

Rapid Quantum Search Algorithm

Yehuda Roth

Oranim Academic College, K. Tivon, Israel

Email: yudroth@gmail.com

Received April 9, 2013; revised May 16, 2013; accepted June 12, 2013

Copyright © 2013 Yehuda Roth. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

ABSTRACT

The collapse phenomenon, the parallelism principle and states correlation are used to define a type of a Grover rapid search engine. In our approach, the observer's query and the Grover-unsorted-data are stored in different memories where the global state is represented by a tensor product of the associated states. In the proposed formalism, each **query-state** input activates an adjusted operator that implements the unsorted state in an appropriate 2-D Grover representation. It will be shown that once the representation is set, it takes mainly two operations to complete the whole query search. This seems to be a very efficient search algorithm.

Keywords: Grover Algorithm; Parallel Space

1. Introduction

Quantum coherence, together with the superposition principle, gives rise to the parallelism concept for which processing a single state is like acting simultaneously on all states that participate in the superposition [1,2]. It is also known that a key role in speeding up quantum algorithms is played by multi-particle entanglement [3]. These entanglement and parallelism concepts enable quantum algorithms such as Shor's factoring, which provide options for very fast computers [4].

In addition, quantum superposition of coherence qubits has the advantage of maintaining enormous databases by a single state. It is shown that this superposition of qubits can be applied to an efficient database-finding algorithm and in particular the Grover algorithm that defines a possibility for an efficient search engine [5-7]. The advantage of using quantum memory is shown in [8-10] with the definition of a model for which binary patterns of n-bits are stored in the quantum superposition of the appropriate subset of the computational basis of n-qubits.

The grover algorithm divides the Hilbert space into two segments: the requested query (denoted by the state $|k\rangle$) and all the other records $\left(|\neq k\rangle = \sum_{i \neq k} |i\rangle\right)$. The initial state is a maximal unsorted state $\frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle$ while the Grover Oracle is a unitary transformation (logical gate)

that rotates the unsorted state until after a relatively small number of iteration ($\sim \sqrt{N}$). The probability of detecting the requested query is almost one. Then, a measurement terminates a successful search.

The selection of the 2-D basis $|k\rangle$ and $|\neq k\rangle$ yields that each search has to be associated with an exclusive 2-D representation such that preparing the appropriate representation corresponds with a primary knowledge about the searched state. In that sense, the Grover algorithm is not a pure search algorithm. Actually, it's more likely an "inverting a function" meaning that if we have a function $y = f(x)$ that can be evaluated on a quantum computer, this algorithm allows us to calculate x when given y [11]. If we consider the $|\neq k\rangle$ and $|k\rangle$ as vectors, the algorithm corresponds with rotating the $|\neq k\rangle$ state with respect to the $|k\rangle$ state. This is an equivalent way to represent the constrain of a primary knowledge concerning the searched state $|k\rangle$ [12]. In order to overcome this obstacle, we introduce a parallel space (like another memory component) which is also spanned by the records states type $|i\rangle_o$ (the subscript o stand for the observer which determines the searched item) but unlike the register state (which belongs to what we refer to a r -space) that is occupied with the unsorted state, this o -space represents the observer search selection of a definite single record state, say, $|k\rangle_o$. Thus, eventually, the searched item is well defined but only among the observer parallel space and definitely not to the other search component. We will present a scheme which shows that by correlating

the observer state with the unsorted state, the later will be represented in the appropriate 2D representation. Once the unsorted state is prepared, we introduce a rapid search algorithm.

2. Concepts Formulation

Searching a record corresponds with introducing the search machine with a query. A successful search ends with the presentation of the related data. In our approach the query is introduced thorough a state while the related data output is associated with an operator-eigenvalues rather than eigenstates.

To be more specific, suppose that the search machine possesses N states $|i\rangle$ where each state presents a possible query. The measurement output is associated with the projective operator:

$$\hat{O} = \sum_{i=1}^N \lambda_i |i\rangle\langle i| \tag{1}$$

with the corresponding eigenvalues λ_i .

A measurement result is the readable content, namely the eigenvalues rather than the eigenstates. Therefore, we propose that while the query request is introduced through a state, the relevant record information is expressed through the eigenvalues.

