Generalized Quantum Deutsch-Jozsa Algorithm

Tomasz Arodz

Department of Computer Science, Virginia Commonwealth University,
Richmond, VA 23284, USA
tarodz@vcu.edu

Abstract. Quantum computing aims to provide algorithms and hardware that allows for solving computational problems asymptotically faster than on classical computers. Yet, design of new, fast quantum algorithms is not straightforward, and the field faces high barriers of entry for traditional computer scientists. One of the main didactic examples used to introduce speedup resulting from quantum computing is the Deutsch-Jozsa algorithm for discriminating between constant and balanced functions. Here, we show a generalization of the Deutsch-Jozsa algorithm beyond balanced functions that can be used to further illustrate the design choices underpinning quantum algorithms.

Keywords: Quantum speedup · Promise problems · Didactics of quantum computing.

1 Introduction

Quantum computing studies algorithms and hardware for performing computation using systems that exploit quantum physics. In the gate model of quantum computing [6], the information storage and processing are done using tools from linear algebra. The information is stored in a quantum register composed of quantum bits. Each quantum bit is represented as a unit-norm vector in a two-dimensional complex Hilbert space. A multi-qubit register is modeled as a tensor product space arising from the individual quantum bits. The information in the quantum register can evolve in time according to invertible, inner product-preserving transformations, modeled mathematically in the gate model of quantum computing as unitary operators. The information in the quantum register can be accessed, to some extent, through quantum measurement, typically modeled as a randomized projection on vectors from the computational basis of the Hilbert space.

The key promise of quantum computing is to achieve speedup compared to classical computers [10], leading to faster algorithms in many domains, including linear algebra [5], database search [4], or machine learning [1, 9, 3]. The study of quantum algorithms can also lead to more efficient classical methods [11, 7].

One of the first, didactic example of a problem for which a quantum computer abstracted using the gate model shows speedup is a promise problem known as the Deutsch-Jozsa problem [2]. Assume that you are given a Boolean function

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36

on n-bits: $f : \{0,1\}^n \to \{0,1\}$ with a black-box access. That is, on a classical computer, you can call it on any bit string and see the result, but you cannot decompile it. On a quantum computer, you have access to an oracle, a unitary transformation U_f that performs the function using some form of input and output encoding.

In the Deutsch-Jozsa problem, we are promised that the function is of one of two types

- *constant*, that is, always returns 0, or always returns 1,
- *balanced*, that is, for half of the 2^n possible inputs it returns 0, for the remaining 2^{n-1} inputs, it returns 1.

The task is to use the ability to execute $f(x)$ for any x to figure out if f is constant, or balanced.

Let $N = 2^n$ be the number of distinct inputs x. On a classical computer, we need at least two calls to f to be able to decide if the function is constant or balanced – in the most optimistic scenario when first call returns 0 and the second returns 1, we know the answer after these two calls to f. But if we see zeros all the time, we need $N/2+1$ calls to have the answer – if value number $N/2+1$ is also 0, it is a constant function, if the value is 1, it is a balanced function. Thus, pessimistically, we need $O(N) = O(2^n)$ calls to f to solve the Deutsch-Jozsa problem on a classical computer. It is well-known that on a quantum computer, we can solve the Deutsch-Jozsa problem much quicker, using just one call to the unitary oracle U_f.

The Deutsch-Jozsa algorithm has been recently generalized to include discrimination between balanced functions and almost-constant functions [8], with the query complexity increasing with the distance from a constant function. Here, we show that it can be generalized in a different way, to a family of promise problems involving discrimination between a specific constant function f_l, for example an all-zero function, and a family of functions f_k that have fixed level of imbalance, that is, have exactly k outputs equal to one, as long as $k \geq N/2$. We show that this problem can be solved with only one query to the function oracle.

2 Quantum Deutsch-Jozsa Algorithm

To define the Deutsch-Jozsa algorithm, we need to have ability to evaluate Boolean functions using unitary operators, and the ability to use Boolean function evaluation to provide the answer whether the function is constant or balanced.

