WILFRID LAURIER UNIVERSITY

LAURIER *Inspiring Lives.*

Computer Algebra Research Group
of
Wilfrid Laurier University

A Quantum Computing Algorithm for a Combinatorial Search Problem

George Lifchits, BSc. Computer Science
Supervised by Dr. Shohini Ghose & Dr. Ilias Kotsireas
Department of Physics & Computer Science
March 23, 2016

fqssa
Faculty of Science
Students' Association
of Wilfrid Laurier University

SHARCNET™

1 Motivation

Quantum computing is a theoretical framework for processing information that is based on the postulates of quantum mechanics. By exploiting quantum mechanical phenomena, researchers have developed several fascinating algorithms that solve problems faster than any "classical" computer can ever possibly achieve.

Grover's search is one such algorithm. It performs search through an unsorted database with $O(\sqrt{N})$ resource complexity. [5] How can we use Grover's search in an applied context? What would it look like? How fast would it be? Can we do it naïvely, avoiding the use of esoteric tricks and domain expertise?

To investigate the research question, we focused on how we can use Grover's search to solve the problem of finding **D-optimal sequences**.

2 Background

2.1 D-optimal sequences

D-optimal sequences are a pair of {-1, 1} sequences which can be used to construct a $2N \times 2N$ matrix (with elements -1 and 1) that has the maximum possible determinant.

For a D-optimal sequence pair A and B, the matrix is: [2]

$$H = \begin{pmatrix} A & B \\ -B^T & A^T \end{pmatrix}$$

Let $A = [a_1, a_2, \ldots, a_n]$ be a {-1, 1} sequence of length N. The periodic autocorrelation function (PAF) associated with A is defined as $P_A(i) = \sum_{k=1}^{N} a_k a_{k+i}, \quad i = 0, \ldots, N-1$. Consider a similar sequence B. The two sequences A and B are D-optimal if the sum of their periodic autocorrelation functions is 2 for all lag values. Formally: [3]

$$P_A(i) + P_B(i) = 2, \quad i = 1, \ldots, \frac{N-1}{2} \tag{1}$$

This is sufficient to identify D-optimal {-1, 1} sequence pairs, so we refer to formula 1 as the 'PAF constraint'.

2.2 Quantum bits

- Instead of bits (0 or 1) we have quantum bits ('qubits'), which have a *probability* of being *either* 0 or 1.
- Until we measure the qubit, it is *simultaneously* 0 *and* 1. This is *superposition* (think of Schrödinger's cat).
- A qubit is denoted $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$, which means the qubit has a probability of $|\alpha|^2$ of being measured as 0, and a probability of $|\beta|^2$ of being measured as 1. α and β are complex values.
- To perform operations with qubits, we use unitary operators (matrices, with norm equal to 1).
- A system with n qubits is a superposition of 2^n possible states. Such a system is the Kronecker product of those n individual qubits. e.g. for two qubits:

$$|\psi_1\rangle|\psi_2\rangle = \alpha_1\alpha_2|00\rangle + \alpha_1\beta_2 01 + \beta_1\alpha_2|10\rangle + \beta_1\beta_2|11\rangle$$

2.3 Grover's quantum search algorithm

In 1996, Grover published a paper [5] which describes the now-famous algorithm for searching an (unsorted) database with computational complexity $O(\sqrt{N})$. The algorithm makes clever use of quantum phenomena.

With n qubits we can store $N = 2^n$ unique binary strings.

1. Start with n qubits in the $|0\rangle$ state: $|\gamma\rangle = |000\ldots0\rangle$
2. Apply a Hadamard gate to each qubit, which puts all of the qubits in an equal superposition (each state is equally likely): $|\gamma\rangle = \frac{1}{\sqrt{N}} \sum_{i=0}^{N-1} |i\rangle$
3. Perform the 'Grover iteration' $(\pi/4)\sqrt{N}$ times:
   (a) Apply a unitary operator which has the effect of flipping each state that satisfies the search. (Due to "quantum parallelism" this only requires one query!)

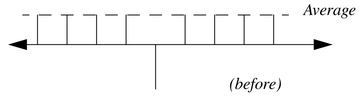

Figure 1: After applying the oracle function to the quantum state. The flipped line is a qubit state which corresponds to a solution.

