A Note on the Quantum Query Complexity of Permutation Symmetric Functions

André Chailloux
Inria de Paris, EPI SECRET, Paris, France
andre.chailloux@inria.fr

Abstract

It is known since the work of [1] that for any permutation symmetric function f, the quantum query complexity is at most polynomially smaller than the classical randomized query complexity, more precisely that $R(f) = \widetilde{O}\left(Q^7(f)\right)$. In this paper, we improve this result and show that $R(f) = O\left(Q^3(f)\right)$ for a more general class of symmetric functions. Our proof is constructive and relies largely on the quantum hardness of distinguishing a random permutation from a random function with small range from Zhandry [11].

2012 ACM Subject Classification Theory of computation → Quantum query complexity

Keywords and phrases quantum query complexity, permutation symmetric functions

Digital Object Identifier 10.4230/LIPIcs.ITCS.2019.19

Acknowledgements The author wishes to thank Anthony Leverrier and anonymous referees for helpful comments on this paper.

1 Introduction

The black box model has been a very fruitful model for understanding the possibilities and limitations of quantum algorithms. In this model, we can prove some exponential speedups for quantum algorithms, which is notoriously hard to do in standard complexity theory. Famous examples are the Deutsch-Josza problem [7] and Simon's problem [10]. There has been a great line of work to understand quantum query complexity, which developed some of the most advanced algorithms techniques. Even Shor's algorithm [9] for factoring fundamentally relies on a black box algorithm for period finding.

We describe here the query complexity model in a nutshell. The idea is that we have to compute $f(x_1, \ldots, x_n)$ where each $x_i \in [M]$ can be accessed via a query. We consider decision problems meaning that $f : S \to \{0,1\}$ with $S \subseteq [M]^n$. In this paper, we will consider inputs $x \in [M]^n$ equivalently as functions from $[n] \to [M]$. We are not interested in the running time of our algorithm but only want to minimize the number of queries to x, which in the quantum setting consists of applying the unitary $\mathcal{O}_x : |i\rangle|j\rangle \to |i\rangle|j + x_i\rangle$. $D(f), R(f)$ and $Q(f)$ represent the minimal amount of queries to compute f with probability greater than $2/3$ (or $= 1$ for the case of $D(f)$) using respectively a deterministic algorithm with classical queries, a randomized algorithm with classical queries and a quantum algorithm with quantum queries.

As we said before, the query complexity is great for designing new quantum algortihms. It is also very useful for providing black box limitations for quantum algorithms. There are some cases in particular where we can prove that the quantum query complexity of f is at most polynomially smaller than classical (deterministic or randomized) query complexity. For example:

10th Innovations in Theoretical Computer Science (ITCS 2019).
Editor: Avrim Blum; Article No. 19; pp. 19:1–19:7
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

- for specific functions such as search [6] or element distinctness (ED) [2, 8, 4], we have respectively $Q(Search) = \Theta(n^{1/2}), D(Search) = \Theta(n)$ and $Q(\text{ED}) = \Theta(n^{2/3}), D(\text{ED}) = \Theta(n)$.
- For any total function f *i.e.* when its domain $S = [M]^n$, Beals *et al.* [5] proved using the polynomial method that $D(f) \leq O(Q^6(f))$.

Another case of interest where we can lower bound the quantum query complexity is the case of permutation symmetric functions. There are several ways of defining such functions and we will be interested in the following definitions for a function $f : S \to \{0, 1\}$ with $S \subseteq [M]^n$.

▶ **Definition 1.**
- f permutation symmetric of the first type iff. $\forall \pi \in S_n,\ f(x) = f(x \circ \pi)$.
- f is permutation symmetric of the second type iff. $\forall \pi \in S_n,\ \forall \sigma \in S_M$, $f(x) = f(\sigma \circ x \circ \pi)$.

where S_n (resp. S_M) represents the set of permutations on $[n]$ (resp. $[M]$) and $\circ$ is the usual function composition.