2.1 Quantum Boolean Function Evaluation

A Boolean function on n bits returning an m bit string is a function $f : \{0,1\}^n \to \{0,1\}^m$. Consider $n = m$, and a function $f(x) = NOTx$ that negates all bits of x. It is a permutation - a one-to-one mapping - on the set $\{0,1\}^n$. Now consider

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36

a function $f_y(x) = y$ XOR $x = y \oplus x$, where $\oplus$ is a modulo-two addition. If a bit in y is 0, the corresponding bit in x is not changed – an identity mapping, a particular form of permutation, on those bits. If a bit in y is 1, the corresponding bit in x is negated - a permutation on those bits. That is, for arbitrary n-bit y, $x \to y \oplus x$ is a permutation on the set $\{0,1\}^n$ - $|y \oplus x\rangle$ vectors for all possible x are mutually orthogonal, same as $|x\rangle$ are. Then, a linear mapping $U_y = \sum_{x \in \{0,1\}^n} |y \oplus x\rangle \langle x|$ will produce $|y \oplus x\rangle$ when we perform $U_y |x\rangle$. Since $y \oplus x$ is a permutation on $\{0,1\}^n$, it is a one-to-one mapping. It also preserves the inner product. The inner product among basis vectors $|x\rangle$, $x \in \{0,1\}^n$ is null, and the inner products among the outputs $U_y |x\rangle$ are also null. This mapping is inner-product-preserving, and thus compatible with the rules of quantum mechanics.

Not all functions $f : \{0,1\}^n \to \{0,1\}^n$ are permutations; a constant function that always returns all bits set to 0 is not a permutation. Consider an arbitrary function $f : \{0,1\}^n \to \{0,1\}^m$. Let us have an $n+m$-qubit system $\mathcal{H} = (\mathbb{C}^2)^n \otimes (\mathbb{C}^2)^m$. Consider a state $|x\rangle |z\rangle$, where $|x\rangle$ is one of the 2^n basis states of $(\mathbb{C}^2)^n$, and $|z\rangle$ is one of the 2^m basis states of $(\mathbb{C}^2)^m$. We wish to have a linear transformation U_f, a unitary operator, that takes the vector, and produces, on output, a state $|x\rangle |z \oplus f(x)\rangle$, in particular, for $z = 0^m$, it produces $|x\rangle |f(x)\rangle$. For each x, $z \to z \oplus f(x)$ is a permutation on the basis states of $(\mathbb{C}^2)^m$, and thus $|x\rangle |z \oplus f(x)\rangle$ is a permutation on the basis states of $\mathcal{H}$. Hence,

$$U_f = \sum_{x,z} \left(|x\rangle |z \oplus f(x)\rangle\right) \left(\langle x| \langle z|\right),$$

which performs $|x\rangle |0\rangle \to |x\rangle |f(x)\rangle$, is a one-to-one mapping that preserves the inner product – and thus can describe unitary evolution of a quantum system.

Consider $m = 1$, arbitrary n, and an $n+1$-qubit system in an arbitrary state of the form $|\psi\rangle |0\rangle$. Let $E = HX$ be a Pauli X gate followed by Hadamard gate; it transforms $|0\rangle$ to $|-\rangle$ and $|1\rangle$ to $|+\rangle$, and is Hermitian. Let $E_n = I^{\otimes n} \otimes E$ denote application of E to the last, $n+1$ qubit. We have

$$E_n |\psi\rangle |0\rangle = |\psi\rangle |-\rangle, \qquad E_n |\psi\rangle |-\rangle = |\psi\rangle |0\rangle,$$
$$U_f |x\rangle |-\rangle = |x\rangle \frac{|f(x)\rangle - |1 \otimes f(x)\rangle}{\sqrt{2}} = |x\rangle \frac{(-1)^{f(x)}(|0\rangle - |1\rangle)}{\sqrt{2}} = (-1)^{f(x)} |x\rangle |-\rangle$$

We obtained the result by seeing that $|f(x)\rangle - |1 \otimes f(x)\rangle$ differs by a sign depending on the value of $f(x)$; if $f(x) = 0$, it is $|0\rangle - |1\rangle$, if $f(x) = 1$ it is $|1\rangle - |0\rangle$. Then we used phase kickback to the top qubits. Value of $f(x) = 1$ is reflected in change of phase of $|x\rangle$ by π, whereas $f(x) = 0$ results in no change.

