ELSEVIER

Information Sciences 128 (2000) 271–293

INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL

www.elsevier.com/locate/ins

Quantum associative memory with distributed queries

A.A. Ezhov [a,*], A.V. Nifanova [a], Dan Ventura [b]

[a] *Troitsk Institute of Innovation and Fusion Research, 142092 Troitsk, Moscow Region, Russian Federation*

[b] *Graduate Faculty of Computer Science and Engineering, The Pennsylvania State University, University Park, PA 16802, USA*

Received 8 October 1999; accepted 4 July 2000

Abstract

This paper discusses a model of quantum associative memory which generalizes the completing associative memory proposed by Ventura and Martinez. Similar to this model, our system is based on Grover's well-known algorithm for searching an unsorted quantum database. However, the model presented in this paper suggests the use of a distributed query of general form. It is demonstrated that spurious memories form an unavoidable part of the quantum associative memory model; however, the very presence of these spurious states provides the possibility of organizing a controlled process of data retrieval using a specially formed initial state of the quantum database and also of the transformation performed upon it. Concrete examples illustrating the properties of the proposed model are also presented.

Keywords: Quantum associative memory; Distributed query; Spurious memory

1. Introduction

Development of efficient and/or biologically plausible models of associative memory forms a very important part of neurocomputing research [1]. Among the various approaches to this problem Hopfield's model of content-addressable memory [2], Kosko's bidirectional autoassociative memory (BAM) [3] and

* Corresponding author.

0020-0255/00/$ - see front matter
PII: S 0 0 2 0 - 0 2 5 5 (0 0) 0 0 0 5 7 - 8

Pollack's recursive auto-associative memory (RAAM) [4] are most widely known and comprehensively studied. One of the most important characteristics of neural models of associative memory is capacity. Unfortunately, the capacity of existing neural memories grows only linearly with the number of neurons in a network (bits in a pattern) [1].

Recently, studies in quantum computation have been enriched by new works addressing the idea of developing quantum neural networks [5–8]. These quantum neural networks have many promising characteristics, both in the case of supervised and unsupervised learning. In particular, an associative memory based on the use of Grover's quantum search algorithm [9] has been proposed by Ventura and Martinez [10–12]. This network solves the *completion* problem; that is, it can restore the full pattern when initially presented with just a part of that pattern. One of the most attractive properties of this memory is its exponential capacity.

This paper presents a generalization of this model which is able to retrieve a memory state when presented with a corrupted (noisy) version of the pattern. So, the model presented here solves the problem of associative search for which no part of the input stimulus is guaranteed to be noise free.

Section 2 briefly outlines Grover's algorithm and discusses its main features and possible interpretations. It is argued that this algorithm in some sense already solves the completion problem for the simplest case of a database having a full set of coding fragments in its stored patterns. Section 3 gives a short review of the model of quantum associative memory proposed by Ventura and Martinez and discusses its peculiarities. Section 4 generalizes Grover's algorithm to the case of distributed query and derives an analytical solution for the iterated amplitudes of the basis states whose superposition describes the state of a quantum database. Section 5 describes a generalized quantum associative memory which utilizes Grover's algorithm and compares two distinct approaches to its implementation. The second method, which uses the trick of exchanging stored memories for spurious ones, provides controlled iteration of the system and reaches a quantum state for which the probabilities of retrieving a given memory state take prescribed values defined by the form of the distributed query. Analytical solutions for the iterated state and its different averages are used to analyze these variants of the presented model. The conclusion summarizes the main results presented in the article.

2. Grover's algorithm

This algorithm, proposed by Grover [9], effectively searches for an entry in an unsorted quantum database of N entries, requiring only $\mathrm{O}(\sqrt{N})$ steps to produce the appropriate item with high probability. This compares favorably with the classical lower bound of $N/2$ for the number of queries needed to find an item in an unsorted classical database containing N records. Further, it has been proved that the efficiency of Grover's algorithm is optimal for quantum search [13].

Here, the algorithm is briefly described so that some kind of equivalence between Grover's approach and the quantum associative memory proposed in [10,11] may be established. Consider the database (phone book) containing the four records shown in Table 1. Each record consists of two parts – name and phone number. The records are ordered by name and unordered with respect to the phone numbers. The *direct problem* of searching for the associated phone number given a name is obviously very easy. On the contrary, the *inverse problem* of searching for the associated name given a phone number is difficult. The most efficient classical solution is random search which demands in the worst case $N-1$ queries and on average $N/2$ queries.

This classical random search can be thought of as testing each entry with probability N^{-1}. This situation may be described in a quasi-quantum manner. Before testing every entry, *NameFinder* is in the indefinite state $|\Psi\rangle$, which can be represented as a *superposition* of all possibilities:

$$|\Psi\rangle = \sum_{number=0}^{N-1} \frac{1}{N} |name, number\rangle, \tag{1}$$

Each testing corresponds to the *collapse* of the superposition (1) to one of its items with equal probability N^{-1} (this collapsing just means that *NameFinder* makes a decision to explore a particular entry in the phone book). In a classical statement of the problem, the coefficients of each entry in (1) are real numbers (probabilities) and the entries do not *interfere*. Also the state of *NameFinder* $|\Psi\rangle$ *does not evolve* during the search process.

In a truly quantum approach to the problem, the state of *NameFinder* can be described as a *superposition* of quantum basis states (*wavefunction*)

$$|\Psi\rangle = \sum_{number=0}^{N-1} c_{number} |name, number\rangle, \tag{2}$$

Table 1
The records in a phone book are ordered by names coded using five bits[a]

Name	Code	Number	Code
A	00001	3	11
B	00010	0	00
C	00011	2	10
D	01010	1	01

[a] Phone numbers (unordered) are coded using two bits. In the quantum analog of a phone book, bits are replaced by qubits, and the states of qubits corresponding to names and numbers are entangled.

where c_k are complex numbers, and the probability of the system to be found after the measurement in a given basis state k (probability to choose entry k) is given by $|c_k|^2$. The full database record $|name, number\rangle$ is represented by two parts (sub-records) consisting of sets of qubits which code the name and phone number, respectively. For example, in Table 1 each phone number can be coded by two qubits and each name (one character) by five qubits (sufficient to represent any character in the English alphabet). The states of these two parts of record are *entangled* . This means that if the wavefunction collapses to the basis state having a given phone number value, the measurement of the state of the qubits which encode the name will produce the name corresponding to that phone number.

Typically, the first sub-record of qubits corresponding to the name is omitted and the problem is formulated as the search for the entry with a given (marked) second record $|number\rangle \stackrel{def}{=} |x\rangle$ value known to *Oracle*. In other words, *Oracle* knows the number and asks *NameFinder* to find the corresponding name in his phone book. So in the language of associative memory, *Oracle* presents an external stimulus and *NameFinder* retrieves a memory item. Further, the wavefunction $|\Psi\rangle$ will be referred to as describing the state of *NameFinder* in a *database* or in a *memory*. These two cases are differentiated only for convenience in order to specify correspondingly the search for a record in a database having an enormous number (all 2^d patterns consisting of d qubits) of data items, for which Grover's algorithm was originally developed, and in a memory, which ordinarily includes a rather more restricted number of patterns ($p \ll 2^d$).

