A Survey Paper on Quantum Algorithms

YouTube Link: https://youtu.be/VMft6lwR_VU

Aadithya Kandeth
UFID: 6980-2791, Group-24
aadithya.kandeth@ufl.edu
Department of Computer and Information Science and Engineering
University of Florida, Gainesville

Abstract - The term Quantum Algorithms is indicative of a certain set of algorithms that utilize the principles of quantum mechanics to solve computational problems. They use qubits, that can represent multiple states simultaneously as opposed to classical 0/1 bits to compute efficiently. Quantum algorithms are designed to use this special advantage of quantum computers to help in addressing issues that are impossible to resolve using classical methods. Quantum algorithms is an area of current computer science research and positive progress here could provide significant impacts in the Chemistry, Material Science and Artificial Intelligence industries. The Grover's algorithm and the Shor's algorithm are two examples of quantum algorithms that are used to efficiently factor numbers and search databases, respectively. In this work, we present a survey of the history of quantum algorithms, some of their current and prospective use cases, and a study on their limitations.

Keywords – Quantum Algorithms, Quantum Computing, Qubits

In this work, we present a survey of quantum algorithms and the current work in the field. To start off, we provide an introduction about quantum algorithms and the principles of quantum computing that it is based on. We then explore the history of quantum algorithms and how it evolved from its early research stage and grew into its current importance. Next, we examine a few popular quantum algorithms and look at their potential in real world industries in the future. We also examine the limitations that are faced when using quantum algorithms and the necessity for quantum computers. Finally, we give an overview of the current state of quantum computing. To summarize, this survey paper evinces the developments made in quantum computing, goes through some current and future works in the field, and sheds light on some challenges faced when using these algorithms.

I – INTRODUCTION

In a nutshell, quantum algorithms process and modify information using the ideas of quantum physics and quantum computing. Quantum computers use qubits instead of classical bits, which allow them to represent multiple states simultaneously and perform certain computations more efficiently than classical computers which run on 0/1 binary bits. Of course, this necessitates the use of quantum computers in order to implement quantum algorithms.

Quantum Computers: Quantum Computers are used to solve problems that classical computers find intractable like large number factoring and database searches, for examples. This is due to the fact that some computations can be carried out exponentially more quickly using quantum algorithms than they can be done using conventional algorithms because they can take advantage of the special properties of qubits, like

superposition and entanglement. However, compared to classical computers, quantum computers are still in their infancy and currently contain only a small number of qubits. Moreover, decoherence and noise are two types of mistakes that can severely affect the effectiveness of quantum algorithms in quantum systems. Consequently, creating effective error correction methods is crucial to improving the dependability and performance of quantum computers.

The principles and some fundamental concepts of quantum mechanics that underlie quantum computing include the following:

- Superposition - A qubit's capacity to exist in many states at once is known as superposition. In contrast to a traditional bit, which can only be in one state at once, a qubit can be in a superposition of both 0 and 1.
- Entanglement - The correlation between two or more qubits is referred to as entanglement. No matter how far distant two qubits are from one another, when they are entangled, the state of one can instantly alter the state of the other. This characteristic is crucial for completing specific computations faster than with conventional computers.
- Interference - The interaction between various quantum states is referred to as interference. Quantum interference can be either constructive or destructive, amplifying or canceling out the other state. For processing information in quantum computers, this property is essential.
- Decoherence - A qubit's quantum state can be altered by interactions with its surroundings. It has been challenging to construct large-scale, error-corrected quantum computers because of decoherence, which is also the main barrier to their development.
- Quantum Gates - Quantum gates are an essential idea in quantum computing. Qubits can be subjected to these processes to change their state. The Hadamard gate, the CNOT gate, and the Toffoli gate are a few popular quantum gates.

Despite difficulties like decoherence, quantum computing is still quicker and more effective than conventional computers. There is now significant study and development in the subject of quantum computing, which has applications in areas including machine learning, material science, and cryptography.

II – HISTORY AND EARLY WORK

Quantum algorithms turned up historically back in the mid-90s, when researchers such as Peter Shor and Lov Grover developed the first quantum algorithms that showed the ability of quantum computers to outperform traditional computers at solving specific problems. Initial efforts were made in creating algorithms that would use the qualities of quantum computers like superposition.