We now see two ways of representing the result of applying $f(x)$ as encoded by $U_f = \sum_{x,z} (|x\rangle |z \oplus f(x)\rangle)(\langle x| \langle z|)$:

$$U_f |x\rangle |0\rangle = |x\rangle \otimes |f(x)\rangle, \qquad E_n U_f E_n |x\rangle |0\rangle = \left((-1)^{f(x)} |x\rangle\right) \otimes |0\rangle.$$

The first option is to encode the result in the state of the last qubit, the second is to encode it in the phase of the input qubits.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36

2.2 The Deutsch-Jozsa Problem

Consider four query states $\frac{|0u\rangle+|1v\rangle}{\sqrt{2}}$ for arbitrary binary u, v. Two are simple, $|FF\rangle = |+\rangle|0\rangle = \frac{|00\rangle+|10\rangle}{\sqrt{2}}$ and $|TT\rangle = |+\rangle|1\rangle = \frac{|01\rangle+|11\rangle}{\sqrt{2}}$, and correspond to two possible one-bit constant functions. Two other are entangled, $|FT\rangle = \frac{|00\rangle+|11\rangle}{\sqrt{2}}$ and $|TF\rangle = \frac{|01\rangle+|10\rangle}{\sqrt{2}}$, and correspond to the two one-bit balanced functions.

If someone prepares a two-qubit system in one of the four query states, can we distinguish whether it is any of these two $|FF\rangle$ and $|TT\rangle$, or any of these two $|TF\rangle$ and $|FT\rangle$? We have $\langle TT\,|\,FF\rangle = \langle TF\,|\,FT\rangle = 0$, that is, it is easy to distinguish $|TT\rangle$ from $|FF\rangle$, and $|TF\rangle$ from $|FT\rangle$. All other inner products of these four states are equal to 1/2. Any unitary transformation has to preserve the dimensionality of the space, and the inner products. Thus, there is no mapping that would map $|TT\rangle$ to be orthogonal to $|TF\rangle$ or $|FT\rangle$; same for $|FF\rangle$. If we want to use orthogonality of any member from one group of states to any member from the other group of states to reliably distinguish states from one group from the other, then a group composed of $|TT\rangle$ and $|FF\rangle$ cannot be reliably distinguished from group consisting of $|TF\rangle$ and $|FT\rangle$.

We have seen above two ways of representing the result of applying a binary function $f(x)$ given a state $|x\rangle$: encoding the result as an additional qubit, or as a phase change of the input qubit. Consider a n-bit binary function f. We can construct a state $|+\rangle^{\otimes n} = \sum_{x=0}^{2^n-1}|x\rangle$ using Hadamard transform. Then, we can construct a state $|+^{\otimes n}, 0\rangle = |+\rangle^{\otimes n}|0\rangle$, and apply our two options of evaluating f quantum mechanically to this $n+1$-qubit state. We will get

$$U_f\left|+^{\otimes n}, 0\right\rangle = \frac{1}{2^{\frac{n}{2}}}\sum_{x=0}^{2^n-1}\left(|x\rangle\otimes|f(x)\rangle\right),$$
$$E_n U_f E_n\left|+^{\otimes n}, 0\right\rangle = \left(\frac{1}{2^{\frac{n}{2}}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\right)\otimes|0\rangle.$$

These two options are not equivalent. The four query states we have seen above can be seen as the first option for $n = 1$ if we equate $u = f(0)$ and $v = f(1)$; indeed $\sum_{x=0}^{1}|x\rangle|f(x)\rangle = \frac{|0u\rangle+|1v\rangle}{\sqrt{2}}$. We have seen that we cannot use this representation to decide, based on orthogonality, if we got a function that is constant, or that returns equal number of 0's and 1's.

