Analysis of Deutsch-Jozsa Quantum Algorithm

Zhengjun Cao, Lihua Liu

Abstract. The Deutsch-Jozsa quantum algorithm is of great importance to modern quantum computation, but we find it is flawed. It confuses two unitary transformations: one is performed on a pure state, and the other on a superposition. In the past decades, no constructive specification on the unitary operator performed on involved superposition has been found, and no experimental test on the algorithm has been practically carried out. We think it needs more constructive specifications on the algorithm so as to confirm its correctness.

Keywords: pure state, superposition, tensor product, quantum computation.

1 Introduction

The Deutsch-Jozsa algorithm [5] is one of the first examples of a quantum algorithm that is exponentially faster than any possible deterministic classical algorithm, which has become the cornerstone for quantum computation and inspired Grover's algorithm [7] and Shor's algorithm [13]. In this note, we want to point out that Deutsch-Jozsa algorithm did confuse two unitary transformations: one was performed on a pure state, and the other on a superposition. So far, no constructive specification on the essential unitary transformation performed on involved superposition has been found. This fact renders the algorithm somewhat dubious. We think it needs more constructive specifications on the algorithm so as to check its correctness and physical complexity.

2 Preliminaries

A qubit is a quantum state $|\Psi\rangle = a|0\rangle + b|1\rangle$, where the amplitudes $a, b \in \mathbb{C}$ such that $|a|^2 + |b|^2 = 1$, $|0\rangle = \binom{1}{0}$ and $|1\rangle = \binom{0}{1}$ are basis vectors of Hilbert space. Two quantum mechanical systems are combined using tensor product. For example, a system of two qubits $|\Psi\rangle = a_1|0\rangle + a_2|1\rangle$

Z. Cao is with Department of Mathematics, Shanghai University, Shanghai, 200444, China. caozhj@shu.edu.cn
L. Liu (corresponding author) is with Department of Mathematics, Shanghai Maritime University, Shanghai, 201306, China. liulh@shmtu.edu.cn

and $|\Phi\rangle = b_1|0\rangle + b_2|1\rangle$ can be written as

$$|\Psi\rangle|\Phi\rangle = \begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \otimes \begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = \begin{pmatrix} a_1 b_1 \\ a_1 b_2 \\ a_2 b_1 \\ a_2 b_2 \end{pmatrix}$$

Its shorthand notation is $|\Psi, \Phi\rangle$.

Operations on a qubit are described by 2×2 unitary matrices. Of these, the most important is Hadamard gate $H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$. Clearly,

$$H|0\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle),\ H|1\rangle = \frac{1}{\sqrt{2}}(|0\rangle - |1\rangle),\ H^2|0\rangle = |0\rangle,\ H^2|1\rangle = |1\rangle.$$

Mathematically, a unitary operator used to modulate a superposition should be written as the product of some basic operations. To this day, at each stage of the creation of a superposition such a program performs a unitary operation on at most three particles at once. These basic operators are listed as follows.

$$\text{Hadamard}: \quad \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad \text{Pauli}-X: \quad \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$$

$$\text{Pauli}-Y: \quad \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad \text{Pauli}-Z: \quad \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$$

$$\text{Phase}: \quad \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \qquad \pi/8: \quad \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$$

$$\text{controlled-NOT}: \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}, \qquad \text{swap}: \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{controlled}-Z: \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, \qquad \text{controlled-phase}: \quad \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & i \end{bmatrix}$$

$$\text{Toffoli}: \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}, \quad \text{Fredkin}: \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Example 1. If the *swap* operator is performed on the state $|01\rangle$, then

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} |01\rangle = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \left(\begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right)$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = |10\rangle$$

3 Review of Deutsch-Jozsa algorithm

Let $f : \{0,1\}^n \to \{0,1\}$ be a map with only two images 0 or 1. The Deutsch-Jozsa algorithm needs a quantum oracle computing $f(x)$ from x which doesn't decohere x. It begins with the $n+1$ bit state $|0\rangle^{\otimes n}|1\rangle$. That is, the first n qubits are each in the state $|0\rangle$ and the final qubit is in the state $|1\rangle$.

