Discrete Quantum Walk for Solving Nonlinear Equations over Finite Fields

Essam F. Al-Daoud
Zarka Private University, Zarka, Jordan

Received: 25/9/2006                                                                                              Accepted: 2/5/2007

Abstract: This paper reviews some of quantum computation basics: qubit definition, quantum operations, quantum algorithms, measurement and quantum walk, and then applies the discrete quantum hypercube walk to the nonlinear systems over Galois Finite Field GF (2^n). The modified quantum walk algorithm needs $O(\sqrt{m})$ iteration, while the best classical general solution takes $O(m^{\frac{3}{2}})$ iteration. The paper also compares the classical and the modified algorithms numerically by using the Quack library.

Keywords: Quantum Walk, Superposition, Nonlinear Equations, Quantum Simulator.

Introduction

Quantum computation and quantum information are new fields in computer science, which have rapidly gained popularity and earned a lot of attention in the last decade. The main advantage of the quantum technology is its ability to efficiently solve hard problems, such as integer factorization, finding the hidden subgroup, lattice problems, and in general solving the NP complete problems in polynomial time. Furthermore, quantum information offers a new cryptosystem suitable for the modern communication and computation. In 2001, IBM's Almaden Research Center demonstrated the execution of Shor's algorithm using 7-qubit NMR computer. The number 15 was factored using identical molecules, each containing 7 atoms. [1]

Classical random walks are very well studied processes, and have had several applications in computer science, such as estimating the volume of convex body, approximating the permanent and solving the 3SAT problem. In the simplest classical walks variation, a single walker moves either forward or backward, depending on the outcome of corresponding coin. [2,3] In contrast, quantum walks allow for superposition of classical random walks and, due to interference effects, can exhibit different features and offer advantages when compared to the classical case. [4]

The quantum walks expected distance from origin after $t$ steps is $\Omega(t)$; therefore, it spreads quadratically faster than the classical walks. [5]

* Author's e-mail address: essamauto@yahoo.com
The remainder of this paper is organized as follows. Section 2 presents the basic concepts of the quantum computer, the superposition representation of the qubit, and derives the quantum operations or gates from Schrödinger equation. Section 3 introduces the principles of the quantum algorithms, such as quantum parallelism, interference, and measurement. Section 4 discusses the discrete quantum walk on one dimension and compares the probability distribution of classical and quantum walks. In section 5 we apply the discrete quantum hypercube walk to the nonlinear systems over GF($2^n$). Finally, in Section 6, we implement the suggested quantum algorithm by using the Quack library.

**Quantum Operations**

The basic unity information in the quantum computer is the qubit, which has two possible states $|0\rangle$ or $|1\rangle$. Unlike classical bits, a qubit can be forced into a superposition of the two states which is often represented as linear combination of states:

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle$$

for some $\alpha$ and $\beta$ such that $|\alpha|^2 + |\beta|^2 = 1$. The states can be represented as vectors in Hilbert space, and $n$ qubits (a register) is the tensor product of the states. The second postulate of quantum mechanics describes the evolution of a closed system by the Schrödinger equation:

$$i\hbar \frac{\partial}{\partial t} |\psi\rangle = H |\psi\rangle$$

where $H$ is the Hamiltonian operator and $\hbar$ is Planck’s constant. In quantum physics, it is common to use a system of measurement where $\hbar = 1$, so the discrete-time solution of Schrödinger equation is:

$$|\psi\rangle = U |\psi_0\rangle$$

where $U$ is a unitary matrix. A general 2-dimensional complex unitary matrix $U$ can be written as:

$$U = e^{i\theta H}$$

The common single qubit unitary operations or gates on registers contain a qubit:

**Pauli gates**

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

**Hadamard gate**

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

**Phase- and $\pi/8$-gate:**

$$S = \begin{pmatrix} 1 & 0 \\ 0 & i \end{pmatrix}, \quad T = \begin{pmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{pmatrix}$$

The previous gates can be generalized to registers containing $n$ qubit by applying tensor product $n$ times on the unitary operation itself. On the other hand, the common two qubit operations or gates:

**controlled-not gate**
\[ \text{CNot} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} \quad \text{or CNot} \left| x, y \right\rangle \rightarrow \left| x \oplus y, y \right\rangle \]

\[ \text{swap-gate} \]

\[ \text{Swap} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{or Swap} \left| x, y \right\rangle \rightarrow \left| y, x \right\rangle \]

However, the universal set of quantum gates can be built by using CNot and single qubit operations which can be implemented by using a beam splitter and applying a radio frequency pulse.