Here, recall that we consider strings $x \in [M]^n$ as functions from $[n] \to [M]$. Notice also that this definition implies that S is stable by permutation, meaning that $x \in S \Leftrightarrow \forall \pi \in S_n,\ x \circ \pi \in S$. We already know from the work of Aaronson and Ambainis the following result:

▶ **Theorem 2** ([1]). *For any permutation symmetric function f of the second type (Definition 1), $R(f) \leq \widetilde{O}(Q^7(f))$.*

In a recent survey on quantum query complexity and quantum algorithms [3], Ambainis writes:

> “It has been conjectured since about 2000 that a similar result also holds for f with a symmetry of the first type.”

Contribution

The contribution of this paper is to prove the above conjecture. We show the following:

▶ **Theorem 3.** *For any permutation symmetric function f of the first type, $R(f) \leq O(Q^3(f))$.*

This result not only generalizes the result for a more general class of permutation symmetric function, but also improves the exponent from 7 to 3. In the case where $M = 2$, this result was already known [1] with an exponent of 2, which is tight from Grover’s algorithm.

The proof technique is arguably simple, constructive and relies primarily on the quantum hardness of distinguishing a random permutation from a random function with small range from Zhandry [11]. We start from a permutation symmetric function f. At high level, the proof goes as follows:

- We start from an algorithm $\mathcal{A}$ that outputs $f(x)$ for all x with high (constant) probability. Let q the number of quantum queries to $\mathcal{O}_x$ performed by $\mathcal{A}$.
- Instead of running $\mathcal{A}$ on input x, we choose a random function $C : [n] \to [n]$ with a range of small size r (from a distribution specified later in the paper) and apply the algorithm $\mathcal{A}$ where we replace calls to $\mathcal{O}_x$ with calls to $\mathcal{O}_{x \circ C}$. We note that there is a simple procedure to compute $\mathcal{O}_{x \circ C}$ from $\mathcal{O}_x$ and $\mathcal{O}_C$.

- If we take $r = \Theta(q^3)$, we can use Zhandry's lower bound, we show that for each x, the output will be close to the output of the algorithm $\mathcal{A}$ where we replace calls to $\mathcal{O}_{x \circ C}$ with calls to $\mathcal{O}_{x \circ \pi}$ for a random permutation π. Using the fact that f is permutation symmetric, the latter algorithm will output with high probability $f(x \circ \pi) = f(x)$. In other words, if the algorithm $\mathcal{A}$ that calls $\mathcal{O}_{x \circ C}$ wouldn't output $f(x)$ for a random C and a fixed x then we would find a distinguisher between a random C and a random permutation π, which is hard from Zhandry's lower bound.
- The above tells us that applying $\mathcal{A}$ where we replace calls to $\mathcal{O}_x$ with calls to $\mathcal{O}_{x \circ C}$ gives us output $f(x)$ with high probability. Knowing C, we can construct the whole string $x \circ C$ by querying x on inputs $i \in Im(C)$ which can be done with $Im(C) \leq r$ classical queries which allows us to construct the unitary $\mathcal{O}_{x \circ C}$. This means we can emulate $\mathcal{A}$ on input $x \circ C$ with r classical queries to x and this gives us $f(x)$ with high probability.

After presenting a few notations, we dive directly into the proof of our theorem.

2 Preliminaries

2.1 Notations

For any function f, let $Im(f)$ be its range (or image).

Query algorithms

A query algorithm $\mathcal{A}^{\mathcal{O}}$ is described by a algorithm that calls another function $\mathcal{O}$ in a black box fashion. We will never be interested in the running time or the size of $\mathcal{A}$ but only in the number of calls, or queries, to $\mathcal{O}$. We will consider both the cases where the algorithm $\mathcal{A}^{\mathcal{O}}$ is classical and quantum. In the latter $\mathcal{O}$ will be a quantum unitary. In both cases, we only consider algorithms that output a single bit.

