Applied Soft Computing 132 (2023) 109844

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Rapid solution of logical equivalence problems by quantum computation algorithm

Mohammed Zidan [a,b,*,1], Salem F. Hegazy [c,**,1], Mahmoud Abdel-Aty [d,1], Salah S.A. Obayya [e,1]

[a] Department of Artificial Intelligence, Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada Branch, Egypt
[b] Faculty of Computers and Information Technology, The Egyptian E-Learning University, Cairo, Egypt
[c] National Institute of Laser Enhanced Sciences, Cairo University, Giza 12613, Egypt
[d] Department of Mathematics, Faculty of Science, Sohag University, Sohag, Egypt
[e] Centre for Photonics and Smart Materials, Zewail City of Science and Technology, October Gardens, 6th of October City, Giza 12578, Egypt

ARTICLE INFO

Article history:
Received 22 February 2022
Received in revised form 10 September 2022
Accepted 11 November 2022
Available online 24 November 2022

Keywords:
Quantum computing
Quantum algorithms
Logical equivalence
Quantum verification

ABSTRACT

We present a quantum computation algorithm that enables solving the problem of logical equivalence verification in exponentially less time than the classical deterministic computation. In this novel quantum algorithm, the oracles of the two evaluated functions are executed in series to yield a common target qubit which then interacts with an ancillary qubit. We found that the degree of entanglement (measured by the concurrence) of the target and ancillary qubits is a reliable witness for the logical equivalence property of the two functions. The steps number of the quantum algorithm is inversely proportional to the square of the standard error ϵ^2 of the measured concurrence value, with no dependence on the input size $'n'$ of each function. This corresponds to a number of evaluations of the two functions: $O(\epsilon^{-2})$ for the quantum algorithm compared with $O(2^n)$ for the classical approach. To assess the algorithm performance, two sets of experiments are conducted using the IBM Q Experience simulator for input sizes: 2 and 12 variables per function. While the former verifies that the results of the experiment are in a good match with the theory, the latter showcases the quantum supremacy of the presented algorithm. In particular, The latter shows that the quantum algorithm requires only 200 oracles queries compared with 2^{13} queries for the classical algorithm.

1. Introduction

Quantum phenomena have inspired many scientists to introduce a plethora of groundbreaking technologies such as quantum computing [1–7], quantum memories [8], quantum teleportation [9], quantum key distribution [10–12], and quantum machine learning models [13]. Quantum computing, for example, exploits the advantage of the quantum superposition behavior of microscopic systems [2,3] to solve traditional and intractable problems more efficiently compared with classical computing [5,6]. Its unique capabilities were initially discovered by R. Feynman in 1982 when he attempted to numerically solve the system of equations modeling quantum systems on classical computers [3]. More attention has been extensively drawn to quantum computing after treating some problems known to be intractable, particularly, the co-prime number factorization and discrete logarithm [5,6]. Although building a reliable universal quantum computer is still a far goal, there is a good progress in this direction [14].

Quantum entanglement is arguably the most widely known quantum feature that has no classical counterpart [15–17]. It is a cornerstone in many quantum technology applications with practical realizations using photons [18–22], atoms [23,24], and electrons [25]. Recently, it was shown that entanglement measures, like the concurrence [26], can be employed to solve various quantum computing problems [27].

One of the substantial problems encountered in several engineering fields is the logical equivalence of two unknown identities with well-known applications in Artificial Intelligence [28], circuit design and optimization [29], and software engineering [30]. In Artificial Intelligence, particularly, in expert systems and data mining fields, one of the vital tasks is to determine the most effective association rules that construct the domain

* Corresponding author at: Department of Artificial Intelligence, Faculty of Computers and Artificial Intelligence, South Valley University, Hurghada Branch, Egypt.
** Corresponding author at: National Institute of Laser Enhanced Sciences, Cairo University, Giza 12613, Egypt.
E-mail addresses: comsi2014@gmail.com (M. Zidan), salem@niles.cu.edu.eg (S.F. Hegazy), sobayya@zewailcity.edu.eg (S.S.A. Obayya).
[1] All authors contributed equally in this paper.

https://doi.org/10.1016/j.asoc.2022.109844
1568-4946/

knowledge. Moreover, the logical equivalence is exploited in an approach to discover coherent rules (see, e.g., [28]). Therefore, association rules can be derived objectively and directly without knowing the level of minimum support threshold required. The logical equivalence is also used to determine whether two classical circuits are equivalent or not in order to improve circuits design [29,31]. In software engineering, the logical equivalence offers an efficient model to detect the paradigm of software plagiarism and reserve copyrights of software computing systems [30]. Additionally, it offers a method for program synthesis and generation [32]. Quite recently, the logical equivalence is applied to design models for measuring the complexity of structures in civil engineering [33].

Let us consider that we have two unknown functions given *solely* by calling two oracles which have inaccessible internal workings. These two oracles are therefore much like black boxes (can be two computer programs or access memories [7]) which ideally give an output in response to an input. Classically, the verification of the logical equivalence of the two functions requires to pass through the whole truth table; that is 2^n oracle queries for a function of n variables [31]. This means an exponential complexity $O(2^n)$ for the classical approach. The logical equivalence is thus a complex problem in the classical domain which implies a dramatic increase of the number of oracles queries with the increase of the number n.

In this paper, we present a novel quantum algorithm that determines the logical equivalence of two unknown functions in a number of steps independent of the input size of the two functions, which is exponentially less than the classical method. To realize this, the presented quantum algorithm adopts the quantum interference of all outcomes of the problem. The computational complexity of the presented quantum algorithm is investigated based on the number of oracles queries needed to estimate the concurrence value up to a specific bound of standard error. The algorithm performance is evaluated through several experiments using IBM Q Experience simulator.

The paper is organized as follows: Section 2 defines the problem statement, and hypothesis and limitations. Section 3 explains the classical algorithm for logical equivalence verification. In Section 4, the methodology and analysis of the presented algorithm are extensively explained. The complexity of the quantum algorithm versus its traditional counterpart is addressed in Section 5. Section 6 shows the experimental realization of the quantum algorithm. Section 7 comprises the main results of this paper.