A Hadamard gate is applied to each qubit to obtain the following state

$$|0\rangle^{\otimes n}|1\rangle \longrightarrow \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|0\rangle - |1\rangle). \tag{1}$$

Suppose that the oracle $\mathcal{U}_f : |x\rangle|y\rangle \longrightarrow |x\rangle|y \oplus f(x)\rangle$ is available, where the notation $\oplus$ represents bitwise XOR. Applying the quantum oracle, it gives

$$\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|0\rangle - |1\rangle) \longrightarrow \frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} |x\rangle(|f(x)\rangle - |1 \oplus f(x)\rangle). \tag{2}$$

For each x, $f(x)$ is either 0 or 1. The state can be written as $\frac{1}{\sqrt{2^{n+1}}} \sum_{x=0}^{2^n-1} (-1)^{f(x)} |x\rangle(|0\rangle - |1\rangle)$.

Ignoring the last qubit and applying the Hadamard gate to each of the first n qubits, it gives

$$\frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle \longrightarrow \frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\left[\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right] \tag{3}$$

where $x\cdot y = x_0y_0 \oplus x_1y_1 \oplus \cdots \oplus x_{n-1}y_{n-1}$ is the sum of the bitwise product. The above new superposition can be written as

$$\frac{1}{2^n}\sum_{y=0}^{2^n-1}\left[\sum_{x=0}^{2^n-1}(-1)^{f(x)}(-1)^{x\cdot y}\right]|y\rangle. \tag{4}$$

Then the probability of measuring the state $|0\rangle^{\otimes n}$ is $|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|^2$.

4 Analysis of Deutsch-Jozsa algorithm

The whole process of Deutsch-Jozsa algorithm can be described as follows

$$\begin{aligned}
|\underbrace{00\cdots0}_{n}\rangle|1\rangle &\xrightarrow{\text{apply the operator } H^{\otimes(n+1)}} \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|0\rangle-|1\rangle)\\
&\xrightarrow{\text{apply the operator } \mathcal{W}} \frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|f(x)\rangle-|1\oplus f(x)\rangle)\\
&\xrightarrow[\text{and obtain the state}]{\text{ignore the last qubit}} \frac{1}{\sqrt{2^n}}\sum_{x=0}^{2^n-1}(-1)^{f(x)}|x\rangle\\
&\xrightarrow{\text{apply the operator } H^{\otimes n}} \frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\left[\sum_{y=0}^{2^n-1}(-1)^{x\cdot y}|y\rangle\right]\\
&\xrightarrow[\text{to obtain its probability}]{\text{measure the state}} |\underbrace{00\cdots0}_{n}\rangle.
\end{aligned}$$

4.1 How to construct the oracle performed on a pure state

In Deutsch-Jozsa algorithm, the oracle $\mathcal{U}_f : |x\rangle|y\rangle \longrightarrow |x\rangle|y\oplus f(x)\rangle$ must be of the form

$$\mathcal{U}_f = I_2^{\otimes n}\otimes\mathcal{V}_f^{(1)}, \text{ or } I_2^{\otimes(n-1)}\otimes\mathcal{V}_f^{(2)}, \text{ or } I_2^{\otimes(n-2)}\otimes\mathcal{V}_f^{(3)},$$

where I_2 is the 2×2 identity matrix, $\mathcal{V}_f^{(1)}$ is a 2×2 unitary matrix, $\mathcal{V}_f^{(2)}$ is a 4×4 unitary matrix, and $\mathcal{V}_f^{(3)}$ is an 8×8 unitary matrix.

Case-1: Suppose that $\mathcal{V}_f^{(1)} = \begin{pmatrix} X_1 & X_2 \\ X_3 & X_4 \end{pmatrix}$. We have

$$\mathcal{V}_f^{(1)}|y\rangle = \begin{pmatrix} X_1 & X_2 \\ X_3 & X_4 \end{pmatrix}|y\rangle = |y\oplus f(x)\rangle$$

If $y=0$, then $|y\rangle=\binom{1}{0}$. It gives $\binom{X_1}{X_3}=|f(x)\rangle$. Since $f(x)\in\{0,1\}$, we obtain $X_1, X_3\in\{0,1\}$. If $y=1$, then $|y\rangle=\binom{0}{1}$. It gives $\binom{X_2}{X_4}=|1\oplus f(x)\rangle$. Since $f(x)\in\{0,1\}$, we obtain $X_2, X_4\in\{0,1\}$. Thus, $\mathcal{V}_f^{(1)}$ is in the set

$$\left\{\begin{pmatrix}1&0\\0&1\end{pmatrix},\begin{pmatrix}0&1\\1&0\end{pmatrix},\begin{pmatrix}1&1\\0&1\end{pmatrix},\begin{pmatrix}1&1\\1&0\end{pmatrix},\begin{pmatrix}0&1\\1&1\end{pmatrix},\begin{pmatrix}1&0\\1&1\end{pmatrix}\right\}.$$