On the other hand, consider the second representation, or actually just its first qubit $\frac{1}{\sqrt{2}}\sum_{x=0}^{1}(-1)^{f(x)}|x\rangle = \frac{(-1)^u|0\rangle+(-1)^v|1\rangle}{\sqrt{2}}$. A constant function will have $\pm|+\rangle$, while a balanced function will have $\pm|-\rangle$. We can ignore the global phase, that is, the sign, and we end up with two orthogonal states, $|+\rangle$ for constant and $|-\rangle$ for balanced function. As we can see, while unitary actions after applying the black-box unitary oracle cannot change the distinguishability of states because they must preserve the inner products, differences prior to applying the function oracle can affect our ability to distinguish states.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36

3 Generalized Deutsch-Jozsa Problem

We show here that the ability to distinguish all-zeros from all-ones that we get from $U_f = U_f(I^{\otimes n} \otimes I)$ and the ability to distinguish all-zeros and all-ones from a balanced function that we get from $U_f E_n = U_f(I^{\otimes n} \otimes E)$ are not the only possibilities for quickly solving promise problems involving two groups of binary functions. We can form a class of single-qubit unitary operators A defined by the condition $|a|^2 + |b|^2 = 1$,

$$I = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad A = \begin{bmatrix} a & -b^* \\ b & a^* \end{bmatrix}, \quad E = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \\ -\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} \end{bmatrix},$$

and observe that unitaries I and E are at the extreme ends of the family, with a spectrum of other operators in between. These operators can each solve a different promise problem of discriminating between two classes of functions.

Consider two functions f_l and f_k with the corresponding black-box unitaries U_l and U_k, and let $y_j = f_k(j)$ and $x_j = f_l(j)$. Also, let $\xi_j = 1$ if $y_j = x_j$ and $\xi_j = 0$ otherwise, and $\delta_j = 1 - \xi_j$; we will use Ξ to denote the number of outputs on which the functions agree, and Δ to denote the Hamming distance between the outputs, that is, the number of outputs that differ; we have $\Xi + \Delta = 2^n$.

Let $A_n = I^{\otimes n} \otimes A$ denote application of A to the last, $n+1$-st qubit of an $n+1$-qubit system. Then, we have

$$\begin{aligned} U_k A_n \left|+^{\otimes n}, 0\right\rangle &= U_k \left(\frac{1}{2^{\frac{n}{2}}} \sum_{j=0}^{2^n-1} |j\rangle \otimes A|0\rangle \right) = \frac{1}{2^{\frac{n}{2}}} \sum_{j=0}^{2^n-1} \left(a|j\rangle|y_j\rangle + b|j\rangle|\bar{y_j}\rangle \right) \\ &= \frac{1}{2^{\frac{n}{2}}} \sum_{j=0}^{2^n-1} \left(a\bar{y_j}|j\rangle|0\rangle + ay_j|j\rangle|1\rangle + by_j|j\rangle|0\rangle + b\bar{y_j}|j\rangle|1\rangle \right) \\ &= \frac{1}{2^{\frac{n}{2}}} \sum_{j=0}^{2^n-1} \left((a\bar{y_j} + by_j)|j\rangle|0\rangle + (ay_j + b\bar{y_j})|j\rangle|1\rangle \right) \end{aligned}$$

and a similar expression for U_l, with y_j replaced by x_j.

