. Quantum vs. classical communication and computation. In *Proceedings of the Thirtieth Annual ACM Symposium on the Theory of Computing (STOC)*, pages 63–68, 1998.
4. Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002.
5. Bala Kalyanasundaram and Georg Schintger. The probabilistic communication complexity of set intersection. *SIAM Journal on Discrete Mathematics*, 5(4):545–557, 1992.
6. Claire Kenyon and Samuel Kutin. Sensitivity, block sensitivity, and l-block sensitivity of boolean functions. *Information and Computation*, 189(1):43–53, 2004.
7. Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, Cambridge, UK, 1997.

8. Alexander Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106:385–390, 1992.
9. Alexander Razborov. Quantum communication complexity of symmetric predicates. *Izvestiya: Mathematics*, 67(1):145–159, 2003.
10. Alexander Sherstov. On quantum-classical equivalence for composed communication problems. *arXiv:quant-ph/0906.1399*, 2009.
11. Alexander A. Sherstov. The pattern matrix method for lower bounds on quantum communication. In *Proceedings of the 40th Annual ACM Symposium on the Theory of Computing (STOC)*, pages 85–94, 2008.
12. Andrew Yao. Some complexity questions related to distributive computing. In *Proceedings of the Eleventh Annual ACM Symposium on Theory of Computing (STOC)*, pages 209–213, 1979.