Clearly, *to determine the matrix $\mathcal{V}_f^{(1)}$, one has to invoke the classical computational result $f(x)$.* That means the unitary matrix $\mathcal{V}_f^{(1)}$ should be further specified as $\mathcal{V}_{f(x)}^{(1)}$. The notation is very useful because it indicates the constructive specification of the involved unitary matrix. So it is better to rewrite the oracle as

$$\mathcal{U}_{f(x)}=I_2^{\otimes n}\otimes\mathcal{V}_{f_{(x)}}^{(1)}.$$

Note that the construction of the oracle depends essentially on the classical computational result $f(x)$. Besides, the oracle is performed on the pure state $|x\rangle|y\rangle$.

Case-2: The operator $\mathcal{V}_f^{(2)}$ is performed on the last two qubits. Since it keeps the state of the first qubit and changes that of the second qubit, we know it must be of the form

$$\text{controlled-NOT}: \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}, \quad \begin{bmatrix}1&0\\0&1\end{bmatrix}\otimes\begin{bmatrix}X_1&X_2\\X_3&X_4\end{bmatrix}=\begin{bmatrix}X_1&X_2&0&0\\X_3&X_4&0&0\\0&0&X_1&X_2\\0&0&X_3&X_4\end{bmatrix},$$

$$\text{controlled}-Z: \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&-1\end{bmatrix}, \quad \text{controlled-phase}: \begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&i\end{bmatrix}$$

By the similar argument in **Case-1**, we know *it has to invoke the classical computational result $f(x)$ to construct the operator $I_2^{\otimes(n-1)}\otimes\mathcal{V}_f^{(2)}$.*

Case-3: The operator $\mathcal{V}_f^{(3)}$ is performed on the last three qubits. It is easy to check that both Toffoli operator and Fredkin operator cannot generate the wanted quantum state.

4.2 Impossibility to create the oracle performed on involved superposition

The unitary operator W is performed on the superposition $\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|0\rangle-|1\rangle)$ and keeps the states of the first n qubits. Hence, if it can be decomposed as $\mathsf{W}=I_2^{\otimes n}\otimes\Gamma$, where Γ is a 2×2 unitary matrix, then by the original description of Deutsch-Jozsa algorithm and the above analysis, we have

$$\mathsf{W}=I_2^{\otimes n}\otimes\Gamma=\mathsf{U}_{f(x)}=I_2^{\otimes n}\otimes\mathsf{V}_{f_{(x)}}.$$

That means one has to extract a classical computational result $f(x)$ from the superposition

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|0\rangle-|1\rangle)$$

in order to construct the operator W practically. Since x runs through all values $0,1,\cdots,2^n-1$, one has to measure the superposition so as to obtain a value $\hat{x}$.

Once the value $\hat{x}$ is measured, applying $\mathsf{W}=I_2^{\otimes n}\otimes\mathsf{V}_{f(\hat{x})}$ to $\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|0\rangle-|1\rangle)$ will produce the state

$$\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\begin{bmatrix}1&0\\0&1\end{bmatrix}(|0\rangle-|1\rangle),\text{ or }\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\begin{bmatrix}0&1\\1&0\end{bmatrix}(|0\rangle-|1\rangle),$$

$$\text{or }\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\begin{bmatrix}1&1\\0&1\end{bmatrix}(|0\rangle-|1\rangle),\text{ or }\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\begin{bmatrix}1&1\\1&0\end{bmatrix}(|0\rangle-|1\rangle),$$

$$\text{or }\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\begin{bmatrix}0&1\\1&1\end{bmatrix}(|0\rangle-|1\rangle),\text{ or }\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle\begin{bmatrix}1&0\\1&1\end{bmatrix}(|0\rangle-|1\rangle),$$

instead of the wanted state $\frac{1}{\sqrt{2^{n+1}}}\sum_{x=0}^{2^n-1}|x\rangle(|f(x)\rangle-|1\oplus f(x)\rangle)$.