2. Definitions

2.1. Problem statement

The problem under scrutiny is defined as follows:
Given: two oracles U_{f_i}, where $i \in \{1, 2\}$, each oracle encodes unknown Boolean function, logical expressions (compound propositions), f_i: $\{0, 1\}^n \rightarrow \{0, 1\}$, where n is the number of input Boolean variables (propositions).
Goal: check whether the oracles U_{f_1} and U_{f_2} are logically equivalent, or not logically equivalent in a constant time with some predefined error ϵ.

2.2. Hypothesis and limitations

Let U_{f_1}, U_{f_2} be devices that compute the two functions $f_1, f_2 : Z_n \rightarrow Z_1$. Given some input j, U_{f_1} and U_{f_2} will, after some time, give the values of $f_1(j)$ and $f_2(j)$ at the output. The class of computational task that we consider involves – in general – being given U_{f_1} and U_{f_2} and then using it to obtain some property $W[f_1, f_2]$ (which is, some function W of all row sequences of f_1 and f_2) in the least possible time. The analysis is based on the approximation that the internal work of both U_{f_1} and U_{f_2} is inaccessible, which defines U_{f_1} and U_{f_2} as oracles for f_1 and f_2 (this is an excellent approximation in the analysis of these types of tasks [7]). If U_{f_1} and U_{f_2} were two computer programs that evaluates f_1 and f_2, this approximation is equivalent to an assumption that there is no faster approach of evaluating $W[f_1, f_2]$ than actually executing the programs U_{f_1} and U_{f_2} to obtain all values of f_1 and f_2 to determine the property $W[f_1, f_2]$. This assumption appears quiet realistic, if U_{f_1} and U_{f_2} were two blocks of memory locations that register two sequences f_1 and f_2. In this case, the need for reading all memory locations is undeniable.

We assume that the coherence of every quantum state is maintained through the operation, that is the purity of each quantum state does not deteriorate by time. We also assume perfect fidelity of quantum measurements, which means that the outcome of measurement depends on the probability defined solely by ideal projective measurement.

3. Classical computation approach

In proofs, a mathematical statement can be replaced with another statement if they have the same truth value. Therefore, various approaches are utilized to discover logical expressions (compound propositions) of the same truth value [31]. However, if two logical expressions A and B are unknown (as the case we consider here), one classical approach exists to check their logical equivalence using the truth table as depicted in Algorithm 1.

Algorithm 1 : Logical equivalence verification using truth table [34].

Step 1: The truth table is constructed with one column for the truth values of A and another one for the truth values of B.
Step 2: Each combination of truth values must be checked for the same logical expression variables (propositions) to see whether the truth values of A and B are the same.
Step 3: If in each row, the truth values of A and B are the same, A and B are logically equivalent.
Step 4: If in some rows, A and B have a different truth value, A and B are not logically equivalent.
Step 5: If in each row, A and B have different truth value, A and B are the negation of each other.

Remarks. There are two remarks to be recorded on Algorithm 1. First: If the two logical expressions A and B were defined, other methods (e.g., the textual analysis, propositional logic, and Boolean algebra [31,34]) could be used to verify the logical equivalence. However, in this paper it is assumed that the logical expressions A and B are unknown, so that the only classical algorithm that can be used to solve the defined problem is the mentioned algorithm [31,32,34]. Second: the number of evaluations of the functions is given by $O(2^n)$ since it needs to check all 2^n rows, where n is the input size of each function. Thus, this allgorithm is hardly practical to be used to evaluate the logical equivalence in case of large n.

4. Quantum computation approach

4.1. Methodology

The concurrence is one of the measures used to quantify the degree of entanglement of two qubits [26,35]. The concurrence value C for a state in the form $|\psi\rangle = s_1|00\rangle + s_2|01\rangle + s_3|10\rangle + s_4|11\rangle$ is $C = 2|s_1 s_4 - s_2 s_3|$, $0 \leq C \leq 1$. If the system has the state $|\psi\rangle = s_1|00\rangle + s_4|11\rangle$, then the concurrence value

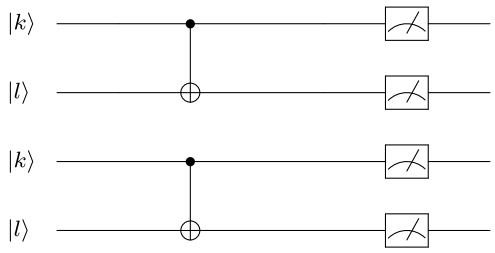

Fig. 1. The circuit model of the operator M_z. It entangles two replicas of the qubits $|k\rangle$ and $|l\rangle$ separately based on the state of the qubit $|k\rangle$, and also measures the degree of entanglement in between.

is $C = 2|s_1 s_4|$. Recently, the effectiveness of using the concurrence measure to solve quantum computing problems was demonstrated [26]. Also, the quantum computing model based on the degree of entanglement was proposed [26]. This model uses an operator M_z to find the solution of the problem at hand using one of two techniques. The *first technique* finds the solution of the quantum problem based on the degree of entanglement between two auxiliary qubits. The *second technique* finds the solution of the quantum problem by checking the presence or absence of the entanglement between the two auxiliary qubits. The circuit-model implementation of this model is explained in [26]. The operator M_z is implemented in the circuit model through the quantum circuit shown in Fig. 1.

This circuit receives two decoupled replicas of the two qubits $|k\rangle \otimes |l\rangle$, where qubit $|k\rangle$ has an arbitrarily unknown state in the form $|k\rangle = s_1|0\rangle + s_4|1\rangle$ and the qubit $|l\rangle$ is initialized by a state $|0\rangle$. The operator M_z executes two operations.

