International Journal of Theoretical Physics, Vol. 42, No. 9, September 2003 (© 2003)

Quantum Algorithms in Hilbert Database

Michail Zak[1,2]

Received August 20, 2002

A fast quantum algorithm for a search and pattern recognition in a Hilbert space memory structure is proposed. All the memory information is mapped onto a unitary operator acting upon a quantum state which represents a piece of information to be retrieved. As a result of only one quantum measurement, the address of the required information encoded in a number of the corresponding row of the unitary matrix is determined. By combining direct and dot products, the dimensionality of the memory space can be made exponentially large, using only linear resources. However, since the preprocessing, i.e., mapping the memory information into a Hilbert space can appear to be exponentially expensive, the proposed algorithm will be effective for NASA applications when the preprocessing is implemented on the ground, while the memory search is performed on remote objects.

KEY WORDS: quantum algorithms; Hilbert database.

1. INTRODUCTION

Recent advances in quantum information theory have inspired an explosion of interest in new quantum algorithms for solving hard computational problems. They include a special class of so-called NP-complete problems which are considered to be intractable by most of the theoretical computer scientists. One of the oldest (and still unsolved) problems of this class is a search problem: find one item in an unsorted database (for instance, find the name that matches a telephone number in a telephone book). As for any of NP-complete problems, here the algorithm for solution is very simple: try each item and compare with the sought one. Classically, this would require an average $0(N)$ queries to the database if N is the number of items. Therefore, for exponentially large N, the problem becomes intractable. In terms of quantum computing, finding the item in the database corresponds to measuring the system and having it collapse to the state which represents that

[1] Ultracomputing Group, Jet Propulsion Laboratory, California Institute of Technology, California; e-mail: zak@solstice.jpl.nasa.gov.

[2] Oak Ridge National Laboratory, Center for Engineering Science Advanced Research, Oak Ridge, Tennessee.

0020-7748/03/0900-2061/0 © 2003 Plenum Publishing Corporation

item. As shown by Grover (1996), the quantum algorithm requires only $0(\sqrt{N})$ queries, but its complexity is still exponential.

Further progress in reducing the search complexity can be associated with incorporating a structure into databases. Indeed, unsorted databases have limited practical importance and they usually serve as benchmarks for rating the problem's complexity. In this note we will be concerned with memory databases whose structures can be predesigned to make anticipated information processing as effective as possible.

2. QUANTUM SEARCH

Suppose that a memory is organized as a normalized relational database, i.e., it is stored in a set of N N-dimensional real orthonormal vectors:

$$U_i = U_i^{(1)}, U_i^{(2)}, \ldots, U_i^N; \quad i = 1, 2, \ldots, N \tag{1}$$

$$U_i \bullet U_j = \delta_{ij} \tag{2}$$

The components of these vectors form a unitary matrix

$$U = \begin{pmatrix} U_{11} & \cdots & U_{1N} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ U_{N1} & \cdots & U_{NN} \end{pmatrix}, \quad U^{-1} = U^T \tag{3}$$

which is characterized by $N(N-1)/2$ independent parameters. Each row of the matrix (3) corresponds to a vector from the set (1), and therefore, the number of that row can be associated with the vector address.

Now the problem can be posed as follows: given a vector U_k as a memory state from the set (1), find its address, i.e., find the corresponding number of the row in the matrix (3).

It turns out that utilizing quantum parallelism, this problem can be solved in one computational step.

Indeed, let us build a quantum system whose Hamiltonian is derived from the unitary matrix (3) as the evolutionary operator, i.e.,

$$U(t_*) = e^{-iHt\cdot|h}, \quad i = \sqrt{-1}, h = 1.054 \times 10^{-34}\,\mathrm{j\,s} \tag{4}$$

for a fixed time interval t_*.

