A Note About Claw Function with a Small Range

Andris Ambainis
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Kaspars Balodis
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Jānis Iraids
Center for Quantum Computer Science, Faculty of Computing, University of Latvia, Riga, Latvia

Abstract

In the claw detection problem we are given two functions $f : D \to R$ and $g : D \to R$ ($|D| = n$, $|R| = k$), and we have to determine if there is exist $x, y \in D$ such that $f(x) = g(y)$. We show that the quantum query complexity of this problem is between $\Omega(n^{1/2}k^{1/6})$ and $O(n^{1/2+k/4})$ when $2 \leq k < n$.

1 Introduction

In this note we study the CLAW problem in which given two discrete functions $f : D \to R$ and $g : D \to R$ ($|D| = n$, $|R| = k$) we have to determine if there is a collision, i.e., inputs $x, y \in D$ such that $f(x) = g(y)$. In contrast to the ELEMENT-DISTINCTNESS problem, where the input is a single function $f : D \to R$ and we have to determine if $f$ is injective, CLAW is non-trivial even when $k < n$. This is the setting we focus on.

Both CLAW and ELEMENT-DISTINCTNESS have wide applications as useful subroutines in more complex algorithms [5, 12] and as a means of lower bounding complexity [10, 1].

CLAW and ELEMENT-DISTINCTNESS were first tackled by Buhrman et al. in 2000 [8] where they gave an $O(n^{1/4})$ algorithm and $\Omega(n^{1/2})$ lower bound. In 2003 Ambainis, introducing a novel technique of quantum walks, improved the upper bound to $O(n^{3/5})$ in the query model [4]. It was soon realized that a similar approach works for CLAW [9, 13, 15]. Meanwhile Aaronson and Shi showed a lower bound $\Omega(n^{2/3})$ that holds if the range $k = \Omega(n^2)$ [2]. Eventually Ambainis showed that the $\Omega(n^{2/3})$ bound holds even if $k = n$ [3]. The same lower bound has since been reproved using the adversary method [14]. Until now, only the $\Omega(n^{1/2})$ bound based on reduction of searching was known for CLAW with $k = o(n)$ [8].

We consider quantum query complexity of CLAW where the input functions are given as a list of their values in black box. Let $Q(f)$ denote the bounded error quantum query complexity of $f$. For a short overview of black box model refer to Buhrman and de Wolf’s survey [7]. Let $[n]$ denote $\{1, 2, \ldots, n\}$. Let $\text{CLAW}_{n \to k} : [k]^{2n} \to \{0, 1\}$ be defined as

$$\text{CLAW}_{n \to k}(x_1, \ldots, x_n, y_1, \ldots, y_n) = \begin{cases} 1, & \text{if } \exists i, j \ x_i = y_j \\ 0, & \text{otherwise} \end{cases}.$$

Our contribution is a quantum algorithm for $\text{CLAW}_{n \to k}$ with quantum query complexity $Q(\text{CLAW}_{n \to k}) = O(n^{1/2+k/4})$ and a lower bound $Q(\text{CLAW}_{n \to k}) = \Omega(n^{1/2+k/6})$. In section 2 we describe the algorithm, and in section 3 we give the lower bound.
2 Results

Theorem 1. For all \( \epsilon > 0 \), we have \( Q(\text{CLA}\text{W}_{n\to k}) = O\left(n^{3/2+\epsilon}k^{1/4}\right) \).

Proof. Let \( X = (x_1, \ldots, x_n) \), \( Y = (y_1, \ldots, y_n) \) be the inputs of the function. We denote \( k = n^\alpha \).

Consider the following algorithm parametrized by \( \alpha \in [0, 1] \).