   (b) Apply diffusion operator (inversion about the mean):

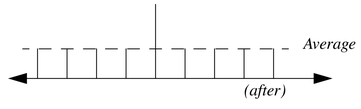

Figure 2: After inversion about the mean. The flipped qubit has brought the mean of all the qubits downward. When the mean is subtracted from all qubits, it has the effect of bringing non-solution qubits closer to probability 0.

4. Solutions will have high probability. Measure the state.

3 Simulating quantum algorithms

Here, we describe how we implemented a program which modeled a quantum state with a vector containing complex values. The program allows the user to perform basic quantum computation operations on that state vector. We used Numpy, which is a library for scientific computing and linear algebra in Python.

```
import numpy as np
from math import floor, sqrt
from quantum import Q, kron, zero, H, I
qubits = 10  # n
N = 2 ** qubits
# |00...0> (length n)
state = Q(kron(*(zero for _ in range(qubits))))
# Apply H^n to the input register.
input_H = kron(*(H for _ in range(qubits)))
state = state.apply_unitary(input_H)
# Define D (the diffusion matrix aka. inversion about the mean)
P = 1/N * np.ones((N, N))
D = -np.eye(N) + 2*P
# Now we perform the Grover iteration.
for iteration in range(floor((np.pi/4) * sqrt(N).real)):
    state = state.apply_func(oracle) # step 1 (flip solutions)
    state = state.apply_unitary(D) # step 2 (inversion about mean)
```

Figure 3: Python implementation of the Grover's search algorithm. Besides the declaration of `oracle`, this is a working skeleton.

4 Grover's search for D-optimal sequences

For sequences of length N, we use a quantum system with $2N$ qubits. This gives us a state space of size 2^{2N}, which encodes all possible sequence pairs length N.

4.1 Oracle design for the D-optimal problem

Grover's search needs an 'oracle' that it can query for solutions in parallel. We needed a closed-form equation that would model the PAF constraint (i.e. equation 1):

$$S = \sum_{i=1}^{M} |P_A(i) + P_B(i) - 2| \tag{2}$$

S is guaranteed to be 0 if sequences A and B are D-optimal. Otherwise, S is a positive number. Now apply X (NOT) to all qubits then perform AND between them all, and store the result in a final qubit. It will be 1 if the sequences are D-optimal, and 0 if not.

4.2 Simulation results

I implemented this oracle and simulated Grover's search. Figure 4 shows how our simulation was used to find D-optimal sequences of length 5.

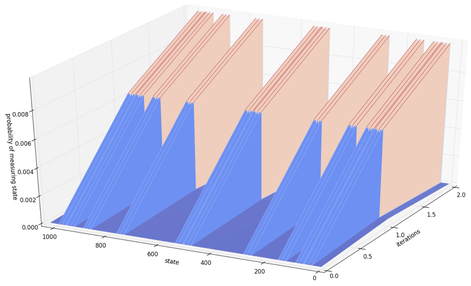

Figure 4: The plot shows the squared amplitude of each possible state along the axis labelled 'state'. The 'iterations' axis shows how the probabilities change after each Grover iteration.

This system contains $2 * 5 = 10$ qubits, yielding $2^{10} = 1024$ possible sequence pairs. 100 of the 1024 states have squared amplitude equal to 1.0%. Each corresponds to a D-optimal sequence pair. It takes 2 Grover iterations to reach this state.

5 Results: Complexity of Grover's search

- The oracle itself has a space complexity of $O(\log_2 N)$.
- The oracle itself has a runtime complexity of $O(N^2)$.
- Since Grover's search calls the oracle $O(\sqrt{N})$ times, we get a total runtime of $O(N^{5/2})$.

References

[1] Lifchits, G. "Directed Research Project". (2015) [Online]. Available: `http://georgelifchits.ca/DRP-GeorgeLifchits.pdf`
[2] Đoković, D.Ž. & Kotsireas, I.S. "New Results on D-Optimal Matrices." (2012). *Journal of Combinatorial Designs*. (20).
[3] Kotsireas, I.S. "Algorithms and Metaheuristics for Combinatorial Matrices." (2013). *Handbook of Combinatorial Optimization*.
[4] Ghose, S. *CP310A: Intro to Quantum Computing - Lectures*. (2015).
[5] Grover, Lov K., "A fast quantum mechanical algorithm for database search". (1996). *Proceedings of the 28th Annual ACM Symposium on Theory of Computing*.
[6] Mermin, N. (2007). *Quantum computer science: An introduction*. Cambridge, UK: Cambridge University Press.

This poster has been realised with LaTeX using Lars Nummendal A0 poster template