If the operator (4) is applied to a quantum state $|\psi_k(0)\rangle$ represented by a unit vector $U_k(0)$ then the new quantum state at $t = t_*$ will be

$$|\psi_k(t_*)\rangle = U|_\psi(0)\rangle \tag{5}$$

The new unit vector $U_k(t_*)$ has the following components:

$$U_k(t_*) = \begin{pmatrix} U_{11} & \cdots & U_{1N\cdots} \\ \cdots & \cdots & \cdots \\ \cdots & \cdots & \cdots \\ U_{N1} & \cdots & U_{NN} \end{pmatrix} \begin{pmatrix} U_{k1} \\ \vdots \\ \\ \vdots \\ U_{KN} \end{pmatrix} = \begin{pmatrix} U_1 \bullet U_k \\ \cdots \\ U_k \bullet U_k \\ \cdots \\ U_n \bullet U_k \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{pmatrix} \tag{6}$$

and therefore, it is in eigenstate which corresponds to its address. This address can be determined by one quantum measurement.

Thus, because of quantum parallelism, all the queries are performed simultaneously.

3. PATTERNS CLASSIFICATION

The same procedure described by Eq. (6) can be reinterpreted as pattern recognition and classification. Indeed, let us assume that $|\psi_k(0)\rangle$ represents an unknown vector $U_k(0)$, and one wishes to find out what is the closest vector among the set (1) to it. Then the vector (6) gives the closest address of $U_k(0)$. Actually the quantum evolution (5) and the measurement (6) represent a dynamical system with N static attractors. However unlike recurrent neural nets, the basins of attractors are not defined in a deterministic way. To show that, let us assume that

$$U_i \bullet U_k = \varepsilon_{ik}, \quad i = 1, 2, \ldots, N, \sum_{i=1}^{N} \varepsilon_{ik}^2 = 1 \tag{7}$$

where ε_{ik} characterizes the distance between the input vector U_k and the vectors of the set (1). Then the vector U_q has the highest probability ε_{qk}^2 to become an attractor

$$\varepsilon_{qk} = \max_{i \neq q}(\varepsilon_{ik}) \tag{8}$$

However, with a smaller probability $\varepsilon_{jk}^2 < \varepsilon_{qk}^2 (i \neq q)$, any vector $U_j \neq U_q$ can become the attractor. Moreover, with the probability

$$\varepsilon_{ik}^2 = 1 - \varepsilon_{iq}^2, \quad i = 1, 2, \ldots, N \neq q \tag{9}$$

the vector U_q will not be the attractor.

To remove such an uncertainty, one has to repeat the algorithm (5), (6) several times. Indeed, considering each quantum simulation as a Bernoulli trial in which the probability of the correct answer is f, the answer is incorrect if

$$P = 1 - f \tag{10}$$

Then the probability of the correct answer after M simulations is

$$f_m = 1 - (1 - f)^M = 1 - \varepsilon_{ik}^{2M} \to 1 \, at \, M \to \infty \tag{11}$$

Hence, eventually, the vector U_q becomes a clear winner. As in recurrent neural nets, here a static attractor represents an abstraction of the pattern which is free of insignificant features such as ε_{ik} (see Eq. (7)). The role of synaptic interconnections are played by the components U_{ij} of the unitary matrix (3) which as "wirelessly" implemented via quantum physics; the dynamical evolution is governed by the Schrödinger Eq. (5), and the sigmoid function is implemented by the measurement operator in (5). An easy reconfigurability of the "net" with change of the "synaptic interconnections" U_{ij} (but without change of the hardware) will be demonstrated later.

4. EXPONENTIAL CAPACITY

For an effective implementation of the evolutionary operator U, one has to impose some constraints upon the orthonormal vectors (1), namely they must be representable in the direct product form:

$$U_i = |\psi^{(i)}\rangle = \left|\psi_1^{(i)}\right\rangle \otimes \left|\psi_2^{(i)}\right\rangle \otimes \cdots \otimes \left|\psi_n^{(i)}\right\rangle \tag{12}$$

where each multiplier is a superposition of two qubits:

$$\left|\psi_1^{(i)}\right\rangle = \omega'_{1i} \begin{pmatrix} 1 \\ 0 \end{pmatrix} + \omega_{1i} \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} \omega'_{1i} \\ \omega''_{1i} \end{pmatrix} \tag{13}$$

Then the evolutionary operator U has the form:

$$U = U^{(1)} \otimes U^{(2)} \otimes \cdots \otimes U^{(n)} \tag{14}$$

where $U^{(i)}$ are the 2×2 unitary operators acting on the corresponding single qubits (13).