1. a. Select a random sample \( A = \{a_1, \ldots, a_\ell\} \subseteq [n] \) of size \( \ell = 4 \cdot n^\alpha \cdot \ln n \) and query the variables \( x_{a_1}, \ldots, x_{a_\ell} \).
Denote by \( X_A = \{x_a \mid a \in A\} \) the set containing their values. Do a Grover search for an element \( y \in Y \) such that \( y \in X_A \). If found, output 1.

b. Select a random sample \( A' = \{a'_1, \ldots, a'_\ell\} \subseteq Y \) of size \( \ell \) and query the variables \( y_{a'_1}, \ldots, y_{a'_\ell} \).
Denote by \( Y_{A'} = \{y_{a'} \mid a' \in A'\} \) the set containing their values. Do a Grover search for an element \( x \in X \) such that \( x \in Y_{A'} \). If found, output 1.

2. Run CLA\text{W}_{4b\ln n\to k} \) algorithm (with the value of \( b \) specified below) with the following oracle:

a. To get \( x_i \): do a pseudorandom permutation on \( x_1, \ldots, x_n \) using seed \( i \) and using Grover’s minimum search return the first value \( x_j \) such that \( x_j \notin X_A \).

b. To get \( y_i \): do a pseudorandom permutation on \( y_1, \ldots, y_n \) using seed \( i \) and using Grover’s minimum search return the first value \( y_j \) such that \( y_j \notin Y_{A'} \).

Let \( B = \{i \in [n] \mid x_i \notin X_A\} \), \( B' = \{i \in [n] \mid y_i \notin Y_{A'}\} \) be the sets containing the indices of the variables which have values not seen in the steps 1a and 1b. We denote \( |B| = b = n^\beta \).

Let us calculate the probability that after step 1a there exists an unseen value \( v \) which is represented in at least \( n^{1-\alpha} \) variables, i.e., \( v \notin X_B \land \{|i \in [n] \mid x_i = v\} \geq n^{1-\alpha} \). Consider an arbitrary value \( v^* \in [k] \) such that \( \{|i \mid x_i = v^*\} \geq n^{1-\alpha} \). For \( i \in [\ell] \), let \( Z_i \) be the event that \( x_{a_i} = v^*. \forall i \in [\ell] \) \( \Pr[Z_i] \geq \frac{2^{1-\alpha}}{n^{\alpha}} \).

Let \( Z = \sum_{i \in [\ell]} Z_i \). Then \( \mathbb{E}[Z] = \ell \cdot \mathbb{E}[Z_i] \geq 4 \cdot n^\alpha \cdot \ln n \cdot \frac{n^{1-\alpha}}{n} = 4 \ln n \). Using Chernoff inequality (see e.g. [11]),

\[
\Pr[Z = 0] \leq \exp\left(-\frac{1}{2} \mathbb{E}[Z]\right) \leq \exp(-2 \ln n) = \frac{1}{n^2}.
\]

The probability that there exists such \( v^* \in [k] \) is at most \( \frac{n^\alpha}{n^2} = o(1) \). Therefore, with probability \( 1 - o(1) \) after step 1a, every value \( v \in X_B \) is represented in the input less than \( n^{1-\alpha} \) times. The same reasoning can be applied to step 1b and the set \( B' \). Therefore, with probability \( 1 - o(1) \) both \( b \) and \( b' \) are at most \( k \cdot n^{1-\alpha} = n^{\alpha+1-\alpha} \).

Similarly, we show that with probability \( 1 - o(1) \) each \( x \in B \) appears as the first element from \( B \) in at least one of the permutations of the oracle in step 2. Let \( W^x_i \) be the event that \( x \in B \) appears in the \( i \)-th permutation as the first element from \( B \). \( \mathbb{E}[W^x_i] = \frac{1}{n} \).

Let \( W^x = \sum_{i \in [4b \ln n]} W^x_i \). \( \mathbb{E}[W^x] = 4b \ln n \cdot \frac{1}{n} = 4 \ln n \). \( \Pr[W^x = 0] \leq \exp(-2 \ln n) = \frac{1}{n^2} \).

\( \Pr[\exists x \in B : W^x = 0] \leq \frac{2^\alpha}{n^\alpha} = \frac{1}{n^\alpha} = o(1) \). The same argument works for \( B' \). Therefore, if there is a collision, it will be found by the algorithm with probability \( 1 - o(1) \).