**Quantum Algorithm Principles**

The superposition of \( n \) qubits (or a register) allows each operation or quantum gate to act on all basic states simultaneously. This type of computation is the basis for quantum parallelism which leads to a completely new model of data processing. Shor’s algorithm is a good example of quantum superposition and parallelism. Let \( \left| \psi \right\rangle = \left| 0 \right\rangle \left| 0 \right\rangle \) be the initial state of a quantum computer. The Hadamard operation on the first register will then leave the quantum computer in the following superposition state:

\[ \left| \psi \right\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left| i \right\rangle \left| 0 \right\rangle \]

Quantum parallelism is exploited by applying a reversible function \( f \) on all states from \( \left| 0 \right\rangle \) to \( \left| 2^n-1 \right\rangle \), simultaneously. In Shor’s algorithm \( f(x) = x \mod n \), and the computer state becomes:

\[ \left| \psi \right\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left| i \right\rangle \left| x' \mod n \right\rangle \]

However, the observation of the superposition of states makes it collapse to one of the states with a certain probability. For example, if we like to measure the quantum register:

\[ \psi = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{2^n-1} \left| i \right\rangle \]

then the superposition states will collapse to the state \( \left| x \right\rangle \) with probability:

\[ p(x) = \left< \psi \left| M_x^\dagger M_x \right| \psi \right> \]

and the state of the register after measurement

\[ \left| \psi \right\rangle' = \frac{M_x \left| \psi \right\rangle}{\left< \psi \left| M_x^\dagger M_x \right| \psi \right>} , \]

where \( M_x = \left| x \right\rangle \left< x \right| \). Fortunately, quantum interference can be used to improve the probability of obtaining a desired result by constructive interference and minimize the probability of obtaining an unwanted result by destructive interference. Thus, the challenge is to design quantum algorithms which utilize the interaction of the superposition states to maximize the chance of the interesting states.\(^{[8,9]}\)
Discrete Quantum Walk

The discrete quantum walk can be described by the repeated application of a unitary operator

\[ U = S (I \otimes C) \]

on a Hilbert space \( H^S \otimes H^C \), where \( S \) performs a controlled shift based on the state of the quantum coin and acts on a Hilbert space \( H^S \) which spanned by the nodes, \( C \) flips the quantum coin and acts on the Hilbert space \( H^C \) which spanned by the directions, and \( I \) is the identity operator on \( H^S \). We can select any shift or coin operators, but both must be unitary matrices. For example, we can use the Hadamard operator as a coin to get asymmetric distribution:

\[ H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix}, \]

or the unitary operator:

\[ C = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & i \\ i & 1 \end{pmatrix} \]

to get symmetric distribution. The associated one dimension shift operator is:

\[ S = (|1><1| \otimes \sum_i |i+1><i|) + (|0><0| \otimes \sum_i |i-1><i|) \]

where the index \( i \) runs over \( Z \) in the case of a line, or \( 0 \leq i \leq N-1 \) in the case of a circle. To illustrate the difference between quantum walk and classical walk, let us consider the first three steps of a two-way quantum walk, where the used coin is Hadamard operator and starting in state \(|0><0|\):

\[ \begin{align*}
|0><0| & \xrightarrow{U} \frac{1}{\sqrt{2}} |-1><0| + \frac{1}{\sqrt{2}} |1><1| \\
\xrightarrow{U} & \frac{1}{\sqrt{2}} |2><0| + \frac{1}{\sqrt{2}} |0><1| + \frac{1}{2} |0><0| - \frac{1}{2} |2><1| \\
\xrightarrow{U} & \frac{1}{2\sqrt{2}} |-3><0| + \frac{1}{2\sqrt{2}} |-1><1| + \frac{1}{2\sqrt{2}} |1><0| - \frac{1}{2\sqrt{2}} |3><1|
\end{align*} \]

However, two effects remain for almost any coin and starting state: the maximum probability is reached for \(|n| = \pm \frac{t}{\sqrt{2}}\), and the expected distance from the starting location and the location which we measure after \( t \) steps is \( \Omega(t) \). Figures 1 and 2 show the probability distribution of quantum and classical walks.
In general, we can walk in $d$ directions, but we must find a suitable unitary shift operator associated with the coin which spanned by the $d$ directions. Hypercube is a famous structure and has been used in classical and quantum walks.\cite{6,7}

**Discrete Quantum Walk Algorithm for NLS over GF (q)**

In this section, we will introduce the nonlinear system over GF (q) and then apply the hypercube quantum walk to find the solution of the nonlinear systems.

Solving a system of nonlinear equations over Galois Finite Field GF(q) (NLS) is an NP hard problem, and the best known general solution is brute force. One of the most important applications of NLE is the cryptanalysis, where many cryptography methods can be expressed as a system of quadratic equations over GF(q).

