A Polynomial Time Bounded-error Quantum Algorithm for Boolean Satisfiability

Ahmed Younes*[1,2] and Jonathan E. Rowe†[2]

[1]Department of Mathematics and Computer Science, Faculty of Science, Alexandria University, Egypt
[2]School of Computer Science, University of Birmingham, Birmingham, B15 2TT, United Kingdom

July 20, 2015

Abstract

The aim of the paper to answer a long-standing open problem on the relationship between NP and BQP. The paper shows that BQP contains NP by proposing a BQP quantum algorithm for the MAX-E3-SAT problem which is a fundamental NP-hard problem. Given an E3-CNF Boolean formula, the aim of the MAX-E3-SAT problem is to find the variable assignment that maximizes the number of satisfied clauses. The proposed algorithm runs in $O(m^2)$ for an E3-CNF Boolean formula with m clauses and in the worst case runs in $O(n^6)$ for an E3-CNF Boolean formula with n inputs. The proposed algorithm maximizes the set of satisfied clauses using a novel iterative partial negation and partial measurement technique. The algorithm is shown to achieve an arbitrary high probability of success of $1-\epsilon$ for small $\epsilon > 0$ using a polynomial resources. In addition to solving the MAX-E3-SAT problem, the proposed algorithm can also be used to decide if an E3-CNF Boolean formula is satisfiable or not, which is an NP-complete problem, based on the maximum number of satisfied clauses.

Keywords: Quantum Algorithm, MAX-E3-SAT,E3-SAT, Amplitude Amplification, BQP, NP-hard, NP-complete.

1 Introduction

Long-standing open problem in quantum computing is the relationship between the classes NP and BQP [2, 4]. Decision problems are in NP if yes-instances have witnesses that can be checked in polynomial time [12]. The class BQP is the quantum computing analogue of the classical class BPP (bounded error probabilistic polynomial) [6]. A problem is in BPP if there is a probabilistic classic algorithm (Turing machine with access to random bits) which makes

*ayounes@alexu.edu.eg or ayounes2@yahoo.com
†J.E.Rowe@cs.bham.ac.uk

errors (for either yes or no instances) with probability of given a wrong answer at most 1/3. The value 1/3 is arbitrary - all that is required is that the value is bounded away from 1/2. By repeated runs, the probability of failure can be made exponentially small. The problem class BQP replaces the classical algorithm with a quantum algorithm [1]. Thus a decision problem is in BQP if there is a quantum algorithm for it with probability of being wrong less than 1/3.

The common belief concerning the relationship between NP and BQP was that NP is not contained in BQP (e.g see chapter 15 of [8]. However, a recent paper by one of the current authors has shown that an NP-hard problem (Graph Bisection) can be efficiently solved, with low failure probability, by a quantum algorithm [13]. This implies that NP is in fact contained within BQP, as any NP problem can be polynomially reduced to an NP-hard problem. The Graph Bisection problem is, perhaps, somewhat obscure and much of the presentation of that result involves ensuring the balance of the partition, which detracts from the main features of the approach. Consequently in the current paper, we will directly address the classic Boolean Satisfiability problem (SAT) to show precisely how constraints, expressed as Boolean formula, can be encoded into quantum algorithm. The constraints are entangled with the superposition of all possible truth-value assignments and a probability amplification technique applied to amplify the assignment which maximizes the number of satisfied clauses.

In particular, we will focus on MAX-E3-SAT [11], in which each clause contains exactly three literals, and we will show that our quantum algorithm will solve this maximization problem with high probability of success. In particular, it can then be used to solve the decision problem (with high probability). Iterating the process allows the probability of failure to be made exponentially small.

A key fact about the MAX-E3-SAT problem is that random truth assignments will satisfy, in expectation, 7/8 of the clauses. A consequence of the PCP Theorem is that this cannot be improved upon (more precisely, there is no $(7/8 + \epsilon)$ approximation algorithm for constant $\epsilon > 0$) unless P=NP [7]. That MAX-E3-SAT can be solved (with high probability) in polynomial time by a quantum computer is therefore all the more remarkable.

The result shown this paper doesn't contradict with that shown [2] where it was shown that BQP does not contain NP relative to a random unitary oracle with probability one. This argument does not imply that BQP does not contain NP in a non-relativized world which is the novel feature in the proposed algorithm where partial measurement is used in the amplitude amplification process instead of the usual unitary amplitude amplification techniques that use iterative calls to an oracle to amplify the required solution .

The aim of the paper is to propose a quantum algorithm for the MAX-E3-SAT problem. The algorithm prepares a superposition of all possible variable assignments, then the algorithm evaluates the set of clauses using all the possible variable assignments simultaneously and then amplifies the amplitudes of the state(s) that achieve(s) the maximum satisfaction to the set of clauses using a novel amplitude amplification technique that applies an iterative partial negation and partial measurement. The proposed algorithm runs in $O(m^2)$ for an E3-CNF Boolean formula with m clauses and in the worst case runs in $O(n^6)$ for an E3-CNF Boolean formula with n Boolean variables to achieve an arbitrary high probability of success of $1-\epsilon$ for small $\epsilon > 0$ using a polynomial resources.

The paper is organized as follows; Section 2 shows the data structures and the quantum circuit for encoding an E3-CNF Boolean formula. Section 3 presents the proposed algorithm with analysis on time and space requirements. Section 4 concludes the paper.