We also show that with probability \( 1 - o(1) \), in all permutations the first element from \( B \) appears no further than in position \( \frac{4b \ln n}{n} \) (and similarly for \( B' \)). We denote \( P_{i,j} \) the event that in the \( i \)-th permutation in the \( j \)-th position is an element from \( B \). \( \mathbb{E}[P_{i,j}] = \frac{1}{n} \).

We denote \( P_i = \sum_{j \in [\frac{4b \ln n}{n}]} P_{i,j} \). \( \mathbb{E}[P_i] = 4 \cdot \ln n \). \( \Pr[P_i = 0] \leq \exp(-2 \ln n) = \frac{1}{n^2} \).

\( \Pr[\exists i \in [4b \ln n] : P_i = 0] \leq \frac{4b \ln n}{n^2} \leq \frac{4n \ln n}{n^2} = o(1) \). Therefore, the Grover’s minimum search will use at most \( \tilde{O}\left(\sqrt{n^2}\right) \) queries.
The steps 1a and 1b use $\tilde{O}(n^\alpha)$ queries to obtain the random sample, and $O(\sqrt{n})$ queries to check if there is a colliding element on the other side of the input. The oracle in step 2 uses $\tilde{O}(\sqrt{\frac{\kappa}{\alpha+1}})$ queries to obtain one value of $x_i$ or $y_i$.

Therefore the total complexity of the algorithm is

$$\tilde{O}\left(n^\alpha + n^{\frac{3}{2}} + Q(\text{CLAW}_{4b \ln n - k}) \cdot n^{\frac{\beta}{2} - \frac{1}{2}\beta}\right).$$

By using the $O(n^{2\beta})$ algorithm in step 2,

$$Q(\text{CLAW}_{4b \ln n - k}) \cdot n^{\frac{3}{2} - \frac{1}{2}\beta} = n^{\frac{3}{2}\beta + \frac{1}{2} - \frac{1}{2}\beta} = n^{\frac{3}{2} + \beta} \leq n^{\frac{2}{\alpha+\frac{\kappa}{\alpha+1}}} \cdot n^{\frac{\alpha+\kappa}{\alpha+1}} = n^{\frac{\alpha+\kappa}{\alpha}},$$

and the total complexity is minimized by setting $\alpha = \frac{4+\kappa}{7}$. However, we can do better than that. Notice that the $O(n^{2\beta})$ algorithm might not be the best choice for solving CLAW$_{4b \ln n - k}$ in step 2.

Let $A_0$ denote the regular $O(n^{2\beta})$ CLAW$_{n-k}$ algorithm. For $i > 0$, let $A_i$ denote a version of algorithm from Theorem 1 that in step 2 calls $A_{i-1}$. Then we show that for all $n$ and all $0 \leq \varkappa \leq \frac{2}{3}$,

$$Q(A_i) = \tilde{O}\left(n^{T_i(\varkappa)}\right),$$

where $T_i(\varkappa) = \frac{(2^{i-1})\varkappa + 2^{i+1} + 1}{2^{i+1} - 1}$.

The proof is by induction on $i$. For $i = 0$, we trivially have that $Q(A_0) = \tilde{O}(n^{2\beta})$. For the inductive step, consider the analysis of our algorithm. Let us set $\alpha = T_i(\varkappa)$. First, notice that $T_i(\varkappa)$ is non-decreasing in $\varkappa$ and $T_i\left(\frac{2}{3}\right) = \frac{2}{3}$ for all $i$. Thus for all $\varkappa \leq \frac{2}{3}$, we have $T_i(\varkappa) \leq \frac{2}{3}$, hence $\alpha \leq \frac{2}{3}$ and $\frac{\varkappa}{1-\alpha+\varkappa} \leq \frac{2}{3}$. Second, since the coefficient of $\varkappa$ is $\frac{2^{i-1}}{2^{i+1} - 1}$ the function $T_i(\varkappa)$ is above $\varkappa$ for $\varkappa \leq \frac{2}{3}$, establishing $\alpha - \varkappa \geq 0$. This confirms that $\alpha = T_i(\varkappa)$ is a valid choice of $\alpha$.

