Journal of Machine Learning Research 13 (2012) 3177-3206 Submitted 1/12; Revised 10/12; Published 11/12

Quantum Set Intersection and its Application to Associative Memory

Tamer Salman TAMER@CS.TECHNION.AC.IL
Yoram Baram BARAM@CS.TECHNION.AC.IL
Department of Computer Science
Technion - Israel Institute of Technology
Haifa, 32000, Israel

Editor: Manfred Opper

Abstract

We describe a quantum algorithm for computing the intersection of two sets and its application to associative memory. The algorithm is based on a modification of Grover's quantum search algorithm (Grover, 1996). We present algorithms for pattern retrieval, pattern completion, and pattern correction. We show that the quantum associative memory can store an exponential number of memories and retrieve them in sub-exponential time. We prove that this model has advantages over known classical associative memories as well as previously proposed quantum models.

Keywords: associative memory, pattern completion, pattern correction, quantum computation, quantum search

1. Introduction

The introduction of Shor's algorithm for factoring numbers in polynomial time (Shor, 1994) has demonstrated the ability of quantum computation to solve certain problems more efficiently than classical computers. This perception was ratified two years later, when Grover (1996) introduced a sub-exponential algorithm for quantum searching a database.

The field of quantum computation is based on the combination of computation theory and quantum mechanics. Computation theory concerns the design of computational models and the study of their time and space complexities. Quantum mechanics, on the other hand, concerns the study of systems governed by the rules of quantum physics. The combination of the two fields addresses the nature of computation in the physical world. However, there is still no efficient reduction of quantum mechanical behavior to classical computation.

Quantum mechanics is a conceptual framework that mathematically describes physical systems. It is based on four postulates, known as the postulates of quantum mechanics (Nielsen and Chuang, 2000), which provide a connection between the physical world and mathematical formalism. Through these postulates, it is possible to better understand the nature of physical computation and what can be physically computed.

The first postulate states that a physical system is completely described by a state in a Hilbert space known as the state space. The second states that the evolution of a closed quantum system is described by a unitary transformation. The third states that a quantum measurement is described by a collection of measurement operators that satisfy the completeness equality, that is, the sum of all possible measurements adds up to 1. The forth states that the state space of a composite physical system is the tensor product of the state spaces of its components.

Next, we introduce the basic building blocks of quantum computation. The reader is referred to Nielsen and Chuang (2000) for a detailed introduction.

1.1 States and Qubits

The basic entity of classical computation is the classical bit. Each classical bit can have one of two values, 0 and 1. The state of any finite physical system that can be found in a finite number of states can be described by a string of bits. A string of n bits represents one of 2^n possible states of a system enumerated 0, ..., $2^n - 1$.

In quantum computation, the basic entity is called a qubit (quantum bit). The qubit can have the analogue values $|0\rangle$ and $|1\rangle$, known as the computational basis states, where $|\cdot\rangle$ is the Dirac notation. Yet, the qubit can also have any other value that is a linear combination of $|0\rangle$ and $|1\rangle$:

$$|\Psi\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$$

where α and β are any complex numbers (called the amplitudes of the basis states 0 and 1, respectively), such that $|\alpha|^2 + |\beta|^2 = 1$. Consequently, the qubit can be in any one of an infinite number of states described by unit vectors in a 2-dimensional complex vector space. The unary representation of a qubit can be given as a vector of two values

$$|\Psi\rangle = \begin{pmatrix}\alpha\\ \beta\end{pmatrix}.$$

Analogously to the classical bit string, qubit strings (or quregisters) describe the state of a system. A two qubit system comprising two qubits $|a\rangle = \alpha\,|0\rangle + \beta\,|1\rangle$ and $|b\rangle = \gamma|0\rangle + \delta\,|1\rangle$ is described by the tensor product of the two qubits $|a,b\rangle \equiv |a\rangle \otimes |b\rangle = \alpha\gamma|00\rangle + \alpha\delta\,|01\rangle + \beta\gamma|10\rangle + \beta\delta\,|11\rangle$.

1.2 Measurement

The measurement of a qubit reveals only one of two possible outcomes. The value of α and β cannot be extracted from the measurement of a qubit. Instead, when measuring the qubit $\alpha\,|0\rangle + \beta\,|1\rangle$ in the computational basis, the result can be either 0 or 1 with probabilities $|\alpha|^2$ and $|\beta|^2$ respectively. For example, the state $\left(\frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle\right)$, when measured, yields any of the two results 0 or 1 with probability $1/2$. The measurement operation is not reversible and, once made, the qubit no longer exists in its state before the measurement. Measurements can be performed in different bases. For example, measuring the qubit $\alpha\,|0\rangle + \beta\,|1\rangle$ in the Hadamard basis defined by the two basis states $|+\rangle \equiv \left(\frac{1}{\sqrt{2}}\,|0\rangle + \frac{1}{\sqrt{2}}\,|1\rangle\right)$ and $|-\rangle \equiv \left(\frac{1}{\sqrt{2}}\,|0\rangle - \frac{1}{\sqrt{2}}\,|1\rangle\right)$ gives $|+\rangle$ with probability $\frac{(\alpha+\beta)^2}{2}$ and $|-\rangle$ with probability $\frac{(\alpha-\beta)^2}{2}$, since $\alpha\,|0\rangle + \beta\,|1\rangle = \frac{\alpha+\beta}{\sqrt{2}}\,|+\rangle + \frac{\alpha-\beta}{\sqrt{2}}\,|-\rangle$.

An n-qubit system can be either measured completely or partially. When measured partially, the unmeasured subsystem can retain quantum superposition and further quantum manipulations can be performed upon it. However, any measurement can be delayed to the end of the computation process.

1.3 Operators

In quantum computation, a system changes its state under a unitary quantum operator U from $|\Psi\rangle$ to $U\,|\Psi\rangle$. An operator U can be described as a $2^n \times 2^n$ matrix operating on the unary representation

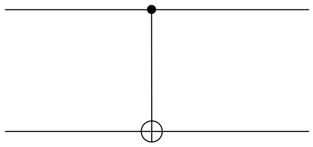

Figure 1: The 2-qubit Controlled-Not (CNOT) quantum gate.

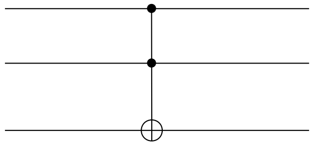

Figure 2: The 3-qubit Toffoli quantum gate.

of the system state. A unitary operator satisfies $UU^\dagger = I$, where $U^\dagger$ is the conjugate transpose of U (transpose the matrix U then substitute the conjugate complex of each element in the matrix).

Quantum operators can be implemented using quantum gates, which are the analogue of the classical gates that compose classical electrical circuits. In this analogy, the wires of a circuit carry the information on the system's state, while the quantum gates manipulate their contents to different states. For example, the Hadamard operator

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}$$

transforms a qubit in the state $|0\rangle$ into the state $|+\rangle$ and the state $|1\rangle$ into the state $|-\rangle$.

Operators can also be quantum gates operating on multiple qubits. An n-qubit quantum operator has n inputs and n outputs. For example, the 2-qubit controlled-not (CNOT) gate depicted in Figure 1, flips the target (second) qubit if the control qubit (first) has value $|1\rangle$ and leaves it unchanged if the control qubit has the value $|0\rangle$. Specifically, the CNOT gate performs the following transformations on the four computational basis states: $|00\rangle \rightarrow |00\rangle$, $|01\rangle \rightarrow |01\rangle$, $|10\rangle \rightarrow |11\rangle$, and $|11\rangle \rightarrow |10\rangle$. It can be described as a unitary matrix operating on the unary representation of the state as follows:

$$CNOT = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Another gate is the 3-qubit controlled-controlled-not gate also known as the Toffoli gate depicted in Figure 2, which flips the target (third) qubit if the two control bits (first and second) both have the values $|1\rangle$ and leaves it unchanged otherwise.

The Hadamard operator can also be seen as operating on n-qubits by the tensor product of n single qubit Hadamard operators. Each qubit is then transformed according to the single qubit Hadamard transform, that is,

$$H^{\otimes n} |x_{n-1}, ..., x_1, x_0\rangle = H |x_{n_1}\rangle \otimes ... \otimes H |x_1\rangle \otimes H |x_0\rangle .$$

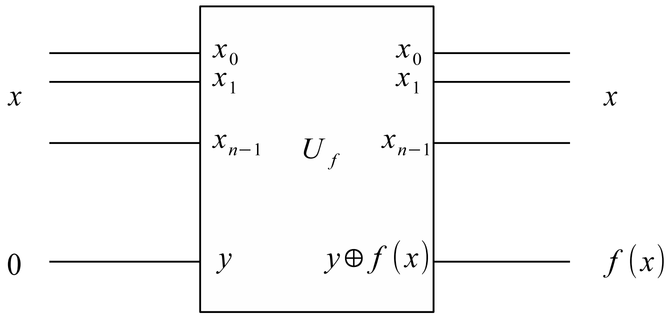

Figure 3: A classical reversible Oracle.

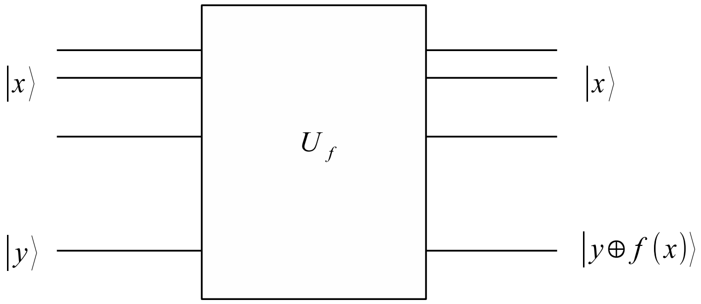

Figure 4: A quantum Oracle.

1.4 Quantum Parallelism and Interference

Consider an oracle of an n-dimensional function f implemented by a classical circuit, which, for every input x, produces $f(x)$ at the output. The oracle can be made reversible by adding an additional bit to the input, initialized by 0, and letting the output be $x, 0 \oplus f(x)$ as depicted in Figure 3, where $\oplus$ is the addition modulo 2. Initializing the additional bit with 1 produced $\overline{f(x)}$ at the output.

A quantum oracle is a reversible oracle that accepts a superposition of inputs and produces a superposition of outputs as depicted in Figure 4. When the additional qubit is initialized by $|0\rangle$, the oracle performs the following transformation: $|x\rangle|0\rangle \to |x\rangle|0 \oplus f(x)\rangle$. When the additional qubit is initialized by $|-\rangle$ the oracle is called a quantum phase oracle that gives $f(x)$ in the phase of the state $|x\rangle$ as follows: $|x\rangle|-\rangle \to (-1)^{f(x)}|x\rangle|-\rangle$.