Equation (14) can be generalized by including dot products and identity operators, for instance

$$U = \left(U_1^{(1)} \otimes U_1^{(2)} \otimes \cdots \otimes U_1^{(n)}\right) \bullet \left(U_2^{(1)} \otimes \cdots \otimes U_2^{(n)}\right) \bullet \cdots \bullet \left(U_n^{(1)} \otimes \cdots \otimes U_n^{(n)}\right) \tag{15}$$

or

$$U = \left(I \otimes U^{(3)}\right) \bullet U^{(2)} \bullet \left(U^{(1)} \otimes I\right) \tag{16}$$

In terms of circuit representation, Eq. 16 can be implemented as shown in Fig. 1. Here the role of the identity operator I is to introduce a 2-qubit logic gate known as a sufficient tool for universal quantum computations.

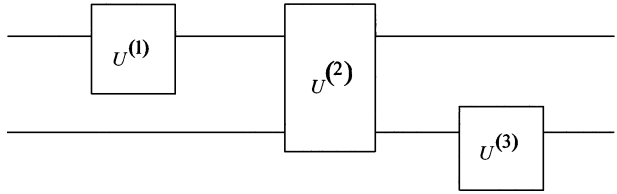

Fig. 1.

Let us return to Eq. (14). The hardware "resources" can be characterized by the number of matrix elements to be implemented

$$r = 4n \tag{17}$$

The information capacity can be characterized by the number of the matrix element in the total direct product in (14):

$$c = 2^n = 2^{r/4} \tag{18}$$

As follows from (18), the information capacity grows exponentially with linear growth of the resources.

5. RECONFIGURABILITY

Let us represent a 2×2 unitary matrix U_p in (14) in the form

$$U_p = \nu_p \begin{pmatrix} \lambda_1^{(p)} & 0 \\ 0 & \lambda_2^{(p)} \end{pmatrix} \nu_p^{-1}, \quad \lambda_j^{(p)} = e^{i\varphi_j^{(p)} t_p^*/\hbar} \tag{19}$$

where $\varphi_j^{(p)}$ are the eigenvalues of the corresponding Hamiltonian, and t^* is the duration of the computation.

As follows from Eq. (19), each componenet of the matrix U_p is a sum of two periodic functions with the periods

$$T_j^{(p)} = \frac{2\pi\hbar}{\varphi_j^{(p)}}, \quad j = 1, 2; \tag{20}$$

The number of different values which $\lambda_j^{(p)}$ takes during the corresponding period $T_j^{(p)}$ is

$$\ell_j^{(p)} = \frac{T_j^{(p)}}{t_p^*} = \frac{2\pi\hbar}{t_p^* \varphi_j^{(p)}} \tag{21}$$

where t_p^* is the duration of the computational step.

If $\varphi_1^{(p)}$ and $\varphi_2^{(p)}$ do not commensurate (for instance, if they are relatively prime numbers), then the number of different configurations which the total matrix U_p can take is

$$\ell^{(p)} = \ell_1^{(p)}\ell_2^{(p)} = \frac{4\pi^2\hbar^2}{t^{*2}\varphi_1^{(p)}\varphi_2^{(p)}} \tag{22}$$

Since all the unitary matrices U_p in the direct product (14) are independent, the corresponding time-steps t_p^* can be chosen arbitrarily, and therefore, the number of different configurations of the product (14) will be

$$\ell = \prod_{p=1}^{n} \ell^{(p)} = \frac{\hbar^{2n}(2\pi)^{2n}}{\prod_{p=1}^{n} t_p^{*2}\varphi_1^{(p)}\varphi_2^{(p)}} \tag{23}$$