Algorithm 2 The quantum algorithm for Logical Equivalence Verification

Step 1: The quantum system is established as a tensor product of two registers. The size of the first register $|q\rangle$ is n qubits, and the second register is composed of two qubits $|k\rangle$ and $|l\rangle$ such that each register is initialized as follows:

$$|\psi_0^{c1}\rangle = |q\rangle \otimes |kl\rangle = |0\rangle^n \otimes |00\rangle.$$

Step 2: Hadamard operator is applied on the n qubits of the first register $|q\rangle$, which gives $|\psi_1^{c1}\rangle = (H^{\otimes n} \otimes I^{\otimes 2})|0\rangle^n|00\rangle$,
Step 3: The two oracles are executed in series. This yields $|\psi_2^{c1}\rangle = \prod_{i=1}^{2}(U_{f_i} \otimes I)|\psi_1^{c1}\rangle$.
Step 4: Steps 1-3 are re-executed to obtain another copy of the system $|\psi_2^{c2}\rangle$ (because M_z operator works on two replicas), where the superscripts "$c1$" and "$c2$" represent the two copies of the quantum states in steps 1–4.
Step 5: The operator M_z is applied on the qubits $|klkl\rangle$. The probabilities P_{0011} and P_{1100} are estimated which determine the concurrence value C as in Eq. (2).

(a) If C > 0 then

the functions f_1 and f_2 are not logically equivalent.

(b) If $C = 0$ and the state of qubits $|klkl\rangle = |0000\rangle$ then

the functions f_1 and f_2 are logically equivalent.

(c) If $C = 0$ and the state of qubits $|klkl\rangle = |1111\rangle$ then

f_1 is the negation of f_2.

In the first operation, two *CNOT* operators are applied which entangle each copy of the two qubits $|kl\rangle$ with a degree depending on the state of the qubit $|k\rangle$ and yields the four-qubit state

$$(CNOT \otimes CNOT)(|kl\rangle \otimes |kl\rangle) = s_1^2|0000\rangle + s_1 s_4|0011\rangle + s_4 s_1|1100\rangle + s_4^2|1111\rangle. \quad (1)$$

Then, in the second operation, projective measurements (in the computational basis) are applied on this four-qubit state which gives the outcomes $|0011\rangle$ and $|1100\rangle$ with probabilities $P_{0011} = P_{1100} = |s_1 s_4|^2$. This quantifies the concurrence value for each pair of the entangled qubits $|k\rangle$ and $|l\rangle$ as [26]:

$$C = 2\sqrt{P_{0011}} = 2\sqrt{P_{1100}}, \text{ or } C = \sqrt{2(P_{0011} + P_{1100})}. \quad (2)$$

The second technique of the model [26] uses the quantum circuit of the operator M_z shown in Fig. 1 to differentiate between the non-orthogonal states in the form $e_1|0\rangle + e_4|1\rangle$, (where $e_1, e_4 \in R^+$), the state $|0\rangle$, and the state $|1\rangle$ using the concurrence value C. The algorithm presented in the following subsection is based on this second technique to solve the problem stated in Section 2.1.

4.2. Quantum algorithm

Algorithm 2 depicts the quantum algorithm that solves the logical equivalence problem for two unknown functions defined by the oracles U_{f_1} and U_{f_2}. Fig. 2 shows the quantum circuit of the algorithm which can be discussed as follows.

Step 1: The quantum system is composed of two registers: the n-qubit register $|q\rangle$ which represent the n propositions (n Boolean variables) of the oracles U_{f_1} and U_{f_2}, and the two-qubit register $|kl\rangle$. The initial state of the system is $|\psi_0^{c1}\rangle = |0\rangle^n \otimes |00\rangle$.

Step 2: The Hadamard gate H is applied on each qubit of the register $|q\rangle$ to generate all possible 2^n rows of the truth table. Hence, the state of the system is $|\psi_1^{c1}\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle|00\rangle$. This step creates and superposes every possible input for the oracles U_{f_1} and U_{f_2} in step 3 (In other words, for each input the oracles U_{f_1} and U_{f_2} need to be recalled in terms of classical algorithms [34]). Therefore, the computational cost of the classical algorithm is $O(2^n)$. Conversely, due to the superposition of all inputs in Step 2, the quantum algorithm will recall these oracles in constant time to estimate the concurrence in Step 5.

Step 3: The oracles U_{f_1} and U_{f_2} are applied on the first $n+1$ qubits of the system which are the register $|q\rangle$ and the qubit $|k\rangle$ as in Fig. 2. After applying the two oracles on the first $n+1$ qubits, the XOR operation: $f_1 \oplus f_2$ is performed directly between the outcome of the Boolean functions f_1: $\{0,1\}^n \rightarrow \{0,1\}$, and f_2: $\{0,1\}^n \rightarrow \{0,1\}$ with the result being registered in the qubit $|k\rangle$ (which has the initial state $|0\rangle$). This is expressed as follows:

$$|\psi_2^{c1}\rangle = U_{f_2}(U_{f_1}|q\rangle|k\rangle)|l\rangle = U_{f_2}\left[\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|j\rangle|0 \oplus f_1(j)\rangle\right]|l\rangle$$
$$= U_{f_2}\left[\frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|j\rangle|f_1(j)\rangle\right]|l\rangle$$
$$= \frac{1}{\sqrt{2^n}}\sum_{j=0}^{2^n-1}|j\rangle|f_1(j) \oplus f_2(j)\rangle|l\rangle.$$

Consequently, there are three possible cases:

Case (A): The oracles U_{f_1} and U_{f_2} do not have the same truth values for all possible cases which indicates that f_1 and f_2 are not