Suppose that we have constructed a quantum circuit U_f that implements a function $f : \{0,1\}^n \to \{0,1\}$, such that when introduced with an input $|x\rangle|y\rangle$, the output of the circuit would be $|x\rangle|y \oplus f(x)\rangle$. Quantum parallelism is the ability of the quantum circuit to process many inputs simultaneously and receive all the outcomes at the output. Consider the case where $|y\rangle = |0\rangle$, and the $|x\rangle = H^{\otimes n}|0\rangle^{\otimes n}$. Applying the n-qubit Hadamard operator to the $|0\rangle$ state yields a superposition of all basis states

$\frac{1}{\sqrt{2^n}} \sum i = 0^{2^n} |i\rangle$. The superposition will be maintained through the quantum circuit and the resulting output would be $\frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n} |i\rangle |f(i)\rangle$. This can be computed as follows:

$$U_f^{\otimes n+1} \left(\left(H^{\otimes n} \otimes I \right) |0\rangle^{\otimes n} |0\rangle \right) = U_f^{\otimes n+1} \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n} |i\rangle |0\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n} |i\rangle |f(i)\rangle .$$

Although this computes f for all values of i simultaneously there is no immediate way of seeing them all together, because once the output state is measured, only one value of $f(i)$ is revealed and the rest vanish. However, further quantum computation allows the different values to interfere together and reveal some information concerning the function f.

It has been shown (Deutsch, 1985; Deutsch and Jozsa, 1992; Simon, 1994) that the revealed information can be considerably more than what classical computation would achieve after one query of the circuit that implements the function f. Deutsch and Jozsa (1992) showed that if a binary n-dimensional function is guaranteed to be either constant or balanced, then determination can be done using only one query to a quantum oracle implementing it, while a classical solution would require an exponential number of queries. Another example of the advantage of quantum algorithms over classical ones was presented by Simon (1994), in which a function is known to have the property that there exists some $s \in \{0,1\}^n$ for which for all $x, y \in \{0,1\}^n$ it hold that $f(x) = f(y)$ if and only if $x = y$ or $x \oplus y = s$. Simon (1994) proved that using quantum computations, s can be found exponentially faster than with any classical algorithm, including probabilistic algorithms.

1.5 Solving Problems using Quantum Computation

According to the above definitions of a system state, measurement, and operators, a quantum computer drives the dynamics of the quantum system through operators that change its state and measures the final state to reveal classical information. In the general case, one might think of the process of quantum computation as a multi-phase procedure, which performs some classical computation on the data at hand, creates a quantum state describing the system, drives it through quantum operators, which might depend on the data, to the target state, measures the outcome, and performs some more classical computation to receive a result.

Consequently, we can describe a schematic process for solving problems using quantum computation as follows:

A general solution scheme using quantum computations

Given:

Classical input data

1. Preliminary classical computation of the input
2. Create initial quantum state
3. Apply quantum circuit to the initial quantum state
4. Measure the resulting state
5. Apply classical computation to the measured state

1.6 Grover's Quantum Search Algorithm

Given a database of $N \equiv 2^n$ unsorted elements of n bits each, any classical search would require $O(N)$ queries to find a desired element. In 1996, Grover presented a quantum computational algorithm that searches an unsorted database with $O(\sqrt{N})$ operations (Grover, 1996; Boyer et al., 1996).

The algorithm performs a series of $O(\sqrt{N})$ unitary operations on the superposition of all basis states that amplify the solution states causing the probability of measuring one of the solutions at the end of the computation to be close to 1.

Suppose that the search problem has a set X of r solutions and that we own an oracle function f_X that identifies the solution $x \in X$ according to the following:

$$f_X(x) = \begin{cases} 1, & x \in X \\ 0, & x \notin X \end{cases}.$$

Any classical algorithm that attempts to find the solution clearly needs to query the oracle N times in the worst case. Grover's algorithm shows that we can find the solution with the help of the oracle by querying it only $O(\sqrt{N})$ times.

The quantum phase oracle of the function f_X flips (rotates by π) the amplitude of the states of X, while leaving all other states unchanged. This is done by the operator $I_X = I - 2\sum_{x\in X} |x\rangle \langle x|$. In matrix formulation, I_X is similar to the identity matrix I except it has -1 on the xth elements of the diagonal.

Grover's algorithm starts with the superposition of all basis states created by applying the Hadamard operator on the zero state, $H^{\otimes n} |0\rangle^{\otimes n}$ (shortened by $H|0\rangle$), and goes about performing multiple iterations, in which each iteration consists of applying the phase oracle followed by the operator HI_0H, where I_0 flips the phase of the state $|0\rangle^{\otimes n}$. Grover's iterator is thus defined as

$$Q = -HI_0HI_X \tag{1}$$

where the sign "$-$" stands for the global phase flip that has no physical meaning and is performed only for analytical convenience.

The operator in Equation 1 can be viewed in the space defined by the two basis states

$$|l_1\rangle = \frac{1}{\sqrt{r}} \sum_{i\in X} |i\rangle$$

and

$$|l_2\rangle = \frac{1}{\sqrt{N-r}} \sum_{i\notin X} |i\rangle$$

as the rotation

$$\begin{pmatrix} 1 - \frac{2r}{N} & 2\frac{\sqrt{r(N-r)}}{N} \\ -2\frac{\sqrt{r(N-r)}}{N} & 1 - \frac{2r}{N} \end{pmatrix}$$

which is depicted in Figure 5, where the rotation angle is

$$w = \arccos\left(1 - \frac{2r}{N-r}\right).$$

The initial state has an angle

$$\phi = \arctan\left(\sqrt{\frac{r}{N-r}}\right)$$

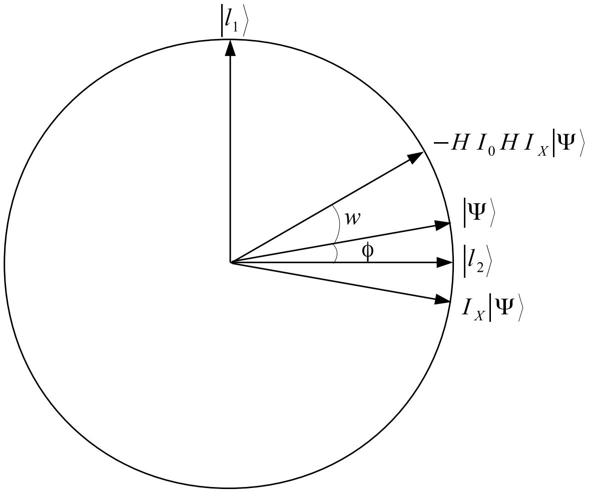

Figure 5: The effect of Grover's rotation on the state $|\Psi\rangle$.

with $|l_2\rangle$, and after some analysis one can find that applying the operator T times starting from the initial state yields a solution state with a maximal probability that is very close to 1 upon measurement when

$$T = \left\lfloor \frac{\pi}{4}\sqrt{\frac{N}{r}} \right\rfloor = O\left(\sqrt{\frac{N}{r}}\right).$$

The algorithm performs $O\left(\sqrt{\frac{N}{r}}\right)$ iterations, where $r = |X|$ is the number of marked states. Additional improvements were made by Boyer et al. (1996) and Brassard et al. (1998) coping with an unknown number of marked states in complexity $O\left(\sqrt{\frac{N}{r}}\right)$. Biham et al. (1999) introduced a third improvement that outputs a marked state when initiated with a state of an arbitrary amplitude distribution.

In the next sections, we present our algorithm for set intersection and its use in a quantum model of associative memory. We focus our analysis on the auto-associative memory model. However, presenting the algorithm with the quantum superposition of pairs x_i and y_i created by the use of the oracle B makes it a model for general associative memory with no additional cost, as a quantum search can yield y_i upon measurement of x_i.

1.7 Associative Memory

Associative memory stores and retrieves patterns with error correction or pattern completion of the input. The task can be defined as memorizing m pairs of data $\left(x^i, y^i\right)$, where x^i is an n dimensional vector and y^i is a q dimensional vector, and outputting y^i when presented with $\tilde{x}$, which is a faulty or a partial version of x^i. A specific case of associative memory is the auto-associative memory, in which $y^i \equiv x^i$, $\forall i \in \{1,\ldots,m\}$. Associative memory can be defined over a continuum, where $\left(x^i, y^i\right) \in R^n \times R^q$, or over a binary space, where $\left(x^i, y^i\right) \in \{0,1\}^n \times \{0,1\}^q$. In this paper, we concentrate on the binary model.

Algorithms for implementation of associative memory (Hopfield, 1982; Kanerva, 1993) have been extensively studied in the neural networks literature. The Hopfield model (Hopfield, 1982) consisting of n threshold neurons stores n-dimensional patterns $x \in \{\pm 1\}^n$ by the sum of outer products and retrieves a stored pattern when presented with a partial or a noisy version of the pattern.

The maximal storage capacity for which the stored patterns will be retrieved correctly with high probability (McEliece et al., 1987) is

$$M_{max} = \frac{n}{2 \ln n}.$$

Sparse encoding has been shown to increase the storage capacity considerably (Baram, 1991). A capacity exponential in the input dimension has been shown to result from a network size also exponential in the input dimension (Baram and Sal'ee, 1992).

2. Quantum Intersection

Given a set of marked states K of size k, Grover's quantum search algorithm for multiple marked states yields any member of the set with probability $O\left(k^{-1/2}\right)$ when given a phase version I_K of an oracle f_K of the form

$$f_K(x) = \begin{cases} 1, & x \in K \\ 0, & x \notin K \end{cases}. \tag{2}$$

Consequently, given a phase oracle I_K, Grover's algorithm chooses a member of the subset K. Suppose that, in addition to the oracle f_K, we have an oracle f_M, such that

$$f_M(x) = \begin{cases} 1, & x \in M \\ 0, & x \notin M \end{cases} \tag{3}$$

where M is another set of size m of marked states.

We define the problem of quantum intersection as the choice of any member of the intersection set $K \cap M$ of size r with probability $O\left(r^{-1/2}\right)$.

A straightforward algorithm for finding a member of the intersection between two sets of marked states K and M, based on the oracles f_K and f_M involves the use of the intersection oracle, comprising the oracles in Equation 2, Equation 3 in sequence, and a Toffoli gate as depicted in Figure 6.

When the input to the oracle is $|x\rangle \otimes |0\rangle \otimes |0\rangle \otimes |0\rangle$, then the output is $|x\rangle \otimes |f_K(x)\rangle \otimes |f_M(x)\rangle \otimes |f_{K\cap M}(x)\rangle$. The intersection phase oracle is realized when $|b\rangle = |-\rangle$, then, the input $|x\rangle \otimes |0\rangle \otimes |0\rangle \otimes |-\rangle$ will cause the output to be $|x\rangle \otimes |f_K(x)\rangle \otimes |f_M(x)\rangle \otimes \left[(-1)^{f_{K\cap M}(x)} |-\rangle\right]$.

Retrieving a state in the intersection between the two sets is accomplished by using Grover's quantum search algorithm with the phase version $I_{K\cap M}$ of the oracle $f_{M\cap K}$.