Let us evaluate this number. As follows from the time–energy uncertainty relationship:

$$\Delta E\,\Delta t \geq \hbar \tag{24}$$

the upper bound of (23) is

$$\ell \leq (2\pi)^{2n} = (2\pi)^{r/2} \tag{25}$$

since

$$\varphi_i^{(p)} t_p^* \geq \hbar \tag{26}$$

Thus, Eq. (25) expresses the maximum number of different unitary matrix configurations which can be achieved with the same hardware, i.e., only by changing the values of the computation times τ_p, and each configuration is encoded by n numbers $t_1^*, t_2^*, \ldots, t_n^*$.

6. EXTENSION TO COMPLEX DOMAIN

So far we have not exploited the fact that each element of a unitary matrix is characterized by two numbers: the real and the imaginary. Here we will show how one can double the information capacity of the proposed quantum device.

Consider a set of n n-dimensional vectors

$$S_i = S_1, S_2, \ldots, S_n; \quad i = 1, 2, \ldots, n \tag{27}$$

which are not necessarily unit or orthogonal. Such vectors form an affine matrix with elements S_{ij}. Let us map this matrix onto the following Hermitian matrix:

$$S = \begin{pmatrix} S_{11} & S_{12} & \cdots & S_{1n} \\ S_{21} & S_{22} & \cdots & S_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ S_{n1} & S_{2n} & \cdots & S_{nn} \end{pmatrix} \rightarrow \begin{pmatrix} H_{11,} & H_{12} + iH_{21} & \cdots & H_{1n} + iH_{n1} \\ H_{12} - iH_{21}, & H_{22}, & \cdots & H_{2n} - iH_{n2} \\ \cdots & \cdots & \cdots & \cdots \\ H_{1n} - iH_{n1}, & \cdots & \cdots & H_{nn} \end{pmatrix} = H \cdots \tag{28}$$

which we associate with the underlying Hamiltonian. Then the corresponding unitary matrix (3) will be uniquely defined by Eq. (4)

$$U = e^{-iHt\cdot/\hbar} \tag{29}$$

All the three matrices (28) and (29) have N^2 independent parameters since the matrix S can be uniquely mapped onto the unitary matrix U in Eq. (29).

7. CONCLUSION

There are three basic properties of the proposed algorithm which distinguishes it from existing quantum algorithms.

Firstly, in contradistinction to the Grover (1996) or Shor (1997) algorithms, it is fully deterministic: the answer results from only one measurement.

Secondly, the answer is stored not in a quantum state, but rather in the evolutionary operator while the quantum state represents the "question."

Thirdly, the algorithm itself is very simple, however the complexity of the preprocessing, in general, may occur to be exponential. Indeed, although a normalized relational database is a preferred objective (regardless of quantum implementation), it is not always easy to organize memory that way.

It should be noticed that the direct and dot product decomposability of the evolutionary operator expressed by Eqs. (9 and 10) presents another restriction imposed upon the database; without this restriction, an arbitrary Hamiltonian could not be built using linear resources.

The last property suggests that the proposed algorithm will be effective for such applications in which the available resources for the preprocessing stage is much bigger than those for the stage of performance. This requirement perfectly fits NASA applications when the preprocessing, i.e., organizing memory structure is implemented on the ground and can take months or years, while the memory search is performed in seconds on remote objects (spacecrafts, robots) where the resources are extremely limited.

ACKNOWLEDGMENTS

The research described in this paper was performed by the Jet Propulsion Laboratory, California Institute of Technology. This research was performed in association with the Center for Engineering Science Advanced Research 9CESAR), Oak Ridge National Laboratory. Funding was provided by the Engineering Research Program of the DOE Office of Science, under contract DE-AC05-00Or22725 with UT-Battelle LLC.

REFERENCES

Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. In *Proceedings of the 23rd Annual ACM Symposium on the Theory of Computing ACM*, New York, pp. 212–219.

Shor, P. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal of Computation*, **26**(5), 1473–1483.