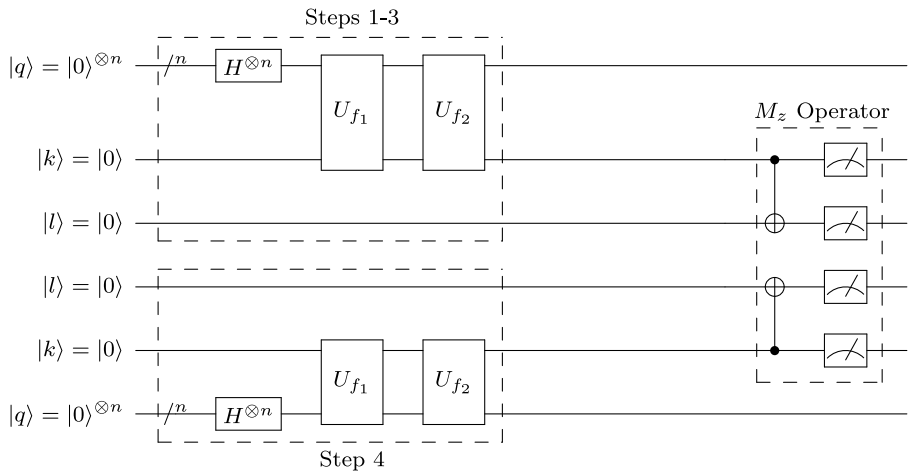

Fig. 2. The quantum circuit of the algorithm that verifies the logical equivalence between two unknown oracles U_{f_1} and U_{f_2}.

logically equivalent. In this case, the output state is

$$|\psi_2^{c1}\rangle = |\theta_1\rangle|00\rangle + |\theta_2\rangle|10\rangle, \tag{3}$$

where
$|\theta_1\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=\{j|f_1(j)\oplus f_2(j)=0\}} |j\rangle$, and $|\theta_2\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=\{j|f_1(j)\oplus f_2(j)=1\}} |j\rangle$.

Case (B): If the oracles U_{f_1} and U_{f_2} have the same truth values in all possible cases indicating that f_1 and f_2 are logically equivalent, the state of the system will be as follows

$$|\psi_2^{c1}\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle|00\rangle. \tag{4}$$

Case (C): If the oracle U_{f_1} is the negation of the oracle U_{f_2}, the system will be in the state

$$|\psi_2^{c1}\rangle = \frac{1}{\sqrt{2^n}} \sum_{j=0}^{2^n-1} |j\rangle|10\rangle. \tag{5}$$

Step 4: Steps 1-3 are re-executed to obtain another decoupled replica $|\psi_2^{c2}\rangle$, of the system $|\psi_2^{c1}\rangle$, because M_z operator works on two decoupled copies of $|kl\rangle$ to retrieve the degree of entanglement between these two qubits. Now, the two copies of the two qubits $|k\rangle$ and $|l\rangle$ are available, where the state of the entire system is expressed as

$$\begin{aligned}|\xi\rangle = {} & |\theta_1\rangle^{\otimes 2} |0000\rangle + |\theta_1\rangle|\theta_2\rangle |0010\rangle \\ & + |\theta_2\rangle|\theta_1\rangle |1000\rangle + |\theta_2\rangle^{\otimes 2} |1010\rangle.\end{aligned} \tag{6}$$

Step 5: By applying the first operation of the operator M_z on the four qubits $|klkl\rangle$ in the closed system described by Eq. (6), the three cases of *Step 3* proceed as follows:

Case (A): If the oracles U_{f_1} and U_{f_2} are not logically equivalent, the state of the entire system will be defined by Eq. (6) with the probabilities of the states $|0000\rangle$, $|0011\rangle$, $|1100\rangle$ and $|1111\rangle$ being greater than zero. This leads to a concurrence value $C > 0$ in a way similar to Eq. (2).

Case (B): If the oracles U_{f_1} and U_{f_2} are logically equivalent, the four qubits $|klkl\rangle$ will be in the state $|0000\rangle$, and the probabilities of basis states $|0011\rangle$, $|1100\rangle$, $|1111\rangle$ in Eq. (6) vanish. In this case, the concurrence value $C = 0$.

Case (C): If the oracle U_{f_1} is the negation of the oracle U_{f_2}, the four qubits $|klkl\rangle$ will be in the state $|1111\rangle$, and the other probabilities of basis states $|0000\rangle$, $|0011\rangle$, and $|1100\rangle$ vanish. Also in this case, the concurrence $C = 0$.

5. Computational complexity

Here, the complexity of the presented algorithm is investigated based on the number of measurements M needed to estimate the concurrence value C up to a standard error ϵ. The concurrence value C is estimated quantum mechanically using Eq. (2) by performing projective measurements in the computational basis (including the basis states $\{|0000\rangle, |1100\rangle, |0011\rangle, |1111\rangle\}$) on an ensemble of M systems. The concurrence value C can be quantified experimentally by measuring the probability that the two qubits of the first pair of qubits $|kl\rangle$ anti-correlates with the second pair (the probability that the state is projected into one of the basis states $|0011\rangle$ or $|1100\rangle$). As expressed in Eq. (2), the anti-correlation probability is given as $P_{ac} = P_{0011} + P_{1100} = \frac{1}{2}C^2$.

The measurement event of either correlation or anti-correlation of the two pair of qubits can be considered as a binomial random variable with standard deviation for one trial (or run) being $\sqrt{P_{ac}(1-P_{ac})}$. The standard deviation of the number of anti-correlations over M trials is then $\sqrt{MP_{ac}(1-P_{ac})}$. In actual experiments, the value of the probability P_{ac} is accessible by normalizing the number of anticorrelations to the total number of trials M. This gives the standard error of measuring P_{ac} as

$$\sigma_{P_{ac}} = \frac{1}{M}\sqrt{MP_{ac}(1-P_{ac})} = \sqrt{\frac{1}{M}P_{ac}(1-P_{ac})}.$$

As in Eq. (2), the value of the concurrence can be estimated by the relation $C = \sqrt{2P_{ac}}$, then the standard error in estimating the concurrence value C is given by

$$\epsilon \approx \frac{\partial C}{\partial P_{ac}}\sigma_{P_{ac}} = \sqrt{\frac{1}{2M}(1-P_{ac})}. \tag{7}$$

Consequently, the number of measurements M that need to be performed in order to estimate the concurrence value C, for the computing model [26], up to a standard error ϵ can be determined by

$$M = \left\lceil \frac{1}{2\epsilon^2}(1 - C^2/2) \right\rceil. \tag{8}$$

Here, in the presented quantum algorithm, each of the oracles U_{f_1} and U_{f_2} needs to be called M times (total queries are $4M$), which is inversely proportional to ϵ^2 in the concurrence value. The upper bound of the oracles-queries number equals $2/\epsilon^2$ which takes place if the functions f_1 and f_2 are either logically equivalent or the negation of each other [That is, Cases (B) and (C)]. The lower bound of the oracles-queries number is $1/\epsilon^2$, which applies when f_1 and f_2 are not logically equivalent [as in Case (A)].