2 Data Structures and Clause Encoding

Given an n inputs k-CNF Boolean formula,

$$f(x_0, x_1, \ldots, x_{n-1}) = c_0 \wedge c_1 \wedge \ldots \wedge c_{m-1}, \tag{1}$$

where m is the number of clauses. A k-CNF formula is a conjunction (AND) of m clauses, each clause c_j is represented by a disjunction (OR) of exactly $k \leq n$ literals, $c_j = \left(l_{j_0} \vee l_{j_1} \vee l_{j_2} \ldots \vee l_{j_{k-1}}\right)$, such that a literal $l_{j,a}$ in clause c_j with $0 \leq a \leq k-1$ and $0 \leq j \leq m-1$ equals to an input variable in its true form x_i or its complemented form $\neg x_i$, i.e. $l_{j,a} = \overset{\bullet}{x}_i$, where $\overset{\bullet}{x}_i$ can be replaced by x_i or $\neg x_i$ such that $\neg x_i$ is the negation of x_i with $0 \leq i \leq n-1$. The first aim is to decide whether f is satisfiable or not (deciding f). The second aim is to find a variable assignment for $x_0, x_1, \ldots$ and x_{n-1} that satisfies f if it is satisfiable (solve f) or to find a variable assignment that satisfies the maximum possible number of clauses if f is unsatisfiable (maximize f).

The problem of deciding whether a k-CNF Boolean formula is satisfiable or not is NP-complete and is known as k-SAT or Ek-SAT problem. The optimization problem associated with the k-SAT problem to find a variable assignment to satisfy a satisfiable k-CNF formula is NP-hard. If the k-CNF is unsatisfiable, then the problem of finding a variable assignment to maximize the number of satisfied clauses is known as MAX-Ek-SAT problem which is NP-hard problem [11]. The maximum number of clauses for a k-CNF Boolean formula is $2^k \begin{pmatrix} n \\ k \end{pmatrix} = O(n^k)$. Without loss of generality, this paper targets the E3-SAT and MAX-E3-SAT problems where $k=3$, so the maximum number of clauses m for the MAX-3E-SAT is $\frac{4}{3}n(n-1)(n-2)$.

2.1 Encoding of a Solution

A candidate solution S to the MAX-E3-SAT problem is a vector of variable assignment $A = (x_0, x_1, \ldots, x_{n-1}) \in \{0,1\}^n$, and each vector A is associated with a vector of the truth values $C(A) = (c_0, c_1, \ldots, c_{m-1}) \in \{0,1\}^m$ of the m clauses sorted in order, i.e. $S = (A, C(A)) \in \{0,1\}^{n+m}$. The optimal solution $S_{max} = (A_{max}, C(A_{max}))$ is the solution that contains a vector of variable assignment A_{max} that maximizes the number of 1's in the vector of the truth values $C(A_{max})$ of the m clauses. For short, the number of 1's in the vector of clauses C, i.e. the number of satisfied clauses, will be referred to as the 1-density of C so that the 1-density for a satisfiable formula must be equal to m. For example, consider the E3-CNF Formula with $n=3$ and $m=4$,

$$f(x_0, x_1, x_2) = c_0 \wedge c_1 \wedge c_2 \wedge c_3, \tag{2}$$

where,

$$\begin{aligned}c_0 &= (\neg x_0 \vee \neg x_1 \vee \neg x_2),\\ c_1 &= (\neg x_0 \vee x_1 \vee x_2),\\ c_2 &= (x_0 \vee \neg x_1 \vee x_2),\\ c_3 &= (x_0 \vee x_1 \vee x_2),\end{aligned} \tag{3}$$

then a solution to this formula will be encoded as $S = (A, C(A))$, where $A = (x_0, x_1, x_2)$ and $C(A) = (c_0, c_1, c_2, c_3)$. This formula is satisfiable when $(x_0, x_1, x_2) = (0,0,1), (0,1,1), (1,0,1)$, or $(1,1,0)$, and an instance of an optimal solution will be $S_{max} = ((0,0,1),(1,1,1,1))$ with $A_{max} = (0,0,1)$ and $C(A_{max}) = (1,1,1,1)$. For $n \geq 3$ and $m > 7$, the E3-CNF formula might not be satisfied [5] where the 1-density of the $C(A_{max})$ vector will give the maximum number of satisfied clauses and the order of 1's will show the satisfied clauses using the variable assignment A_{max}.

For $n \geq 3$ and $m = \frac{4}{3}n(n-1)(n-2)$, the 1-density of $C(A)$ will be 7/8m which represents the worst case for the 1-density of C where the 3-CNF formula will be unsatisfied for an arbitrary variable assignment A [14].

2.2 Encoding of a Clause

An E3-CNF formula with n inputs and m clauses will be encoded as an $n+m$ inputs/outputs quantum circuit. Every E3-CNF clause $c = (l_0 \vee l_1 \vee l_2)$ will be encoded using a 4×4 quantum gate. The GT^4 (4×4 Generalized Toffli) gate [9] is the main primitive gate that will be used to encode a clause. The GT^4 gate is defined as follows:

Definition 2.1 *(GT^4 gate)*

GT^4 gate is a reversible gate denoted as,

$$(y_0, y_1, y_2; f_{out}) = GT^4(x_0 \oplus \delta_0, x_1 \oplus \delta_1, x_2 \oplus \delta_2; f_{in}), \tag{4}$$

where x_a, δ_a, f_{in} and $f_{out} \in \{0,1\}$ with $a \in \{0,1,2\}$. The GT^4 gate has 4 inputs: x_0, x_1, x_2 (known as control qubits) and f_{in} (known as target qubit). Each control qubit x_a is associated with a condition δ_a, such that if $\delta_a = 1$ then the condition on x_a is satisfied if $x_a = 0$, i.e. $x_a \oplus 1 = \neg x_a$, and if $\delta_a = 0$ then the condition on x_a is satisfied if $x_a = 1$. The GT^4 gate has 4 outputs: y_0, y_1, y_2 and f_{out}. The operation of the GT^4 gate is defined as follows,

$$\begin{aligned}&y_a = x_a, \text{for } a = \{0,1,2\},\\ &f_{out} = f_{in} \oplus ((x_0 \oplus \delta_0) \wedge (x_1 \oplus \delta_1) \wedge (x_2 \oplus \delta_2)),\end{aligned} \tag{5}$$

where $\oplus$ is the XOR logic operation, i.e. the target qubit f_{in} will be flipped if and only if each control qubits x_a satisfies its associated condition δ_a. For example, $f_{out} = \neg f_{in}$ for the gate $GT^4(x_0 \oplus 1, x_1, x_2 \oplus 1; f_{in})$ if and only if $x_0 = 0$, $x_1 = 1$ and $x_2 = 0$.

A GT^4 gate with its target qubit, f_{in}, initialized to state $|1\rangle$ can be used to encode a clause $c = (l_0 \vee l_1 \vee l_2)$ using the Boolean algebraic identity,

$$c = (l_0 \vee l_1 \vee l_2) = ((l_0 \oplus 1) \wedge (l_1 \oplus 1) \wedge (l_2 \oplus 1)) \oplus 1, \tag{6}$$

so that $f_{out} = c$, where $l_a = \dot{x}$, and $\dot{x}$ can be replaced by x or $\neg x$ such that $\neg x = x \oplus 1$ is the negation of x, so that,

$$(x_0, x_1, x_2; c) = GT^4(l_0 \oplus 1, l_1 \oplus 1, l_1 \oplus 1; 1). \tag{7}$$

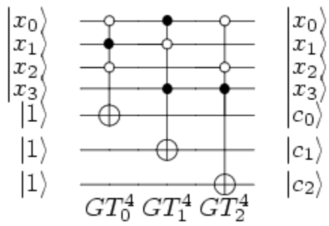

Figure 1: A quantum circuit using GT^4 gates for the E3-CNF formula shown in Eqn. 8 and 9, where $\bullet$ on a control qubit means that the associated condition is 1 while $\circ$ on a control qubit means that the associated condition is 0.

For example, consider the following E3-CNF Boolean formula with $n = 4$ and $m = 3$,

$$f(x_0, x_1, x_2, x_3) = c_0 \wedge c_1 \wedge c_2, \tag{8}$$

with

$$c_0 = (x_0 \vee \neg x_1 \vee \neg x_2), c_1 = (\neg x_0 \vee x_1 \vee \neg x_3), c_2 = (x_0 \vee x_2 \vee \neg x_3). \tag{9}$$

Apply the Boolean algebraic identity shown in Eqn.6 on each clause, then,

$$\begin{array}{l} c_0 = (x_0 \vee \neg x_1 \vee \neg x_2) = \neg(\neg x_0 \wedge x_1 \wedge x_2) = (\neg x_0 \wedge x_1 \wedge x_2) \oplus 1, \\ c_1 = (\neg x_0 \vee x_1 \vee \neg x_3) = \neg(x_0 \wedge \neg x_1 \wedge x_3) = (x_0 \wedge \neg x_1 \wedge x_3) \oplus 1, \\ c_2 = (x_0 \vee x_2 \vee \neg x_3) = \neg(\neg x_0 \wedge \neg x_2 \wedge x_3) = (\neg x_0 \wedge \neg x_2 \wedge x_3) \oplus 1, \end{array} \tag{10}$$

then each clause can be encoded using a GT^4 gate as follows,

$$\begin{array}{l} (x_0, x_1, x_2; c_0) \equiv GT_0^4 (x_0 \oplus 1, x_1, x_2, 1), \\ (x_0, x_1, x_3; c_1) \equiv GT_1^4 (x_0, x_1 \oplus 1, x_3, 1), \\ (x_0, x_2, x_3; c_2) \equiv GT_2^4 (x_0 \oplus 1, x_2 \oplus 1, x_3, 1). \end{array} \tag{11}$$

To construct a quantum circuit for this E3-CNF formula, prepare a quantum register with 4 qubits to be loaded with the values of x_0, x_1, x_2 and x_3, and add 3 extra qubits initialized with the quantum state $|1\rangle$ so that GT_0^4 uses the first extra qubit as the target qubit, GT_1^4 uses the second extra qubit as the target qubit, and so on, as shown in Fig. 1. Let U be a quantum circuit on 7 qubits defined as $U = GT_0^4 GT_1^4 GT_2^4$, then,

$$(x_0, x_1, x_2, x_3; c_0, c_1, c_2) = U(x_0, x_1, x_2, x_3; 1, 1, 1). \tag{12}$$

3 The Algorithm

Given an E3-CNF formula f with n inputs and m clauses. The proposed algorithm is divided into three stages, the first stage prepares a superposition of all possible variable assignments for the n variables. The second stage evaluates the m clauses for every variable assignment and stores the truth values of the

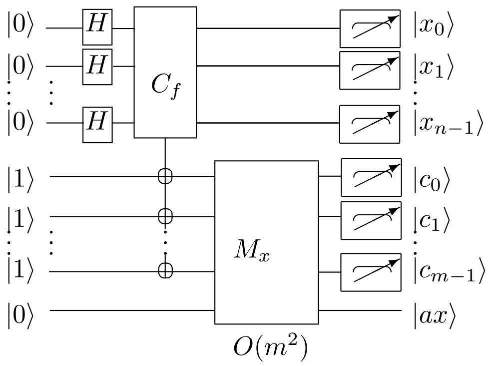

Figure 2: A quantum circuit for the proposed algorithm.