It remains to show that the complexity of step 2 does not exceed $\tilde{O}(n^{T_i(\varkappa)})$. By the inductive assumption and analysis of the algorithm, the complexity (up to logarithmic factors) of the second step is $n$ to the power of $(1 - \alpha + \varkappa) \cdot T_{i-1}\left(\frac{\varkappa}{1-\alpha+\varkappa}\right) + \frac{\alpha - \varkappa}{2}$. Finally, we have to show that

$$(1 - T_i(\varkappa) + \varkappa) \cdot T_{i-1}\left(\frac{\varkappa}{1 - T_i(\varkappa) + \varkappa}\right) + \frac{T_i(\varkappa) - \varkappa}{2} \leq T_i(\varkappa).$$

By expanding $T_{i-1}(\varkappa)$ and with a slight rearrangement, we obtain

$$\frac{(2^{i-1} - 1)\varkappa + 2^i(1 - T_i(\varkappa) + \varkappa)}{2^{i+1} - 1} \leq T_i(\varkappa) + \varkappa.$$

We can further rearrange the required inequality by bringing $T_i(\varkappa)$ to right hand side and everything else to the other. Then we get

$$\frac{(2^{i-1} - 1 + 2^i - 2^{i+1} - 1)\varkappa + 2^i}{2^{i+1} - 1} \leq T_i(\varkappa)\left(\frac{1}{2} + \frac{2^i}{2^{i+1} - 1}\right).$$

After simplification we obtain $\frac{(2^{i-1})\varkappa + 2^{i+1}}{2^{i+1} - 1} \leq T_i(\varkappa)$, which is true.

Since $\lim_{i \to \infty} \frac{2^i - 1}{2^{i+1} - 1} = \frac{1}{4}$ and $\lim_{i \to \infty} \frac{2^{i+1}}{2^{i+1} - 1} = \frac{1}{2}$, the result follows. ▶
3 Lower Bound

We show a $\Omega(n^{1/2}k^{1/6})$ quantum query complexity lower bound for $\text{CLAW}_{n \rightarrow k}$.

**Theorem 2.** For all $k \geq 2$, we have $Q(\text{CLAW}_{n \rightarrow k}) = \Omega(n^{1/2}k^{1/6})$.

**Proof.** Let $p\text{Search}_m : (\ast \cup [k])^m \rightarrow [k]$ be the partial function defined as

$$p\text{Search}_m(x_1, x_2, \ldots, x_m) = \begin{cases} x_i, & \text{if } x_i \neq \ast, \forall j \neq i : x_j = \ast \\ \text{undefined}, & \text{otherwise} \end{cases}$$

Consider the function $f_{n,k} = \text{CLAW}_{k \rightarrow k} \circ p\text{Search}_{\lfloor n/k \rfloor}$. One can straightforwardly reduce $f_{n,k}(x,y)$ to $\text{CLAW}_{n \rightarrow k+2}(x', y')$ by setting

$$x'_i = \begin{cases} x_i, & \text{if } x_i \neq \ast \\ k+1, & \text{if } x_i = \ast \end{cases}$$

and

$$y'_i = \begin{cases} y_i, & \text{if } y_i \neq \ast \\ k+2, & \text{if } y_i = \ast \end{cases}$$

Now we show that $Q(f_{n,k}) = \Omega(k^{2/3} \sqrt{n/k}) = \Omega(n^{1/2}k^{1/6})$. The fact that $Q(\text{CLAW}_{k \rightarrow k}) = \Omega(k^{2/3})$ has been established by Zhang [16]. Furthermore, thanks to the work done by Brassard et al. in [6, Theorem 13] we know that for $p\text{Search}_m$ a composition theorem holds:

$$Q(h \circ p\text{Search}_m) = \Omega(Q(h) \cdot Q(p\text{Search}_m)) = \Omega(Q(h) \cdot \sqrt{m})$$

Therefore,

$$Q(\text{CLAW}_{n \rightarrow k}) \geq Q(\text{CLAW}_{k \rightarrow k+2} \circ p\text{Search}_{\lfloor n/k \rfloor}) = \Omega(k^{2/3} \sqrt{n/k}) = \Omega(n^{1/2}k^{1/6})$$