There are few possibilities to represent a data. It can be presented numerically or through a string of symbols. Indeed the current mathematical formulation allows us to introduce a symbolic eigenvalues-*eigensymbols*, that is, a string of symbols instead of a numeric value as shown in the following example:

Suppose that we search for information concerning Albert Einstein from an N number of records. The request is then coded into a state $|E\rangle$. We can present the 2D-projective-operator:

$$\hat{E} = \begin{pmatrix} \text{Albert} \\ \text{Einstein} \end{pmatrix} |E\rangle\langle E| + \begin{pmatrix} \text{Record} \\ \text{not} \\ \text{found} \end{pmatrix} \sum_{i \neq E} |i\rangle\langle i| \tag{2}$$

If $|E\rangle$ is the input state then the result will be the $\begin{pmatrix} \text{Albert} \\ \text{Einstein} \end{pmatrix}$, that is

$$\hat{E}|E\rangle = \begin{pmatrix} \text{Albert} \\ \text{Einstein} \end{pmatrix} |E\rangle. \tag{3}$$

All other states of the orthogonal basis, $|i \neq E\rangle$, will

yields the string $\begin{pmatrix} \text{Record} \\ \text{not} \\ \text{found} \end{pmatrix}$:

$$\hat{E}|i \neq E\rangle = \begin{pmatrix} \text{Record} \\ \text{not} \\ \text{found} \end{pmatrix} |i \neq E\rangle \tag{4}$$

Note that this mathematical formalism allows us to present the output with many ways. For example the output can be a link to a relevant computer site. For that case the link can be represented by a **Links-Operator-EinsteinLink** such that:

$$\hat{E} = \underline{\text{EinsteinLink}} |E\rangle\langle E| + \begin{pmatrix} \text{Record} \\ \text{not} \\ \text{found} \end{pmatrix} \sum_{i \neq E} |i\rangle\langle i| \tag{5}$$

If the observer query is $|E\rangle$, then we obtain

$$\hat{E}|E\rangle = \underline{\text{EinsteinLink}} |E\rangle. \tag{6}$$

Once the machine recognize the $|E\rangle$ state it activates the link operator EinsteinLink to present or link to the relevant site. For other cases the observer receive the text message “Record not found”.

3. The Machine Processor

3.1. The Observer Input

As in the grover algorithm [6] the register is occupied with an N unsorted records state:

$$|\rho\rangle_r = \frac{1}{\sqrt{N}} \sum_{i=1}^N |i\rangle_r \tag{7}$$

where states that are related to the register component are denoted by the subscript r .

The Grover search algorithm shrinks the N -records basis into a 2-D space, spanned by the states $|k\rangle$ —the state under the search and the state $|\neq k\rangle$ —a superposition of all the other states. Consequently each search is associated with an exclusive 2-D representation such that preparing the appropriate representation corresponds with a primary knowledge about the searched state. In that sense the Grover algorithm is not a pure search algorithm. Actually it more accurate to describe the Grover algorithm as “inverting a function” meaning that if we have a function $y = f(x)$ that can be evaluated on a quantum computer, this algorithm allows us to calculate x when given y [11].

If we consider the $|\neq k\rangle$ and $|k\rangle$ as vectors, the algorithm corresponds with rotating the $|\neq k\rangle$ state with respect to the $|k\rangle$ state. This is an equivalent way to represent the constrain of a primary knowledge concerning the searched state $|k\rangle$ as nicely described in ref. [12]:

“Each Classical Walk step is almost Target-blind; meaning that it does not depend on what the searched

target is, except in deciding whether to stop or go on with the search. It's as if we were moving from some original binary code 0000 to the searched target in a dark room, taking small blind steps, until we hit the target. Once we hit the target, we recognize it as such and stop.

Now note that each Grover step is NOT an almost Target-blind; in fact, each Grover step is a rotation by a small angle α in the plane that contains the unsorted state vector $|\rho\rangle$ and the target state $|k\rangle$, where α depends on the angle between $|\rho\rangle$ and $|k\rangle$. So the Grover steps are not almost Target-blind. Far from it."

As mentioned before, transforming the algorithm into a real search engine corresponds with introducing a parallel space, that is, an additional memory component. Likewise the records space it is spanned by the records states type $|i\rangle_o$ (the subscript o stand for the observer which determine the searched item) but unlike the r -space, this o -space represents the observer search selection of a definite single record state $|k\rangle_o$. Thus, eventually, the searched item is well know but only in the observer parallel space and definitely not by the other search component.