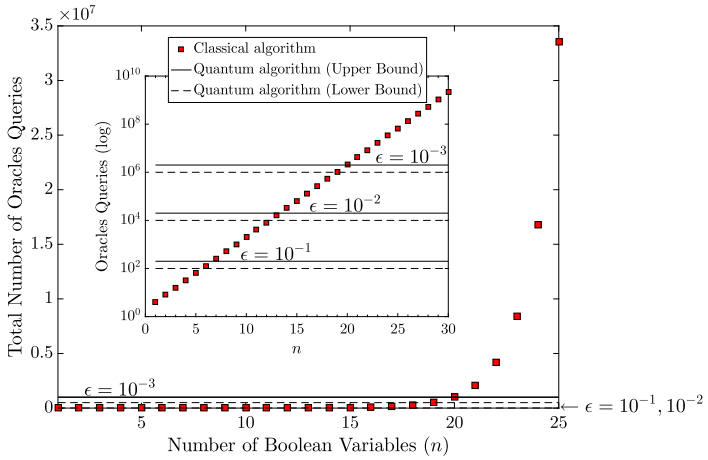

Fig. 3. Comparison between the total number of oracles queries of the quantum algorithm – defined by upper and lower bounds – and that of the classical algorithm. The number of queries are plotted – in linear and logarithmic scale (in the main figure and inset, respectively) – versus the number of Boolean variables n of the two functions f_1 and f_2 checked for logical equivalence. The upper and lower bounds for quantum algorithm are given in the cases when the concurrence value C encounters standard error values $\epsilon = \{10^{-1}, 10^{-2}, 10^{-3}\}$.

Table 1
Comparison between the classical algorithm and quantum algorithm for logical equivalence verification.

Method name:	Classical algorithm	quantum algorithm
Techniques:	Boolean logic.	Quantum computation.
Type of dataset:	Unknown Boolean expressions.	Unknown Boolean expressions.
Complexity:	$O(2^n)$.	$O(\epsilon^{-2})$.
Advantages:	Deterministic computing.	Computation time is independent of the input size.
Limitations:	Computation time is exponential of the input size.	Errors may exist (but can be negligible with additional queries).

Eq. (8) also explains that the complexity of the presented quantum algorithm (and all algorithms based on the second technique of the quantum computing model in [26]) is $O(\epsilon^{-2})$ where the number of oracle(s) queries does not depend on the input size of the functions. This is compared with a complexity $O(2^n)$ for the classical algorithm.

Fig. 3 shows a comparison between the number of oracles queries of the presented quantum algorithm and that of the classical algorithm to verify the logical equivalence property of two functions. The comparison is made along a range of the Boolean variables number $'n'$ of the two functions. The queries number for the quantum algorithm is bounded by an upper and lower levels when the standard error ϵ takes the values 10^{-1}, 10^{-2}, and 10^{-3}, respectively. Recalling Eq. (8), these error levels correspond respectively to total numbers of queries: $4M = 200, 2 \times 10^4$, and 2×10^6. It is remarkable that the queries number of the quantum algorithm is generally independent of the number of Boolean variables n. Classically, in contrast, the functions f_1 and f_2 need to be called 2^{n+1} times to solve the same problem [31,34]. Fig. 3 implies that the speed of the presented algorithm elevates exponentially compared to the classical algorithm as the number of propositions increases. However, the speed of the quantum algorithm depends *solely* on the level of the standard error of our measured quantum witness; the concurrence value. A comparison between the classical algorithm and the quantum algorithm is illustrated in Table 1.

6. Experimental realization of the quantum algorithm

6.1. Experimental setup

The quantum algorithm determines the logical equivalence of two functions f_1 and f_2 by calling the oracles U_{f_1} and U_{f_2}, where the number of calls depends only on the level of allowed error and independent of the number of variables of the two functions. To assess the performance of the presented algorithm experimentally, we assume that the preparation of the oracles U_{f_1} and U_{f_2} is a secret kept by the preparation operator and not known to anybody else. Here, two sets of experiments are addressed. The first set considers 2-variable functions for f_1 and f_2, and the second addresses the 12-variable case. These experiments are conducted using the IBM's high-performance simulation framework called Qiskit Aer [36].

The number of oracles calls per each experiment run, M, takes the value 8192 in the first set of experiments (2-variable case) and only 200 in the second set (12-variable case). We aim from this wide span of M value to demonstrate the effect of the number of oracles calls on the reliability of algorithm decision regarding the logical equivalence. That is important since, as mentioned in Section 5, the number of oracles calls is not constrained by the number of propositions (the function variables), but by the standard error of the concurrence value. In order to evaluate the performance in each set of experiments, the preparation operator assigns functions f_1 and f_2 based on one of the three scenarios with f_1 and f_2 being either logically equivalent, not logically equivalent, or negation of each other.

