Quantum algorithm for solving the test-suite minimization problem

Hager Hussein1*, Ahmed Younes2,3 and Walid Abdelmoez1

Abstract: Test-suite minimization problem is an essential problem in software engineering as its application helps to improve the software quality. This paper proposes a quantum algorithm to solve the test-suite minimization problem with high probability in $O(\sqrt{2^n})$, where $n$ is the number of test cases. It generates an incomplete superposition to find the best solution. It also handles the non-uniform amplitudes' distribution case for the system with multisolutions. The proposed algorithm uses amplitude amplification techniques to search for the minimum number of test cases required to test all the requirements. The proposed algorithm employs two quantum search algorithms, Younes et al. algorithm for quantum searching via entanglement and partial diffusion to prepare incomplete superpositions that represent different search spaces such that the number of test cases is incremented in each search space, and updated Arima's algorithm to handle the multisolutions case. The updated Arima's algorithm searches for a quantum state that satisfies an oracle that represent the instance of the test-suite minimization problem.

ABOUT THE AUTHORS

Hager Hussein is an Assistant Lecturer in Software Engineering Department, College of Computing and Information Technology (CCIT), Arab Academy for Science, Technology and Maritime Transport (AASTMT). She received her MSc degree in software engineering from CCIT in AASTMT in 2012. Her research interests include quantum algorithms, software engineering.

Ahmed Younes is a Professor of Computer Science at Alexandria University and Honorary Research Fellow at School of Computer Science, University of Birmingham, United Kingdom. He is the founder and leader of Alexandria Quantum Computing Group. He obtained his PhD from University of Birmingham, United Kingdom, in 2004. He published many papers on Quantum Algorithms and Reversible Circuits.

Walid M. Abdelmoez is an Associate Professor in Software Engineering Department, CCIT, AASTMT. He received PhD degree in computer engineering at Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown, WV, USA, in 2006. His research interests are software engineering, software quality assurance, software risk assessment.

PUBLIC INTEREST STATEMENT

Test-suite minimization problem is an essential software engineering problem that has special importance in software testing. Evolutionary algorithms and other algorithms have been proposed to solve this problem. In this paper, a quantum algorithm is proposed to solve the test-suite minimization problem with high probability. Applying quantum algorithms to software engineering problems gives better results than that obtained using classical methods. This paper proposes a quantum algorithm that uses amplitude amplification techniques to search for the minimum number of test cases required to test all the requirements. The proposed algorithm employs two quantum search algorithms, Younes et al. algorithm for quantum searching via entanglement to prepare incomplete superposition, and an updated Arima's algorithm for searching the prepared search space for a possible solution to the given problem instance.
1. Introduction

Software testing is an essential stage in the software development life cycle. Redundant test cases can arise through the development of software versions. New test cases may arise for new versions of the software that cover the requirements tested by previous test cases for earlier software versions. Thus, software testing minimization techniques are required to reduce the number of test cases that cover the same requirements. Software testing minimization techniques are also required to utilize resources efficiently to reduce the cost and time requirements and to detect software bugs (Harman et al., 2012; Rothermel et al., 1998). There are many metaheuristic algorithms used to solve optimization problems, for example, quantum algorithms, quantum inspired algorithms, genetic algorithms, differential evolution, simulated annealing, ant and bee algorithms, bat algorithm, particle swarm optimization, and others as surveyed (Ahmed, 2016) and discussed next in this section.

Reduction can be performed classically using clustering such as Subashini and Jeyamala in 2014 (Subashini & Jeyamala, 2014) and Wang, Qu, Lu in 2015 (Wang et al., 2015) such that the program can be tested using any one of the clustered test cases. These clusters can be classified with the similarity in profiling (Alian et al., 2016) or by using the data mining approach of clustering (Wang et al., 2015). Many quantum algorithms are proposed to solve minimization problems for various applications, such as Anindita and Anirban (Banerjee & Pathak, 2009) who proposed a minimization algorithm for quantum circuit cost in 2010, where they minimize the number of gates in various selected quantum circuits. This algorithm can be applied on other circuits, but to do so, templates have to be developed for the corresponding gate library or the circuit has to be converted into other gate library with a previously prepared template. In 2020, Hussein et al. (Hussein et al., 2020) proposed a quantum inspired algorithm that performs better than the classical genetic version of the algorithm. Orsan Ozener and Hasan Sozer (Orsan Ozener & Sozer, 2020) proposed a formulation of the test-suite minimization problem in 2020 that solves the issues in heuristic techniques or integer linear programming focusing on a specific criterion or bi-criteria. In 2020, Yinxing Xue, and Yan Li (Xue & Li, 2020) proved that integer linear programming models multi-criteria test-suite minimization then they proposed a multi-objective integer programming approach to solve it.

The aim of the paper is to propose a quantum algorithm to solve the test-suite minimization problem. The proposed algorithm consists of two stages: the first stage prepares an incomplete superposition of a search space with certain properties using Younes et al. algorithm for quantum searching via entanglement (Younes et al., 2008). The second stage searches for a solution of the test-suite minimization problem in the prepared incomplete superposition using an updated version of Arima's algorithm for incomplete superposition searching (Arima et al., 2009).

This paper is organized as follows: Section 2 reviews the required background and discusses the quantum search algorithms used in the proposed algorithm. Section 3 proposes the quantum algorithm to solve the test-suite minimization problem. Section 4 shows the analysis of the searching phase. Section 5 discusses the proposed algorithm. Section 6 concludes the paper and mentions the future work.
2. Background

2.1. Test-Suite minimization problem
The aim of the test-suite minimization problem is to cover a given set of requirements with the smallest number of tests. The problem can be defined as follows (Panda & Mohapatra, 2017).