The essence of the quantum approach to the search is constructing an iteration scheme consisting of transformations performed by *Oracle* and *NameFinder* which change the *NameFinder* wavefunction in such a manner that the amplitude of the marked state (stimulus) increases and the amplitudes of other states decrease. Then the collapse of *NameFinder's* wave function (choice of entry in the database) will give, with high probability, the desired marked entry. Finally, the entanglement of the states of the two parts of the record will permit the retrieval of the desired associated name. Formally, this algorithm can be described as follows:

- Initialize the state of *NameFinder-in-Database* $|\Psi\rangle = \sum_{x=0}^{2^d-1} a_x|x\rangle$ as an equiprobable superposition $|s\rangle$ of all entries

  $$|\Psi^{(0)}\rangle = |s\rangle = \sum_{x=0}^{2^{d-1}} \frac{1}{\sqrt{2^d}}|x\rangle, \tag{3}$$

  where d denotes the number of qubits used to represent a phone number and $|x\rangle$ denotes a basis state whose binary form corresponds to the phone number $\mathbf{x}$.

- Iterate the state $|\Psi\rangle$ using the sequence of two transformations U_p and U_s:
  1. U_p: *Oracle* inverts the phase of marked state $|p\rangle$

$$a_x \to \begin{cases} -a_x, & \text{if } x = p, \\ a_x, & \text{if } x \neq p. \end{cases} \tag{4}$$

  2. U_s: *NameFinder-in-Database* inverts the amplitudes of all states around their average value

$$a_x \to 2\langle a\rangle - a_x, \tag{5}$$

where

$$\langle a\rangle = 2^{-d} \sum_{x=0}^{2^d} a_x.$$

It has been shown [15] that after $T = \mathrm{O}(\sqrt{N})$ $(N = 2^d)$ iterations the amplitude a_p becomes very close to unity while the amplitudes of the other states almost vanish. At this point, measuring the state of *NameFinder-in-Database* will, with high probability, produce that basis state $|p\rangle$ and the entangled name coded in the remaining qubits can be directly retrieved.

Note that the database considered here is complete: it contains all possible 2^d phone numbers. So, the collapse of the system into any basis state will provide a valid database record. This situation corresponds to the absence of spurious memories in quantum models of associative memory.

3. Completing quantum associative memory

A quantum associative memory with a capacity exponential in the number of qubits and based on Grover's algorithm has been proposed by Ventura and Martinez [10–12]. This kind of memory solves the problem of *pattern completion*. It can restore the full pattern when presented with only a partial one. It is crucial that this partial pattern would exactly coincide with some part of a valid full pattern. Then, in the recall phase only the remainder of the pattern is reconstructed while the initially presented partial pattern remains intact. This kind of associative memory clearly differs from the general statement of associative search. Indeed, general associative memory should also retrieve valid memory items when presented with noisy versions of these patterns. In fact, the original Grover's algorithm also solves the completion problem – in the interpretation described above it retrieves a full pattern (*name, number*) when presented with a partial pattern (*?, number*).

The main difference between the quantum associative memory developed in [10–12] and Grover's search algorithm of an unsorted database is that for the quantum associative memory the number of entries is smaller than 2^d and they form a set M of so-called *memory states*. This is an important distinction for at least two reasons. First, while the creation of a quantum superposition containing all 2^d possible basis states is a straightforward operation [9], the creation of a quantum superposition containing only those basis states that correspond to valid memories is a non-trivial task, the solution to which is detailed in [14]. Second, the superposition of all 2^d basis states is a special case of the general wave function and researchers have to this point had difficulty in discovering a practical use for Grover's algorithm, theoretically spectacular as it is. The quantum associative memory of [10–12] provides perhaps the first such practical application.

Hence in the case of the quantum associative memory, the initial state of *NameFinder-in-Memory* is described as a superposition of these states

$$\left|\Psi^{(0)}\right\rangle = |m\rangle = \sum_{x\in M} \frac{1}{\sqrt{P}}|x\rangle, \tag{6}$$

where P denotes the number of memorized patterns.

Also, the transformation performed by *NameFinder-in-Memory* now suggests the inversion of the amplitudes around the average *of only the amplitudes of memorized patterns* (zero amplitudes of the other ones play no role) [10]:

$2'$. $U_s \Rightarrow U_m$: *NameFinder-in-Memory* inverts amplitudes of all states around the average value for memorized patterns.

$$\begin{aligned} &a_x \to 2\langle a\rangle_M - a_x, \\ &\langle a\rangle_M = \frac{1}{P}\sum_{x\in M} a_x. \end{aligned} \tag{7}$$

(Note, however, that in other versions of the model the original transformation U_s is used [11,12]). Further, it is assumed that *Oracle* knows *a part of one of the memorized patterns* and no other pattern has the same part. Then Grover's algorithm is used to find the pattern having this part and entanglement permits the restoration of the remainder of this pattern.

3.1. Completing vs general associative memory

In general, associative search suggests the possibility of retrieving valid memory items when presented with any possible external stimulus, including noisy stimuli. It is desirable to retrieve the memory state which is most similar to the given stimulus, i.e., that memory state which differs from the stimulus in the minimal number of bits (qubits). In the binary case, this corresponds to minimizing of the Hamming distance between query and memory states. So, this kind of memory is a *correcting* memory rather than a *completing* one. However, the quantum associative memory proposed in [10–12] does not actually take into account the distance between states but only uses information about the presence of some prescribed bit values in the memory state. Moreover, the quantum state describing the simplest form of completing memory [10] will not evolve while performing iterations if the query (part of memory item) presented by *Oracle* does not coincide with some part of a valid memory state. In order to overcome these difficulties, it is possible to introduce a metric into the quantum search algorithm in the form of *distributed queries*. Further, this complication of a query permits a quantum state to evolve in the case of the use of the transformation U_m; however it also leads to the appearance of spurious memories. Hence, while the completing associative memory proposed by Ventura and Martinez *can* be free from spurious memories (and can also contain them in other model variants [11,12]), general correcting associative memory should contain spurious states. Really, the appearance of spurious states in the model proposed in [12] is due to the traditional form of the transformation U_s. In a quantum associative memory of the general form discussed here, spurious memories arise even in the case when the transformation U_m is used. (Tables 2 and 3 illustrate the difference between these two

Table 2
A *completing* associative memory *can* use a database (phone book) free from spurious memories (it contains only some of the possible numbers) and any query must represent an *exact* part of a phone number in the database, for example (*11), (01*), etc.[a]

Record	Name	Phone
1	Alice	010
2	Bob	111

[a] Any query of a form similar to (10*) or (001) etc. will not cause any evolution of the quantum state describing a completing memory if transformation U_m is used.