Suppose that by searching a k -record, the observer defines the state $|k\rangle_o$. The searching machine first step is to correlate between the observer state $|k\rangle_o$ with the processor unsorted state $|\rho\rangle_r$ yielding:

$$|\tau\rangle = |\rho\rangle_r |k\rangle_o, \tag{8}$$

with $|\rho\rangle_r$ being the unsorted state of Equation (7).

Assuming that the machine processor is fed with the observer selection. In response the machine processor exhibit the **register state** of Equation (7) in the associated representation:

$$\begin{aligned} |\rho\rangle_r &= \frac{1}{\sqrt{N}} |k\rangle_r + \sqrt{\frac{N-1}{N}} |\neq k\rangle_r, \\ |\neq k\rangle_r &= \frac{1}{\sqrt{N-1}} \sum_{i \neq k} |i\rangle_r \end{aligned} \tag{9}$$

We refer this representation as the **k -basis-representation**. The question we address is how does the observer $|k\rangle_o$ -selection can be followed by the r -space such that the unsorted state $|\rho\rangle_r$ will be represented by the k -basis?

Suppose that $\hat{M}_{i,r}$ is the operator that operates within the r -space and can be coupled to any observer selection $|i\rangle_o$. Clearly there are N operators of the kind. The global processor task is to associate each operator with its compatible o -state. This operation is activated by the following $\hat{\mathcal{M}}$ -operator:

$$\hat{\mathcal{M}} = \sum_i |i\rangle_o \hat{M}_{i,r} \langle i|_o \tag{10}$$

Once the observer select his query, say, the state $|k\rangle_o$ and after the states are correlated as shown in Equation (8), the processor operation provides us with the output:

$$\hat{\mathcal{M}}|\rho\rangle_r |k\rangle_o = \hat{M}_{k,r} |\rho\rangle_r |k\rangle_o, \tag{11}$$

meaning that among all other possibilities the r -processor is now operating within the k -representation only. We note that $\hat{M}_{k,r}$ can be regarded as an "eigenoperator" [13,14].

3.2. The $\hat{M}_{k,r}$ Operator

A simple measurement of the record k can be represented with the following projective operator:

$$\hat{M}_{r,k} = \left(\begin{array}{c} \text{Relevant} \\ \text{information} \end{array} \right) |k\rangle_r \langle k|_r + \left(\begin{array}{c} \text{not} \\ \text{found} \end{array} \right) |\neq k\rangle_r \langle \neq k|_r \tag{12}$$

Applying the operator $\hat{M}_{r,k}$ on the unsorted state $|\rho\rangle_r$ (see Equation (9)) and considering this operation by means of conducting a macroscopic measurement, we

will probably obtain a $\left(\begin{array}{c} \text{not} \\ \text{found} \end{array} \right)$ output and conse-

quently the remaining state will be the $|\neq k\rangle$ -state. Therefore in order to improve the odds of detecting the required $|k\rangle$ associated data, we apply the Pauli gate that switches between the $|k\rangle_r$ $|\neq k\rangle_r$ -states while leaving the "eigensymbols" unchanged.

This improved operator is:

$$\hat{\mathcal{M}}_{r,k} = \hat{P}_{r,k} \hat{M}_{r,k} \hat{P}_{r,k}^{-1} \tag{13}$$

where the Pauli gate is

$$\hat{P}_{r,k} = |k\rangle_r \langle \neq k|_r + |\neq k\rangle_r \langle k|_r \tag{14}$$

This yields:

$$\hat{\mathcal{M}}_{r,k} = \left(\begin{array}{c} \text{Relevant} \\ \text{information} \end{array} \right) |\neq k\rangle_r \langle \neq k|_r + \left(\begin{array}{c} \text{not} \\ \text{found} \end{array} \right) |k\rangle_r \langle k|_r \tag{15}$$