Given: A test suite $T$ with a set of $n$ test cases ($t_1, t_2, \ldots, t_n$) and a set $R$ of $m$ test requirements ($R_1, R_2, R_3, \ldots, R_m$) that must be satisfied. Each test case must cover one or more requirements and a requirement can be covered by one or more test cases.

Required: Find the minimum subset of $T$ that tests all the requirements.

For example, Table 1 illustrates a given test suite showing the requirements that are covered by each test case. Many solutions can be found that cover all the requirements, but the target is to find the minimum number of tests that cover all the requirements. For example, all the requirements can be covered with the test set ($t_1, t_2, t_3$) or ($t_1, t_2, t_4$), but the minimum set which covers all the requirements is ($t_1, t_2, t_4$). It becomes more complicated with large data sets. The TestNo column represents the test case number, while the $R$’s columns represent the requirements that are satisfied by each test case.

A test requirement matrix (TR) has to guarantee covering all the requirements, for example, to minimize the number of test cases shown in the test-suite minimization problem shown in Table 1, we start to represent the problem by its TR matrix as follows:

$$TR = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 \end{bmatrix}. \quad (1)$$

To solve this instance of the test-suite minimization problem that is shown in Table 1, the minimal number of true assignments that satisfy the following Boolean formula should be found as follows:

$$f_{TR} = (t_4) \land (t_1 \lor t_2 \lor t_3) \land (t_2) \land (t_3 \lor t_4) \land (t_2) \land (t_1 \lor t_2 \lor t_4),$$  \quad (2)

which is a reduction of the test-suite minimization problem to the SAT problem (satisfiability problem).

2.2. Quantum computing
The elementary unit of data in quantum computing is the quantum bit or qubit. The qubit can be in a superposed state that is a combination of state $|0\rangle$ and state $|1\rangle$ and when a measurement is performed, the superposition is collapsed to one of the states in a probabilistic way. A combination

| Table 1. Example for test-suite minimization problem |
|-----------------------------------------------|
| TestNo | $r_1$ | $r_2$ | $r_3$ | $r_4$ | $r_5$ | $r_6$ |
| $t_1$  | 0     | 1     | 0     | 1     | 1     | 1     |
| $t_2$  | 0     | 1     | 1     | 0     | 1     | 1     |
| $t_3$  | 0     | 1     | 0     | 1     | 0     | 0     |
| $t_4$  | 1     | 0     | 0     | 1     | 0     | 1     |

Page 3 of 16
of $|0\rangle$ and $|1\rangle$ before measurement is called a superposition as shown in Equation 3 where $\alpha, \beta$ are complex numbers representing the probabilistic amplitudes of $|0\rangle$ and $|1\rangle$ respectively. The amplitudes must satisfy the condition in Equation 4:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle,$$  \hfill (3)

and

$$|\alpha|^2 + |\beta|^2 = 1.$$  \hfill (4)

One of the quantum properties is entanglement, which means that each object of the quantum system can't be described independently, and instead, the quantum state has to be described for the whole system (Menon & Ritwik, 2014). Another feature in quantum computing is the parallelism where it takes a quantum computer a single step to operate on $N$ inputs with a single gate, while the classical computer takes $N$ steps for the same input size. Parallelism does not require additional hardware or to wait for other processes to complete, but it performs multiple operations at a time. Quantum gates are unitary operators, considering that a gate has $s$ inputs, and then it can be represented as $2^s \times 2^s$ unitary matrix assuming that state $|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$ and state $|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$.

Examples of such gates are:

The Hadamard gate with the following effect on $|0\rangle$ and $|1\rangle$:

$$H|0\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}} \left( |0\rangle + |1\rangle \right).$$  \hfill (5)

$$H|1\rangle = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \frac{1}{\sqrt{2}} \left( |0\rangle - |1\rangle \right).$$  \hfill (6)

The X gate which is equivalent to the NOT gate in classical computers maps $|0\rangle$ to $|1\rangle$ and $|1\rangle$ to $|0\rangle$ as shown in the following equation:

$$X|0\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix} = |1\rangle.$$  \hfill (7)

$$X|1\rangle = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle.$$  \hfill (8)

The Z gate that does not change the state $|0\rangle$ but it converts $|1\rangle$ to $-|1\rangle$ as shown in the following equation:

$$Z|0\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} = |0\rangle.$$  \hfill (9)

$$Z|1\rangle = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} = -|1\rangle.$$  \hfill (10)

Quantum circuits are combinations of elementary quantum gates to perform a certain task.
2.3. Quantum searching algorithms

The aim of this section is to review the three quantum algorithms that will be used in the proposed algorithm: Grover’s algorithm for searching an unstructured list via phase shift (Younes, 2008), Younes et al. algorithm for searching an unstructured list via entanglement (Younes et al., 2008), and Arima’s algorithm for searching in an incomplete superposition (Arima et al., 2009).

Grover’s algorithm is a fast searching algorithm, Ventura added the feature of searching in a subset of the whole superposition to Grover’s algorithm, Arima amended Ventura’s algorithm to guarantee high probability results, and the Partial diffusion operator that updated Grover’s algorithm to guarantee high probability results in case of having multiple copies of the searched data.