Table 3
The *general* (correcting) form of associative memory which uses distributed queries suggests that the database (phone book) includes a full set of numbers (8) but some of them are not used, i.e., correspond to spurious memories[a]

Record	Name	Phone
1	Not used	000
2	Not used	001
3	Alice	010
4	Not used	011
5	Not used	100
6	Not used	101
7	Not used	110
8	Bob	111

[a] Collapse of the wave function into one of the basis states corresponding to such a spurious memory does not provide useful information.

situations.) The following discussion will be restricted to considering the case of the use of the transformation U_m in constructing a scheme for a general quantum associative memory. More complex cases will be considered in the future.

Distributed query means a query, or stimulus to the system, has the form of a superposition, just as a quantum memory does,

$$|b^p\rangle = \sum_{x=0}^{2^d-1} b_x^p |x\rangle, \tag{8}$$

which in general includes *all* basis states. (Index p marks one of these states, $|p\rangle$, which plays the role of the *center* of the distribution). However, the introduction of distributed query demands the modification of the memory (phone book) in such a manner that it has *every* possible phone number (basis state), despite the fact that most of them have no corresponding names.

This modification introduces the possibility of the quantum state collapsing into a basis state corresponding to an entry having the code "*not used*" in the *Name* field. In other words, such a memory will have so-called spurious memory states. However, before considering such a generalized associative memory, the application of Grover's algorithm to the case is when *Oracle* defines not a single query (*marked state*) or finite set of such states [15] but rather defines a distributed *fuzzy* query.

First, suppose that in distributed query (8) real amplitudes are distributed such that the maximal value occurs for some definite state $|x\rangle = |p\rangle : |b_p^p|^2 = \max_x |b_x^p|^2$, and the amplitudes of the other basis states decrease monotonically with Hamming distance $|x-p| : |x-p| \uparrow \Rightarrow |b_x^p|^2 \downarrow$. From here on, $|p\rangle$ shall be referred to as the *query center*.

One way to satisfy these conditions follows from the binomial distribution

$$|b_x^p|^2 = q^{|p-x|}(1-q)^{d-|p-x|}, \tag{9}$$

where $|p-x|$ denotes Hamming distance between $|p\rangle$ and $|x\rangle$, d is the number of qubits needed to code a phone number and $0 < q < 1/2$ is an arbitrary value which tunes the width of the distribution. For example, $d = 2$; $q = 1/4$; $|p\rangle = |11\rangle$, produces the following distributed query

$$|b^p\rangle = \frac{3}{4}|11\rangle + \frac{\sqrt{3}}{4}|01\rangle + \frac{\sqrt{3}}{4}|10\rangle + \frac{1}{4}|00\rangle.$$

It is important to note once more that introducing a distributed query with Hamming distance-dependent amplitudes for the basis states incorporates a metric into the model which permits comparison of the similarity of the stimulus and the retrieved memory. This is a necessary condition for associative searching. For this type of query the transformation performed by *Oracle* will have the form

$$U_b = 1 - 2|b^p\rangle\langle b^p|. \tag{10}$$

It is well-known that in the case of the traditional Grover's algorithm the transformation performed by *NameFinder-in-Database* inverts the amplitudes of the basis states around their mean value

$$a_x \to 2\langle a\rangle - a_x. \tag{11}$$

In the case of the simple *completing* associative memory [10], *only memory states* are used in building the transformation performed by a network

$$U_m = 2|m\rangle\langle m| - 1. \tag{12}$$

Correspondingly, this transformation inverts the amplitudes of the memory states (their equiprobable superposition ordinarily forms the initial quantum state) around the average value of *only these memories* (remember, no additional spurious memories arise).

In the case of a distributed query, the *Oracle* transformation is defined by

$$\begin{aligned} &U_b = 1 - 2|b^p\rangle\langle b^p|, \\ &U_b : \Psi \to \Psi - 2|b^p\rangle\langle b^p|\Psi\rangle. \end{aligned} \tag{13}$$

Since

$$\begin{aligned} \langle b^p|\Psi\rangle &= \sum_x b_x^p\langle x|\sum_y a_y|y\rangle = \sum_x b_x^p \sum_y a_y\langle x|y\rangle = \sum_x b_x^p \sum_y a_y\delta_{xy} \\ &= \sum_x a_x b_x^p \equiv \langle a|b^p\rangle, \end{aligned} \tag{14}$$

where δ is the Kronecker delta function and $\langle a \mid b^p\rangle$ represents an overlapping of the current quantum state with the query state,

$$\Psi - 2|b^p\rangle\langle b^p|\Psi\rangle = \sum_x a_x|x\rangle - 2|b^p\rangle\langle a|b^p\rangle = \sum_x \left(a_x - 2\langle a|b^p\rangle b_x^p\right)|x\rangle$$

or

$$a_x \to a_x - 2\langle a \mid b^p\rangle b_x^p. \tag{15}$$

Note, that expression (15) can be transformed into the classical Grover's transformation U_s if $b_x^p = \delta_{xp}$.

In this case $\langle a|b^p\rangle = \sum_x a_x b_x^p = \sum_x a_x\delta_{xp} = a_p$ and (15) takes a familiar form

$$a_x \to a_x - 2a_p\delta_{xp}, \quad \text{i.e.,}\ a_x = \begin{cases} a_x & \text{if}\ \ x \neq p, \\ -a_x & \text{otherwise.} \end{cases}$$

4. Grover's search in unsorted database with distributed query

Before considering the quantum associative memory with distributed queries, a generalization of Grover's original algorithm in the context of distributed queries will be presented. In general, a quantum database can have an arbitrary initial quantum state. In fact, the algorithm proposed in [14] is used to create database states representing quantum memory of arbitrary sets of patterns. How will Grover's original algorithm perform with such quantum states using a distributed query model?

4.1. Deriving the equations for averages

If for some iteration τ the state of the system is described by the superposition

$$\Psi^{(\tau)} = \sum_x a_x^{(\tau)} |x\rangle, \tag{16}$$

then after the transformation U_b (first sub-step of an iteration), the superposition becomes

$$\Psi^{(\tau+1/2)} = \sum_x a_x^{(\tau+1/2)} |x\rangle = \sum_x \left(a_x^{(\tau)} - 2\langle a|b^p\rangle^{(\tau)} b_x^p \right) |x\rangle. \tag{17}$$

After U_s, the transformation performed by the *NameFinder-in-Memory* (second sub-step of an iteration), which inverts amplitudes around their average value,

$$\begin{aligned} \Psi^{(\tau+1)} &= \sum_x \left(2\left\langle a^{(\tau+1/2)} \right\rangle - a_x^{(\tau+1/2)} \right) |x\rangle \\ &= \sum_x \left\{ 2\left\langle a_x^{(\tau)} - 2\langle a|b^p\rangle^{(\tau)} b_x^p \right\rangle - a_x^{(\tau)} + 2\langle a|b^p\rangle^{(\tau)} b_x^p \right\} |x\rangle. \end{aligned} \tag{18}$$