4 Open Problems

Can we show that $Q(\text{CLAW}_{n \rightarrow n^{2/3}}) = \Omega(n^{2/3})$? In particular, our algorithm struggles with instances where there are $\frac{n^{2/3}}{2}$ singletons only two (or none) of which are matching and the remaining variables are evenly distributed with $\Theta(n^{1/3})$ copies each, such that none are matching. Thus our algorithm then either has to waste time sampling all the high-frequency decoy values or have most variables not sampled by step 2. If this lower bound held, it would imply a better lower bound for evaluating constant depth formulas and Boolean matrix product verification [10, Theorem 5].

References

1. Scott Aaronson, Nai-Hui Chia, Han-Hsuan Lin, Chunhao Wang, and Ruizhe Zhang. On the Quantum Complexity of Closest Pair and Related Problems. In Shubhangi Saraf, editor, 35th Computational Complexity Conference (CCC 2020), volume 169 of Leibniz International Proceedings in Informatics (LIPIcs), pages 16:1–16:43, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CCC.2020.16.

2. Scott Aaronson and Yaoyun Shi. Quantum lower bounds for the collision and the element distinctness problems. Journal of the ACM (JACM), 51(4):595–605, 2004.
Andris Ambainis. Polynomial degree and lower bounds in quantum complexity: Collision and element distinctness with small range. *Theory of Computing*, 1(1):37–46, 2005.

Andris Ambainis. Quantum walk algorithm for element distinctness. *SIAM Journal on Computing*, 37(1):210–239, 2007.

Daniel J. Bernstein, Stacey Jeffery, Tanja Lange, and Alexander Meurer. Quantum algorithms for the subset-sum problem. In Philippe Gaborit, editor, *Post-Quantum Cryptography*, pages 16–33, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

Gilles Brassard, Peter Høyer, Kassem Kalach, Marc Kaplan, Sophie Laplante, and Louis Salvail. Key establishment à la merkle in a quantum world. *Journal of Cryptology*, 32(3):601–634, 2019. doi:10.1007/s00145-019-09317-z.

Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. Complexity and Logic. doi:10.1016/S0304-3975(01)00144-X.

Harry Buhrman, Christoph Dürr, Mark Heiligman, Peter Høyer, Frédéric Magniez, Miklos Santha, and Ronald de Wolf. Quantum algorithms for element distinctness. *SIAM Journal on Computing*, 34(6):1324–1330, 2005. doi:10.1137/S0097539702402780.

Andrew M. Childs and Jason M. Eisenberg. Quantum algorithms for subset finding. *Quantum Inf. Comput.*, 5(7):593–604, 2005.

Andrew M. Childs, Shelby Kimmel, and Robin Kothari. The quantum query complexity of read-many formulas. In *Proceedings of the 20th Annual European Conference on Algorithms, ESA’12*, pages 337–348, Berlin, Heidelberg, 2012. Springer-Verlag. doi:10.1007/978-3-642-33090-2_30.

Fan Chung and Linyuan Lu. Concentration inequalities and martingale inequalities: a survey. *Internet Mathematics*, 3(1):79–127, 2006.

François Le Gall and Saeed Seddighin. Quantum meets fine-grained complexity: Sublinear time quantum algorithms for string problems, 2020. arXiv:2010.12122.

Frédéric Magniez, Miklos Santha, and Mario Szegedy. Quantum algorithms for the triangle problem. *SIAM Journal on Computing*, 37(2):413–424, 2007. doi:10.1137/050643684.

Ansis Rosmanis. Adversary lower bound for element distinctness with small range, 2014. arXiv:1401.3826.

Seiichiro Tani. Claw finding algorithms using quantum walk. *Theoretical Computer Science*, 410(50):5285–5297, 2009. Mathematical Foundations of Computer Science (MFCS 2007). doi:10.1016/j.tcs.2009.08.030.

Shengyu Zhang. Promised and distributed quantum search. In Lusheng Wang, editor, *Computing and Combinatorics*, pages 430–439, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.