Oracles

We use oracles to perform black box queries to a function. For any function g, $\mathcal{O}_g^{\text{Classical}}$ is a black box that on input i outputs $g(i)$ while $\mathcal{O}_g$ (without any superscript) is the quantum unitary satisfying

$$\mathcal{O}_g : |i\rangle|j\rangle \to |i\rangle|j + g(i)\rangle.$$

Query complexity

Fix a function $f : S \to \{0,1\}$ where $S \subseteq [M]^n$.

▶ **Definition 4.** The randomized query complexity $R(f)$ of f is the smallest integer q such that there exists a classical randomized algorithm $\mathcal{A}^{\mathcal{O}}$ performing q queries to $\mathcal{O}$ satisfying:

$$\forall x \in S,\ \Pr[\mathcal{A}^{\mathcal{O}_x^{\text{Classical}}} \text{ outputs } f(x)] \geq 2/3.$$

▶ **Definition 5.** The quantum query complexity $Q(f)$ of f is the smallest integer q such that there exists a quantum algorithm $\mathcal{A}^{\mathcal{O}}$ performing q queries to $\mathcal{O}$ satisfying:

$$\forall x \in S,\ \Pr[\mathcal{A}^{\mathcal{O}_x} \text{ outputs } f(x)] \geq 2/3.$$

2.2 Hardness of distinguishing a random permutation from a random function with small range

Our proof will use a quantum lower bound on distinguishing a random permutation from a random function with small range proven in [11]. Following this paper, we define, for any $r \in [n]$, the following distribution D_r on functions from $[n]$ to $[n]$ from which can be sampled as follows.

- Draw a random function g from $[n] \to [r]$.
- Draw a random injective function h from $[r] \to [n]$.
- Output the composition $h \circ g$.

Notice that any function f drawn from D_r is of small range and satisfies $|Im(f)| \leq r$. Let also D_{perm} be the uniform distribution on permutations on $[n]$. Zhandry's lower bound can be stated as follows:

▶ **Proposition 6** ([11]). *There exists an absolute constant Λ such that for any $r \in [n]$ and any quantum query algorithm $\mathcal{B}^{\mathcal{O}}$ performing at most $\lceil \Lambda r^{1/3} \rceil$ queries to $\mathcal{O}$:*

$$\forall b \in \{0,1\},\ \left|\mathbb{E}_{\pi \leftarrow D_{perm}} \Pr[\mathcal{B}^{\mathcal{O}_\pi} \text{ outputs } b] - \mathbb{E}_{C \leftarrow D_r} \Pr[\mathcal{B}^{\mathcal{O}_C} \text{ outputs } b]\right| \leq \frac{2}{27}.$$

This is obtained immediately by combining Theorem 8 and Lemma 1 of [11][1].

3 Proving our main theorem

The goal of this section is to prove Theorem 3. Fix a function $f : S \to \{0,1\}$ where $S \subseteq [M]^n$ with $Q(f) = q$. This means there exists a quantum query algorithm $\mathcal{A}^{\mathcal{O}}$ performing q queries to $\mathcal{O}$ such that

$$\forall x \in S,\ \Pr[\mathcal{A}^{\mathcal{O}_x} \text{ outputs } f(x)] \geq 2/3.$$

We first amplify the success probability to 20/27.

▶ **Lemma 7.** *There exists a quantum query algorithm $\mathcal{A}_3^{\mathcal{O}}$ that performs $3q$ queries to $\mathcal{O}$ such that*

$$\forall x \in S,\ \Pr[\mathcal{A}_3^{\mathcal{O}_x} \text{ outputs } f(x)] \geq \frac{20}{27}.$$

Proof. $\mathcal{A}_3^{\mathcal{O}}$ will consist of the following: run $\mathcal{A}^{\mathcal{O}}$ independently 3 times and take the output that occurs the most. For each x, each run of $\mathcal{A}^{\mathcal{O}_x}$ outputs $f(x)$ with probability greater than $2/3$. The probability that the correct $f(x)$ appears at least twice out of the 3 results is therefore greater than $\frac{8}{27} + 3 \cdot \frac{4}{27} = \frac{20}{27}$. ◀