2.3.1. Grover’s Algorithm

Grover’s algorithm can search an unstructured database of size $N$ in $O(\sqrt{N})$ given that $N=2^n$. Grover’s algorithm requires the following initialization (Younes, 2008) of qubits where the first $n$ qubits are initialized to state $|0\rangle$ and an extra workspace qubit initialized to state $|1\rangle$,

$$|\psi_{G0}\rangle = |0\rangle^\otimes n \otimes |1\rangle,$$  

(11) then it applies $H^{\otimes n+1}$ on each of the $n+1$ qubits so that the first $n$ qubits contain the $N$ states representing the list and the extra qubit will be in state $(|0\rangle - |1\rangle)/\sqrt{2}$ as follows:

$$|\psi_G\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes \frac{|0\rangle - |1\rangle}{\sqrt{2}}.$$  

(12)

Grover’s algorithm then applies the oracle operator $O_g$ that evaluates a Boolean function $f_g: \{0, 1\}^n \rightarrow \{0, 1\}$. The operator $O_g$ gives the amplitudes of the matches a phase shift of $e^{i\pi}$ (Younes, 2008) as shown in Equation 13, and the system can be written as shown in Equation 14.

$$O_g|\psi_G\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle \otimes \left[ \frac{0 + f_g(i)|0\rangle - |1\rangle}{\sqrt{2}} \right].$$  

(14)

Next, it applies the diffusion operator $D$ on the first $n$ qubits. The diffusion operator $D$ is as follows:

$$D = H^{\otimes n}(2|0\rangle^\otimes n \langle 0|\otimes n - I_n)H^{\otimes n},$$  

(15)

where $I_n$ is the identity matrix of size $N \times N$. Consider a general system $|\psi_{G2}\rangle$ of $n$ qubits as follows:

$$|\psi_{G2}\rangle = \sum_{i=0}^{N-1} \alpha_i |i\rangle.$$  

(16)

Applying $D$ on the general system $|\psi_{G2}\rangle$ has the following effect:

$$D|\psi_{G2}\rangle = \sum_{i=0}^{N-1} \left[ -\alpha_i + 2\langle \alpha \rangle \right] |i\rangle,$$  

(17)

where $\langle \alpha \rangle = \frac{1}{n} \sum_{i=0}^{N-1} \alpha_i$ represents the mean of the states’ amplitudes.
The oracle operator $O_q$ and diffusion operator $D$ must be iterated approximately $\frac{\pi^2}{4N}$ times. Measurement is then performed on the first $n$ qubits to reveal one of the searched items.

2.3.2. Younes et al. Algorithm

An unstructured database of size $N$ can be searched with higher probability of success with an algorithm that uses the partial diffusion operator $(D_p)$ in $O(\sqrt{N/M})$ where $M$ is the number of matches satisfying $1 \leq M \leq N$. This algorithm amplifies the amplitudes of the solutions via entanglement of the search space with the extra qubit that is useful in the preparation of the incomplete superposition that can be done by applying the measurement on the extra qubit (Younes et al., 2003).

Younes et al. algorithm requires the following initialization of $n+1$ qubits where the $n+1$ qubits are initialized to state $|0\rangle$ (Younes et al., 2008):

$$|\psi_{Y0}\rangle = |0\rangle^\otimes n \otimes |0\rangle. \tag{18}$$

then applies $H$ on the first $n$ qubits as follows:

$$|\psi_{Y1}\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{2^n-1} |i\rangle \otimes |0\rangle. \tag{19}$$

This algorithm then applies the oracle operator $O_Y$ on $n+1$ qubits where $O_Y$ evaluates the Boolean function $f_Y: \{0, 1\}^n \rightarrow \{0, 1\}$ as shown in the following equation:

$$O_Y|x, 0\rangle = \begin{cases} |x, 0\rangle, & f_Y(x) = 0 \\ |x, 1\rangle, & f_Y(x) = 1. \end{cases} \tag{20}$$

Next, it applies the $D_p$ operator which can be described in the following form (Younes et al., 2008):

$$D_p = (H^\otimes n \otimes I_1)(2/|0\rangle^\otimes n+1)(|0\rangle^\otimes n+1 - I_{n+1})(H^\otimes n \otimes I_1). \tag{21}$$

where $|0\rangle$'s size is $2N = 2^{n+1}$, and the identity matrix $I_k$ is of size $2^k \times 2^k$.

Consider a general system $|\psi_{Y2}\rangle$ of $n+1$ qubits as follows:

$$|\psi_{Y2}\rangle = \sum_{k=0}^{2^{n-1}} \delta_k |k\rangle. \tag{22}$$

The state $|\psi_{Y2}\rangle$ can be re-written to

$$|\psi_{Y2}\rangle = \sum_{k=0}^{N-1} \alpha_k |k\rangle \otimes |0\rangle + \sum_{k=0}^{N-1} \beta_k |k\rangle \otimes |1\rangle. \tag{23}$$

where $\alpha_k = \delta_k : k \text{ even}$ and $\beta_k = \delta_k : k \text{ odd}$.

Applying $D_p$ on the general system has the following effect:

$$D_p|\psi_{Y2}\rangle = \sum_{k=0}^{N-1} (2(\alpha_k - \alpha_k)(|k\rangle \otimes |0\rangle) - \sum_{k=0}^{N-1} \beta_k (|k\rangle \otimes |1\rangle). \tag{24}$$

where $\langle \alpha \rangle = \sum_{k=0}^{2N-1} \alpha_k / N$ is the mean of the amplitudes of the subspace entangled with the $|0\rangle$ of the extra qubit. The $O_Y$ and $D_p$ operators are iterated $q_p$ times where $q_p$ is as follows:
\[ q_p = \frac{\pi}{2\sqrt{2}} \sqrt{\frac{N}{M}}, \quad 1 \leq M \leq N. \]  