However, under certain conditions, one might not have the two oracles at hand, and, thus, the use of the intersection oracle might not be feasible. Another usage is when one constant oracle is available and the second oracle needs to be changed with each activation. We present an algorithm that applies a series of computations carried out by the two owners of the oracles in an alternating fashion.

Algorithm 1 : Quantum Set Intersection

```
Given: Phase oracles $I_M$and $I_K$
Denote: $Q_M \equiv -HI_0HI_M$, $Q_K \equiv -HI_0HI_K$
1. Let $|\Psi\rangle = H|0\rangle^{\otimes n}$
```

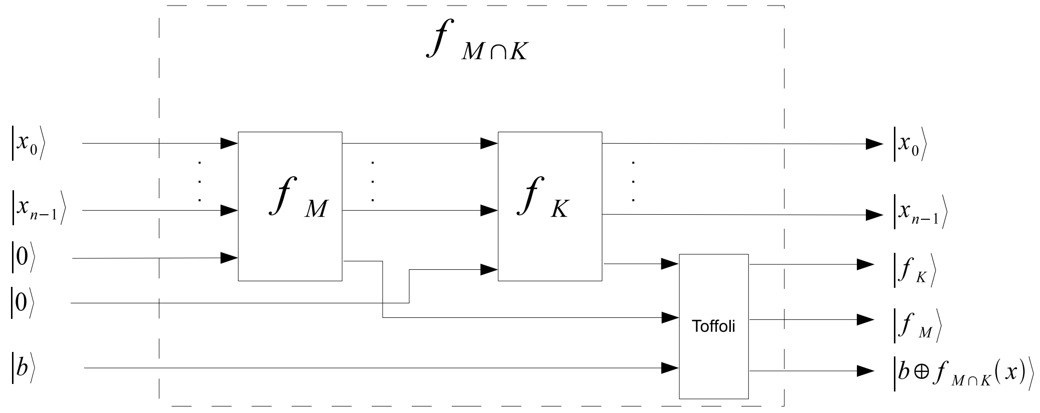

Figure 6: The intersection oracle $F_{K\cap M}$ created using f_M, f_K, and $Toffoli$.

```
2. Repeat
        $|\Psi\rangle = Q_K|\Psi\rangle$
        $|\Psi\rangle = Q_M|\Psi\rangle$
    $T = O\left(\sqrt{\frac{N}{|K\cap M|}}\right)$ times
3. Measure $|\Psi\rangle$
```

Algorithm 1 assumes that the size of the intersection set $|K\cap M|$ is known in order to determine the number of iterations. In the more general case where $|K\cap M|$ is unknown we apply the modification for an unknown number of marked states (Boyer et al., 1996). Alteratively, we can use the quantum counting algorithm (Brassard et al., 1998) for the apriori estimation of the number of marked states. In both cases the time complexity of the set intersection algorithm is sub-exponential. Next, we present two theorems that prove that Algorithm 1 measures a member of the intersection set with maximal probability and this probability is asymptotically close to 1.

Theorem 1 *Let I_K and I_M be phase oracles that mark two sets of n-qubit states K and M with $|K|,|M| << N$. Let us denote $|K| = k$, $|M| = m, |K\cap M| = r$,*

$$Q \equiv Q_M Q_K = (HI_0HI_MHI_0HI_K) \tag{4}$$

and

$$|\Psi(t)\rangle = Q^T H|0\rangle .$$

Then, the maximal probability of measuring a state in the intersection $K\cap M$ is achieved at

$$T = \arg\max_t \sum_{x\in K\cap M} |\langle x|\Psi(t)\rangle|^2 = \left\lfloor \frac{\pi/2 - \arctan\left(\sqrt{\frac{r}{N-r}}\right)}{\arccos\left(\frac{4km}{N^2} - \frac{4r}{N} + \Gamma\right)} \right\rfloor \tag{5}$$

where

$$\Gamma = \sqrt{1 - \frac{8rN^3 + 8kmN^2 - 16rkN^2 - 16rmN^2 + 32rkmN - 16k^2m^2}{N^4}}.$$

Proof The Hilbert space spanned by all computational basis states of the n-qubit register can be divided into four different subspaces: the subspace spanned by the states in $K \cap M$, the subspace spanned by the states in $K \backslash M$, the subspace spanned by the states in $M \backslash K$, and the subspace spanned by the states in $\overline{K \cup M}$. Accordingly, we select an orthonormal basis for representing the operator, in which the first four basis states are:

$$\begin{aligned}
\left|l_1\right\rangle &\equiv \frac{1}{\sqrt{|K \cap M|}} \sum_{i \in K \cap M} |i\rangle, &(6)\\
\left|l_2\right\rangle &\equiv \frac{1}{\sqrt{|K \backslash M|}} \sum_{i \in K \backslash M} |i\rangle, &(7)\\
\left|l_3\right\rangle &\equiv \frac{1}{\sqrt{|M \backslash K|}} \sum_{i \in M \backslash K} |i\rangle, &(8)\\
\left|l_4\right\rangle &\equiv \frac{1}{\sqrt{|\overline{K \cup M}|}} \sum_{i \in \overline{K \cup M}} |i\rangle &(9)
\end{aligned}$$

and the rest of the basis consists of orthonormal extensions of these states, then

$$H|0\rangle = \sqrt{\frac{r}{N}}\left|l_1\right\rangle + \sqrt{\frac{k-r}{N}}\left|l_2\right\rangle + \sqrt{\frac{m-r}{N}}\left|l_3\right\rangle + \sqrt{\frac{N-m-k+r}{N}}\left|l_4\right\rangle .$$

The operator Q_K affects only the four states $\left(\left|l_1\right\rangle, \left|l_2\right\rangle, \left|l_3\right\rangle, \left|l_4\right\rangle\right)$ as follows:

$$\begin{aligned}
Q_K\left|l_1\right\rangle = \quad & -HI_0HI_K\left|l_1\right\rangle = -H\left(I - 2|0\rangle\langle 0|\right)H\left(I - 2\sum_{i \in K}|i\rangle\langle i|\right)\left|l_1\right\rangle = \\
& \left(1 - \frac{2r}{N}\right)\left|l_1\right\rangle + \left(-\frac{2\sqrt{r(k-r)}}{N}\right)\left|l_2\right\rangle \\
+ \quad & \left(-\frac{2\sqrt{r(m-r)}}{N}\right)\left|l_3\right\rangle + \left(-\frac{2\sqrt{r(N-m-k+r)}}{N}\right)\left|l_4\right\rangle ,
\end{aligned}$$

$$\begin{aligned}
Q_K\left|l_2\right\rangle = \quad & -HI_0HI_K\left|l_2\right\rangle = -H\left(I - 2|0\rangle\langle 0|\right)H\left(I - 2\sum_{i \in K}|i\rangle\langle i|\right)\left|l_2\right\rangle = \\
& \left(-\frac{2\sqrt{r(k-r)}}{N}\right)\left|l_1\right\rangle + \left(1 - \frac{2(k-r)}{N}\right)\left|l_2\right\rangle \\
+ \quad & \left(-\frac{2\sqrt{(k-r)(m-r)}}{N}\right)\left|l_3\right\rangle + \left(-\frac{2\sqrt{(k-r)(N-m-k+r)}}{N}\right)\left|l_4\right\rangle ,
\end{aligned}$$

$$\begin{aligned}
Q_K\left|l_3\right\rangle = \quad & -HI_0HI_K\left|l_3\right\rangle = -H\left(I-2\left|0\right\rangle\left\langle 0\right|\right)H\left(I-2\sum_{i\in K}\left|i\right\rangle\left\langle i\right|\right)\left|l_3\right\rangle = \\
& \left(\frac{2\sqrt{r(m-r)}}{N}\right)\left|l_1\right\rangle + \left(\frac{2\sqrt{(k-r)(m-r)}}{N}\right)\left|l_2\right\rangle \\
+ \quad & \left(\frac{2(m-r)}{N}-1\right)\left|l_3\right\rangle + \left(\frac{2\sqrt{(m-r)(N-m-k+r)}}{N}\right)\left|l_4\right\rangle,
\end{aligned}$$

$$\begin{aligned}
Q_K\left|l_4\right\rangle = \quad & -HI_0HI_K\left|l_4\right\rangle = -H\left(I-2\left|0\right\rangle\left\langle 0\right|\right)H\left(I-2\sum_{i\in K}\left|i\right\rangle\left\langle i\right|\right)\left|l_4\right\rangle = \\
& \left(\frac{2\sqrt{r(N-m-k+r)}}{N}\right)\left|l_1\right\rangle + \left(\frac{2\sqrt{(k-r)(N-m-k+r)}}{N}\right)\left|l_2\right\rangle \\
+ \quad & \left(\frac{2\sqrt{(m-r)(N-m-k+r)}}{N}\right)\left|l_3\right\rangle + \left(\frac{2(N-m-k-+r)}{N}-1\right)\left|l_4\right\rangle
\end{aligned}$$

yielding Q_K in matrix form

$$Q_K = \begin{pmatrix}
1-\frac{2r}{N} & \frac{-2\sqrt{r}\sqrt{k-r}}{N} & \frac{2\sqrt{r}\sqrt{m-r}}{N} & \frac{2\sqrt{r}\sqrt{N-k-m+r}}{N} \\
\frac{-2\sqrt{r}\sqrt{k-r}}{N} & 1-\frac{2(k-r)}{N} & \frac{2\sqrt{k-r}\sqrt{m-r}}{N} & \frac{2\sqrt{k-r}\sqrt{N-k-m+r}}{N} \\
\frac{-2\sqrt{r}\sqrt{m-r}}{N} & \frac{-2\sqrt{k-r}\sqrt{m-r}}{N} & \frac{2(m-r)}{N}-1 & \frac{2\sqrt{m-r}\sqrt{N-k-m+r}}{N} \\
\frac{-2\sqrt{r}\sqrt{N-k-m+r}}{N} & \frac{-2\sqrt{k-r}\sqrt{N-k-m+r}}{N} & \frac{2\sqrt{m-r}\sqrt{N-k-m+r}}{N} & \frac{2(N-k-m+r)}{N}-1
\end{pmatrix}. \tag{10}$$

Similarly, we obtain the matrix form of Q_M

$$Q_M = \begin{pmatrix}
1-\frac{2r}{N} & \frac{2\sqrt{r}\sqrt{k-r}}{N} & \frac{-2\sqrt{r}\sqrt{m-r}}{N} & \frac{2\sqrt{r}\sqrt{N-k-m+r}}{N} \\
\frac{-2\sqrt{r}\sqrt{k-r}}{N} & \frac{2(k-r)}{N}-1 & \frac{-2\sqrt{k-r}\sqrt{m-r}}{N} & \frac{2\sqrt{k-r}\sqrt{N-k-m+r}}{N} \\
\frac{-2\sqrt{r}\sqrt{m-r}}{N} & \frac{2\sqrt{k-r}\sqrt{m-r}}{N} & \frac{1-2(m-r)}{N} & \frac{2\sqrt{m-r}\sqrt{N-k-m+r}}{N} \\
\frac{-2\sqrt{r}\sqrt{N-k-m+r}}{N} & \frac{2\sqrt{k-r}\sqrt{N-k-m+r}}{N} & \frac{-2\sqrt{m-r}\sqrt{N-k-m+r}}{N} & \frac{2(N-k-m+r)}{N}-1
\end{pmatrix}. \tag{11}$$