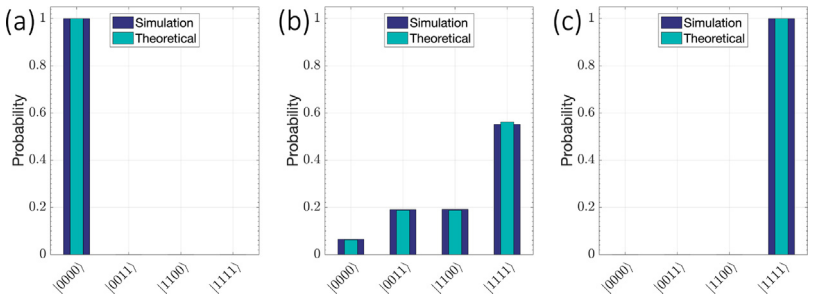

Fig. 4. Theoretical and simulation results for logical equivalent verification of two-variable functions. (a) Logical equivalence scenario: $f_1 \equiv \neg(p \vee q)$; $f_2 \equiv \neg p \wedge \neg q$, (b) No logical equivalence scenario: $f_1 \equiv p \vee q$; $f_2 \equiv \neg q$, (c) Negation scenario: $f_1 \equiv p \vee q$ and $f_2 \equiv \neg p \wedge \neg q$. The green histograms represent the theoretical probabilities of the states $|0000\rangle$, $|0011\rangle$, $|1100\rangle$, and $|1111\rangle$ of the qubits $|klkl\rangle$, where the blue histograms represent the simulation probabilities.

6.2. Results analysis and discussion

The results of the first set of experiments (in which, f_1, f_2 are functions of two variables) are depicted in Fig. 4. In the first scenario, the preparation operator assigns the two logically equivalent functions: $f_1 \equiv \neg(p \vee q)$ and $f_2 \equiv \neg p \wedge \neg q$. In this case, Fig. 4(a) shows that the qubits $|klkl\rangle = |0000\rangle$ with certainty and the probabilities of the basis states $|0011\rangle$, $|1100\rangle$, and $|1111\rangle$ are zero. This yields a concurrence value $C = 0$ as given in Eq. (2). Fig. 4(a) shows that the theoretical results exactly match the simulation results with a fidelity $F = 1$. The fidelity is expressed as $F = \sum_{j=0}^{2^n-1} \sqrt{p_j^{sim} p_j^{th}}$, where n is the number of qubits and p_j^{sim}, p_j^{th} are the simulation and theoretical probabilities, respectively, for the state that has index j [1].

In the second scenario, the operator prepares $f_1 \equiv p \vee q$; $f_2 \equiv \neg q$, which are not logically equivalent. In this case, Fig. 4(b) depicts that all probabilities of the states $|0000\rangle$, $|0011\rangle$, $|1100\rangle$, and $|1111\rangle$ are greater than zero, and accordingly the concurrence value C takes a value above zero. Theoretically, it can be determined that the probability of the state $|0011\rangle$ is 0.19 which implies that concurrence value is $C = 0.87$ based on Eq. (2). The simulation results shown in Fig. 4(b) implies that concurrence value is $C = 0.87$. The theoretical and simulation results significantly match with a fidelity $F = 0.9999$. The negligible difference between both results is because the number of calls is 8192 ($F \to 1$ iff $M \to \infty$).

In the third Scenario, the two functions are prepared as $f_1 \equiv p \vee q$ and $f_2 \equiv \neg p \wedge \neg q$ which are the negation of each other. Fig. 4(c) shows that the qubits $|klkl\rangle$ are in the state $|1111\rangle$ with certainty; indicating that concurrence value is $C = 0$ upon Eq. (2). Fig. 4(c) exhibit a perfect match between the theoretical and simulation results with fidelity $F = 1$. In this first set of experiments, the quantum algorithm exhibited no supremacy compared to its classical counterpart.

Now, let us move to the second set of computing experiments which verify the logical equivalence of two 12-variable functions. In the first scenario, the operator assigns the two logically equivalent functions: $f_1 = 1 \oplus (\wedge_{i=1}^{12} x_i)$, and $f_2 = \neg(\wedge_{i=1}^{12} x_i)$, where the logical operator $\wedge_{i=1}^{12}$ denotes the recursive operation over the 12 variables x_i. For the second scenario, the two functions are $f_1 = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus (x_5 \wedge x_6) \oplus (x_7 \wedge x_8) \oplus (x_9 \wedge x_{10}) \oplus (x_{11} \wedge x_{12}) \oplus (\wedge_{i=1}^{12} x_i)$, and $f_2 = \wedge_{i=1}^{12} x_i x_1$, which are not logically equivalent. For the third scenario, The functions: $f_1 = \wedge_{i=1}^{12} x_i$, and $f_2 = \vee_{i=1}^{12} \neg x_i$ are assigned, which are the negation of each other.

The quantum algorithm is run with $M = 50$ which gives a total number of 200 oracles queries per experiment. This is compared to the classical algorithm which requires 8192 queries to reach a logical-equivalence decision. The results of 10 experiments are shown in Fig. 5, which comply with the theoretical expectations for the three scenarios. The quantum algorithm achieves a fidelity