Assume that \( \Sigma ' \) is the sum over the desired \( M \) matches, while \( \Sigma '' \) represents the sum over the \( N - M \) undesired matches. The system after \( q_p \geq 1 \) iterations can be described as follows:

\[ |\psi_{y3}\rangle = a_q \sum_{j=0}^{N-1} n(j) \otimes |0\rangle + b_q \sum_{i=0}^{N-1} n(i) \otimes |0\rangle \]

\[ + c_q \sum_{j=0}^{N-1} (\bar{\alpha}_j \otimes |1\rangle), \]

where \( a_q, b_q, \) and \( c_q \) values are as follows: Let \( y = 1 - \frac{M}{N} \) and \( \nu = \frac{1}{\sqrt{N}}, \) and then (Younes et al., 2008),

\[ \langle \alpha_q \rangle = y a_{q-1} + (1 - y) c_{q-1}. \]

\[ a_0 = \nu, \quad a_1 = \nu(2y - 1), \quad a_q = 2 \langle \alpha_q \rangle - a_{q-1}. \]

\[ b_0 = \nu, \quad b_1 = 2\nu y, \quad b_q = 2 \langle \alpha_q \rangle - c_{q-1}. \]

\[ c_0 = 0, \quad c_1 = -\nu, \quad c_q = -b_{q-1}. \]

Finally, the first \( n \) qubits are measured to get one of the searched items.

2.3.3. Arima’s Algorithm

Grover’s algorithm is effective when the initial amplitude distribution of dataset is uniform, which means that \( N \) is equal to the number of stored data, but it is not always effective in the non-uniform cases where \( N \) is not equal to the number of stored data. Arima’s algorithm was proposed to solve the search in an incomplete superposition.

Given an incomplete superposition \( |\psi_{in}\rangle \) and an oracle \( U_{TR} \) that evaluates to \( |1\rangle \) for states in \( |\psi_{in}\rangle \), and an oracle \( U_f \). It is required to find any state in \( |\psi_{in}\rangle \) that makes \( U_f \) evaluate to 1. Arima’s algorithm is summarized as (Arima et al., 2009),

- Initial state \( |\psi_{in}\rangle \).
- Repeat \( q_o \) times, where \( q_o = \frac{\pi}{\sqrt{N}} \).
- \( |\psi_n'\rangle = U_{TR} |\psi_{in}\rangle \).
- \( |\psi_n''\rangle = D |\psi_{n'}\rangle \).
- \( |\psi_n'''\rangle = U_f |\psi_{in}\rangle \).
- \( |\psi_{in}\rangle = D |\psi_{n''}\rangle \).
- Observe the system.
The test-suite minimization problem definition has been illustrated. Quantum computing basics have been shown. Quantum searching algorithms that will be used in the proposed algorithm have been explained in detail.

3. The proposed algorithm

3.1. Methodology

The main idea of the proposed algorithm is to search the power set of the set of test cases in rounds, excluding the empty set since the problem is promised to have a solution. Each round searches simultaneously the set of all subsets from the power set that have the same cardinality, and if no solution is found, then increment the cardinality of the subsets in the search space next round. For example, given the set of test cases \( \{t_1, t_2, t_3, t_4\} \), the first round searches in a search space that contains a single test, i.e. \( \{\{t_1\}, \{t_2\}, \{t_3\}, \{t_4\}\} \), the second round searches in a search space that contains two tests, i.e. \( \{\{t_1, t_2\}, \{t_1, t_3\}, \{t_1, t_4\}, \ldots, \{t_3, t_4\}\} \), and so on.

The proposed algorithm prepares an incomplete superposition using Younes et al. algorithm where all the states in the incomplete superposition have the same Hamming distance with the state \( |0^{\otimes n}\rangle \). Starting with hamming distance equals to 1, the proposed algorithm searches for a state that satisfies the oracle that represents the TR using an updated version of Arima's algorithm. If the TR oracle is satisfied with a state in the prepared incomplete superposition, then the algorithm terminates and the truth assignment from the incomplete superposition is reported as the minimum solution, and otherwise prepare the next incomplete superposition with an incremented hamming distance and restart the algorithm.

The proposed algorithm has a maximum of \( O(\log_2 N) \) rounds, where the first round will search in \( nC_1 \) states and the second round will search in \( nC_2 \) states and so on. After \( \log_2 N \) rounds in the worst case, the algorithm will search the \( N \) states since \( \sum_{i=0}^{n} nC_i = N \), where state \( |0^{\otimes n}\rangle \) is excluded since it is guaranteed that the \( m \) requirements will be covered with a subset of the \( n \) test cases.

3.2. Problem encoding

Given a test-suite minimization problem as shown in Table 1. Assuming that all the requirements will be satisfied by a subset of the set of test cases, the problem can be represented as TR matrix, for example, the problem shown in XTable 1 can be represented as shown in the TR matrix shown in Equation 1. The TR matrix will be the input to the proposed algorithm.

The idea of the proposed algorithm is to represent the tests to be applied as quantum states, for example, given a set of tests \( \{t_1, t_2, t_3, t_4\} \). Applying the subset \( \{t_1, t_2, t_4\} \) is represented as the quantum state \( |1101\rangle \). The proposed algorithm prepares an incomplete superposition of states with the same Hamming distance as will be illustrated next, for example, \( |0001\rangle \), \( |0010\rangle \), \( |0100\rangle \), and \( |1000\rangle \) have the same Hamming distance with state \( |0000\rangle \), which is equal to 1. The algorithm then searches if the operator representing \( f_R \) is satisfied by one of these states, otherwise, increment the number of 1's in the prepared states and repeat the algorithm until a solution with the minimum number of tests is reached.

3.3. Preparation of states with a given hamming distance

A TR matrix of size \( n \times m \) is considered as an input, and this means we will need to prepare \( n + 1 \) qubit system. The quantum circuit shown in Figure 1 illustrates the algorithm steps, which will be explained in detail in this section.