Substituting Equation 10 and Equation 11 into Equation 4 yields

$$\begin{aligned}
Q \quad = \quad & \begin{pmatrix}
1-\frac{8r(N-m)}{N^2} & \frac{-4\sqrt{r}\sqrt{k-r}(N-2m)}{N^2} & 0 & 0 \\
\frac{-4\sqrt{r}\sqrt{k-r}(N-2m)}{N^2} & \frac{8m(k-r)}{N^2}-1 & 0 & 0 \\
\frac{-8\sqrt{r}\sqrt{m-r}(N-m)}{N^2} & \frac{-4\sqrt{k-s}\sqrt{m-r}(N-2m)}{N^2} & 0 & 0 \\
\frac{-4\sqrt{r}\sqrt{N-k-m+r}(N-2m)}{N^2} & \frac{8m\sqrt{k-r}\sqrt{N-k-m+r}}{N^2} & 0 & 0
\end{pmatrix} \\
+ \quad & \begin{pmatrix}
0 & 0 & \frac{8\sqrt{r}\sqrt{m-r}(N-m)}{N^2} & \frac{4\sqrt{r}\sqrt{N-k-m+r}(N-2m)}{N^2} \\
0 & 0 & \frac{4\sqrt{k-r}\sqrt{m-r}(N-2m)}{N^2} & \frac{-8m\sqrt{k-r}\sqrt{N-k-m+r}}{N^2} \\
0 & 0 & \frac{8(N-m)(m-r)}{N^2}-1 & \frac{4\sqrt{m-r}\sqrt{N-k-m+r}(N-2m)}{N^2} \\
0 & 0 & \frac{4\sqrt{m-r}\sqrt{N-k-m+r}(N-2m)}{N^2} & 1-\frac{8m(N-k-m+r)}{N^2}
\end{pmatrix}.
\end{aligned}$$

The compound operator Q is a rotation in the 4-dimensional space spanned by Equations 6 - Equation 9. Finding the rotation angles requires the diagonalized matrix of Q. Let V be the matrix whose columns are the eigenvectors of Q, then $Q^D = V^{-1}QV$ is a diagonal matrix whose diagonal components are the eigenvalues of Q as follows:

$$Q^D = \begin{pmatrix} e^{-iw_1} & 0 & 0 & 0 \\ 0 & e^{iw_1} & 0 & 0 \\ 0 & 0 & e^{-iw_2} & 0 \\ 0 & 0 & 0 & e^{iw_2} \end{pmatrix}$$

where

$$\begin{aligned} e^{-iw_1} &= \frac{4km}{N^2} - \frac{4r}{N} + \Gamma - \Delta \\ e^{iw_1} &= \frac{4km}{N^2} - \frac{4r}{N} + \Gamma + \Delta \\ e^{-iw_2} &= \frac{4km}{N^2} - \frac{4r}{N} - \Gamma - \Delta \\ e^{iw_2} &= \frac{4km}{N^2} - \frac{4r}{N} - \Gamma + \Delta \end{aligned}$$

and Γ and Δ are given by

$$\begin{aligned} \Gamma &= \sqrt{1 - \frac{8rN^3 + 8kmN^2 - 16rkN^2 - 16rmN^2 + 32rkmN - 16k^2m^2}{N^4}}, \\ \Delta &= 2\sqrt{\frac{2N^2r(r+k+m) - N^3r(1+\Gamma) + N^2km\Gamma - N^2km - 8Nkmr + 4k^2m^2}{N^4}} \end{aligned}$$

which implies

$$\begin{aligned} w1 &= \arccos\left(\frac{4km}{N^2} - \frac{4r}{N} + \Gamma\right), \\ w2 &= \arccos\left(\frac{4km}{N^2} - \frac{4r}{N} - \Gamma\right). \end{aligned}$$

The amplitude of $|l_1\rangle$ is given by

$$a(t) = A\sin(w_1t + \phi) \tag{12}$$

where A is the maximal amplitude, to be found, and ϕ is the angle between the initial state $H|0\rangle^{\otimes n}$ and $|l_1\rangle$:

$$\phi = \arctan\sqrt{\frac{r}{N-r}}.$$

The maximal probability A^2 that a measurement of the system will produce a state in $K \cap M$ will be obtained at time

$$T = \arg\max_t A^2 sin^2(w_1t + \phi)$$

yielding

$$T = \left\lfloor \frac{\pi/2 - \arctan\left(\sqrt{\frac{r}{N-r}}\right)}{\arccos\left(\frac{4km}{N^2} - \frac{4r}{N} + \Gamma\right)} \right\rfloor$$

as asserted (Equation 5). ■

Theorem 1 suggests a way for approximating the time complexity of the algorithm when $m,k << N$. Employing the Taylor series, the second order approximation of the rotation angle is

$$w_1 = O\left(\sqrt{\frac{r}{N}}\right)$$

and the number of iterations can be approximated by

$$T \approx O\left(\frac{\pi/2 - \sqrt{\frac{r}{N-r}}}{\sqrt{\frac{r}{N}}}\right) = O\left(\sqrt{\frac{N}{r}}\right). \tag{13}$$

Theorem 2 *The maximal probability, A^2 from Equation 12, of measuring a marked state in $K \cap M$ from Theorem 1 is approximately* 1 *when $|K|,|M| << N$ and N is large.*

Proof The amplitude $a(t)$ of $|l_1\rangle$ behaves as in Equation 12 and the maximal amplitude A can be obtained from

$$\begin{aligned} A\sin(w_1+\phi) = & \sqrt{\frac{r}{N}}\left(1 - \frac{8r(N-m)}{N^2}\right) \\ + & \sqrt{\frac{k-r}{N}}\left(\frac{-4\sqrt{r}\sqrt{k-r}(N-2m)}{N^2}\right) \\ + & \sqrt{\frac{m-r}{N}}\left(\frac{8\sqrt{r}\sqrt{m-r}(N-m)}{N^2}\right) \\ + & \sqrt{\frac{N-m-k+r}{N}}\left(\frac{4\sqrt{r}\sqrt{N-m-k+r}(N-2m)}{N^2}\right). \end{aligned} \tag{14}$$

Substituting w_1 and ϕ in the left hand side of Equation 14 yields

$$
\begin{aligned}
A \approx \quad & \frac{\sqrt{\frac{r}{N}}\left(1-\frac{8r(N-m)}{N^2}\right)}{\sqrt{2-2\Gamma-\frac{8km}{N^2}+\frac{8r}{N}}+\sqrt{\frac{r}{N-r}}} \\
+ \quad & \frac{\sqrt{\frac{k-r}{N}}\left(\frac{-4\sqrt{r}\sqrt{k-r}(N-2m)}{N^2}\right)}{\sqrt{2-2\Gamma-\frac{8km}{N^2}+\frac{8r}{N}}+\sqrt{\frac{r}{N-r}}} \\
+ \quad & \frac{\sqrt{\frac{m-r}{N}}\left(\frac{8\sqrt{r}\sqrt{m-r}(N-m)}{N^2}\right)}{\sqrt{2-2\Gamma-\frac{8km}{N^2}+\frac{8r}{N}}+\sqrt{\frac{r}{N-r}}} \\
+ \quad & \frac{\sqrt{\frac{N-m-k+r}{N}}\left(\frac{4\sqrt{r}\sqrt{N-m-k+r}(N-2m)}{N^2}\right)}{\sqrt{2-2\Gamma-\frac{8km}{N^2}+\frac{8r}{N}}+\sqrt{\frac{r}{N-r}}}
\end{aligned}
$$

which, under the assumption $r,k,m << N$, is close to 1.

■

3. Quantum Associative Memory

In this section we introduce our associative memory model and the retrieval procedure with pattern completion and correction abilities. We present the concept of memory as a quantum operator that flips the phase of the memory patterns. The operator is based on an oracle that identifies, or ”marks”, memory patterns. This allows the initial state of our algorithm to be independent of the memory set. The input of our algorithms is an n-qubit register that contains the superposition of all basis states. This input is created by applying the Hadamard operation on an n-qubit register set to zeros, $H|0\rangle$.

3.1 Pattern Completion

Let I_M be a phase oracle on a set M, called the memory set, of m n-qubit patterns and let x' be a version of a memory pattern $x \in M$ with d missing bits. We are required to output the pattern x based on I_M and x'. The partial pattern is given as a string of binary values 0 and 1 and some unknown bits marked ’?’. Denoting the set of possible completions of the partial pattern K and its size k, the completion problem can be reduced to the problem of retrieving a member x of the intersection between two sets K and M, $x \in K \cap M$. For example, let $M = \{0101010, 0110100, 1001001, 1111000, 1101100, 1010101, 0000111, 0010010\}$ be a 7-bit memory set of size 8 and let ”0110?0?” be a partial pattern with 2 missing bits, so the completion set is $K = \{0110000, 0110001, 0110100, 0110101\}$. Pattern completion is the computation of the intersection between K and M, which is the memory pattern 0110100.

Pattern completion can use either the intersection oracle presented in Figure 6 or the quantum intersection presented in Algorithm 1. In either case, we need to create the completion operator f_K or its phase version I_K that marks the states of the set K, which can be implemented by checking whether a state is a completion of the partial patterns x' represented by the set K. Such an implementation is shown in Figure 7, where for each given bit in the pattern an appropriate control is added

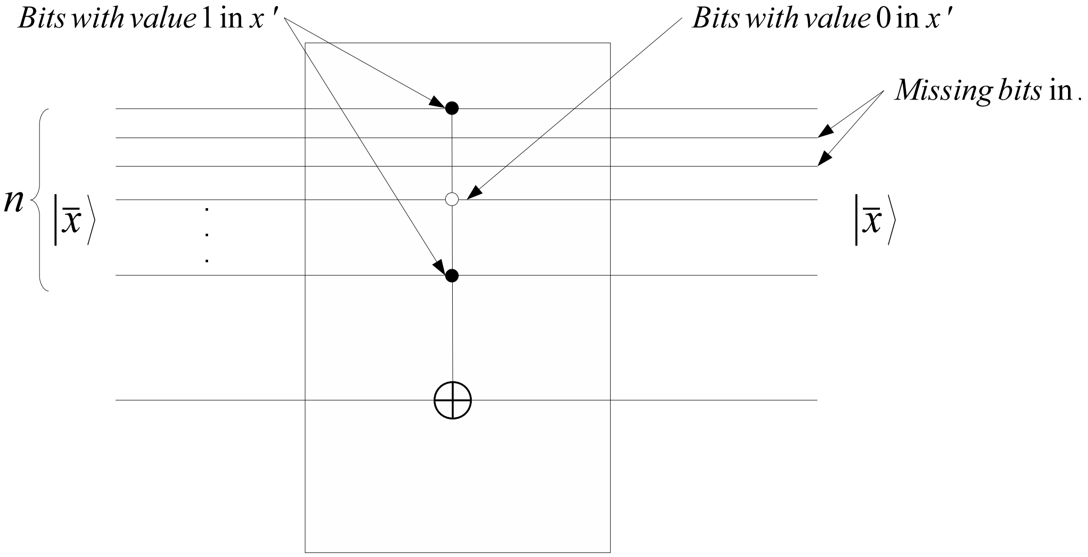

Figure 7: An implementation of a completion operator with an up to $n+1$ dimensional controlled-not operator.

to the corresponding qubit. The dark circle means that the control is activated when the bit value is 1 and the empty circle means that the control is activated when the bit value is 0. For each missing bit, no control is added for the corresponding qubit. A single n qubit operator can be implemented by $O(n)$ 2 and 3-qubit operators (Barenco et al., 1995).