A system of nonlinear equations with $m$ variables can be written as follows:

$$f_j(i_1, i_2, \ldots, i_m) = 0, \text{ for all } j=1, 2, \ldots, k.$$ 

and the variables $i_1, i_2, \ldots, i_m$ are in the finite field $GF(q)$, where $q=2^t$. The best solution of above systems on the classical computers takes $O(2^m)$ iteration, while this complexity can be reduced to $O(\sqrt{2^m})$ iteration by using the discrete quantum walk algorithm on a hypercube of $(nm)$ dimensions.
Let \( x \) be a number of length \( d = nm \) bits and \( x = y_1 \mid y_2 \mid \ldots \mid y_m \) (the operator \( \mid \) means bits concatenation), where \( y_k \) are elements in \( GF(2^n) \) for all \( k = 1, 2, \ldots, m \). Let \( H^C \) be a Hilbert space spanned by \( \{|i>: i = 1, 2, \ldots, d\} \) and let \( H^S \) be a Hilbert space spanned by \( \{|x>: x \in GF(2^d)\} \), thus the search algorithm acts on the space \( H^S \otimes H^C \). One frequently chosen coin is Grover's "diffusion" operator, given by:

\[
C_0 = G_{dxd} = \begin{pmatrix}
 a & b & b & \ldots & b & b \\
 b & a & b & \ldots & b & b \\
 b & b & a & \ldots & b & b \\
 \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\
 b & b & b & \ldots & a & b \\
 b & b & b & \ldots & b & a
\end{pmatrix}
\]

where \( a = (2/d) - 1 \) and \( b = (2/d) \), this operator being the farthest away from the identity operator. To amplify the amplitude of the correct solution we must apply another coin \( C_1 = I \) to some states. A suitable shift operator in the hypercube walk is the map:

\[
S: |i>|x> \rightarrow |i>|z>
\]

and the equivalence unitary operator is

\[
S = \sum_i \sum_i |i><i| \otimes |z><x|
\]

where \( z = x \oplus e \) and \( e \) is the \( i^{th} \) basis vector.

The following algorithm is the discrete quantum \( d \)-dimension hypercube walk for solving a nonlinear system over \( GF(2^n) \):

1- Initialize the coin register to the superposition states.
2- Initialize the elements register to the superposition states, using \( d \) qubits.
3- Repeat steps 4-6 \( T \) times
4- Apply the coin \( C_1 \) to the coin register if the associated state is \( x \) where \( f(y_1, y_2, \ldots, y_m) = 0 \) such that \( x = y_1 \mid y_2 \mid \ldots \mid y_m \).
5- Apply the coin \( C_0 \) to the coin register if the associated state is \( x \) where \( f(y_1, y_2, \ldots, y_m) \neq 0 \) such that \( x = y_1 \mid y_2 \mid \ldots \mid y_m \).
6- Apply the shift operator \( S \) to the element register.
7- Measure the element register.

According to[3] the general hypercube search algorithm will collapse to the correct solution with probability \( p \approx \frac{1}{2} \) after \( T = \frac{\pi}{2} \sqrt{2^d} \) times.

**Discrete Quantum Walk Simulation**

Although the quantum computers at this point in time are not efficient, many quantum simulators have been developed, such as Quack, Open Qubit, QCL and Quantum Octave.[11,8,9] QCL (Quantum Computation Language), a high level, architecture independent programming language for quantum computers, includes program files for simulation of an implementation of Shor's algorithm, and files for simulating other aspects of quantum computation. Quack is a MATLAB package for simulating simple quantum circuits, such as single qubit, Bell state preparation, unitary gates, Hadamard, controlled-NOT,
controlled-sign, swap, Toffoli, and general controlled-unitary gates. Table 1 compares the number of iterations in the classical algorithm and the number of iterations in the suggested algorithm, where the simulation of the suggested algorithm done by using the Quack library and random nonlinear systems over GF(2^n).

Table 1: The number of iterations in classical and quantum algorithms

<table>
<thead>
<tr>
<th>Field size (n)</th>
<th>Number of variables (m)</th>
<th>Number of equations</th>
<th>Classical algorithm (brute force)</th>
<th>Suggested quantum algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>2</td>
<td>2</td>
<td>224</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>2</td>
<td>8192</td>
<td>568</td>
</tr>
<tr>
<td>6</td>
<td>3</td>
<td>3</td>
<td>65642</td>
<td>804</td>
</tr>
<tr>
<td>6</td>
<td>4</td>
<td>3</td>
<td>332327</td>
<td>12867</td>
</tr>
<tr>
<td>6</td>
<td>5</td>
<td>5</td>
<td>178956954</td>
<td>51471</td>
</tr>
<tr>
<td>7</td>
<td>4</td>
<td>4</td>
<td>53687091</td>
<td>25734</td>
</tr>
</tbody>
</table>

Conclusion

Quantum computing outperforms the best classical techniques for some hard computation problems. The universal set of quantum gates can be built by using CNot and single qubit operations which can be implemented by using a beam splitter and applying a radio frequency pulse. This paper has introduced a discrete quantum d-dimension hypercube walk algorithm to solve a system of nonlinear equations over Galois Finite Field GF(q) in $O(\sqrt{2^m})$ iteration, which is considered faster than its classical counterpart by a square-root. The suggested algorithm can be implemented by using the Quack library.

References