Thus, one iteration causes the following change of amplitudes:

$$a_x^{(\tau+1)} = 2\left\langle a_x^{(\tau)} - 2\langle a|b^p\rangle^{(\tau)} b_x^p \right\rangle - a_x^{(\tau)} + 2\langle a|b^p\rangle^{(\tau)} b_x^p \tag{19}$$

or equivalently,

$$a_x^{(\tau+1)} = 2\langle a\rangle^{(\tau)} - 4\langle a|b^p\rangle^{(\tau)} \langle b^p\rangle - a_x^{(\tau)} + 2\langle a|b^p\rangle^{(\tau)} b_x^p. \tag{20}$$

Note, that Grover's original iteration scheme follows from (20) if

$$\begin{aligned} \langle a \mid b^p\rangle &= \sum_x a_x b_x^p = \sum_x a_x \delta_{xp} = a_p, \\ \langle b^p\rangle &= \frac{1}{2^d} \sum_x b_x^p = \frac{1}{2^d} \sum_x \delta_{xp} = \frac{1}{2^d}. \end{aligned} \tag{21}$$

Then, $a_x^{(\tau+1)} = 2\langle a\rangle^{(\tau)} - 4N^{-1} a_p^{(\tau)} - a_p^{(\tau)} + 2a_p^{(\tau)} \delta_{xp}$,

$$a_x^{(\tau+1)} = \begin{cases} 2\langle a\rangle^{(\tau)} - (1 + 4N^{-1}) a_p^{(\tau)} & \text{if } x \neq p, \\ 2\langle a\rangle^{(\tau)} + (1 - 4N^{-1}) a_p^{(\tau)} & \text{otherwise,} \end{cases} \tag{22}$$

where $N = 2^d$ denotes all possible states for a register consisting of d qubits. The particular case of initial state $|\Psi\rangle = 1/2(|00\rangle + |01\rangle + |10\rangle + |11\rangle)$, means that $\langle a\rangle^{(0)} = 1/2$ and for any p

$$a_p^{(1)} = \begin{cases} 2 \cdot \frac{1}{2} - (1 + 4/4) \cdot \frac{1}{2} = 0 & \text{if } x \neq p, \\ 2 \cdot \frac{1}{2} + (1 - 4/4) \cdot \frac{1}{2} = 1 & \text{otherwise}, \end{cases}$$

and so follows the well-known result that for these conditions any query will transform the state of the system to that of the marked state (find the requested number in the phone book) after only one iteration.

Now, starting from Eq. (20) it is possible to obtain a closed system of two equations for the average values $\langle a\rangle$ and $\langle a \mid b^p\rangle$ (the approach is analogous to the one used in [16]). Multiplying this equation by N^{-1} and adding the terms corresponding to all basis states results in

$$\langle a\rangle^{(\tau+1)} = 2\langle a\rangle^{(\tau)} - 4\langle a|b^p\rangle^{(\tau)}\langle b^p\rangle - \langle a\rangle^{(\tau)} + 2\langle a|b^p\rangle^{(\tau)}\langle b^p\rangle$$

or equivalently,

$$\langle a\rangle^{(\tau+1)} = \langle a\rangle^{(\tau)} - 2\langle a|b^p\rangle^{(\tau)}\langle b^p\rangle. \tag{23}$$

Multiplying each of the Eq. (20) by its corresponding b_x^p value and summing over x produces

$$\langle a|b^p\rangle^{(\tau+1)} = 2\langle a\rangle^{(\tau)}N\langle b^p\rangle - 4\langle a|b^p\rangle^{(\tau)}N\langle b^p\rangle^2 - \langle a|b^p\rangle^{(\tau)} + 2\langle a|b^p\rangle^{(\tau)}\sum_x (b_x^p)^2.$$

Taking into account that vector $\mathbf{b}^p = (b_0^p, \ldots, b_N^p)$ has a unity norm simplifies the last expression as

$$\langle a|b^p\rangle^{(\tau+1)} = 2N\langle b^p\rangle\langle a\rangle^{(\tau)} + (1 - 4N\langle b^p\rangle^2)\langle a|b^p\rangle^{(\tau)}. \tag{24}$$

Finally, a closed system for the averages can be written as follows:

$$\begin{aligned} \langle a\rangle^{(\tau+1)} &= \langle a\rangle^{(\tau)} - 2\langle b^p\rangle\langle a|b^p\rangle^{(\tau)}, \\ \langle a|b^p\rangle^{(\tau+1)} &= 2N\langle b^p\rangle\langle a\rangle^{(\tau)} + (1 - 4N\langle b^p\rangle^2)\langle a|b^p\rangle^{(\tau)}. \end{aligned} \tag{25}$$

In order to reduce this to Grover's original scheme, assign $\langle a|b^p\rangle = a_p$ and $\langle b^p\rangle = N^{-1}$. Then

$$\begin{aligned} \langle a\rangle^{(\tau+1)} &= \langle a\rangle^{(\tau)} - 2a_p^{(\tau)}/N, \\ a_p^{(\tau+1)} &= 2\langle a\rangle^{(\tau)} + (1 - 4/N)a_p^{(\tau)}. \end{aligned} \tag{26}$$

Thus, for the case of Grover's iterations the second equation in (25) is transformed into the equation for the amplitude of the marked state (*Oracle's* query). In what follows, the convenient and more compact notation below will be used to represent the average values of system (25)

$$\alpha^{(\tau)} \stackrel{def}{=} \langle a \rangle^{(\tau)}, \quad \beta^{(\tau)} \stackrel{def}{=} \langle a \mid b^p \rangle^{(\tau)}.$$

4.2. Solving the equations for averages

An analytical solution of the system for average values will be derived now. Rewriting (25) using the notation introduced above results in

$$\begin{aligned} \alpha^{(\tau+1)} &= \alpha^{(\tau)} - 2\langle b^p \rangle \beta^{(\tau)}, \\ \beta^{(\tau+1)} &= 2N\langle b^p \rangle \alpha^{(\tau)} + (1 - 4N\langle b^p \rangle^2)\beta^{(\tau)}. \end{aligned} \tag{27}$$

Rewriting the second equation in (27) as

$$\beta^{(\tau+1)} = 2N\langle b^p \rangle \left\{ \alpha^{(\tau)} - 2\langle b^p \rangle \beta^{(\tau)} \right\} + \beta^{(\tau)} \tag{28}$$

it may be seen that the expression in curly braces is equivalent to the right-hand side of the first equation of (27). Hence,

$$\beta^{(\tau+1)} = 2N\langle b^p \rangle \alpha^{(\tau+1)} + \beta^{(\tau)}. \tag{29}$$

Manipulating the last equation gives

$$\alpha^{(\tau+1)} = \frac{\beta^{(\tau+1)} - \beta^{(\tau)}}{2N\langle b^p \rangle}. \tag{30}$$

Substituting expression (30) into the first equation of system (27) and using (29) and some algebra gives

$$\beta^{(\tau+1)} + \beta^{(\tau-1)} = 2\beta^{(\tau)}(1 - 2N\langle b^p \rangle^2). \tag{31}$$