The algorithm first initializes \( n + 1 \) qubit system to \( |0\rangle \) so that the system takes the following state:

\[
|\psi_1\rangle = |0^{\otimes n}\rangle \otimes |0\rangle,
\]
then apply $H$ on the first $n$ qubits to prepare the complete superposition. The system will be

$$|\psi_2\rangle = (H^n \otimes I)|\psi_1\rangle = \frac{1}{\sqrt{N}} \sum_{j=0}^{N-1} |j\rangle \otimes |0\rangle. \quad (32)$$

The $TR$ matrix shown in Table 1 will be taken as a working example to illustrate the proposed algorithm. Using the working example, the complete superposition stored in $|\psi_2\rangle$ will be equal to

$$|\psi_2\rangle = \frac{1}{4} (|0000\rangle + |0001\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |1011\rangle + |1100\rangle + |1110\rangle + |1111\rangle) \otimes |0\rangle. \quad (33)$$

The proposed algorithm prepares superposition of states containing a specific $t$ number of 1's that is $nt$ states. For a set of states that have $t$ number of 1's, define $f_n$ as follows:

$$f_n(x) = \begin{cases} 1, & \text{if} \text{ham}(x) = h, h = 1, 2, \ldots, n \\ 0, & \text{o.w.} \end{cases} \quad (34)$$

Using the working example, $f_n$ can be written as follows (Younes & Rowe, 2015), where $n = 4$ and $t = 3$,

$$f_n = (t_1 \wedge t_2 \wedge t_3 \wedge t_4) \lor (t_1 \wedge \bar{t}_2 \wedge t_3 \wedge t_4) \lor (t_1 \wedge t_2 \wedge \bar{t}_3 \wedge t_4) \lor (t_1 \wedge t_2 \wedge t_3 \wedge \bar{t}_4). \quad (35)$$

This can be represented as Reed–Muller expansion (Younes & Miller, 2004) so it can be represented as follows:

$$f_n = (t_2 \wedge t_3 \wedge t_4) \oplus (t_1 \wedge t_3 \wedge t_4) \oplus (t_1 \wedge t_2 \wedge t_4) \oplus (t_1 \wedge t_2 \wedge t_3), \quad (36)$$

then the proposed algorithm applies the oracle $U_{f_n}$ for the function $f_n$ as shown in the quantum circuit in Figure 2 (Younes & Miller, 2004). $U_{f_n}$ is defined as follows:

$$U_{f_n}(|x\rangle \otimes |0\rangle) = \begin{cases} (|x\rangle \otimes |1\rangle), & f_n(x) = 1 \\ (|x\rangle \otimes |0\rangle), & f_n(x) = 0 \end{cases}. \quad (37)$$

In this paper, the quantum circuits for boolean functions will be represented as Reed–Muller expansions for illustration. In the proposed algorithm, the query complexity will be considered where using a boolean function will be counted once since the optimization of the quantum circuits for boolean functions is a different research area irrelevant to the scope of the paper.
After applying $U_h$, we apply the partial diffusion operator as will be shown next. The current system can be rewritten as a summation of two subspaces, the subspace of the non-matches that is denoted as $\sum$ and the subspace of the matches that is denoted as $\sum$.

$$|\psi_3\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} n(|i\rangle \otimes |0\rangle) + \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} (|i\rangle \otimes |1\rangle).$$  \hspace{1cm} (38)

Let us consider the following equation:

$$|\psi_q\rangle = |0\rangle^\otimes n \otimes |0\rangle.$$ \hspace{1cm} (39)

The proposed algorithm applies $D_pU_h$ for $q$ times, where

$$q = \frac{\pi}{2\sqrt{2}} \sqrt{\frac{N}{nt}}.$$ \hspace{1cm} (40)

Then, the system will be updated to be

$$|\psi'_q\rangle = \sum_{i=0}^{N-1} a_i |i\rangle \otimes |0\rangle$$

$$+ \sum_{i=0}^{N-1} b_i |i\rangle \otimes |0\rangle$$

$$+ \sum_{i=0}^{N-1} c_i |i\rangle \otimes |1\rangle.$$ \hspace{1cm} (41)

Using the working example with $t = 3$ and $n = 4$, $q$ will be equal to $\frac{\pi}{\sqrt{2}}$, which is approximately 2 and the system will be updated as follows:

$$|\psi'_q\rangle = \frac{1}{2} (|0000\rangle + |0001\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |0110\rangle + |1000\rangle + |1001\rangle + |1010\rangle + |1100\rangle + |1101\rangle + |1110\rangle) \otimes |0\rangle$$

$$+ \frac{1}{2} (|0111\rangle + |1011\rangle + |1101\rangle + |1110\rangle) \otimes |0\rangle$$

$$- \frac{1}{2} (|0111\rangle + |1011\rangle + |1101\rangle + |1110\rangle) \otimes |1\rangle.$$ \hspace{1cm} (42)

Then, the proposed algorithm applies measurement on the extra qubit and proceeds by applying $Z$ then $H$ on the extra qubit, if the outcome is $|1\rangle$, otherwise restarts the preparation stage. The probability to get $|1\rangle$ on the extra qubit is $nt|c_q|^2$. If the outcome is $|1\rangle$, then the superposition can be represented as

$$|\psi'_q\rangle = \frac{1}{\sqrt{nt}} \sum_{h=0}^{nt} |i\rangle \otimes |1\rangle$$ \hspace{1cm} (43)
Using the working example, assuming that $U_f$ is an oracle to prepare all possible states with $t = 3$, applying $D_q U_f$ for $q = 2$ times prepares all the possible solutions with $t = 3$ resulting in the following incomplete superposition:

$$|\psi'_3\rangle = \frac{1}{2} (|0111\rangle + |1011\rangle + |1101\rangle + |1110\rangle) \otimes |1\rangle. \quad (44)$$

### 3.4. Searching the incomplete superposition

Given an incomplete superposition of states that have a specific $t$ number of 1's, we amended Arima's algorithm (Arima et al., 2009) to search for the assignment that satisfies $U_{TR}$ in the case of having multisolutions. Considering the given matrix $TR$ in Equation 1, $f_{TR}$ can be represented as shown in Equation 2 (Younes & Rowe, 2015). This can be represented as Reed-Muller expansion (Younes & Miller, 2004) so it can be represented as follows:

$$f_{TR} = t_1 \lor t_2 \lor t_4. \quad (45)$$

then use the oracle $U_{TR}$ for the function $f_{TR}$ in the algorithm as shown in the quantum circuit in Figure 3 (Younes & Miller, 2004).