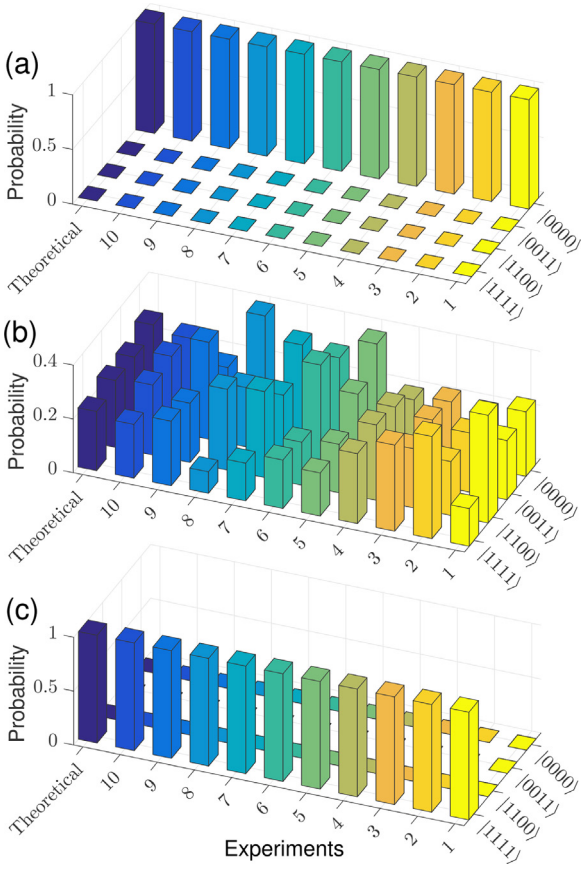

Fig. 5. Theoretical and simulation results for the second set of experiments where logical equivalent verification test is conducted for 12-variable functions. (a) Logical equivalence scenario: $f_1 = 1 \oplus (\wedge_{i=1}^{12} x_i)$, and $f_2 = \neg(\wedge_{i=1}^{12} x_i)$. (b) No logical equivalence scenario: $f_1 = (x_1 \wedge x_2) \oplus (x_3 \wedge x_4) \oplus (x_5 \wedge x_6) \oplus (x_7 \wedge x_8) \oplus (x_9 \wedge x_{10}) \oplus (x_{11} \wedge x_{12}) \oplus (\wedge_{i=1}^{12} x_i)$, and $f_2 = \wedge_{i=1}^{12} x_i x_1$. (c) Negation Scenario; $f_1 = \wedge_{i=1}^{12} x_i$, and $f_2 = \vee_{i=1}^{12} \neg x_i$. The sub-figure of each scenario includes the results of 10 experiments followed by the theoretical results. The results of each experiment are based on 200 calls of the oracles U_{f_1} and U_{f_2}. This corresponds to $\epsilon = 0.1$ of the concurrence value C.

$F = 1$ for the first and third scenarios and $F = 0.998$ for the second scenario. The variation between the results of the 10 experiments demonstrates the statistical nature of the measured concurrence. However, as shown, this variation does not affect the solution of the logical equivalence problem in any of the 10 experiments. The results demonstrate the quantum supremacy by a ratio of 40 for 12 input Boolean variables. This quantum supremacy ratio increases to astronomic values as the input size of the functions extends.

7. Conclusion

In this work, a novel quantum algorithm has been presented for verifying the logical equivalence of two unknown functions in a number of steps that does not depend on the input size of each function n. The number of oracles queries of the quantum algorithm is inversely proportional to the square of the standard error of the concurrence (the quantum measure we used to witness the logical equivalence property). This performance comes in contrast to the classical approach for this problem, which exhibits an exponential increase of the number of steps with n; particularly, $O(2^n)$. The quantum algorithm outperforms its classical counterpart at the large values of n, remarkably, when the complexity of the logical equivalence problem intensively rises. The exponential speedup of the quantum algorithm compared with the classical algorithm has been demonstrated by simulation on the IBM Q Experience simulator; which matched the expected findings. The presented algorithm paves the way to innovative classes of artificial intelligence and machine learning that benefit from the existence of rapid solution for the logical equivalence problem. Interestingly, this will inspire novel proposals of more efficient quantum expert systems [28,32] compared with the classical expert systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] F. Arute, K. Arya, R. Babbush, D. Bacon, J.C. Bardin, R. Barends, R. Biswas, et al., Quantum supremacy using a programmable superconducting processor, Nature 574 (2019) 505–510, http://dx.doi.org/10.1038/s41586-019-1666-5.

[2] K. Rudinger, K. Young, E. Nielsen, R. Blume-Kohout, Measuring the capabilities of quantum computers, Nat. Phys. 18 (2022) 75–79, http://dx.doi.org/10.1038/s41567-021-01409-7.

[3] A.W. Cross, L.S. Bishop, S. Sheldon, P.D. Nation, J.M. Gambetta, Validating quantum computers using randomized model circuits, Phys. Rev. A 100 (2019) 032328, http://dx.doi.org/10.1103/PhysRevA.100.032328.

[4] M.-G. Zhou, Z.-P. Liu, W.-B. Liu, C.-L. Li, J.-L. Bai, Y.-R. Xue, Y. Fu, H.-L. Yin, Z.-B. Chen, Neural network-based prediction of the secret-key rate of quantum key distribution, Sci. Rep. 12 (2022) 8879, http://dx.doi.org/10.1038/s41598-022-12647-x.

[5] W. Peng, B. Wang, F. Hu, Y. Wang, X. Fang, X. Chen, Chao. Wan, Factoring larger integers with fewer qubits via quantum annealing with optimized parameters, Sci. China Phys. Mech. Astron. 62 (2019) 60311, http://dx.doi.org/10.1007/s11433-018-9307-1.

[6] M. M. Mosca, S.R. Verschoor, Factoring semi-primes with (quantum) SAT-solvers, Sci. Rep. 12 (2022) 7982, http://dx.doi.org/10.1038/s41598-022-11687-7.

[7] D. Deutsch, R. Jozsa, Rapid solutions of problems by quantum computation, Proc. R. Soc. Lond. Ser. A 439 (1907) (1992) 553–557, http://dx.doi.org/10.1098/rspa.1992.0167.

[8] M. Zidan, A.-H. Abdel-Aty, A. Khalil, M. Abdel-Aty, H. Eleuch, A novel efficient quantum random access memory, IEEE Access 9 (2021) 151775–151780, http://dx.doi.org/10.1038/s41598-022-11687-7, http://dx.doi.org/10.1109/ACCESS.2021.3119588.

[9] A. Parakh, Quantum teleportation with one classical bit, Sci. Rep. 12 (2022) 3392, http://dx.doi.org/10.1038/s41598-022-06853-w.

[10] K. Nagata, T. Nakamura, The Deutsch-Jozsa algorithm can be used for quantum key distribution, Open Access Library J. 2 (2015) 1798-1–1798-6, http://dx.doi.org/10.4236/oalib.1101798.

[11] D.M. Nguyen, S. Kim, Quantum key distribution protocol based on modified generalization of Deutsch-Jozsa algorithm in d-level quantum system, Internat. J. Theoret. Phys. 58 (1) (2019) 71–82, http://dx.doi.org/10.1007/s10773-018-3910-4.