The inner product of the two states is

$$h_\Delta = \left\langle +^{\otimes n}, 0 \left| A_n{}^\dagger U_l{}^\dagger U_k A_n \right| +^{\otimes n}, 0 \right\rangle = \frac{\Xi - \Delta\alpha}{2^n},$$

where $\alpha = -2\Re(ab^*)$. Indeed, we have

$$\begin{aligned} h_\Delta &= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \{(a^*\bar{x_j} + b^*x_j)\langle j|\langle 0| + (a^*x_j + b^*\bar{x_j})\langle j|\langle 1|\} \\ &\qquad \{(a\bar{y_j} + by_j)|j\rangle|0\rangle + (ay_j + b\bar{y_j})|j\rangle|1\rangle\} \\ &= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \left((a\bar{y_j} + by_j)(a^*\bar{x_j} + b^*x_j)\langle j|j\rangle\langle 0|0\rangle \right. \end{aligned}$$

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36

$$
\begin{aligned}
&\qquad\qquad + (a^* x_j + b^* \bar{x_j})\,(a y_j + b \bar{y_j})\,\langle j \,|\, j\rangle\,\langle 1 \,|\, 1\rangle \Big) \\
&= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \Big(a \bar{y_j} a^* \bar{x_j} + b y_j a^* \bar{x_j} + a \bar{y_j} b^* x_j + b y_j b^* x_j \\
&\qquad\qquad + a^* x_j a y_j + b^* \bar{x_j} a y_j + a^* x_j b \bar{y_j} + b^* \bar{x_j} b \bar{y_j} \Big) \\
&= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \Big((|a|^2 + |b|^2)(\bar{y_j}\bar{x_j} + x_j y_j) + (ab^* + ba^*)(x_j \bar{y_j} + y_j \bar{x_j}) \Big) \\
&= \frac{1}{2^n} \sum_{j=0}^{2^n-1} \Big(\xi_j + 2\Re(ab^*)\delta_j \Big) = \frac{\Xi + 2\Re(ab^*)\Delta}{2^n} = \frac{\Xi - \Delta\alpha}{2^n}.
\end{aligned}
$$

The last equality comes from the fact that $(\bar{y_j}\bar{x_j} + x_j y_j) = 1$ if and only if $x_j = y_j$, and $(x_j \bar{y_j} + y_j \bar{x_j}) = 1$ if and only if $x_j = \bar{y_j}$.

To achieve perfect distinguishability of the resulting states through quantum measurement, which corresponds to having inner product $h_\Delta = 0$, we need to set a, b such that $\alpha = -2\Re(ab^*)$ becomes $\alpha = \Xi/\Delta = 2^n/\Delta - 1$. Since $\alpha \in [-1, 1]$ for any two unit-norm quantum states, we can do that as long as $\Delta \geq \frac{1}{2}2^n$; the promised functions f_k and f_l must differ on at least half of their outputs to be perfectly distinguishable through orthogonality of the results of $U_k A_n \,|+^{\otimes n}, 0\rangle$ and $U_l A_n \,|+^{\otimes n}, 0\rangle$.

For any $k \in [2^{n-1}, 2^n]$, we can perfectly distinguish functions f_k with exactly k outputs of 1 from the all-zero constant function f_l using just one oracle access by using unitary defined by $\Re(ab^*) = -\frac{2^n/k - 1}{2}$; note that here $\Delta = k$. On a classical computer, the same task can be achieved quickly if $k \sim 2^n$, but can take $O(N) = O(2^n)$ function calls if k is close to 2^{n-1}.

As an example, consider a problem involving functions defined over $n = 4$ bits. Let f_l be an all-zero function, that is, for any of the 16 possible input bitstrings, it returns null. Let f_k be an imbalanced function with exactly $k = 12$ out of 16 inputs returning one, and with four null outputs. To discriminate f_l from any possible f_k, we need to find a, b such that $\Re(ab^*) = -(16/12 - 1)/2 = -1/6$ and $|a|^2 + |b|^2 = 1$: we can have $a = 1/\sqrt{6} - 1/\sqrt{3}$ and $b = 1/\sqrt{3} + 1/\sqrt{6}$.