Using the fact that f is permutation symmetric, we get the following corollary:

▶ **Corollary 8.**

$$\forall x \in S,\ \forall \pi \in S_n,\ \Pr[\mathcal{A}_3^{\mathcal{O}_{x\circ\pi}} \text{ outputs } f(x)] = \Pr[\mathcal{A}_3^{\mathcal{O}_{x\circ\pi}} \text{ outputs } f(x \circ \pi)] \geq \frac{20}{27}.$$

[1] Equivalently, this is obtained immediately by combining Lemma 3.2 and Lemma 3.4 from the arXiv version `quant-ph:1312.1027`.

3.1 Looking at a small number of indices of x

The main idea of the proof is to show that $\mathcal{A}_3$ will output $f(x)$ with high probability when replacing queries to $\mathcal{O}_x$ with queries to $\mathcal{O}_{x \circ C}$ for C chosen uniformly from D_r for some $r = \Theta(Q^3(f))$. First notice that for any $x : [n] \rightarrow [M]$ and any $g : [n] \rightarrow [n]$, it is possible to apply $\mathcal{O}_{x \circ g}$ with 2 calls to $\mathcal{O}_g$ and 1 call to $\mathcal{O}_x$ with the following procedure:

$$|i\rangle|j\rangle|0\rangle \rightarrow |i\rangle|j\rangle|g(i)\rangle \rightarrow |i\rangle|j + (x \ \circ g)(i)\rangle|g(i)\rangle \rightarrow |i\rangle|j + (x \circ g)(i)\rangle|0\rangle$$

where we respectively apply $\mathcal{O}_g$ on registers $(1,3)$; $\mathcal{O}_x$ on registers $(3,2)$ and $\mathcal{O}_g^\dagger$ on registers $(1,3)$.

Therefore, for any fixed (and known) x, for any function $g : [n] \rightarrow [n]$, we can look at $\mathcal{A}_3^{\mathcal{O}_{x \circ g}}$ as a quantum query algorithm that queries $\mathcal{O}_g$. In other words, for each $x \in S$, there is a quantum query algorithm $\mathcal{B}_x^{\mathcal{O}}$ such that $\mathcal{B}_x^{\mathcal{O}_g} = \mathcal{A}^{\mathcal{O}_{x \circ g}}$ for any function $g : [n] \rightarrow [n]$. Notice also that since a query to $\mathcal{O}_{x \circ g}$ is done by doing 2 queries to $\mathcal{O}_g$, we have that $\mathcal{B}^{\mathcal{O}}$ uses twice as many queries than $\mathcal{A}_3^{\mathcal{O}}$.

We can now prove our main proposition that shows that we can compute $f(x)$ by looking only at $x \circ C$ for a random C with $|Im(C)| \leq r$.

▶ **Proposition 9.** *Let $f : [M]^n \rightarrow \{0,1\}$ with $Q(f) = q$ and $r = \lceil 216 q^3 \Lambda^{-3} \rceil$ where Λ is the absolute constant from Proposition 6.*

$$\forall x \in S, \ \mathbb{E}_{C \leftarrow D_r} \Pr[\mathcal{A}_3^{\mathcal{O}_{x \circ C}} \textit{ outputs } f(x)] \geq 2/3.$$

Proof. For each $x \in S$, we consider the algorithm $\mathcal{B}_x^{\mathcal{O}}$ described above. Recall that for all $g : [n] \rightarrow [n]$, $\mathcal{B}_x^{\mathcal{O}_g} = \mathcal{A}_3^{\mathcal{O}_{x \circ g}}$. Since $\mathcal{A}_3^{\mathcal{O}}$ uses $3q$ queries, $\mathcal{B}_x^{\mathcal{O}}$ uses $6q$ queries. We first consider the case where g is a random permutation. Using Corollary 8:

$$\forall x \in S, \ \mathbb{E}_{\pi \leftarrow D_{\text{perm}}} \Pr[\mathcal{B}_x^{\mathcal{O}_\pi} \text{ outputs } f(x)] = \mathbb{E}_{\pi \leftarrow D_{\text{perm}}} \Pr[\mathcal{A}_3^{\mathcal{O}_{x \circ \pi}} \text{ outputs } f(x)] \geq \frac{20}{27}$$