Suppose that the solution of Eq. (31) is of the form

$$\beta^{(\tau)} = B\cos(\omega\tau + \varphi). \tag{32}$$

Inserting this expression into Eq. (31) and using some trigonometry, an expression for the frequency may be obtained

$$\cos\omega = 1 - 2N\langle b^p \rangle^2 \tag{33}$$

or because $\cos\omega = 1 - 2\sin^2(\omega/2)$,

$$\omega = 2\arcsin(N\langle b^p \rangle). \tag{34}$$

Now using Eq. (30) produces the analytical form of $\alpha^{(\tau)}$

$$\alpha^{(\tau)} = \frac{\beta^{(\tau)} - \beta^{(\tau-1)}}{2N\langle b^p \rangle} = B\frac{\cos(\omega\tau + \varphi) - \cos(\omega(\tau - 1) + \varphi)}{2N\langle b^p \rangle}. \tag{35}$$

The values of the constants B and φ can be found from the initial conditions

$$\alpha^{(0)} = B\frac{\cos\varphi - \cos(\omega - \varphi)}{2N\langle b^p\rangle}, \tag{36}$$

$$\beta^{(0)} = B\cos\varphi. \tag{37}$$

It follows from (36) and (37), that

$$\alpha^{(0)} = \frac{\beta^{(0)}}{\cos\varphi}\frac{\cos\varphi - \cos(\omega - \varphi)}{2N\langle b^p\rangle}, \tag{38}$$

and after some transformation

$$\tan\varphi = \frac{1}{\sin\omega}\left\{1 - \cos\omega - 2N\frac{\alpha^{(0)}}{\beta^{(0)}}\langle b^p\rangle\right\}. \tag{39}$$

The other constant is expressible in terms of the phase value

$$B = \frac{\beta^{(0)}}{\cos\varphi}. \tag{40}$$

Note that the amplitude of every basis state, including those that have a zero amplitude in the initial state, can take non-zero values during the iteration process. This corresponds to the development of spurious memories.

Example 1. Consider an unsorted database with phone numbers encoded with two qubits ($d=2$) and the distributed query

$$|b^p\rangle = \frac{4}{5}|00\rangle + \frac{2}{5}|01\rangle + \frac{2}{5}|10\rangle + \frac{1}{5}|11\rangle.$$

This distribution of basis state amplitudes is visually represented by the histogram on the top right corner of Fig. 1. Grover's iterations should be continued until the average overlap $\beta^{(\tau)} = \langle a|b^p\rangle^{(\tau)}$ reaches one of the values $\{\pm 1\}$. After this it is necessary to perform a measurement and the probability for the system to be found in a given basis state becomes the prescribed function of Hamming distance from this state to the query center. In this example, if the initial state of the database is an equiprobable superposition of all basis states, the value +1 is reached after three iterations.

For the previous example the period of state oscillations can be found using expression (33). Since $N = 4$, and $\langle b^p\rangle = 1/4(0.8 + 0.4 + 0.4 + 0.2) = 0.45$, then

$$\cos\omega = 1 - 8\cdot(0.45\cdot 0.45) \cong -0.6; \quad \omega \cong 0.7\pi; \quad \frac{2\pi}{T} \cong 0.7\pi; \quad T \cong 3.$$

So, it may be concluded that the state of the system oscillates very quickly.

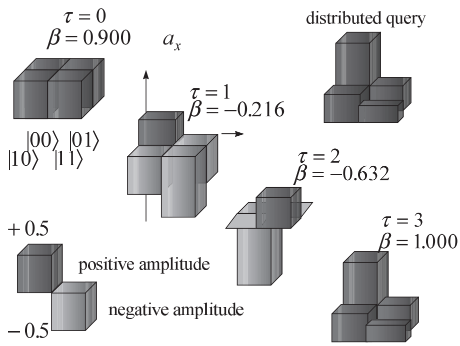

Fig. 1. Right top corner: histogram of query amplitude distribution. Left top corner: initial equally weighted state describing unsorted data base. Histograms of the iterated state amplitudes, with their corresponding β values, are placed along the diagonal. For the third iteration the distribution (lower right corner) of the basis state amplitudes coincides with the distribution of amplitudes in the query (right top corner).

5. Quantum associative memory with distributed query

Now the general form for a quantum associative memory may be considered. Recall that if binary patterns are considered this kind of memory suggests the retrieval of a memory state whose Hamming distance from the presented stimulus is minimal. Also, recall that for the case of memory which includes only a restricted number of memory states it is necessary to change the transformation $U_s = 2|s\rangle\langle s| - 1$ to the transformation $U_m = 2|m\rangle\langle m| - 1$.

Of course, it would be desirable if any quantum state $|\Psi\rangle$ describing our memory would not contain non-zero amplitudes for any basis state not corresponding to one of the memory patterns (the absence of spurious memories). But while this may be possible for a completing associative memory it is impossible in the case of distributed queries because after the transformation performed by *Oracle*

$$a_x \to a_x - 2\langle a|b^p\rangle b^p_x$$

so that in general all amplitudes take non-zero values during the course of the algorithm's iteration.

5.1. Model description

Hence, the state of our system will be described by an arbitrary wave function $|\Psi\rangle$ and the transformation performed by the *NameFinder-in-Memory* will have the form

$$U_m : \Psi \to 2|m\rangle\langle m|\Psi\rangle - |\Psi\rangle. \tag{41}$$

Since

$$\langle m|\Psi\rangle = \frac{1}{\sqrt{P}}\sum_{x\in M}\left(\left\langle x\left|\sum_y a_y\right|y\right\rangle\right) = \frac{1}{\sqrt{P}}\sum_{x\in M}\sum_y a_y\langle x|y\rangle$$
$$= \frac{1}{\sqrt{P}}\sum_{x\in M}\sum_y a_y\delta_{xy} = \frac{1}{\sqrt{P}}\sum_{x\in M} a_x = \sqrt{P}\langle a\rangle_m,$$

where P is the number of patterns in memory and $\langle a\rangle_m = 1/P\sum_{x\in M} a_x$, it can be established that

$$2|m\rangle\langle m|\Psi\rangle - |\Psi\rangle = \sum_{x\in M} 2\langle a\rangle_m|x\rangle - \sum_x a_x|x\rangle,$$

and the *NameFinder-in-Memory* transformation will be defined as

$$a_x \to \begin{cases} 2\langle a\rangle_m - a_x & \text{if } x\in M,\\ -a_x & \text{otherwise.}\end{cases} \tag{42}$$

Hence, the amplitude transformation performed by *NameFinder-in-Memory* will have a form similar to the *NameFinder-in-Database* transformation in Grover's algorithm, but this transformation will be applied only to the basis states corresponding to valid memories. For the rest of the basis states this transformation resembles the *Oracle* transformation (performing phase inversion). Finally, the generalized search algorithm described in the previous section can be adapted to the case in point, taking the following "anti-symmetrical" form:

Oracle transformation:

$$a_x \to a_x - 2\langle a \mid b^p\rangle b_x^p.$$

NameFinder-in-Memory transformation:

$$a_x \to 2\langle a|m\rangle m_x - a_x. \tag{43}$$

Example 2. Consider the case of a memory state containing a single valid pattern $|m\rangle = |01\rangle$ and the distributed query centered on the basis state $|11\rangle$,

$$|b\rangle = \frac{9}{10}|11\rangle + \frac{3}{10}|10\rangle + \frac{3}{10}|01\rangle + \frac{1}{10}|00\rangle.$$

The *Oracle* transformation converts the initial state of the memory $|m\rangle^{(0)} = |01\rangle$ into

$$|m\rangle^{(1/2)} = -0.06|00\rangle + 0.82|01\rangle - 0.18|10\rangle - 0.54|11\rangle,$$

and the *NameFinder-in-Memory* transformation completes a single iteration as

$$|m\rangle^{(1)} = 0.06|00\rangle + 0.82|01\rangle + 0.18|10\rangle + 0.54|11\rangle.$$

Thus, after the first iteration the probability of measuring the system and finding the memory state $|01\rangle$ (which is a Hamming distance of one from the query center $|11\rangle$) takes a value $0.82^2 \approx 0.67$. The probability of collapsing into the spurious state $|11\rangle$ is $0.54^2 \approx 0.29$, and the probabilities for the system to be found in other spurious states are considerably lower.

Example 3. Consider the case of a memory containing two states $|m\rangle = (1/\sqrt{2})|00\rangle + (1/\sqrt{2})|01\rangle$, and suppose that the query has the same form as for the previous example. Let the initial state of our memory be $|a\rangle^{(0)} = |m\rangle$. In this case after one iteration this state will take the form

$$|a\rangle^{(1)} = 0.54|00\rangle + 0.65|01\rangle + 0.17|10\rangle + 0.51|11\rangle.$$

The probability for the system to be found after a measurement in the basis state $|01\rangle$, which is nearest to the query center (in the sense of Hamming distance) takes a value $0.65^2 \approx 0.42$. The probability for the system to be found in the memory state $|00\rangle$, for which query amplitude is minimal is not small: $0.54^2 \approx 0.29$. Also, the probability for the system to be observed in the spurious state $|11\rangle$, which corresponds to the query center, is fairly large. Both these examples demonstrate that the properties of a quantum associative memory with distributed query seem to be reasonable.

In order to transform the expressions obtained earlier for Grover's algorithm with distributed query to the case of a general quantum associative memory for which the transformation *NameFinder-in-Memory* is used, it is necessary to make changes

$$\begin{aligned} &\alpha = \langle a \rangle \Rightarrow \alpha_m = \langle a \mid m \rangle, \\ &\langle b^p \rangle \Rightarrow b_m = \langle b \mid m \rangle, \qquad (44) \\ &2N \Rightarrow 2. \end{aligned}$$

For example , the expressions for the state frequency (33) and (34) take the form

$$\omega = \arccos(1 - 2b_m^2), \qquad (45)$$

$$\omega = 2 \arcsin b_m. \qquad (46)$$

Now using numbers from Example 3 gives $b_m = (0.1 + 0.3)/\sqrt{2} \cong 0.283$ and $\omega = \arccos(1 - 2 \cdot 0.283^2) \cong 0.57$. Therefore $T = (2\pi\ /\omega) \cong 11$, and the memory recall will require 11 iterations of the algorithm to maximize the likelihood of obtaining the correct result (in this case observing the basis state $|01\rangle$).

5.2. Analytical solution for amplitudes

The third example will become clearer after deriving an analytical solution for the amplitudes. From Eq. (43) it may be shown that

$$a_x^{(\tau+1)} = -a_x^{(\tau)} + 2\alpha_m m_{x_x} + 2\beta_m (b_x^p - 2b_m m_x). \tag{47}$$

Taking into account Eq. (44) the following modifications of the expressions for the averages α_m and β_m originally written in expressions (32) and (35) may be obtained:

$$\begin{aligned}\alpha_m &= \frac{B}{2b_m}(\cos(\omega\tau + \varphi) - \cos(\omega(\tau - 1) + \varphi)) \\ &= -\frac{B}{b_m}\sin(\omega\tau + \varphi - \omega/2)\sin(\omega/2),\end{aligned} \tag{48}$$

$$\beta_m = B\cos(\omega\tau + \varphi). \tag{49}$$

Inserting the last expressions into Eq. (47) results in

$$\begin{aligned}a_x^{(\tau+1)} = -a_x^{(\tau)} + 2B\bigg((b_x^p - 2b_m m_x)\cos(\omega\tau + \varphi) \\ - \frac{m_x}{b_m}\sin(\omega/2)\sin(\omega\tau + \varphi - \omega/2)\bigg).\end{aligned} \tag{50}$$

Suppose the solution of the last equation can be cast in the form

$$a_x = A_x \cos(\omega\tau + \delta_x). \tag{51}$$

Inserting (51) in (50) it is possible to derive

$$\begin{aligned}&A_x(\cos(\omega\tau + \omega + \delta_x) + \cos(\omega\tau + \delta_x)) \\ &\quad \equiv 2A_x \cos(\omega/2)\cos(\omega\tau + \delta_x + \omega/2) \\ &\quad = 2B(b_x^p - 2b_m m_x)\cos(\omega\tau + \varphi) - \frac{2Bm_x}{b_m}\sin(\omega/2)\sin(\omega\tau + \varphi - \omega/2) \\ &\quad \equiv 2B(f_x \cdot \cos(\omega\tau + \varphi) + g_x \cdot \sin(\omega\tau + \varphi)),\end{aligned} \tag{52}$$

where

$$f_x \overset{def}{=} \frac{m_x}{b_m}\sin^2(\omega/2) + b_x^p - 2b_m m_x \equiv b_x^p - m_x \sin(\omega/2), \tag{53}$$

$$g_x \overset{def}{=} \frac{m_x}{2b_m}\sin\omega. \tag{54}$$

Introducing a variable

$$\zeta_x = \arccos\left(\frac{f_x}{\sqrt{f_x^2 + g_x^2}}\right). \tag{55}$$

Eq. (52) can be transformed to

$$A_x \cos(\omega/2)\cos(\omega\tau + \delta_x + \omega/2) = B\sqrt{f_x^2 + g_x^2}\cos(\omega\tau + \varphi + \zeta_x). \tag{56}$$

From the last equation it follows immediately that the expressions for coefficients A_x and phases δ_x are

$$A_x = \frac{B\sqrt{f_x^2 + g_x^2}}{\cos(\omega/2)}, \tag{57}$$

$$\delta_x = \zeta_x + \varphi - \omega/2. \tag{58}$$

Expressions (51,57,58) give the analytical form of the amplitudes of the basis states for a general quantum associative memory based on the use of Grover's algorithm with distributed query. Fig. 2 shows the analytical form of the amplitudes for the memory defined in Example 3.