The algorithm for pattern completion through the quantum intersection algorithm is

```
Algorithm 2 : Quantum Pattern Completion
Given: A memory operator $I_M$and a pattern $x' \in \{0,1\}^n$, which is a partial version
    of some memory pattern with up to d missing bits
1. Create the completion operator $I_K$.
2. Apply Algorithm 1 with $I_M$ and $I_K$
```

3.2 Pattern Correction

Let I_M be a phase oracle of a memory set M of size m and let x' be a version of a memory pattern $x \in M$ with up to d faulty bits. We are required to output the pattern x based on I_M and x'. The set K of possible corrections of the faulty pattern consists of all patterns in Hamming distance up to d from x'. The correction problem can be reduced to the problem of retrieving a member x of the intersection between two sets K and M, $x \in K \cap M$. For example, let $M = \{0101010, 0110100, 1001001, 1111000, 1101100, 1010101, 0000111, 0010010\}$ be a 7-bit memory set and let ”0110001” be the input pattern with 2 possible errors. The correction set K

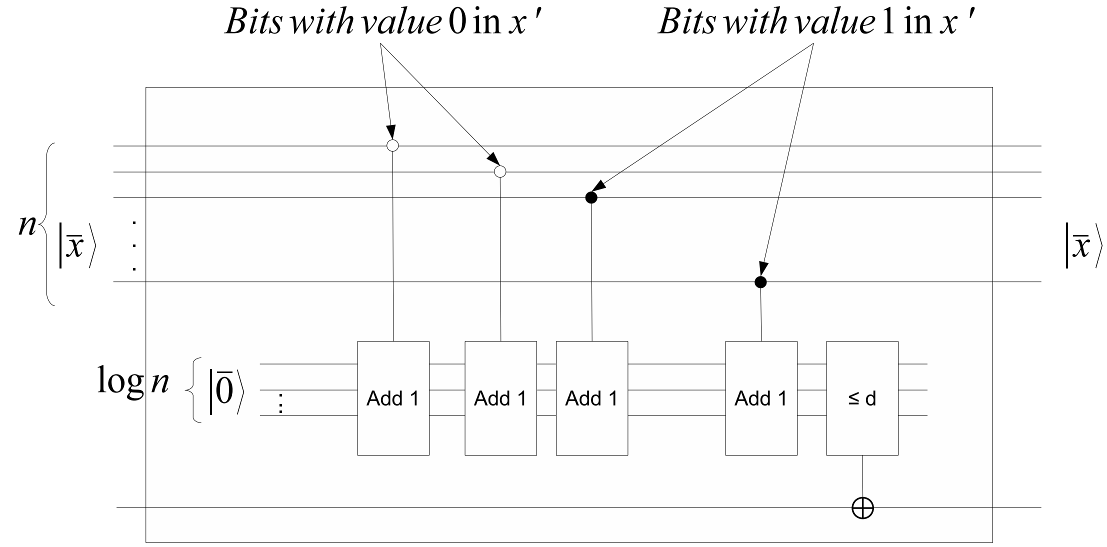

Figure 8: An implementation of a correction operator with a $\lceil \log n \rceil$ additional bits, n controlled operators that add 1, and a threshold operator.

consists of all patterns that are in Hamming distance up to 2 from x'. Pattern correction should then retrieve the memory pattern 0110100.

Pattern correction can be solved using the quantum intersection algorithm, which requires the creation of the correction operator f_K or its phase version I_K. This can be implemented by checking whether the number of different bits between any given state and the faulty pattern x' is less or equal to d. Such an implementation is shown in Figure 8. The operator applies a linear threshold on the hamming distance between x' and any input x. The operator consists of n input qubits, $\lceil \log n \rceil$ qubits for the hamming distance of x from x', and an additional qubit for the output. The operator adds 1 for each bit in x that is not equal to the corresponding bit in x', then applies a controlled not operator if the hamming distance does not exceed the threshold d. The operator can, thus, be implemented using $O(n \log n)$ 2 and 3-qubit operators.

Algorithm 3 : Quantum Pattern Correction

```
Given: A memory operator $I_M$and a pattern $x' \in \{0,1\}^n$, which is a faulty version
        of some memory pattern with up to d faulty bits
1. Create the correction operator $I_K$.
2. Apply Algorithm 1 with $I_M$ and $I_K$
```

A generalization of both Algorithm 2 and Algorithm 3 for the case of unknown number of possible corrections is straight forward using quantum search for an unknown number of marked states (Boyer et al., 1996) or quantum counting (Brassard et al., 1998).

Algorithm 3 finds a memory pattern that is in Hamming distance up to d from x'. If the memory is within the correction capacity bounds, Algorithm 3 finds the correct pattern x with high probability, as will be proved in Section 4. However, if we are interested in ensuring that we find the closest memory pattern to x', with no dependence on the capacity bound, then we can apply Algorithm 3 for $i = 0$ bits and increase it up to $i = d$ bits. In this case we ensure that we find a pattern x, such that

$$\left\{x : (x \in M) \wedge \left(\forall x'' \in M : dist(x'',x) \geq dist(x',x)\right)\right\}$$

where $dist(\cdot,\cdot)$ is the Hamming distance.

4. Analysis of the Quantum Associative Memory

In this section we analyze the time complexity and memory capacity of the proposed quantum associative memory. We first show that the time complexity of retrieval operations is sub-exponential in the number of bits. Then we show that the number of memory patterns that can be stored while the model retains its correction and completion abilities is exponential in the number of bits.

4.1 Time Complexity Analysis

The time complexity of the retrieval procedure with either pattern completion or correction ability is determined by the complexity of the quantum intersection algorithm and the complexity of the completion and correction operators. The two operators can be implemented by a number of operations which is linear in n or in $n \log n$. According to Equation 13, the completion and correction operations are performed in

$$T \approx n \log n + O\left(\sqrt{\frac{N}{|K \cap M|}}\right) = O\left(\sqrt{\frac{N}{|K \cap M|}}\right)$$

operations, which is sub-exponential in number of bits.

4.2 Capacity Analysis

We consider three different capacity measures. The first is the equilibrium capacity M_{eq}, which is the maximal memory size that ensures that all memory patterns are equilibrium points of the model. An Equilibrium point is a pattern that, when presented to the model as an input, is also retrieved as an output with high probability. The second is the pattern completion capacity M_{com}, which is the maximal memory size that allows the completion of any partial pattern with up to d missing bits with high probability. The third is the pattern correction capacity M_{cor}, which is the maximal memory size that allows the correction of any pattern with up to d faulty bits with high probability.

Equilibrium is a special case of completion and correction with neither missing nor faulty bits. Therefore, the equilibrium capacity should be equal to the completion and correction capacities for $d = 0$.

4.2.1 EQUILIBRIUM CAPACITY

The equilibrium capacity of the quantum associative memory is

$$M_{eq} = N$$

because every memory state of an n-bit associative memory M of any size $m \leq N$ is an equilibrium state, that is, if $Q_x = -(HI_0HI_x)$, $T = \left\lfloor \frac{\pi}{4}\sqrt{N} \right\rfloor$, and $|\Psi(T)\rangle = Q_x^T H|0\rangle$, then

$$\forall x \in M : \ |\langle x|\Psi(T)\rangle|^2 \rightarrow 1 \ as \ n \rightarrow \infty.$$

This is a direct consequence of Grover's algorithm (Grover, 1996) and the results obtained by Boyer et al. (1996) concerning the ability to find any member of the size N database with probability close to 1.

4.2.2 COMPLETION CAPACITY

Given a pattern x', which is a partial version of some memory pattern x_c with d missing bits, we seek the maximal memory size, for which the pattern can be completed with high probability from a random uniformly distributed memory set (McEliece et al., 1987; Baram, 1991; Baram and Sal'ee, 1992).

The completion capacity is bounded from above by two different bounds. The first is a result of Grover's quantum search algorithm limitations and the second is a result of the probability of correct completion.

A Bound on Memory Size due to Grover's Quantum Search Limitations. Grover's operator flips the marked states around the zero amplitude (negating their amplitudes) then flips all amplitudes around the average of all amplitudes (Biham et al., 1999). Amplification of the desired amplitudes occurs only when the average of all amplitudes is closer to the amplitudes of the non-marked states than to the marked states. This imposes the following upper bound on the memory size:

$$m < N/2.$$

This can be observed in the first iteration of the quantum search algorithm on a number of marked states. The initial amplitude of all basis states in $H|0\rangle$ is $1/\sqrt{N}$. Flipping the phase of m marked states by I_M to $-1/\sqrt{N}$ yields an average amplitude of

$$(N-m) * \left(1/\sqrt{N}\right) - m * \left(1/\sqrt{N}\right) = \frac{N-2m}{\sqrt{N}}.$$

Then, flipping the phases of all basis states around this average by HI_0H yields two values of amplitudes. The amplitudes of marked and unmarked states become

$$2\left(\frac{N-2m}{\sqrt{N}}\right) \pm \frac{1}{\sqrt{N}} = \left(\frac{2N-4m \pm 1}{\sqrt{N}}\right) \tag{15}$$

where ± correspond to marked and unmarked states respectively.

A necessary condition for the amplification of marked states is that the absolute value of their amplitudes after an iteration of the algorithm is higher than the absolute value of the amplitude of unmarked states. The condition is satisfied if and only if the two equations given in Equation 15 satisfy

$$\left|\left(\frac{2N-4m+1}{\sqrt{N}}\right)\right| > \left|\left(\frac{2N-4m-1}{\sqrt{N}}\right)\right|$$

which holds true if and only if $m < N/2$. Therefore, if $m \geq N/2$ the amplitudes of the marked states will not increase, which gives the following upper bound on the completion capacity:

$$M_{com} < N/2. \tag{16}$$

However, this is a very loose bound and the success probability of the completion will impose a tighter bound.