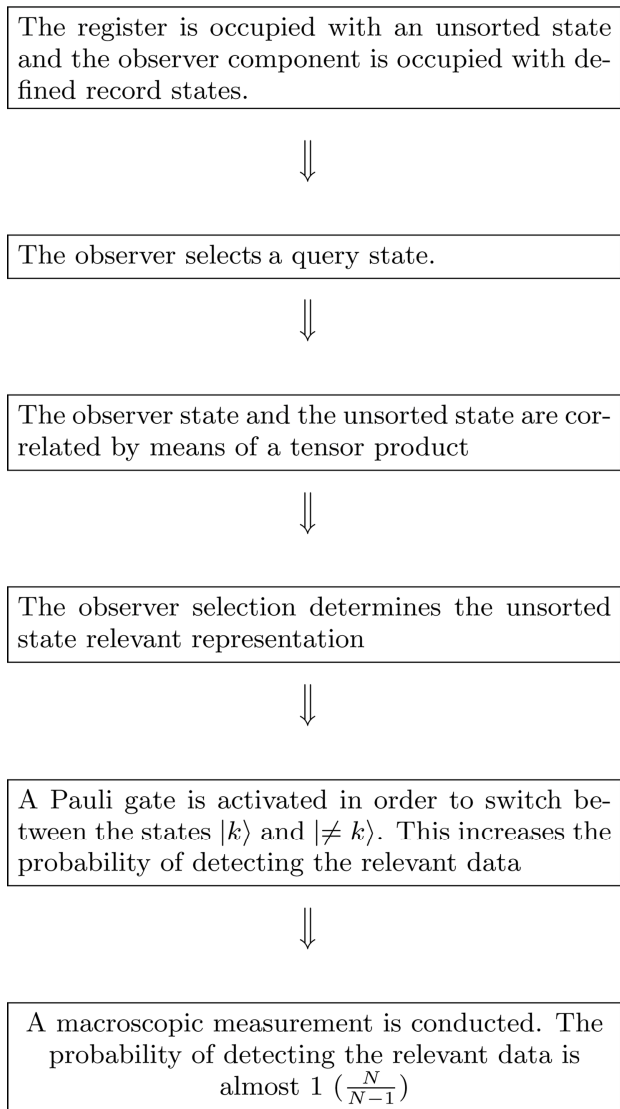
This eigenstates switching causes the high probability state $|\neq k\rangle_r$ to possess the required eigensymbol data

$\left(\begin{array}{c} \text{Relevant} \\ \text{information} \end{array} \right)$ with the high probability $\frac{N-1}{N}$. Thus,

not only we defined a pure searching algorithm by defining the observer query separately from the search engine, we also introduced a very rapid algorithm which composed mainly of two steps: The Pauli gate and a macroscopic measurement.

4. Summary

We summarize this paper with the following flowchart that shows our proposed algorithm:



REFERENCES

- [1] A. Y. Vlasov, *Quantum Physics*, 1996, 9703010v1.
- [2] D. Deutsch, *Proceedings of the Royal Society London: A*, Vol. 400, 1985, pp. 97-117. [doi:10.1098/rspa.1985.0070](https://doi.org/10.1098/rspa.1985.0070)
- [3] R. Jozsa and N. Linden, *Proceedings of the Royal Society London: A*, Vol. 459, 2003, pp. 2011-2032. [doi:10.1098/rspa.2002.1097](https://doi.org/10.1098/rspa.2002.1097)
- [4] P. W. Shor, *SIAM Journal on Computing*, Vol. 26, 1997, pp. 1484-1509. [doi:10.1137/S0097539795293172](https://doi.org/10.1137/S0097539795293172)
- [5] M. A. Nielsen and I. L. Chuang, "Quantum Computation and Quantum Information," Cambridge University Press, Cambridge, 2000.
- [6] L. Grover, *Proceedings of 28th Annual ACM Symposium on the Theory of Computing*, ACM Press, New York, 1996, p. 212.
- [7] D. Deutsch, *Proceedings of the Royal Society London: A* Vol. 425, 1989, p. 73.
- [8] C. A. Trugenberger, *Quantum Physics*, 2006, 0210176v2.
- [9] C. A. Trugenberger, *Physical Review Letters*, Vol. 87, 2001, Article ID: 067801 [doi:10.1137/S0097539795293172](https://doi.org/10.1137/S0097539795293172)
- [10] C. A. Trugenberger, *Physical Review Letters*, Vol. 89, 2002, Article ID: 0277903. [doi:10.1103/PhysRevLett.89.277903](https://doi.org/10.1103/PhysRevLett.89.277903)
- [11] D. F. Floess, E. Andersson and M. Hillery, 2010. arxiv.org/pdf/1006.1423
- [12] R. R. Tucci, 2010. <http://qbnets.wordpress.com>
- [13] Y. Roth, *Europhysics Letters*, Vol. 82, 2008, Article ID: 10006.
- [14] Y. Roth, *International Journal of Theoretical Physics*, Vol. 51, 2012, pp. 3847-3855.