It can be seen in Fig. 2 that the state $|01\rangle$ has a phase delay compared to the other valid memory state, $|00\rangle$. The observed delay is connected with the

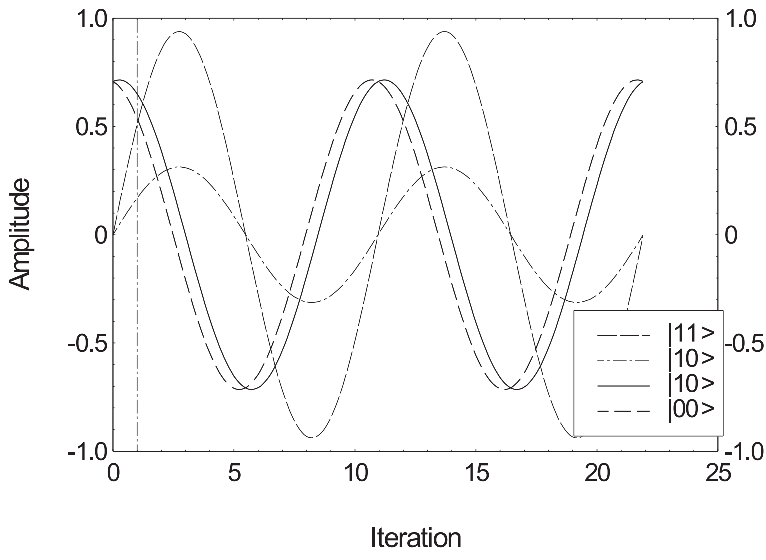

Fig. 2. Dependence of basis state amplitudes on Grover's iterations for the memory defined in Example 3.

greater amplitude which basis state $|01\rangle$ has in the query. The change of phase δ_x due to the change of basis state amplitude b_x^p in a query is dependent only on the variation of ζ_x; considering the derivative and applying some algebra

$$\frac{\partial}{\partial b_x^p}\cos\zeta_x = \frac{\partial}{\partial b_x^p}\left(\frac{b_x^p - m_x b_m}{\sqrt{(b_x^p)^2 - 2b_m b_x^p m_x + m_x^2}}\right)$$

$$= \frac{1}{\sqrt{(b_x^p)^2 - 2b_m b_x^p m_x + m_x^2}} - \frac{(b_x^p - m_x b_m)^2}{((b_x^p)^2 - 2b_m b_x^p m_x + m_x^2)^{3/2}}$$

$$\equiv \frac{\sin^2\zeta_x}{\sqrt{(b_x^p)^2 - 2b_m b_x^p m_x + m_x^2}} \geqslant 0.$$

Hence,

$$-\sin\zeta_x \frac{\partial\zeta_x}{\partial b_x^p} = -\frac{m_x \sin\omega}{2b_m}\cdot\frac{\zeta_x}{\partial b_x^p} \geqslant 0,$$

or

$$\frac{\partial\zeta_x}{\partial b_x^p} \leqslant 0.$$

Thus, the state $|01\rangle$ which has a greater amplitude in the query than does the state $|00\rangle$ will also have a lower phase value ζ_x, and consequently, lower value of δ_x. In query formation, amplitudes of basis states monotonously decrease with Hamming distance from the query center; *therefore, the memory state nearest to this center will have maximal amplitude and consequently, minimal phase value*. It may also be seen in Fig. 2 that those basis states which do not belong to the set of valid memories *(spurious memories) all have the same phase value*.

As will be seen shortly, this fact is very important, and it can be verified using expression (55). Indeed, for any spurious memory, from (53) to (55) it follows that $m_x \equiv 0 \Rightarrow g_x \equiv 0 \Rightarrow \cos\zeta_x \equiv 1 \Rightarrow \zeta_x \equiv 0$. Then, from expression (58) the fact that $\delta_x = \varphi - \omega/2 = \mathrm{const}$ may be derived.

Consider, once again, Fig. 2. Maximal amplitude belongs to the state $|11\rangle$ for which the amplitude of the query is also maximal. On the other hand, the state $|00\rangle$, has both minimal amplitude and minimal query amplitude. However, in general, this relation is not valid. Indeed, the dependence of the basis state amplitude on query amplitude

$$A_x = \frac{B\sqrt{f_x^2 + g_x^2}}{\cos(\omega/2)}$$

can be rewritten using explicit expressions for the parameters f_x and g_x given in (53) and (54) as follows:

$$A_x = \frac{B}{\cos(\omega/2)}\sqrt{(b_x^p)^2 - 2b_m m_x b_x^p + m_x^2}.$$

The expression on the right-hand side is not a monotonic function of b_x^p and takes a minimal value on an internal point on the interval [0,1]. Therefore, basis states with lower query amplitude can have greater amplitude in memory.

Note, that in principle, in a distributed query some amplitudes can have zero values. Then, corresponding basis states will never arise as spurious memories. Thus, spurious memories are generated by the query itself demonstrating the principle: *arise if suggested.*

The evolution of the amplitudes of memory states is rather complicated. In general there exists neither *a priori* knowledge about the structure of memory nor about the correspondence between the location of memories and the query center in configuration space. It is therefore difficult to obtain an analytical expression for the number of iterations needed to reach the maximal values of amplitudes of memories in the vicinity of this center. It is clear that the difficulty in deriving the necessary estimate is connected with the probabilistic character of the parameter b_m. But the situation can be improved considerably by changing the way the memory is structured, taking advantage of the fact that oscillating spurious memories all have the same phase value.

5.3. Memories become easily retrieved when they become spurious

The trick is simply to exchange the valid states to be memorized with other (spurious) states and vice versa. Namely, create a memory whose initial state is

$$|\Psi^{(0)}\rangle = |\tilde{m}\rangle = \sum_{x \notin M} \frac{1}{\sqrt{P}}|x\rangle$$

and correspondingly, the transformation performed by this memory is defined as

$$U_{\tilde{m}} : \Psi = 2|\tilde{m}\rangle\langle\tilde{m}|\Psi\rangle - |\Psi\rangle.$$

In some sense this kind of memory is ideologically similar to the immune system, which includes antibodies corresponding to antigens that *do not belong* to the host organism and has almost no antibodies to its own proteins. Analogically, the associative memory is formed by memorizing patterns which *should not be recalled.*

In this system, all valid memories should be treated as spurious ones. All these memories will be in phase with a phase value of $-\pi/2$ and have initial amplitudes of zero. Therefore, the amplitudes of these states will evolve according to

$$a_x^\tau = A_x \sin\omega\tau,$$

and because for these states $m_x = 0$,

$$A_x = \frac{B}{\cos(\omega/2)} \sqrt{(b_x^p)^2 - 2b_m m_x b_x^p + m_x^2} = \frac{Bb_x^p}{\cos(\omega/2)}$$

based upon the query amplitudes. Applying this trick to Example 3, the initial state of memory is

$$b_m = 2^{-d/2} \sum_{x \notin M} b_x^p \cong \tilde{b}_m = 2^{-d/2} \sum_{x=0}^{2^d - 1} b_x^p,$$

$$|\tilde{m}\rangle = \frac{1}{\sqrt{2}}|10\rangle + \frac{1}{\sqrt{2}}|11\rangle.$$

Then, $\langle b|m\rangle = (0.3 + 0.9)/\sqrt{2} \cong 0.85$ and $\omega \cong 2.03$, and now the states of the system will oscillate with considerably lower (compared to the original 11) period $T = 2\pi/\omega \cong 3.1$.