[12] S.F. Hegazy, S.S.A. Obayya, B.E. A. Saleh, Randomized ancillary qubit overcomes detector-control and intercept–resend hacking of quantum key distribution, J. Lightwave Technol. 40 (21) (2022) 6995–7005, http://dx.doi.org/10.1109/JLT.2022.3198108.

[13] R. Shi, H. Xie, H. Feng, F. Yuan, B. Liu, Quantum zero correlation linear cryptanalysis, Quantum Inf. Process 21 (2022) 293, http://dx.doi.org/10.1007/s11128-022-03642-2.

[14] Manabputra B.K. Behera, P.K. Panigrahi, A simulational model for witnessing quantum effects of gravity using IBM quantum computer, Quantum Inf. Process. 19 (2019) 119, http://dx.doi.org/10.1007/s11128-020-2617-7.

[15] J. Eisert, M.M. Wilde, A smallest computable entanglement monotone, in: 2022 IEEE International Symposium on Information Theory, ISIT, 2022, pp. 2439–2444, http://dx.doi.org/10.1109/ISIT50566.2022.9834375.

[16] Luis Roa, María L. Ladrón de Guevara, Matias Soto-Moscoso, Pamela Catalán, The joint measurement entanglement can significantly offset the effect of a noisy channel in teleportation, J. Phys. A 51 (21) (2018) 215301, http://dx.doi.org/10.1088/1751-8121/aabbb2.

[17] W.S. Gan, Entanglement, in: Quantum Acoustical Imaging, Springer, Singapore, 2022, http://dx.doi.org/10.1007/978-981-19-0983-2_2.

[18] S.F. Hegazy, S.S. Obayya, Tunable spatial–spectral phase compensation of type-i (ooe) hyperentangled photons, J. Opt. Soc. Amer. B 32 (2015) 445–450, http://dx.doi.org/10.1364/JOSAB.32.000445.

[19] S.F. Hegazy, A.B. Yahia, S.A.O. Salah, Relative-phase and time-delay maps all over the emission cone of hyperentangled photon source, Opt. Eng. 56 (2) (2017) 026114, http://dx.doi.org/10.1117/1.OE.56.2.026114.

[20] S.F. Hegazy, S.S. Obayya, B.E. Saleh, Orthogonal quasi-phase-matched superlattice for generation of hyperentangled photons, Sci. Rep. 7 (1) (2017) 1–14, http://dx.doi.org/10.1038/s41598-017-03023-1.

[21] S.F. Hegazy, S.S. Obayya, Extended source of indistinguishable polarization-entangled photons over wide angles of emission, Appl. Phys. Lett. 117 (24) (2020) 244003, http://dx.doi.org/10.1063/5.0022646.

[22] M. Zidan, H. Eleuch, M. Abdel-Aty, Non-classical computing problems: Toward novel type of quantum computing problems, Results Phys. 21 (2021) 103536, http://dx.doi.org/10.1016/j.rinp.2020.103536.

[23] B. Hacker, et al., Deterministic creation of entangled atom–light Schrödinger-cat states, Nat. Photonics 13 (2) (2019) 110–115, http://dx.doi.org/10.1038/s41566-018-0339-5.

[24] D. Bluvstein, et al., A quantum processor based on coherent transport of entangled atom arrays, Nature 604 (7906) (2022) 451–456, http://dx.doi.org/10.1038/s41586-022-04592-6.

[25] T. Takayama, et al., Spin–orbit-entangled electronic phases in 4 d and 5 d transition-metal compounds, J. Phys. Soc. Japan 90 (6) (2021) 062001, http://dx.doi.org/10.7566/JPSJ.90.062001.

[26] M. Zidan, A novel computing model based on entanglement degree, Modern Phys. Lett. B 34 (35) (2020) 2050401, http://dx.doi.org/10.1142/S0217984920504011.

[27] B. Panda, N.K. Tripathy, S. Sahu, B.K. Behera, W.E. Elhady, Controlling remote robots based on zidan's quantum computing model, Comput. Mater. Continua 73 (3) (2022) 6225–6236, http://dx.doi.org/10.32604/cmc.2022.028394.

[28] A.T.H. Sim, M. Indrawan, S. Zutshi, B. Srinivasan, Logic-based pattern discovery, IEEE Trans. Knowl. Data Eng. 22 (6) (2010) 798–811, http://dx.doi.org/10.1109/TKDE.2010.49.

[29] M.E. Amyeen, W.K. Fuchs, I. Pomeranz, V. Boppana, Fault equivalence identification in combinational circuits using implication and evaluation techniques, IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. 22 (7) (2003) 922–936, http://dx.doi.org/10.1109/TCAD.2003.814241.

[30] Dutta R., Efficient approach to detect logical equivalence in the paradigm of software plagiarism, in: Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT), 2015, pp. 1–5, http://dx.doi.org/10.1109/C3IT.2015.7060114.

[31] Rosen K.H., Discrete Mathematics and Its Applications, McGraw-Hill Inc., NewYork, USA, 2012.

[32] K. Miura, K. Akama, H. Mabuchi, Creation of ET rules via logical equivalence, in: Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007), 2007, 468–468, http://dx.doi.org/10.1109/ICICIC.2007.230.

[33] Q. Li, Y. Gong, J. Zang, Y. Zhang, Y. Sun, Measurement method of civil engineering complexity structure based on logical equivalent model, Math. Probl. Eng. 2022 (2022) Article ID 6782822, 13 pages, http://dx.doi.org/10.1155/2022/6782822.
[34] S.S. Epp, Discrete Mathematics with Applications, Richard Stratton, 2010.
[35] M. Zidan, M.G. Eldin, M.Y. Shams, M. Tolan, A. Abd-Elhamed, M. Abdel-Aty, A quantum algorithm for evaluating the hamming distance, Comput. Mater. Continua 71 (1) (2022) 1065–1078, http://dx.doi.org/10.32604/cmc.2022.020103.
[36] IBM Quantum Experience. http://research.ibm.com/ibm-q/.