Using the lower bound of Proposition 6 noticing that $6q \leq \Lambda r^{1/3}$, we have

$$\forall x \in S, \ \left|\mathbb{E}_{\pi \leftarrow D_{\text{perm}}} \Pr[\mathcal{B}_x^{\mathcal{O}_\pi} \text{ outputs } f(x)] - \mathbb{E}_{C \leftarrow D_r} \Pr[\mathcal{B}_x^{\mathcal{O}_C} \text{ outputs } f(x)]\right| \leq \frac{2}{27}.$$

which gives us

$$\forall x \in S, \ \mathbb{E}_{C \leftarrow D_r} \Pr[\mathcal{B}_x^{\mathcal{O}_C} \text{ outputs } f(x)] \geq \frac{20}{27} - \frac{2}{27} = 2/3.$$

Since for each $x \in S$, $\mathcal{B}_x^{\mathcal{O}_C} = \mathcal{A}_3^{x \circ C}$, we can therefore conclude

$$\forall x \in S, \ \mathbb{E}_{C \leftarrow D_r} \Pr[\mathcal{A}_3^{\mathcal{O}_{x \circ C}} \text{ outputs } f(x)] \geq 2/3. \qquad ◀$$

3.2 Constructing a classical query algorithm for f

We can now use the above proposition to prove our main theorem.

▶ **Theorem 2** (Restated). *For any permutation symmetric function f of the first type, $R(f) \leq O(Q^3(f))$.*

Proof. Fix a function $f : S \rightarrow \{0,1\}$ where $S \subseteq [M]^n$ with $Q(f) = q$. This means there exists a quantum query algorithm $\mathcal{A}^{\mathcal{O}}$ performing q queries to $\mathcal{O}$ such that

$$\forall x \in S, \ \Pr[\mathcal{A}^{\mathcal{O}_x} \text{ outputs } f(x)] \geq 2/3.$$

We construct a randomized algorithm that performs $r = \lceil 216 q^3 \Lambda^{-3} \rceil$ classical queries to $\mathcal{O}_x^{\text{Classical}}$ as follows:

1. Choose a random C according to distribution D_r.
2. Query $\mathcal{O}_x^{\text{Classical}}$ to get all values x_i for $i \in Im(C)$. This requires $|Im(C)| \leq r$ queries to $\mathcal{O}_x^{\text{Classical}}$. These queries fully characterize the function $x \circ C$, hence the quantum unitary $\mathcal{O}_{x \circ C}$.
3. From $\mathcal{A}^{\mathcal{O}}$, construct the quantum algorithm $\mathcal{A}_3^{\mathcal{O}}$ as in Lemma 7. Recall that $\mathcal{A}_3^{\mathcal{O}}$ just consists of applying $\mathcal{A}^{\mathcal{O}}$ independently 3 times and output the majority outcome.
4. We consider $\mathcal{A}_3^{\mathcal{O}_{x \circ C}}$ as a quantum unitary circuit acting on t qubits. At each step of the algorithm, we store the 2^t amplitudes. When $\mathcal{O}_{x \circ C}$ is called, we use its representation from step 2 to calculate its action on the 2^t amplitudes. Other parts of $\mathcal{A}_3^{\mathcal{O}_{x \circ C}}$ are treated the same way. While this uses a lot of computing power, it does not require any queries to $\mathcal{O}_x^{\text{Classical}}$ or $\mathcal{O}_x$ other than those used at step 2.