The proposed algorithm amended Arima's algorithm to search for the states that satisfy $f_{TR}$ in the prepared incomplete superposition in the case of having multisolutions, so we are searching for states recognized by $f$, and to satisfy $f_{TR}$ if exists, and otherwise, we prepare another an incomplete superposition with an incremented Hamming distance and repeat the algorithm. The updated Arima's algorithm is applied for $\frac{\sqrt{n}}{4}$ times and report whether or not the prepared superposition contains a state that satisfies $f_{TR}$.

Using the working example with $t = 3$ and $n = 4$, the number of iterations of Arima's algorithm, $q_u = 2$. The system of the working example will be updated as follows:

$$|\psi''_3\rangle = \frac{1}{8\sqrt{2}}(|0000\rangle + |0001\rangle + |0010\rangle + |0011\rangle + |0100\rangle + |0101\rangle + |0110\rangle - |0111\rangle + |1000\rangle + |1001\rangle + |1010\rangle - |1011\rangle + |1100\rangle - |1110\rangle + |1111\rangle) \otimes |0\rangle$$

$$+ \frac{1}{8\sqrt{2}}(-|0000\rangle - |0001\rangle - |0010\rangle - |0011\rangle - |0100\rangle - |0101\rangle - |0110\rangle + |0111\rangle - |1000\rangle - |1001\rangle - |1010\rangle + |1011\rangle - |1100\rangle + |1110\rangle - |1111\rangle) \otimes |1\rangle$$

$$+ \frac{7}{8\sqrt{2}}(|1101\rangle \otimes |0\rangle - \frac{7}{8\sqrt{2}}(|1101\rangle \otimes |1\rangle. \quad (46)$$

**Figure 3.** Quantum circuit for the $U_{TR}$.
The probability of the desired pattern is calculated as in Arima et al., (2009) with a minimized value of the number of stored data because of our incomplete superposition preparation step, which in turn improves the probability. In the working example, the desired pattern of $|1101>$ is obtained with probability $2(\frac{2}{8/2})^2 \approx 0.766$.

The methodology of the proposed algorithm has been illustrated. Problem encoding, preparation phase, and searching the incomplete superposition phase have been explained.

4. Analysis of the searching phase

To analyze the dynamics of the searching phase, $|\psi_{\chi}'>=$ can be generalized to the form:

$$|\psi_{\chi}'> = \frac{1}{\overline{N}} \sum_{k=1}^{h} a_{k}(t_{s})(|i> \otimes |1>) + \frac{1}{m} \sum_{m_{i=1}}^{m} \beta_{i}(t_{s})(|i> \otimes |0>) + \frac{1}{n} \sum_{n_{j=1}}^{n} \gamma_{j}(t_{s})(|i> \otimes |1>)$$

where we define the average of the amplitudes of the subspace $\frac{1}{\overline{N}} \sum_{k=1}^{h} a_{k}(t_{s})(|i> \otimes |1>)$ of possible solutions at time $t_{s}$ by $a_{k}(t_{s})$ where $k$ starts with 1 and increments to $h$ which is the number of possible solutions. The average of the amplitudes of the subspace $\frac{1}{m} \sum_{m_{i=1}}^{m} \beta_{i}(t_{s})(|i> \otimes |0>)$ of non-solutions by $\beta_{i}(t_{s})$ where $i$ takes the range of $h$ to $m$. Thus, $a_{k}(t_{s})$ and $\beta_{i}(t_{s})$ are average amplitudes of the subspace $\frac{1}{n} \sum_{n_{j=1}}^{n} \gamma_{j}(t_{s})(|i> \otimes |1>)$ of the stored data, while the amplitude of the other data that were not in the prepared superposition at time $t_{s}$ is defined by $\gamma_{j}(t_{s})$ where $j$ takes values from $m$ to $N$.

To calculate the averages of such amplitudes, we use the following equations:

$$\overline{a}(t_{s}) = \frac{1}{\overline{N}} \sum_{k=1}^{h} a_{k}(t_{s}).$$

$$\overline{\beta}(t_{s}) = \frac{1}{m - \overline{N}} \sum_{m_{i=1}}^{m} \beta_{i}(t_{s}).$$

$$\overline{\gamma}(t_{s}) = \frac{1}{N - m} \sum_{n_{j=1}}^{n} \gamma_{j}(t_{s}).$$

Considering that the initial distribution at $t_{s} = 0$ is arbitrary. The weighted average over states is calculated as follows:

$$W(t_{s}) = \frac{1}{\overline{N}} \sum_{n_{j=1}}^{n} \gamma_{j}(t_{s}) + \sum_{m_{i=1}}^{m} \beta_{i}(t_{s}) - \sum_{k=1}^{h} a_{k}(t_{s})$$

$$= \frac{1}{\overline{N}} \left( (N - m) \overline{\gamma}(t_{s}) + (m - h) \overline{\beta}(t_{s}) - (h) \overline{a}(t_{s}) \right).$$