clauses in truth vectors entangled with the corresponding variable assignments in the superposition. The third stage amplifies the truth vector of clauses with maximum number of satisfied clauses using a partial negation and iterative measurement technique. The proposed algorithm uses $(n+m+1)$ qubits during the three stages. Each of the first n qubits is initialized to state $|0\rangle$, each of the m qubits is initialized to state $|1\rangle$, and an extra auxiliary qubit, denoted $|ax\rangle$ is initialized to state $|0\rangle$ and will be used during the partial negation and iterative measurement as will be shown later. The system is initially as follows,

$$|\psi_0\rangle = |0\rangle^n \otimes |1\rangle^m \otimes |0\rangle . \tag{13}$$

1- Variable Assignments Preparation. To prepare a superposition of all variable assignments of n qubits, apply $H^{\otimes n} \otimes I^{\otimes m+1}$ on the $n+m+1$ qubits

$$\begin{aligned} |\psi_1\rangle &= \left(H^{\otimes n} \otimes I^{\otimes m+1}\right) |\psi_0\rangle \\ &= \tfrac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |A_k\rangle \otimes |1\rangle^m \otimes |0\rangle , \end{aligned} \tag{14}$$

where H is the Hadamard gate, I is the identity matrix of size 2×2, $N = 2^n$, and $A_k = \left(x_0^k, x_1^k, \ldots, x_{n-1}^k\right) \in \{0,1\}^n$ is the bit-wise representation of an integer k, for $0 \le k \le N-1$, that represents a variable assignment out of the N possible variable assignments.

2- Preparation of the Truth Vectors of Clauses. For every E3-CNF clause $c_j = (l_0 \vee l_1 \vee l_2)$, apply a GT^4 gate taking qubit j in the m qubits register as the target qubit as shown in Section 2.2. The collection of all GT^4 gates applied to evaluate the m clauses is denoted C_f in Figure 2, then the system is transformed to,

$$\begin{aligned} |\psi_2\rangle &= (C_f \otimes I) |\psi_1\rangle \\ &= \tfrac{1}{\sqrt{N}} \sum_{k=0}^{N-1} (|A_k\rangle \otimes |C_k\rangle) \otimes |0\rangle , \end{aligned} \tag{15}$$

where $C_k = \left(c_0^k, c_1^k, \ldots, c_{m-1}^k\right) \in \{0,1\}^m$ is the truth vector for the m clauses associated with variable assignment A_k.

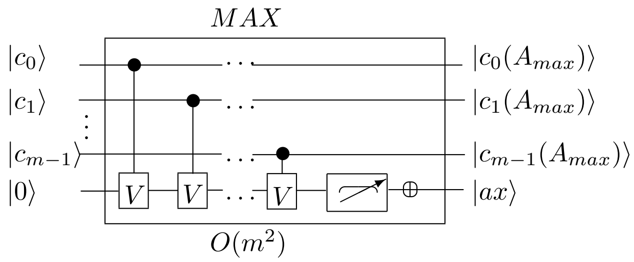

Figure 3: Quantum circuits for the MAX operator followed by a partial measurement then a negation to reset the auxiliary qubit $|ax\rangle$.

3- Maximization of the Number of Satisfied Clauses. The aim of this stage is to find the state $|C_k\rangle$ that contains the maximum number of $|1\rangle$'s, such state will be denoted $|C_{max}\rangle$, where a modified version of the amplitude amplification algorithm shown in [13] will be used for this purpose. Every $|C_k\rangle$ is entangled with the corresponding variable assignment $|A_k\rangle$. The variable assignment $|A_k\rangle$ will not be involved directly in this stage, then for simplicity the system can be re-written as,

$$|\psi_3\rangle = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} |C_k\rangle \otimes |0\rangle . \tag{16}$$

Let $d_k = \langle C_k \rangle$ be the 1-density of the state $|C_k\rangle$ such that $d_k \neq 0$. A solution to the MAX-E3-SAT problem is to find the state $|C_{max}\rangle$ with $d_{max} = max\{d_k,\, 0 \leq k \leq N-1\}$.

The aim is to find $|C_{max}\rangle$ when $|\psi_3\rangle$ is measured. To find $|C_{max}\rangle$, the algorithm applies partial negation on the state of $|ax\rangle$ entangled with $|C_k\rangle$ based on the 1-density of $|C_k\rangle$, i.e. more 1's in $|C_k\rangle$ gives more negation to the state of $|ax\rangle$ entangled with $|C_k\rangle$. If the number of 1's in $|C_k\rangle$ is m, then the entangled state of $|ax\rangle$ will be fully negated.