A Bound on Memory Size due to Pattern Completion. The bound on memory size that ensures a high probability of correct completion depends on the definition of the pattern completion procedure. If one defines pattern completion as the process of outputting any of a number of possible memory patterns when given a partial input, then the capacity bound of our memory is the amplification bound given in Equation 16. However, this is not always the case. Pattern completion capacity is usually defined as the maximal size of a random uniformly distributed memory set that, given a partial version x' of a memory $x_c \in M$ with d missing bits, outputs x_c. The following theorem gives an upper bound on the capacity for pattern completion with high probability:

Theorem 3 *An n-bit associative memory with m random patterns can complete up to d missing bits on average when*

$$m \leq 2^{n-d}$$

with probability higher than

$$\frac{v}{e^v - 1}\sum_{i=1}^{m}\frac{v^{i-1}}{i^i} \tag{17}$$

as n grows to infinity, where

$$v = \frac{m}{2^{n-d}}.$$

Proof Let M be a random uniformly distributed memory set of size $m = v2^{n-d}$, where $0 < v < 1$. Let x' be a partial pattern of $x_c \in M$ with d missing bits. x' induces the set

$$K = \left\{x \middle| x\ is\ a\ completion\ of\ x'\right\}$$

where $|K| = 2^d$. Let Z_i be random indicator variables representing the existence of the ith member of M in K. Denoting $p = Pr(Z_i = 1) = \frac{2^d}{2^n} = \frac{1}{2^{n-d}}$, we have

$$m = v/p. \tag{18}$$

Let us denote $|\Psi(T)\rangle = Q^T H|0\rangle$ and $S = \sum_{i=1}^{m} Z_i$. If there is only one possible memory completion, then it is x_c, and if there are two, then x_c is one of the two, and so on. Therefore, the probability of successfully outputting x_c from the partial pattern x' is the sum of the conditional probabilities that there are i memory completions divided by i:

$$\begin{aligned}
|\langle x_c|\Psi(T)\rangle|^2 &= \sum_{i=1}^{m}\frac{Pr(S=i \mid S\geq 1)}{i} \\
&= \sum_{i=1}^{m}\frac{Pr(S=i)}{iPr(S\geq 1)} \\
&= \sum_{i=1}^{m}\frac{\binom{m}{i}p^i(1-p)^{m-i}}{i(1-(1-p)^m)}.
\end{aligned} \tag{19}$$

Now, this probability can be lower-bounded by

$$|\langle x_c|\Psi(T)\rangle|^2 \geq \sum_{i=1}^{m} \frac{\left(\frac{m}{i}\right)^i p^i (1-p)^{m-i}}{i\left(1-(1-p)^m\right)}. \tag{20}$$

Substituting Equation 18 in Equation 20 we have

$$\begin{aligned}
|\langle x_c|\Psi(T)\rangle|^2 &\geq \sum_{i=1}^{m} \frac{\left(\frac{v^i}{i^i}\right)(1-p)^{v/p-i}}{i\left(1-(1-p)^{v/p}\right)} \\
&\geq \sum_{i=1}^{m} \frac{\left(\frac{v^i}{i^i}\right) e^{-v}\frac{1}{(1-p)^i}}{i\left(1-e^{-v}\right)} \\
&\geq \frac{1}{e^v-1}\sum_{i=1}^{m} \frac{v^i}{(i(1-p))^i} \\
&> \frac{v}{e^v-1}\sum_{i=1}^{m} \frac{v^{i-1}}{i^i}. \qquad (21)
\end{aligned}$$

■

Figure 9 shows that the lower bound with approximation by only three terms of the sum in Equation 17 as a function of v is higher than 75% for all possible sizes of a non-empty memory within the capacity limits.

Theorem 3 implies that $M_{com}(d) = 2^{n-d}$, which agrees with the result concerning the equilibrium capacity, since $M_{com}(0) = 2^{n-0} = 2^n = N = M_{eq}$.

4.2.3 CORRECTION CAPACITY

A bound on the correction capacity of Algorithm 3 is given be the following theorem:

Theorem 4 *An n-bit associative memory with m random patterns can correct up to d faulty bits on average when*

$$m \leq 2^{n-d}/\binom{n}{d}$$

with probability higher than

$$\frac{v}{e^v-1}\sum_{i=1}^{m}\frac{v^{i-1}}{i^i}$$

as n grows to infinity, where

$$v = \frac{m}{2^{n-d}}.$$

Proof Let M be a random uniformly distributed memory set of size

$$m = v2^{n-d}/\binom{n}{d}$$

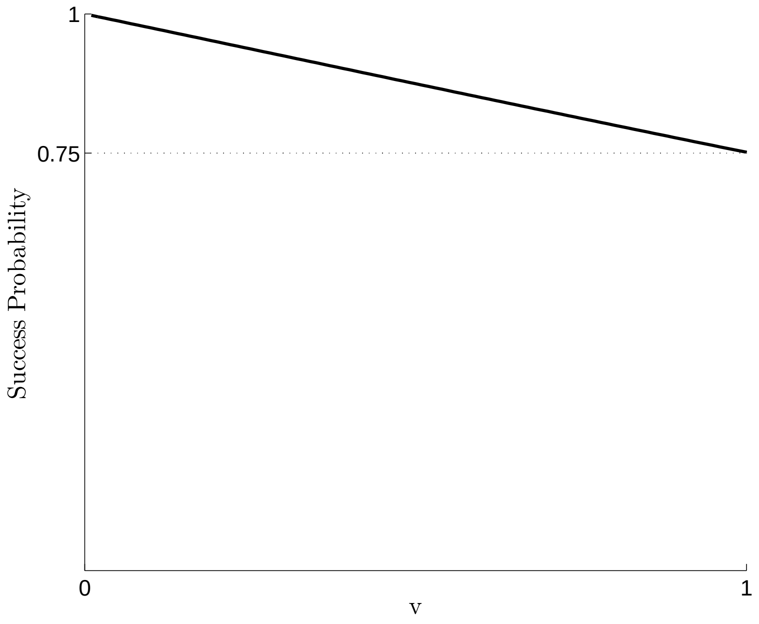

Figure 9: Success probability of pattern completion vs. memory size divided by the maximal completion capacity v for an associative memory with $n = 30$ qubits.

where $0 < v < 1$. Let x' be a pattern $x_c \in M$ with d faulty bits. x' induces a set

$$D = \left\{ x \middle| dist(x, x') \leq d \right\}$$

where $|D| = \binom{n}{d} 2^d$. Let Z_i be random indicator variables representing the existence of the ith member of M in D. Denoting $p = Pr(Z_i = 1) = \binom{n}{d} 2^{d-n}$, we have

$$m = v/p.$$

Let us denote $|\Psi(T)\rangle = Q^T H |0\rangle$ and $S = \sum_{i=1}^{m} Z_i$. The probability of successfully retrieving x_c from the pattern x' is then

$$|\langle x_c|\Psi(T)\rangle|^2 = \sum_{i=1}^{m} \frac{Pr\left(S = i \mid S \geq 1\right)}{i}$$

which, according to Equations 19 - 21, also satisfies

$$|\langle x_c|\Psi(T)\rangle|^2 \geq \frac{v}{e^v - 1} \sum_{i=1}^{m} \frac{v^{i-1}}{i^i}.$$

■

Theorem 4 yields $M_{cor}(d) = \binom{n}{d} 2^{n-d}$, which also agrees with the result concerning the equilibrium capacity, since $M_{cor}(0) = \binom{n}{0} 2^{n-0} = 2^n = N = M_{eq}$.

For example, let M be a memory set over $\{0,1\}^{100}$, then, as long as $|M| < 2^{80}$, we can complete up to $d = 100 - \log|M| = 20$ bits and correct up to $d = 13$ bits.

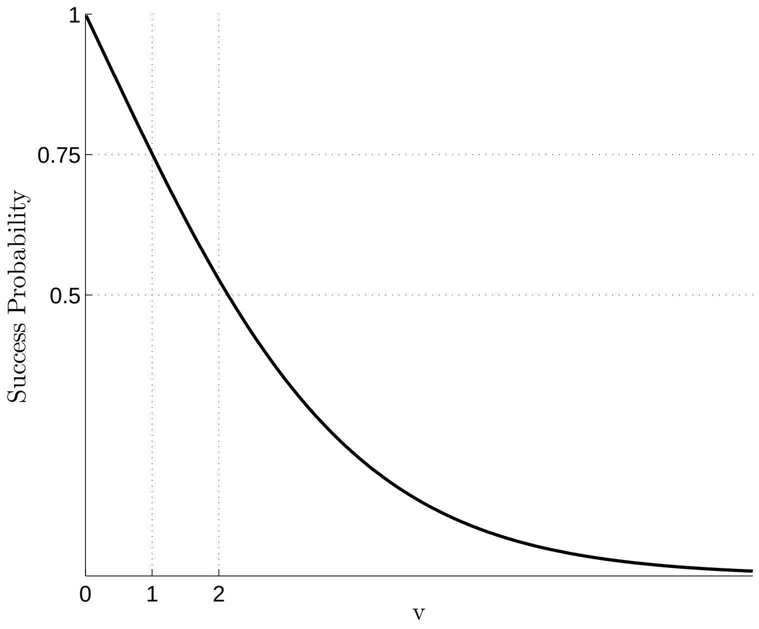

Figure 10: Pattern completion or correction probability vs. the memory size divided by the maximal completion capacity v. For $0 < v < 1$, the probability is above 75% and for $v < 2$ is above 50%.

4.2.4 INCREASING MEMORY SIZE BEYOND THE CAPACITY BOUNDS

The various capacities presented above are exponential in n under the assumption $d << n$. However, an increase of m beyond the capacity bound results in a decay of the correct completion probability as depicted in Figure 10. It can be seen that it is more likely to find the correct completion than not to find it as long as $v < 2$.

In addition, the model can also output a superposition of a number of possible outputs, by skipping the measurement operation in Algorithm 1. This is not true for most classical memory models where spurious memories arise and the output is usually not a memorized pattern, but, rather, some spurious combination of multiple memory patterns (Hopfield, 1982; Bruck, 1990; Goles and Martínez, 1990).

5. Comparison to Previous Works

Quantum computation was previously applied to associative memory by Ventura and Martinez (2000), Ezhov et al. (2000), Howell et al. (2000), and, subsequently, by others. An algorithm based on the model developed by Ventura and Martinez (2000) was proposed by Arima et al. (2008). It was further developed by Arima et al. (2009) and analyzed by Miyajima et al. (2010). We analyze the two main algorithms (Ventura and Martinez, 2000; Arima et al., 2009) and show their differences with respect to our algorithm. These algorithms are given below.