Then, the following relation holds based on (Biron et al., 1998):

$$\overline{a}(t_{s} + 1) = 2W(t_{s}) + \overline{a}(t_{s}).$$

$$\overline{\beta}(t_{s} + 1) = 2W(t_{s}) + \overline{\beta}(t_{s}).$$
\[\gamma(t_s + 1) = 2w(t_s) - \gamma(t_s).\]  \hfill (54)

After solving Equation 52, Equation 53, and Equation 54, the following amplitude averages can be calculated as follows:

\[
\begin{bmatrix}
\gamma(t_s + 1) \\
\bar{\gamma}(t_s + 1) \\
\gamma(t_s + 1)
\end{bmatrix} =
\begin{bmatrix}
\frac{N^2 - 8N - 8(m - h)}{N^2} & \frac{8(m - h)(N - m)}{N^2} & \frac{4(N - m)(N - 2m - 2h + 1)}{N^2} \\
\frac{8(m - h)(N - m)}{N^2} & \frac{8(m - h)(N - m - N)}{N^2} & \frac{4(N - m)(N - 2m - 2h + 1)}{N^2} \\
\frac{4(N - 2m - 2h)}{N^2} & \frac{4(N - 2m - 2h)}{N^2} & \frac{8m - 8m - 4h + 1 - 1}{N^2}
\end{bmatrix}
\begin{bmatrix}
\gamma(t_s) \\
\bar{\gamma}(t_s) \\
\gamma(t_s)
\end{bmatrix}.
\hfill (55)

The analysis of the searching phase has been explained in detail. The amplitude averages can then be calculated at any given time \(t_s\).

5. Discussion

A pseudocode is shown in Algorithm 1 to summarize the proposed algorithm.

**Algorithm 1: Pseudocode for the proposed algorithm**

1. Prepare \(|\psi\rangle\).
2. Apply \(H\) on the first \(n\) qubits.
3. for \(q = 1\) to \(\frac{\sqrt{N}}{2^{\frac{N}{2}}}\) do
4. Apply \(D_pU_h\) on \(|\psi_\ell\rangle\).
5. end for
6. Measure the extra qubit.
7. if The outcome is \(|1\rangle\) then
8. Apply \(Z\) on the extra qubit.
9. Apply \(H\) on the extra qubit.
10. for \(q_o = 1\) to \(\frac{\sqrt{N}}{2^{\frac{N}{2}}}\) do
11. Apply \(DU_hDU_{tr}\) on \(|\psi_\ell\rangle\).
12. end for
13. else
14. Go to step 2.
15. end if

The target of the proposed algorithm is to search for the best solution for the test-suite minimization problem. Equation 31 shows the system initialization. The \(n + 1\) qubits are initialized to \(|0\rangle\) and then \(H\) is applied to the first \(n\) qubits as shown in Equation 32. These steps prepares the complete superposition. A superposition of \(t\) number of 1’s is needed; thus, the oracle \(U_h\) and the partial diffusion operator \(D_p\) are applied for \(q\) times as shown in Equation 40. This step prepares the incomplete superposition that contains a candidate solution we are trying to find. The extra qubit is then measured to amplify the amplitudes of the solutions via entanglement of the search space with this extra qubit. In case that the outcome is \(|1\rangle\), \(Z\) is applied on the extra qubit, and then \(H\) is applied on this extra qubit to update the system to be as shown in Equation 44. The updated version of Arima’s is then applied to search for the best solution in the prepared incomplete superposition. Equation 47 shows the system after applying the updated Arima’s algorithm \(q_o\) times in its general form. Equation 47 also shows the average amplitudes for the possible solutions represented by \(\gamma(t_s)\), the average amplitudes for the non-solutions represented by \(\bar{\gamma}(t_s)\), and the average amplitudes for the other data that were not in the prepared superposition represented by \(\gamma(t_s)\) and the weighted averages are shown in Equation 51. The average of the amplitudes can be calculated at any time \(t_s\) by Equation 55. The proposed algorithm has a maximum of \(O(\log_2 N)\) rounds.
This paper proposes a solution to the test-suite minimization problem. It is crucial to software testing because it helps developers to improve the software quality by finding the minimum test that covers all the requirements. Test sizes increase with software modifications over time in the maintenance phase; thus, the proposed algorithm is significant to find the minimum number of tests through this periodic test size growth. The proposed algorithm saves software testing resources such as time and money by finding the minimum sized solution with high probability.

6. Conclusion and future work
This paper proposed a quantum algorithm to solve the test-suite minimization problem. The proposed algorithm used quantum search algorithms to search for the minimum number of test cases required to test all the requirements. The algorithm encoded a subset of the power set of the set of test cases as quantum states. The two main phases of the algorithm are the states preparation phase and the searching phase. In the preparation phase, the algorithm used Younes et al algorithm for quantum searching via entanglement and partial diffusion to prepare an incomplete superposition that represents a subset of sets from the power set that have the same cardinality. The searching phase was applied by amending Arima’s algorithm to find a possible solution to the given test-suite minimization problem instance in the prepared incomplete superposition. If no solution is found, then the algorithm prepares another incomplete superposition with the cardinality of the subsets in the search space incremented next round. Applying this algorithm to a test-suite minimization problem instance gives the best solution with the minimum required test cases that saves software testing resources such as time and money. The proposed algorithm is crucial while testing the software in the maintenance phase as test sizes increase with software modifications over time. Applying the proposed algorithm is important in case of software growth from a version to another where a requirement that is tested in a previous version will be tested again in the newer version that may result in redundant test cases.