Let X be the Pauli-X gate which is the quantum equivalent to the NOT gate. It can be seen as a rotation of the Bloch Sphere around the X-axis by π radians as follows,

$$X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} . \tag{17}$$

The m^{th} partial negation operator V is the m^{th} root of the X gate and can be calculated using diagonalization as follows,

$$V = \sqrt[m]{X} = \frac{1}{2} \begin{bmatrix} 1+t & 1-t \\ 1-t & 1+t \end{bmatrix} , \tag{18}$$

where $t = \sqrt[m]{-1}$, and applying V for d times on a qubit is equivalent to the operator,

$$V^d = \frac{1}{2}\begin{bmatrix} 1+t^d & 1-t^d \\ 1-t^d & 1+t^d \end{bmatrix}, \tag{19}$$

such that if $d = m$, then $V^m = X$. To amplify the amplitude of the state $|C_{max}\rangle$, apply the operator M_x on $|\psi_3\rangle$ as will be shown later, where M_x is an operator on $m+1$ qubits register that applies V conditionally for m times on $|ax\rangle$ based on 1-density of $|c_0 c_1 \ldots c_{m-1}\rangle$ as follows (as shown in Figure 3),

$$M_x = Cont\_V(c_0; ax) Cont\_V(c_1; ax) \ldots Cont\_V(c_{m-1}; ax), \tag{20}$$

where the $Cont\_V(c_j; ax)$ gate is a 2-qubit controlled gate with control qubit $|c_j\rangle$ and target qubit $|ax\rangle$. The $Cont\_V(c_j; ax)$ gate applies V conditionally on $|ax\rangle$ if $|c_j\rangle = |1\rangle$, so, if d is the 1-density of $|c_0 c_1 \ldots c_{m-1}\rangle$ then,

$$M_x\left(|c_0c_1...c_{m-1}\rangle \otimes |0\rangle\right) = |c_0c_1...c_{m-1}\rangle \otimes \left(\frac{1+t^d}{2}|0\rangle + \frac{1-t^d}{2}|1\rangle\right), \tag{21}$$

and the probabilities of finding the auxiliary qubit $|ax\rangle$ in state $|0\rangle$ or $|1\rangle$ when measured is respectively as follows,

$$\begin{array}{l} Pr(ax=0) = \left|\frac{1+t^d}{2}\right|^2 = \cos^2\left(\frac{d\pi}{2m}\right), \\ Pr(ax=1) = \left|\frac{1-t^d}{2}\right|^2 = \sin^2\left(\frac{d\pi}{2m}\right). \end{array} \tag{22}$$

To find the state $|C_{max}\rangle$ in $|\psi_3\rangle$, the proposed algorithm is as shown in Algorithm 1 and as shown in Figure 3). For simplicity and without loss of generality, assume that a single $|C_{max}\rangle$ exists in $|\psi_3\rangle$, although different variable assignments might be associated with truth vectors with maximum 1-density with different order of 1's, but such information is not known in advance.

Assuming that Algorithm 1 finds $|ax\rangle = |1\rangle$ for r times in a row, then the probability of finding $|ax\rangle = |1\rangle$ after Line:4 in the 1^{st} iteration, i.e. $r = 1$ is given by,

$$Pr^{(1)}(ax=1) = \frac{1}{N}\sum_{k=0}^{N-1} \sin^2\left(\frac{d_k\pi}{2m}\right). \tag{23}$$

The probability of finding $|\psi_r\rangle = |C_{max}\rangle$ after Line:4 in the 1^{st} iteration, i.e. $r = 1$ is given by,

$$Pr^{(1)}(\psi_r = C_{max}) = \frac{1}{N}\sin^2\left(\frac{d_{max}\pi}{2m}\right). \tag{24}$$

Algorithm 1 Amplify $|C_{max}\rangle$ in $|\psi_3\rangle$

1: Let $|\psi_r\rangle = |\psi_3\rangle$
2: **for** *counter* $= 1 \rightarrow r$ **do**
3: Apply the operator M_x on $|\psi_r\rangle$.
4: Measure $|ax\rangle$
5: **if** $|ax\rangle = |1\rangle$ **then**
6: Let $|\psi_r\rangle$ be the system post-measurement of $|ax\rangle$
7: Apply X gate on $|ax\rangle$ {to reset $|ax\rangle$ to $|0\rangle$ for the next iteration}
8: **else**
9: Let $|\psi_r\rangle = |\psi_3\rangle$ and restart the for-loop
10: **end if**
11: **end for**
12: Measure the first m qubits in $|\psi_r\rangle$ to read $|C_{max}\rangle$.
13: **if** $|C_{max}\rangle = |1\rangle^{\otimes m}$ **then**
14: The E3-CNF formula is satisfiable
15: **else**
16: The E3-CNF formula is not satisfiable where number of $|1\rangle$'s in $|C_{max}\rangle$ represents the maximum number of satisfied clauses in order
17: **end if**
18: Measure the first n qubits in $|\psi_2\rangle$ to read the corresponding variable assignment $|A_{max}\rangle$

The probability of finding $|ax\rangle = |1\rangle$ after Line:4 in the r^{th} iteration, i.e. $r > 1$ is given by,

$$Pr^{(r)}(ax = 1) = \frac{\sum_{k=0}^{N-1} \sin^{2r}\left(\frac{d_k \pi}{2m}\right)}{\sum_{k=0}^{N-1} \sin^{2(r-1)}\left(\frac{d_k \pi}{2m}\right)}. \tag{25}$$

The probability of finding $|\psi_r\rangle = |C_{max}\rangle$ after Line:4 in the r^{th} iteration, i.e. $r > 1$ is given by,

$$Pr^{(r)}(\psi_r = C_{max}) = \frac{\sin^{2r}\left(\frac{d_{max} \pi}{2m}\right)}{\sum_{k=0}^{N-1} \sin^{2(r-1)}\left(\frac{d_k \pi}{2m}\right)}. \tag{26}$$

To get the highest probability of success for $Pr(\psi_r = C_{max})$, the for-loop should be repeated until $\left|Pr^{(r)}(ax = 1) - Pr^{(r)}(\psi_r = C_{max})\right| \leq \epsilon$ for small $\epsilon \geq 0$ as shown in Figure 4. This happens when,

$$\sum_{k=0, k \neq max}^{N-1} \sin^{2r}\left(\frac{d_k \pi}{2m}\right) \leq \epsilon, \tag{27}$$

and since the Sine function is a decreasing function then for sufficient large r,