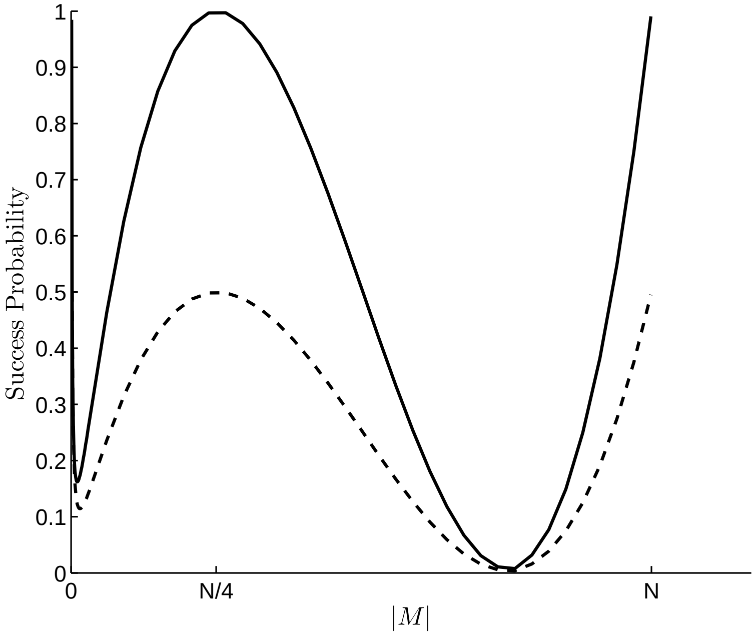

Figure 11: Memory size vs. success probability of Algorithm 4. Optimal results are achieved only when the memory size is close to $\frac{N}{4}$.

Algorithm 4 The algorithm proposed by Ventura and Martinez (2000)

```
Given: Phase oracles $I_M$and $I_K$
1. Denote $Q_M = -HI_0HI_M$ and $Q_K = -HI_0HI_K$
2. Let $|\Psi\rangle = \frac{1}{m}\sum_{i=1}^{m}|i\rangle$.
3. Apply $Q_MQ_K$ on $|\Psi\rangle$
4. Apply $Q_K$ on $|\Psi\rangle$ for $T = \left\lfloor \pi/4\sqrt{N/|K \cap M|} \right\rfloor$-2 times.
5. Measure $|\Psi\rangle$.
```

Algorithm 5 The algorithm proposed by Arima et al. (2009)

```
Given: Phase oracles $I_M$and $I_K$
1. Denote $Q_M = -HI_0HI_M$ and $Q_K = -HI_0HI_K$
2. Let $|\Psi\rangle = \frac{1}{m}\sum_{i=1}^{m}|i\rangle$.
3. Apply $Q_MQ_K$ on $|\Psi\rangle$ for $T$times. ($T$was not found by Arima et al., 2009)
4. Measure $|\Psi\rangle$.
```

Algorithm 4 can find only a single marked state with high probability when the memory size m is close to $\frac{N}{4} - 2$, as shown by the solid line in Figure 11. The probability of measuring this state reduces by a half when there are two marked states and only one of them is a memory pattern, as shown by the dashed line in Figure 11, and so on.

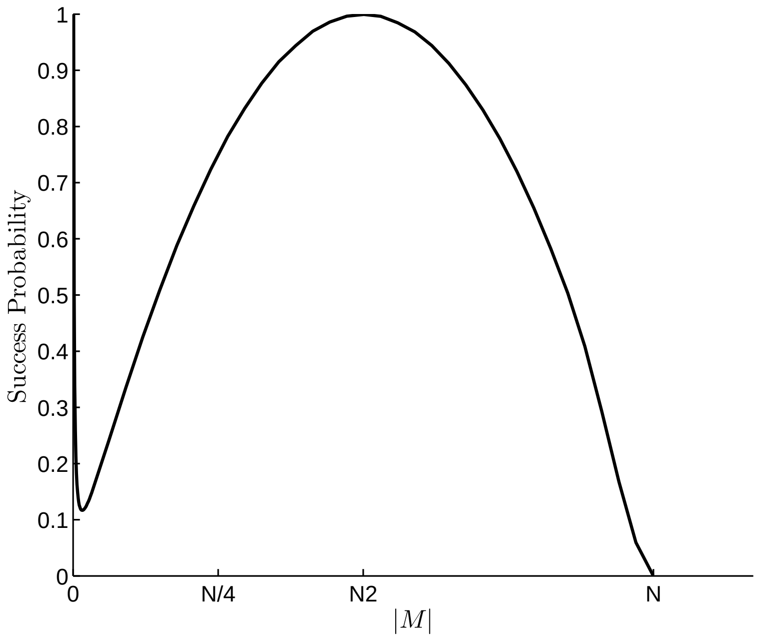

Figure 12: Memory size vs. success probability in Algorithm 5. Satisfactory results are achieved only when the memory size exceeds $\frac{N}{4}$.

Algorithm 5 gives satisfying results only when the memory size exceeds $\frac{N}{4}$, which is exponential in the number of qubits, leaving the possibility of effective pattern completion only for 2 qubits or less. It is therefore not helpful for associative memory with pattern completion and correction abilities. The success probability of Algorithm 5 vs. the memory size is depicted in Figure 12. Miyajima et al. (2010) added a control parameter to tune the algorithm, changing the memory size for which the maximal amplitude is achieved. The algorithm is presented only for one marked state with no completion and correction abilities. The time complexity and stopping criteria were not stated by Arima et al. (2008) and were later found to be $O\left(\sqrt{N}\right)$ (Arima et al., 2009; Miyajima et al., 2010).

Our algorithm, on the other hand achieves high success probability up to memory size $\frac{N}{4}$, as depicted in Figure 13.

Furthermore, both Algorithm 4 and Algorithm 5 need to initialize the system at a superposition of the memory states:

$$|\Psi\rangle = \frac{1}{\sqrt{m}} \sum_{i=1}^{m} |i\rangle$$

which is a shortcoming for two reasons: the time complexity of initialization when the memory size is large and the need for repeated initialization upon every application of the memory. The latter is important as it adds an exponential factor to the query time, for either completion or correction, and an exponential addition to the single query time when amplitude amplification is needed. Amplitude amplification ensures that we pick the correct pattern with probability 1 by performing the algorithm

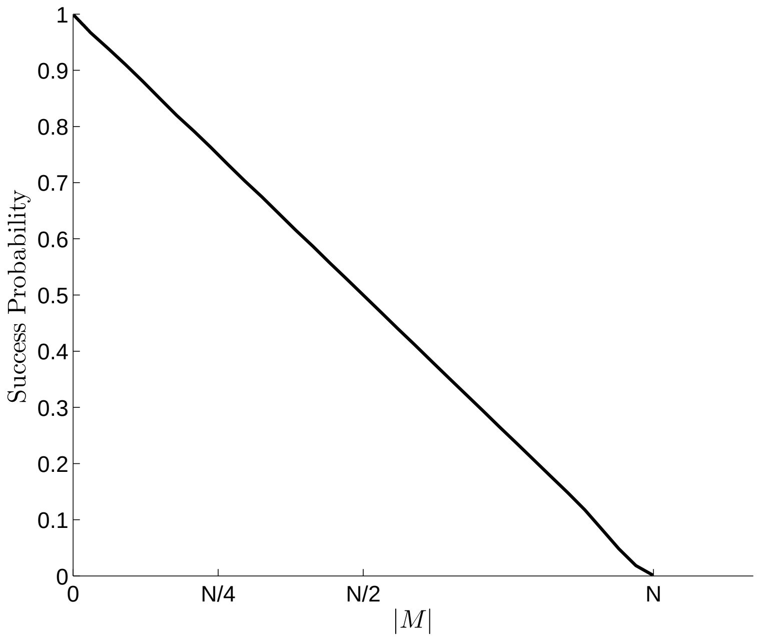

Figure 13: Memory size vs. success probability in Algorithms 2 and 3.

a multiple number of times. Our algorithm's initialization, on the other hand, does not depend on the memory patterns.

6. Numerical Examples and Simulations

Let us first consider an associative memory of 10 qubits. We have randomly chosen a set of 50 patterns M out of the possible 1024 to be stored in memory. We also chose two partial patterns, each with 4 missing qubits, yielding two completion sets K_1 and K_2 of 16 possible completions each. We chose K_1 and K_2 such that they have one and two completions in memory respectively. Figure 14(a) shows the memory set, where each vertical line represents a memory pattern, and Figure 14(b) shows the completion set K_1 in the same manner. The amplitudes of the final state of the completion algorithm are shown in Figure 14(c), where the only possible memory completion has amplitude close to 1. Figure 15 shows the high amplitudes of the two possible memory completions when the completion set is K_2.

As can be seen, applying our algorithm to both completion sets amplified the states that are possible completions in memory. The amplitudes of the desired states reached up to 96.76% in the first case and 68.44% in the second case for each one of the two high amplitudes. Therefore, the probability of measuring the correct completion in the first case is 93.62% and the probability of measuring one of the two correct completions in the second case is 93.67%.

Another simulation was carried out on a 10 qubits associative memory with 2^7 memory patterns and completion queries with 3 missing bits. The behavior of the different subgroups of the basis states is schematically described in Figure 16 for a series of iterations with the completion operator Q of Equation 4. Each amplitude value indicated represents the amplitudes of all the basis states that belong to the corresponding subgroup. It shows the amplification of states in the intersection

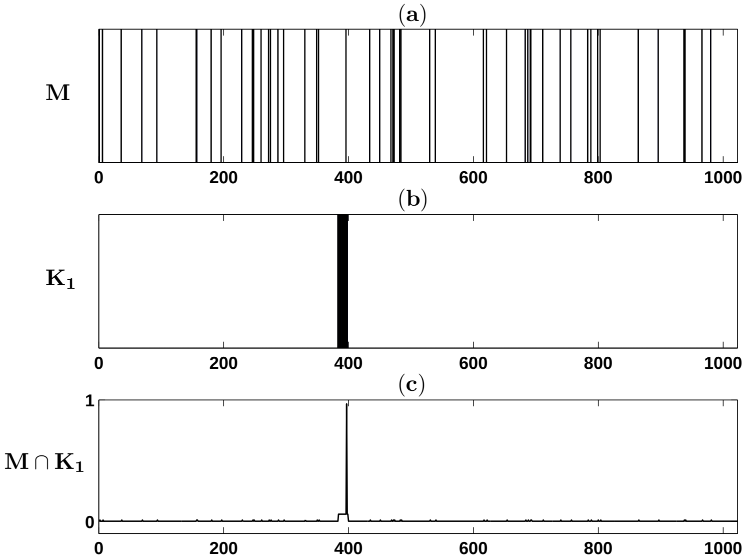

Figure 14: (a) A set of memory patterns M (b) a set of possible completions K_1 to a partial pattern, and (c) the memory completion result in amplitudes.