The algorithm needs $O(\log_2 N)$ rounds to search the power set of all possible solutions. It was shown that the algorithm can find a solution with high probability in $O\left(\sqrt{2^n}\right)$, where $n$ is the number of test cases.

For future work, we intend to propose efficient quantum algorithms to solve other software engineering problems such as next release problem and the structural test data generation problem (Harman et al., 2012).

Author details
Hager Hussein
E-mail: hager.hussein1@aast.edu
ORCID ID: http://orcid.org/0000-0003-2678-8337
Ahmed Younes
Walid Abdelmoez
1 Department of Software Engineering, College of Computing and Information Technology, Arab Academy for Science and Technology, Alexandria, Egypt. 2 Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Alexandria, Egypt. 3 School of Computer Science, University of Birmingham, Birmingham, B15 2TT, UK.

Disclosure statement
No potential conflict of interest was reported by the authors.

Funding
The authors received no direct funding for this research.

Data Availability
The data are readily available upon request.

Citation information
Cite this article as: Quantum algorithm for solving the test-suite minimization problem, Hager Hussein, Ahmed Younes & Walid Abdelmoez, Cogent Engineering (2021), 8: 1882116.

References
Ahmed, B. S. (2016). Test case minimization approach using fault detection and combinatorial optimization techniques for configuration-aware structural testing. International Journal of Engineering Science and Technology, 19(2), 737–753.
Alaun, M., Suleiman, D., & Shaeouat, A. (2016). Test case reduction techniques – Survey. [IJACSA] International Journal of Advanced Computer Science and Applications, 7(5), 264–275.
Arima, K., Shigei, N., & Miyajima, H., A proposal of a quantum search algorithm, Fourth International Conference on Computer Sciences and Convergence Information Technology, Korea. https://doi.org/10.1109/ICCIT.2009.126, (2009).
Banerjee, A., & Pathak, A. (2009). An algorithm for minimization of quantum cost, Applied Mathematics & Information Sciences. arXiv: 0910.2129v2, 6(1).
Biron, D., Biham, O., Biham, E., Grassl, M., & Lidar, D. A. (1998). Generalized grover search algorithm for arbitrary initial amplitude distribution. Lecture Notes In Computer Science, 1509, 140–147.

Harran, M., McMinn, P., de Souza, J. T., & Yoo, S., Search based software engineering: Techniques, taxonomy, tutorial, LASER Summer School 2008–2010, LNCs 700, Springer, pp. 1–59 (2012).

Hussein, H., Younes, A., & AbdelMoez, W. (2020). Quantum- inspired genetic algorithm for solving the test suite minimization problem. WSEAS TRANSACTIONS on COMPUTERS, 19, E-ISSN: 2224-2872. https://doi.org/10.37394/23205.2020.19.20.

Menon, P. S., & Ritwik, M., A comprehensive but not complicated survey on quantum computing. International Conference on Future Information Engineering, China, pp. 144–152 (2014).

Mohapatra, S. K., & Prasad, S. (2015). Test case reduction using ant colony optimization for object oriented program. International Journal of Electrical and Computer Engineering (IJECE), 5(6), 1424–1432. https://doi.org/10.11591/ijece.v5i6.pp1424-1432

Orsan Ötener, O., & Sozer, H. (2020). An effective formulation of the multi-criteria test suite minimization problem. Journal of Systems and Software, 168. ISSN: 0164–1212. https://doi.org/10.1016/j.jss.2020.110632

Pando, S., & Mohapatra, D. P. (2017). Regression test suite minimization using integer linear programming model. Software: Practice & Experience, 1539–1560.

Rothermel, G., Harrold, M. J., Ostrin, J., & Hong, C. (1998). An empirical study of the effects of minimization on the fault detection capabilities of test suites. International Conference on Software Maintenance, Luxembourg. https://doi.org/10.1109/cism.1998.738487 .

Srivasatava, P. R., & Kim, T. (2009). Application of genetic algorithms to software testing. International Journal of Software Engineering and Its Applications, 3(4).

Subashini, B., & Jeyamala, D., Reduction of test cases using clustering technique. International journal of innovative research in science, engineering and technology, vol. 3, Special Issue 3, International Conference on Innovations in Engineering and Technology (ICIET’14), Malaysia, pp. 1992–1995 (2014).

Wang, R., Qu, B., & Lu, Y. (2015). Empirical study of the effects of different profiles on regression test case reduction. IET Software, 9(2), 29–38. https://doi.org/10.1049/iet-sen.2014.0008

Xue, Y., & Li, Y. (2020). Multi-objective integer programming approaches for solving the multi-criteria test-suite minimization problem: Towards sound and complete solutions of a particular search-based software-engineering problem. ACM Transactions on Software Engineering and Methodology, 29(3), 1–50. https://doi.org/10.1145/3392031

You, L., & Lu, Y. (2012). A genetic algorithm for the time-aware regression testing reduction problem, International Conference on Natural Computation, China.

Younes, A. (2008). Strength and weakness in grover’s quantum search algorithm. arXiv:0811.4481v1.

Younes, A., & Miller, J. F. (2004). Representation of boolean quantum circuits as reed-muller expansions. International Journal of Electronics, 91(7), 431–444. https://doi.org/10.1080/00207140420002541285

Younes, A., Rowe, J., & Miller, J. (2003). Quantum search algorithm with more reliable behaviour using partial diffusion, AIP Conference Proceedings, USA. https://doi.org/10.1063/1.1834408

Younes, A., Rowe, J., & Miller, J. (2008). Enhanced quantum searching via entanglement and partial diffusion, Physica D: Nonlinear Phenomena, 237(8), 1074–1078.

Younes, A., & Rowe, J. E. (2015). A polynomial time bounded-error quantum algorithm for boolean satisfiability. arXiv:1507.05061v2.