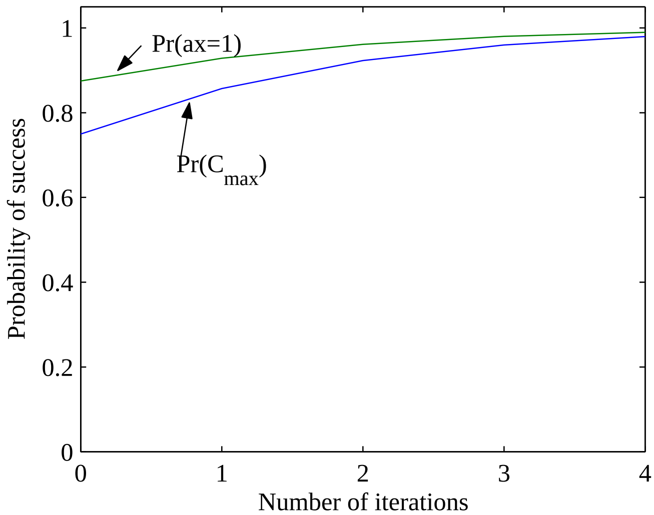

Figure 4: The probability of success for an E3-CNF formula: $(x_0 \vee x_1 \vee x_2) \wedge (\neg x_0 \vee \neg x_1 \vee \neg x_2)$ with $n = 3$ and $m = 2$ with $d_{max} = 2$, i.e. the maximum number of satisfied clauses is 2, where $Pr^{(1)}(\psi_r = C_{max}) = 0.75$, $Pr^{(1)}(ax = 1) = 0.875$, $Pr^{(r)}(\psi_r = C_{max}) = 0.98$, and $Pr^{(r)}(ax = 1) = 0.99$.

$$\sum_{k=0, k\neq max}^{N-1} \sin^{2r}\left(\frac{d_k \pi}{2m}\right) \approx \sin^{2r}\left(\frac{d_{nm}\pi}{2m}\right), \tag{28}$$

where d_{nm} is the next maximum 1-density less than d_{max}. The values of d_{max} and d_{nm} are unknown in advance, so let $d_{max} = m$ be the number of satisfied clauses, then in the worst case when $d_{max} = m$, $d_{nm} = m - 1$ and $m = \frac{4}{3}n(n-1)(n-2)$, the required number of iterations r for $\epsilon = 10^{-\lambda}$ and $\lambda > 0$ can be calculated using the formula,

$$0 < \sin^{2r}\left(\frac{(m-1)\pi}{2m}\right) \leq \epsilon, \tag{29}$$

then,

$$\begin{aligned} r &\geq \frac{\log(\epsilon)}{2\log\left(\sin\left(\frac{(m-1)\pi}{2m}\right)\right)} \\ &= \frac{\log\left(10^{-\lambda}\right)}{2\log\left(\cos\left(\frac{\pi}{2m}\right)\right)} \\ &\geq \lambda\left(\frac{2m}{\pi}\right)^2 = O\left(m^2\right), \end{aligned} \tag{30}$$

where $0 < m \leq \frac{4}{3}n(n-1)(n-2)$. When $m = \frac{4}{3}n(n-1)(n-2)$, then the upper bound for the required number of iterations r is $O\left(n^6\right)$. Assuming that a single $|C_{max}\rangle$ exists in the superposition will increase the required number of iterations, so it is important to notice here that the probability of success will not be over-cooked by increasing the required number of iteration r similar to the common amplitude amplification techniques.

3.1 Tuning the Probability of Success

During the above analysis, two problems might arise during the implementation of the proposed algorithm. The first one is to finding $|ax\rangle = |1\rangle$ for r times in a row which a critical issue in the success of the proposed algorithm to terminate in polynomial time. The second problem is that the value of d_{max} is not known in advance, where the value of $Pr^{(1)}(ax = 1)$ shown in Eqn. 23 plays an important role in the success of finding $|ax\rangle = |1\rangle$ in the next iterations, this value depends heavily on the 1-density of $|C_{max}\rangle$, i.e. the ratio $\frac{d_{max}}{m}$.

Consider the case of a complete E3-CNF formula where the number of clauses is $m = \frac{4}{3}n(n-1)(n-2)$ and all the $|C_k\rangle$'s are equivalent where anyone can be taken as $|C_{max}\rangle$. In this case, each clause c_j will be satisfied by 7 variable assignments out of 8 possible variable assignment, then $d_{max} = \frac{7}{8}m$ for any $|C_k\rangle$ [10], so $Pr^{(1)}(ax = 1)$ is as follows,

$$\begin{aligned} Pr^{(1)}(ax = 1) &= \sin^2\left(\tfrac{d_{max}\pi}{2m}\right) \\ &= \sin^2\left(\tfrac{7\pi}{16}\right) \\ &= 0.9619. \end{aligned} \tag{31}$$

This case is a trivial case for the proposed algorithm by setting $m = d_{max}$ in m^{th} root of X to get a probability of success of certainty after a single iteration. Assuming a blind approach where d_{max} is not known, then this case represents the worst case [14] and iterating the proposed algorithm will not amplify the amplitudes after arbitrary number of iterations. For an arbitrary E3-CNF formula, the actual probability of success will depend of the 1-density of $|C_{max}\rangle$, i.e. the ratio $\frac{d_{max}}{m}$. In the following, a tuning of $Pr^{(1)}(ax = 1)$ will be shown so that we can find $|ax\rangle = |1\rangle$ after the first iteration with an arbitrary higher probability of success close to certainty without a priori knowledge of d_{max}.