Likewise, if $\mathsf{W}=I_2^{\otimes n-1}\otimes\Gamma'$, or $\mathsf{W}=I_2^{\otimes n-2}\otimes\Gamma''$, where Γ' is a 4×4 unitary matrix, and Γ'' is an 8×8 unitary matrix, it is easy to check that both two operators cannot yield the wanted superposition.

All in all, Deutsch and Jozsa have confused a quantum oracle performed on a pure state with a quantum oracle performed on a superposition. So far, no constructive specification on the essential unitary transformation performed on a superposition has been found. Besides, we would like to stress that only the Hadamard gate H is applied to each of the first n qubits twice. Since $H^2|0\rangle=|0\rangle$, we find Deutsch-Jozsa algorithm eventually produces $|\underbrace{00\cdots0}_{n}\rangle|\chi\rangle$, where $\chi\in\{0,1\}$. Their claim that the probability of seeing the state $|0\rangle^{\otimes n}$ is

$$\left|\frac{1}{2^n}\sum_{x=0}^{2^n-1}(-1)^{f(x)}\right|^2,$$

is incorrect.

Notice that the analysis of Deutsch-Jozsa algorithm is just a math problem, having no relation to any physical techniques. All arguments make no sense. It only needs to write down such a matrix so as to cease the controversy about quantum computation.

5 Conclusion

We point out that there are some flaws in Deutsch-Jozsa algorithm. We would like to stress that the construction of a unitary operator performed on a superposition must be compatible with

tensor product [2], which describes the combination of two quantum systems. Some physical experiments [4, 8, 10–12, 14] on Shor's algorithm are criticized for using less qubits in the second register and other deficiencies [1, 3]. So far, the so-called quantum computers, D-wave [6] and IBM [9], have been reported to optimize only some combinatoric problems, not to accelerate any numerical computations. We think Deutsch-Jozsa algorithm needs more specifications so as to facilitate the construction of wanted quantum oracle.

Acknowledgements. We thank professor J. Uhlmann for his discussions.

References

[1] Z. J. Cao and Z. F. Cao: On Shor's Factoring Algorithm with More Registers and the Problem to Certify Quantum Computers. IACR Cryptology ePrint Archive 2014: 721 (2014)

[2] Z. J. Cao, Z. F. Cao and L. H. Liu: Remarks on Quantum Modular Exponentiation and Some Experimental Demonstrations of Shor's Algorithm. IACR Cryptology ePrint Archive 2014: 828 (2014)

[3] Z. J. Cao, Z. F. Cao and L. H. Liu: Comment on Demonstrations of Shor's Algorithm in the Past Decades. IACR Cryptology ePrint Archive 2015: 1207 (2015)

[4] A. Dang, et al.: Optimising Matrix Product State Simulations of Shor's Algorithm, arXiv:1712.07311v2 (2017)

[5] D. Deutsch and R. Jozsa: Rapid solutions of problems by quantum computation. Proceedings of the Royal Society of London A, 439, 553 (1992)

[6] D-Wave Systems, PDF, 01-2017, `http://www.dwavesys.com/sites/default/files/`

[7] L. K. Grover: A fast quantum mechanical algorithm for database search. Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing. pp. 212C219 (1996)

[8] E. Lucero, et al.: Computing prime factors with a Josephson phase qubit quantum processor. Nature Physics 8, 719-723, 2012. arXiv:1202.5707 (2012)

[9] `http://www.research.ibm.com/ibm-q/`

[10] C.Y. Lu, et al.: Demonstration of a Compiled Version of Shor's Quantum Factoring Algorithm Using Photonic Qubits, Physical Review Letters 99 (25): 250504, arXiv:0705.1684 (2007)

[11] B. Lanyon, et al.: Experimental Demonstration of a Compiled Version of Shor's Algorithm with Quantum Entanglement", Physical Review Letters 99 (25): 250505. arXiv:0705.1398 (2007)

[12] E. Martín-López, et al.: Experimental realization of Shor's quantum factoring algorithm using qubit recycling. Nature Photonics. doi:10.1038/nphoton.2012.259 (2012)

[13] P. Shor: Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM J. Comput. 26 (5): 1484-1509 (1997)

[14] L. Vandersypen, et al.: Experimental realization of Shor's quantum factoring algorithm using nuclear magnetic resonance, Nature 414 (6866): 883-887, arXiv:quant-ph/0112176 (2001)