Since the amplitudes of all spurious memories achieve their maximal values at $\tau = T/4$, the system will be in a state for which the amplitudes of these "spurious" memories become proportional to their amplitudes in the query for $\tau \cong 0.77$. Of course, the number of iterations must be integer valued so it is necessary to add some periods for $\tau + nT$ to be as near as possible to an integer value. In this case, for $n = 2, 0.77 + 6.2 = 6.97 \approx 7$, so it will suffice to perform seven iterations of the algorithm before performing a measurement. After these iterations the state of system will be

$$|\Psi\rangle = 0.19|00\rangle + 0.57|01\rangle + 0.6|10\rangle + 0.53|11\rangle.$$

It is evident that the amplitudes of the spurious memories (really the valid memories) in this state are in the same proportion as in the query (a ratio of 1/3). Nevertheless, in general it seems that the difficulty inherent in the previous scheme still remains: a dependence of the frequency ω on the set of valid memory patterns.

However, taking into account that now the number of "valid" patterns is very near 2^d (because $p \ll 2^d$), $b_m = 2^{-d/2} \sum_{x \notin M} b_x^p \cong \tilde{b}_m = 2^{-d/2} \sum_{x=0}^{2^d-1} b_x^p$. Therefore, in contrast to the more intuitive approach (memorizing the valid patterns) the value of b_m can be approximated as *a priori* knowledge, dependent only on the form of the query, and not on the form of the set of memory states. Therefore, using this approximated value of b_m and the corresponding approximation of the frequency $\tilde{\omega} = 2 \arcsin \tilde{b}_m$, the number of iterations $T_{\max}$ needed to transform the system into a state such that the amplitudes of spurious (actually valid) memories become maximal and proportional to their amplitudes in the distributed query is

$$\tilde{\omega} T_{\max} = \frac{\pi}{2} \Rightarrow T_{\max} \approx \frac{\pi}{2\tilde{\omega}}.$$

5.4. A note on initializing quantum states

It has been recently pointed out that there is potentially an inherent problem with current quantum computational algorithms [17]. Quantum computers and quantum algorithms rely heavily on the phase information of quantum states – if the relative phases of the various states in a system are not correct, the computation will not work. Kak discusses the fact that quantum systems can possess random initial phases, whereas quantum algorithms implicitly assume some known initial phase conditions from which the computation begin. The consensus seems to be that it is possible that this initial variability in the state phases may be compensated for by quantum error correction schemes [18,19]. However, these schemes may also be flawed. Classical error correction is based upon the fact that errors in classical systems are discrete – a bit is flipped with some small probability. However, because quantum computational systems contain phase information, they are susceptible to a continuum of possible errors, and quantum error correction schemes developed to date address only a small number of special cases. Therefore, the issue to be resolved is whether or not in practice (that is in constructing a quantum computer) we will encounter mostly those few cases of error which have been treated in the literature or we will see many other possibilities that Kak points out.

6. Conclusion

A model of quantum associative memory which is able to retrieve memory states with probability proportional to the amplitudes these states have in a query has been presented. This quantum memory can retrieve valid stored patterns from arbitrary stimuli represented by a distributed query of general form (fuzzy stimulus). Further investigation of the model is needed to estimate its other possible merits and limitations.

Acknowledgements

We are grateful to our colleagues, V.R. Chechetkin, V.A. Mishchenko, A.G. Khromov and A.G. Tokaev for fruitful discussions at the Seminar on Quantum Neural Networks held at the Troitsk Institute of Innovation and Fusion Research.

References

[1] E. Domany, J.L. van Hemmen, K. Schulten (Eds.), Models of Neural Networks, Springer, Berlin, 1991.

[2] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proc. Nat. Acad. Sci. USA 79 (1982) 2444–2558.
[3] B. Kosko, Bidirectional associative memories, IEEE Trans. Syst., Man, Cybern. 18 (1988) 49–60.
[4] J.B. Pollack, Recursive distributed representations, Arti. Intell. 46 (1990) 77–105.
[5] M. Perus, Neuro-quantum parallelism in brain-mind and computers, Informatica 20 (1996) 173–183.
[6] T. Menneer, A. Narayanan, Quantum-inspired neural networks, technical report R329, Department of Computer Science, University of Exeter, Exeter, United Kingdom, 1995.
[7] D. Ventura, T. Martinez, An artificial neuron with quantum mechanical properties, in: Proceeding of the International Conference on Artificial Neural Networks and Genetics Algorithms, April 1997, pp. 482–485.
[8] R.L. Chrisley, Learning in non-superpositional quantum neurocomputers, in: P. Pylkkanen, P. Pylkko (Eds.), Brain, Mind and Physics, IOS Press, Amsterdam, 1997, pp. 123–139.
[9] L. Grover, A fast quantum mechanical algorithm for database search, in: Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, ACM, New York, 1996, 7pp. 212–219.
[10] D. Ventura, T. Martinez, A quantum associative memory based on Grover's algorithm, in: Proceedings of the International Conference on Artificial Neural Networks and Genetics Algorithms, April 1999, pp. 22–27.
[11] D. Ventura, T. Martinez, Quantum associative memory with exponential capacity, in: Proceedings of the International Joint Conference on Neural Networks, May 1998, pp. 509–513.
[12] D. Ventura, T. Martinez, Quantum associative memory, Inf. Sci. 124 (1–4) (2000) 273–296.
[13] C.H. Bennett, E. Bernstein, C. Brassard, U. Vazirani, Strength and weakness of quantum computing, SIAM J. Comput. 26 (5) (1997) 1510–1523.
[14] D. Ventura, T. Martinez, Initializing the amplitude distribution of a quantum state, Found. Phys. Lett. 12 (6) (1999) 547–559.
[15] M. Boyer, C. Brassard, P. Hoyer, A. Tapp, Tight bounds on quantum searching, Fortschritte der Physik 46 (4&5) (1998) 493–505.
[16] E. Biham, O. Biham, D. Biron et al., Exact solution of Grover's quantum search algorithm for arbitrary initial amplitude distribution, in: Proceedings of the First NASA International Conference on Quantum Computation and Quantum Communications, 1998, pp. 140–147.
[17] S. Kak, The initialization problem in quantum computing, Found. Phys. 29 (1999) 267–279.
[18] A.R. Calderbank, P.W. Shor, Good quantum error-correcting codes exist, Phys. Rev. A 54 (2) (1996) 1098–1106.
[19] J. Preskill, Fault-Tolerant Quantum Computation, in: H.-K. Lo, S. Popescu, T.P. Spiller (Eds.), Introduction to Quantum Computation, World Scientific, 1998 (to appear).