Step 4 outputs the same output distribution than the quantum algorithm $\mathcal{A}_3^{\mathcal{O}_{x \circ C}}$. Using Proposition 9, for all $x \in S$, this algorithm outputs $f(x)$ with probability at least $2/3$, which implies

$$R(f) \leq r = \lceil 216 Q^3(f) \Lambda^{-3} \rceil. \qquad ◀$$

Notice that after step 2, it is not possible to just compute $f(x \circ C)$, and try to show that it is equal to $f(x)$ since we don't even always have $x \circ C \in S$. This is yet another example in query complexity where we use the behavior of a query algorithm on inputs not necessarily in the domain of f.

4 Conclusion

This result extends the class of functions for which we can show a polynomial relationship between the quantum and the randomized query complexity and improves the polynomial in general for permutation symmetric functions.

The first obvious open question is to close the gap between the best known speed-up for permutation symmetric function - which is quadratic - and the cubic lower bound obtained in this paper. Another open question is to see if such techniques can be extended to the case where the domain S is permutation symmetric, which implies the case of total functions. While the techniques seem specific to permutation symmetric functions, using a more powerful lower bound or considering inputs x in superposition (as in [12]) could give interesting results.

Also, we are currently extending those techniques to study the behavior of uniformly random inputs x in particular in the context of the quantum random oracle model. Here, we are interested in the power of quantum attacks on a cryptographic scheme while performing quantum queries to a uniformly random function. This technique seems promising to show that for many attacks, these quantum queries can be replaced with classical queries in the same way as in our steps 2-4.

References

1 Scott Aaronson and Andris Ambainis. The Need for Structure in Quantum Speedups. *Theory of Computing*, 10(6):133–166, 2014. doi:10.4086/toc.2014.v010a006.

2 Scott Aaronson and Yaoyun Shi. Quantum Lower Bounds for the Collision and the Element Distinctness Problems. *J. ACM*, 51(4):595–605, July 2004. doi:10.1145/1008731.1008735.

3 A. Ambainis. Understanding Quantum Algorithms via Query Complexity. *ArXiv e-prints*, December 2017. arXiv:1712.06349.

4 Andris Ambainis. Polynomial Degree and Lower Bounds in Quantum Complexity: Collision and Element Distinctness with Small Range. *Theory of Computing*, 1(3):37–46, 2005. doi:10.4086/toc.2005.v001a003.

5 Robert Beals, Harry Buhrman, Richard Cleve, Michele Mosca, and Ronald de Wolf. Quantum Lower Bounds by Polynomials. *J. ACM*, 48(4):778–797, July 2001. doi:10.1145/502090.502097.

6 Charles H. Bennett, Ethan Bernstein, Gilles Brassard, and Umesh Vazirani. Strengths and Weaknesses of Quantum Computing. *SIAM Journal of Computing*, 26(5):1510–1523, October 1997. doi:10.1137/S0097539796300933.

7 D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 439(1907):553–558, 1992. doi:10.1098/rspa.1992.0167.

8 Samuel Kutin. Quantum Lower Bound for the Collision Problem with Small Range. *Theory of Computing*, 1(2):29–36, 2005. doi:10.4086/toc.2005.v001a002.

9 Peter W. Shor. Algorithms for Quantum Computation: Discrete Logarithms and Factoring. In *IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994. URL: citeseer.ist.psu.edu/14533.html.

10 Daniel R. Simon. On the Power of Quantum Cryptography. In *35th Annual Symposium on Foundations of Computer Science, Santa Fe, New Mexico, USA, 20-22 November 1994*, pages 116–123, 1994. doi:10.1109/SFCS.1994.365701.

11 Mark Zhandry. A Note on the Quantum Collision and Set Equality Problems. *Quantum Info. Comput.*, 15(7-8):557–567, May 2015. URL: http://dl.acm.org/citation.cfm?id=2871411.2871413.

12 Mark Zhandry. How to Record Quantum Queries, and Applications to Quantum Indifferentiability. *IACR Cryptology ePrint Archive*, 2018:276, 2018. URL: https://eprint.iacr.org/2018/276.