In contrast to applying A prior to U_f to help solve the problem of discriminating an all-zero function f_l and a function f_k with k ones and $N - k$ zeros on outputs, we can consider the result of applying I_n or E_n instead. For I_n, we arrive at the state $N^{-1/2} \sum_{|\{j\}|=N} |j\rangle\,|0\rangle$ for the all-zero f_l, while for f_k we arrive at $N^{-1/2} \sum_{|\{j\}|=N-k} |j\rangle\,|0\rangle + N^{-1/2} \sum_{|\{j\}|=k} |j\rangle\,|1\rangle$, which has inner product $\frac{N-k}{N}$. Only the all-one function f_k is distinguishable from the all-zero function f_l. Using E_n instead leads to $N^{-1/2} \sum_{|\{j\}|=N} |j\rangle$ for f_l and to $N^{-1/2} \sum_{|\{j\}|=N-k} |j\rangle - N^{-1/2} \sum_{|\{j\}|=k} |j\rangle$ for f_k. These have inner product of $\frac{N-2k}{N}$. Here, only setting $k = N/2$, that is, using a balanced function f_k as in the original Deutsch-Jozsa problem, leads to the ability to distinguish f_l from f_k without error using a single measurement.

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36

With minor changes, we can also, for any $k \in [0, 2^{n-1}]$, distinguish a function with k zero outputs from an all-one constant function. The original Deutsch-Jozsa choice of $k = 2^{n-1}$ is the only case when we can distinguish f_k from both all-zero and all-one functions f_l.

4 Conclusion

The Deutsch-Jozsa problem is often used to illustrate basic concepts in the gate model of quantum computing. Here, we show that the balance-vs-constant function promise problem lies at the edge of a family of promise problems involving discriminating between two subclasses of Boolean functions. The analysis of this broader family of problems can provide improved understanding of the conditions that lead to quantum speedup in Deutsch-Jozsa algorithm.

Acknowledgements

TA is supported by NSF grant CCF-1617710.

References

1. Arodz, T., Saeedi, S.: Quantum sparse support vector machines. arXiv preprint arXiv:1902.01879 (2019)
2. Deutsch, D., Jozsa, R.: Rapid solution of problems by quantum computation. Proceedings of the Royal Society of London. Series A: Mathematical and Physical Sciences **439**(1907), 553–558 (1992)
3. Dunjko, V., Briegel, H.J.: Machine learning & artificial intelligence in the quantum domain: a review of recent progress. Reports on Progress in Physics **81**(7), 074001 (2018)
4. Grover, L.K.: A fast quantum mechanical algorithm for database search. arXiv preprint quant-ph/9605043 (1996)
5. Harrow, A.W., Hassidim, A., Lloyd, S.: Quantum algorithm for linear systems of equations. Physical Review Letters **103**(15), 150502 (2009)
6. Nielsen, M.A., Chuang, I.L.: Quantum Computation and Quantum Information. Cambridge University Press (2011)
7. Panahi, A., Saeedi, S., Arodz, T.: word2ket: Space-efficient word embeddings inspired by quantum entanglement. In: International Conference on Learning Representations 2020. preprint: arXiv:1911.04975 (2019)
8. Qiu, D., Zheng, S.: Generalized Deutsch-Jozsa problem and the optimal quantum algorithm. Physical Review A **97**(6), 062331 (2018)
9. Rebentrost, P., Mohseni, M., Lloyd, S.: Quantum support vector machine for big data classification. Physical Review Letters **113**(13), 130503 (2014)
10. Rønnow, T.F., Wang, Z., Job, J., Boixo, S., Isakov, S.V., Wecker, D., Martinis, J.M., Lidar, D.A., Troyer, M.: Defining and detecting quantum speedup. Science **345**(6195), 420–424 (2014)
11. Tang, E.: A quantum-inspired classical algorithm for recommendation systems. In: Proceedings of the 51st Annual ACM SIGACT Symposium on Theory of Computing. pp. 217–228. ACM (2019)

ICCS Camera Ready Version 2020
To cite this paper please use the final published version:
DOI: 10.1007/978-3-030-50433-5_36