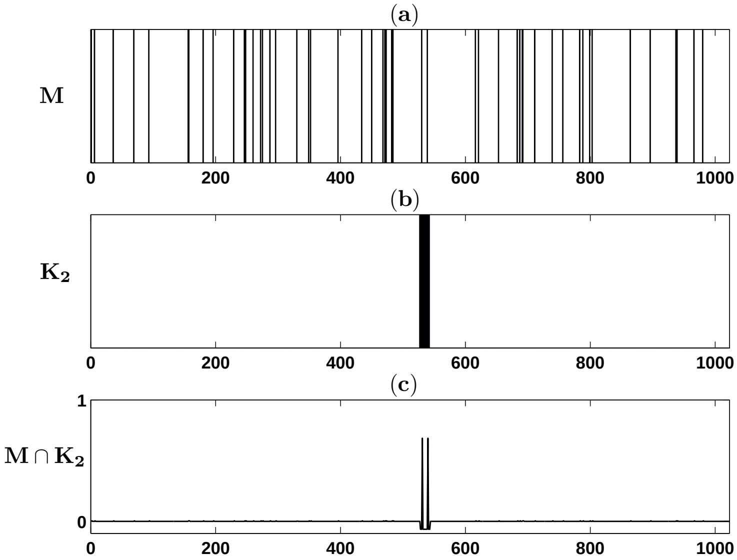

Figure 15: (a)A set of memory patterns M (b) a set of possible completions K_2 to a partial pattern, and the memory completion result in amplitudes.

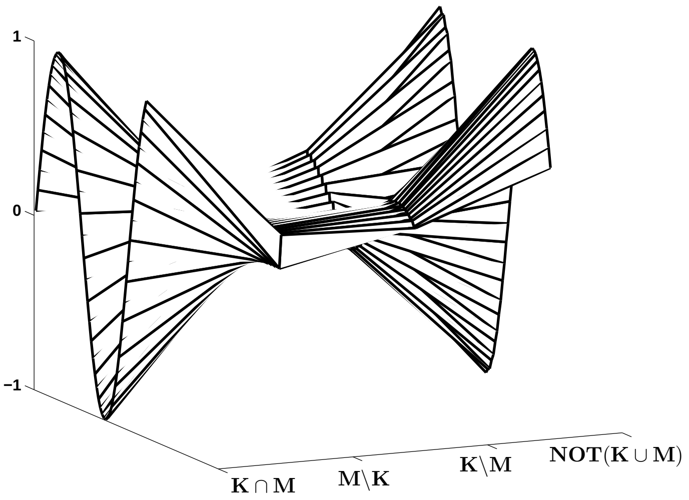

Figure 16: Simulation of a series of iterations of the completion algorithm. The graph shows the different behavior of the different subgroups of basis states. K is the completion set and M is the memory set. The memory completions and the non completions or memories are amplified alternatingly, while the amplitudes of $K \backslash M$ and $M \backslash K$ subgroups stay close to zero.

group $K \cap M$ and in $NOT(K \cup M)$ alternatingly, while the amplitudes of states in $K \backslash M$ and $M \backslash K$ stay close to zero.

We have also tested our algorithms with a larger number of qubits in order to verify that the success rate of retrieval grows asymptotically to 1 as the number of qubits grows. For instance, we tested a 30 qubit system with 2^{25} memory patterns and a completion query that has 8 missing bits. We tested different completions of 8 missing bits so that the intersection set size varied from 1 to 10 patterns. Our algorithm found a member of the memory completion set with probability 96.8%. Increasing the memory size to 2^{26} and 2^{27}, and thereby bringing the capacity close to its limit resulted in completion probabilities of 93.5% and 86.7% respectively. Figure 17 depicts the success rates of pattern completion in a 30 qubit system. An explanation of the different graphs can be found in Table 1. The solid line in Figure 17 depicts the success probability vs. the logarithm of the size of memory with completion queries set to 8 missing bits and the number of possible memory completions set to 1. The dashed line depicts the success probability vs. the logarithm of the completion query size when the memory size is set to 2^{25} patterns and the number of possible memory completions set to 1. The dotted line depicts the success probability vs. the logarithm of the number of possible memory completions when both the memory size and the completion query size are set to 2^{25}. The dash-dotted line depicts the success probability vs. the number of qubits in the system (growing from 5 to 30 qubits) when the memory size, the completion query size, and the number of possible memory completions are small constants.

Figure 17 shows that the deterioration of the success probability vs. the memory size or the completion query size is very slow. For instance, deterioration starts at memory size 2^{26}. Further-

Graph	$\vert N\vert$	$\vert M\vert$	$\vert K\vert$	$\vert K\cap M\vert\vert$
Solid	constant	varying	constant	constant
——	2^{30}	2^3-2^{27}	2^3	1
Dashed	constant	constant	varying	constant
– – –	2^{30}	2^{25}	2^3-2^{25}	1
Dotted	constant	constant	constant	varying
· · ·	2^{30}	2^{25}	2^{25}	$1-2^{25}$
Dash-dotted	varying	constant	constant	constant
– · – · –	2^5-2^{30}	2^3	2^3	1

Table 1: Properties of the four simulations depicted in Figure 17. The x axis in Figure 17 represents the varying set size, while the other set sizes are constant in each simulation.

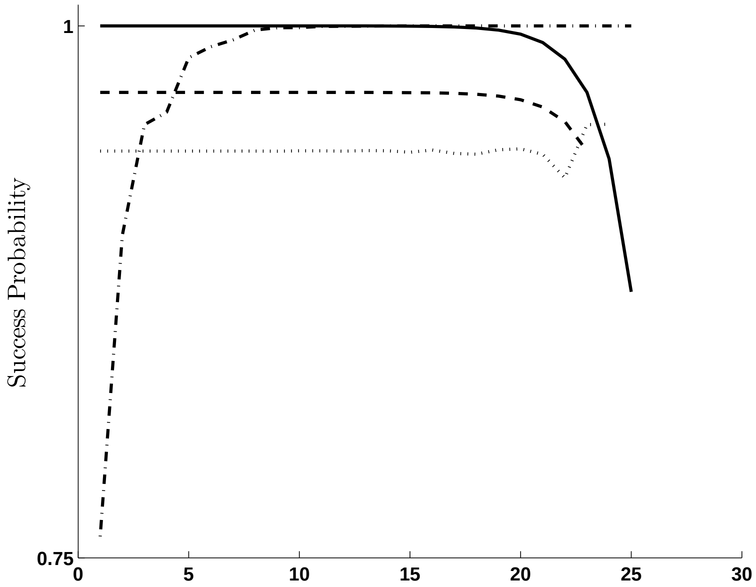

Figure 17: Success probability of measuring a desired memory completion vs. the *log* of the memory size (solid), completion query size (dashed), possible memory completions (dotted), and number of qubits (dash-dotted).

more, the success probability increases when the number of possible memory completions (the size of the intersection set) grows towards the sizes of the completion query and the memory, which indicates that choosing a member of the intersection becomes easy (by randomly choosing a possible completion). Finally, the figure also shows that, as the number of qubits in the system grows, the success probability becomes asymptotically 1, which indicates that, practically, our algorithm produces the intersection when $n >> 1$.

7. Conclusion

We have presented a quantum computational algorithm that computes the intersection between two subsets of n-bit strings. The algorithm is based on a modification of Grover's quantum search. Using the intersection algorithm, we have presented a set of algorithms that implement a model of associative memory via quantum computation. We introduced the notion of memory as a quantum operator, thus avoiding the dependence of the initial state of the system on the memory set. We have shown that our algorithms have both speed and capacity advantages with respect to classical associative memory models, consuming sub-exponential time, and are able to store a number of memory patterns which is exponential in the number of bits. Pattern retrieval algorithms with completion and correction abilities were presented. Bounds relating memory capacity to the maximal allowed signal to noise ratio were found.

References

K. Arima, H. Miyajima, N. Shigei, and M. Maeda. Some properties of quantum data search algorithms. In *Proceedings of the The 23rd International Technical Conference on Circuits/Systems, Computers and Communication (ITC-CSCC2008)*, pages 1169–1172, 2008.

K. Arima, N. Shigei, and H. Miyajima. A proposal of a quantum search algorithm. *International Conference on Convergence Information Technology*, 0:1559–1564, 2009.

Y. Baram. On the capacity of ternary hebbian networks. *Information Theory, IEEE Transactions on*, 37(3):528–534, May 1991.

Y. Baram and D. Sal'ee. Lower bounds on the capacities of binary and ternary networks storing sparse random vectors. *Information Theory, IEEE Transactions on*, 38(6):1633 –1647, nov 1992.

A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. H. Margolus, P. W. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter. Elementary gates for quantum computation. *Physical Review A*, 52 (5):3457–3467, 1995.

E. Biham, O. Biham, D. Biron, M. Grassl, and D. A. Lidar. Grover's quantum search algorithm for an arbitrary initial amplitude distribution. *Physical Review A*, 60(4):2742–2745, 1999.

M. Boyer, G. Brassard, P. H*ö*yer, and A. Tapp. Tight bounds on quantum searching. Technical Report PP-1996-11, 30, 1996.

G. Brassard, P. H*ö*yer, and A. Tapp. Quantum counting. In *ICALP '98: Proceedings of the 25th International Colloquium on Automata, Languages and Programming*, pages 820–831, 1998. ISBN 3-540-64781-3.

J. Bruck. On the convergence properties of the hopfield model. *Proceedings of the IEEE*, 78(10): 1579–1585, oct 1990.

D. Deutsch. Quantum theory, the church-turing principle and the universal quantum computer. *Proceedings of the Royal Society A Mathematical Physical and Engineering Sciences*, 400:97–117, 1985.

D. Deutsch and R. Jozsa. Rapid solution of problems by quantum computation. *Proceedings Mathematical and Physical Sciences*, 439(1907):553–558, 1992.

A. Ezhov, A. Nifanova, and D. Ventura. Quantum associative memory with distributed queries. *Inf. Sci. Inf. Comput. Sci.*, 128(3-4):271–293, 2000.

E. Goles and S. Martínez. *Neural and Automata Networks: Dynamical Behavior and Applications*. Kluwer Academic Publishers, 1990.

L. K. Grover. A fast quantum mechanical algorithm for database search. In *STOC '96: Proceedings of the Twenty-eighth Annual ACM Symposium on Theory of Computing*, pages 212–219, 1996.

J. Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the National Academy of Sciences of the United States of America*, 79:2554–2558, 1982.

J. C. Howell, J. A. Yeazell, and D. Ventura. Optically simulating a quantum associative memory. *Physical Review A*, 62(4):042303, 2000.

P. Kanerva. Sparse distributed memory and related models. In *Associative Neural Memories*, pages 50–76. Oxford University Press, 1993.

R. McEliece, E. Posner, E. Rodemich, and S. Venkatesh. The capacity of the hopfield associative memory. *Information Theory, IEEE Transactions on*, 33(4):461–482, jul 1987.

H. Miyajima, N. Shigei, and K. Arima. Some quantum search algorithms for arbitrary initial amplitude distribution. In *Sixth International Conference on Natural Computation (ICNC)*, volume 8, pages 603–608, 2010.

M.A. Nielsen and I.L. Chuang. *Quantum Computation and Quantum Information*. Cambridge Univ. Press, 2000.

P. W. Shor. Algorithms for quantum computation: Discrete logarithms and factoring. In *Proceedings of the IEEE Symposium on Foundations of Computer Science*, pages 124–134, 1994.

D.R. Simon. On the power of quantum computation. In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 116 –123, nov 1994.

D. Ventura and T. Martinez. Quantum associative memory. *Information Sciences*, 124(1-4):273–296, 2000.