For an arbitrary E3-CNF formula, we could interpret the formula for $Pr(ax = 1)$ in Eqn. (23) as the expected value of the function,

$$\phi(x) = sin^2(\frac{x\pi}{2}), \tag{32}$$

where x is the proportion of clauses satisfied by a random truth assignment, that is, $Pr(ax = 1) = E[\phi(x)]$. The bounds for the probability of finding $|ax\rangle = |1\rangle$ in the first iteration is as shown in the following Lemma,

Lemma 3.1 *The probability of finding $|ax\rangle = |1\rangle$ in the first iteration is bounded as follows,*

$$0.691 < 1 - \frac{\pi^2}{32} \leq Pr^{(1)}(ax = 1) \leq \sin(\frac{7\pi}{16}) < 0.981. \tag{33}$$

Proof

$$Pr^{(1)}(ax = 1) = E\left[\sin^2\left(\frac{d_k\pi}{2m}\right)\right] \leq E\left[\sin\left(\frac{d_k\pi}{2m}\right)\right]. \tag{34}$$

Since $\sin(x)$ is a concave function on $0 \leq x \leq \pi/2$, it follows from Jensen's inequality that,

$$Pr^{(1)}(ax = 1) \leq \sin\left(\frac{E[d_k]\pi}{2m}\right) = \sin(\frac{7\pi}{16}) < 0.981. \tag{35}$$

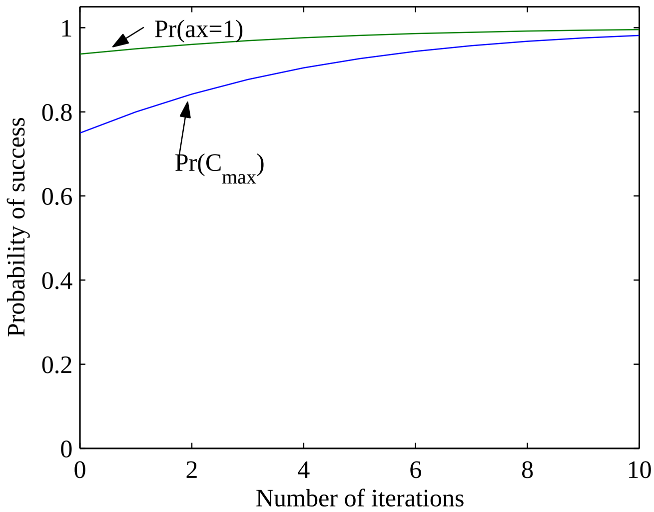

Figure 5: The probability of success for the E3-CNF formula shown in Figure 4 where $Pr^{(1)}(ax=1)$ is raised from 0.875 to 0.94 by adding a single temporary qubit initialized to state $|1\rangle$, i.e. $\mu_{max}=1$ where $Pr^{(1)}(\psi_r = C_{max}) = 0.75$, $Pr^{(1)}(ax=1) = 0.94$, $Pr^{(r)}(\psi_r = C_{max}) = 0.98$, and $Pr^{(r)}(ax=1) = 0.99$.

For the lower bound we use,

$$\sin^2((1-x)\pi/2) \geq 1 - \frac{\pi^2 x^2}{4}, \tag{36}$$

which follows from the Taylor series taken around $x = 0$. Then,

$$\begin{array}{rl} Pr^{(1)}(ax=1) & \geq \frac{1}{N}\sum_{k=0}^{N-1}\left(1-\frac{\pi^2(1-d_k/m)^2}{4}\right) \\ & = 1+\frac{3\pi^2}{16}-\frac{\pi^2}{4Nm^2}\sum_{k=0}^{N-1} d_k^2 \\ & \geq 1+\frac{3\pi^2}{16}-\frac{\pi^2}{4Nm^2}\sum_{k=0}^{N-1} d_k m \\ & = 1+\frac{3\pi^2}{16}-\frac{\pi^2}{4Nm}\sum_{k=0}^{N-1} d_k \\ & = 1+\frac{3\pi^2}{16}-\frac{7\pi^2}{32} \\ & = 1-\frac{\pi^2}{32} > 0.691. \end{array} \tag{37}$$

To overcome the problem of low probability of finding $|ax\rangle = |1\rangle$ in the first iteration, we can add μ_{max} temporary qubits initialized to state $|1\rangle$ to the register $|C_k\rangle$ as follows,

$$|c_0c_1...c_{m-1}\rangle \rightarrow |c_0c_1\ldots c_{m-1}c_mc_{m+1}\ldots c_{m+\mu_{max}-1}\rangle, \tag{38}$$

so that the extended number of clauses m_{ext} will be $m_{ext} = m + \mu_{max}$ and $V = \sqrt[m_{ext}]{X}$ will be used instead of $V = \sqrt[m]{X}$ in the M_x operator, then the density of 1's will be $\frac{\frac{7}{8}m+\mu_{\max}}{m+\mu_{\max}}$. To get a probability of success Pr_{max} to find $|ax\rangle = |1\rangle$ after the first iteration of the for-loop in Algorithm 1,

$$Pr^{(1)}(ax=1) = N\alpha^2 \sin^2\left(\frac{\pi\left(\frac{7}{8}m+\mu_{\max}\right)}{2\left(m+\mu_{\max}\right)}\right) \geq Pr_{\max}, \tag{39}$$

then the required number of temporary qubits μ_{max} is calculated as follows,

$$\mu_{\max} \geq m\left(\frac{\omega-\frac{7}{8}}{1-\omega}\right), \tag{40}$$

where $\omega = \frac{2}{\pi}\sin^{-1}\left(\sqrt{Pr_{\max}}\right)$ and $N\alpha^2 = 1$. For example, if $Pr_{\max} = 0.99$, then $Pr^{(1)}(ax_2 = 1)$ will be in the neighborhood of 99% as shown in Figure 5. To conclude, the problem of low 1-density of $|C_{max}\rangle$ can be solved with a polynomial increase in the number of qubits to get the solution $|C_{max}\rangle$ in $O\left(m_{ext}^2\right) = O\left(n^6\right)$ iterations with arbitrary high probability $Pr_{max} < 1$ to terminate in poly-time, i.e. to read $|ax\rangle = |1\rangle$ for r times in a row.

4 Conclusion

Given an E3-CNF Boolean formula with n inputs, the paper showed that BQP contains NP in a non-relativized world by proposing a BQP quantum algorithm to solve the MAX-E3-SAT problem with m clauses. The proposed algorithm encoded every clause as a GT^4 gate where $O(n+m)$ qubits are used. The algorithm is divided into three stages; the first stage prepares a superposition of all possible variable assignments. In the second stage, the algorithm evaluates the set of clauses for all possible variable assignments using a quantum circuit composed of GT^4 gates so that each variables assignment is entangled with a truth vector of clauses evaluated according to that variables assignment. In the third stage, the algorithm amplified the amplitudes of the truth vector of clauses that achieves the maximum satisfaction to the set of clauses using an amplitude amplification technique that applies an iterative partial negation where partial negation is applied to the state of an auxiliary qubit entangled with the truth vector of clauses based on the number of satisfied clauses, i.e. more satisfied clauses implies more negation to entangled state of the auxiliary qubit. A partial measurement on the auxiliary qubit is then used to amplify the set of clauses with more negation. The third stage requires $O(m^2)$ iterations and in the worst case requires $O(n^6)$ iterations. It was shown that the proposed algorithm achieves an arbitrary high probability of success of $1-\epsilon$ for small $\epsilon > 0$ using a polynomial increase in the resources by adding dummy clauses with predefined values to give more negation to the best truth vector of clauses.

In the same manner, the proposed algorithm can also be used decide if a given E3-CNF Boolean formula is satisfiable or not by checking the truth vector of clauses, if the m clauses are satisfied then the E3-CNF Boolean formula is satisfiable, if not, then the proposed algorithm gives the maximum number of satisfied clauses with the corresponding variable assignment.

The proposed algorithm can easily be extended in a trivial way to solve/decide an Ek-CNF Boolean formula by encoding any Ek-CNF clause as a GT^{k+1} gate where it can be shown that the algorithm will require $O(n^{2k})$ iterations.

References

[1] Adleman, L., DeMarrais, J. and Huang, M.-D., Quantum Computability. SIAM Journal on Computing, 26(5):1524–1540, 1997.

[2] Bennett, C. H., Bernstein, E., Brassard, G. and Vazirani, U., Strengths and Weaknesses of Quantum Computing. SIAM Journal on Computing, 26(5):1510-1523, 1997.

[3] Bernstein, E. and Vazirani, U., Quantum Complexity Theory SIAM Journal on Computing. 26(5): 1411-1473, 1997.

[4] Buhrman, H. and Van Dam, W., Quantum Bounded Query Complexity. In Proceedings of the 14^{th} Annual IEEE Conference on Computational Complexity. 149-156, 1999.

[5] Dimitriou, T., SAT Distributions with Planted Assignments and Phase Transitions Between Decision and Optimization Problems. Discrete Applied Mathematics 153, pp. 58–72, 2005.

[6] Fortnow, L. and Rogers, J., Complexity Limitations on Quantum Computation. Journal of Computer and System Sciences, 59(2):240–252, 1999.

[7] Håstad, J., Some Optimal Inapproximability Results. Journal of the ACM, 48(4):798–859, 2001.

[8] Moore, C. and Martens, S., The Nature of Computation. Oxford University Press. 2011

[9] Toffoli, T., Reversible Computing. In W. de Bakker and J. van Leeuwen, editors, *Automata, Languages and Programming*, page 632. Springer, New York, 1980. Technical Memo MIT/LCS/TM-151, MIT Lab for Computer Science (unpublished).

[10] Trevisan, L., Gadgets, Approximation, and Linear Programming. In Proceedings of the 37^{th} Annual Symposium on Foundations of Computer Science, pp. 617–626, 1996.

[11] Trevisan, L., Paradigms of Combinatorial Optimization: Problems and New Approaches. (Chapter 13: Inapproximability of Combinatorial Optimization Problems), vol. 2, p.381, Vangelis Th. Paschos (Editor), Wiley-ISTE, 2010.

[12] Williamson, D. P. and Shmoys, D. B., The Design of Approximation Algorithms. Cambridge University Press. 2011.

[13] Younes, A., A Bounded-error Quantum Polynomial Time Algorithm for Two Graph Bisection Problems. Quantum Information Processing. doi :10.1007/s11128-015-1069-y, 2015. arXiv:1505.06284 [quant-ph]

[14] Zhang, W., Phase Transitions and Backbones of 3-SAT and Maximum 3-SAT. T. Walsh (Ed.): CP 2001, LNCS 2239, pp. 153—167, 2001.