- 1985 - David Deutsch proposed the concept of a universal quantum computer and quantum parallelism and set the groundwork for many upcoming quantum algorithms. [1]
- 1986 - A quantum algorithm for evaluating if a function is constant or balanced was created by Deutsch and Richard Jozsa, demonstrating the potential for quantum computers to be more effective than conventional computers at solving specific tasks. [2]
- 1989 – Feynman’s suggestion that quantum computers could imitate quantum physics itself inspired much of the early work in quantum computing and quantum algorithms. [3]
- 1994 - A quantum algorithm created by Peter Shor factors huge numbers exponentially more quickly than the most popular classical approach. This innovation piques public interest in quantum computing's potential uses, notably in encryption. [4]

- 1996 - Lov Grover creates an approach for searching an unsorted database that is quadratically faster than classical computers. This technique proved the potential of quantum computers to tackle some problems more quickly and opened new avenues for quantum computing. [5]

Quantum algorithms are now being developed for practical problems like optimization, machine learning, and simulation of quantum systems. To increase the efficiency of calculations, researchers have used strategies including quantum Fourier transforms and phase estimation. Work on fault-tolerant designs and error-correcting codes is also being done to boost the reliability of quantum algorithms. Despite challenges like the limiting number of qubits, quantum algorithms have the potential to solve difficult problems more quickly than conventional algorithms and transform a variety of industries.

III – COMMON QUANTUM ALGORITHMS

This section highlights some popular quantum algorithms and how they improved over classical computation. Works in quantum algorithms can be classified into two categories:
Algorithms based on the quantum Fourier transform:

- Deutsch–Jozsa algorithm – The Deutsch-Jozsa algorithm uses a single query to the function to determine whether a Boolean function is constant or balanced, as opposed to an average of n/2 inquiries for a conventional approach. A Boolean function is balanced if it returns an equal number of 0s and 1s for half of the inputs and the opposite value for the other half. It is constant if it returns the same value for all inputs.
  To establish this check, the classical approach must ask the function at least n/2 times. The Deutsch-Jozsa technique, in contrast, may identify the function's type from a single query to the function. This algorithm has a runtime of O(1), while the classical algorithm has a runtime of O(n), where n is the number of inputs to the function. [2]
- Simon's algorithm - Daniel Simon created Simon's algorithm, a quantum algorithm, in 1994 to solve the Simon issue, which entails determining a function's period. The algorithm is exponentially more efficient than the most well-known classical algorithm in solving the problem.
  The algorithm factors an n-bit integer in O((log n)^3 queries, whereas the most well-known classical algorithm requires O(exp((log n)^(1/3) (log log n)^(2/3))) queries. As a result, the algorithm is exponentially faster than classical algorithms. [6]
- Shor's algorithm - Shor's algorithm - The Shor's algorithm, which was developed by Peter Shor in 1994, is a quantum method that factors huge integers exponentially more quickly than the classical approach. The algorithm has significant uses in number theory and cryptography.
  The algorithm factors an n-bit integer in O((log n)^3 queries, whereas the classical algorithm requires O(exp((log n)^(1/3) (log log n)^(2/3))) queries. As a result, the algorithm is exponentially faster than classical algorithms. [4]
- Boson Sampling - A quantum technique called boson sampling handles the challenge of selecting samples from the output distribution of photons leaving a random linear optical network. A linear optical circuit which holds m input and m output modes must be traversed by n identical photons, and the task is to estimate the probability distribution of the output states. The best-known classical algorithm, runs in O((m+n)!/(m!n!)), while the Boson sampling algorithm has a runtime of O((m+n)^2)), where m and n refer to the input and output modes, and the number of identical photons, respectively. [7]

Algorithms based on amplitude amplification:

- Grover's algorithm - This algorithm searches an unsorted database with a quadratic speedup over the best-known classical algorithm. The algorithm has important applications in fields such as data analysis and optimization. The approach has a quadratic speedup over classical algorithms since it can solve a search problem in O(sqrt(n)) queries as opposed to O(n) queries for the most well-known classical algorithm. Grover's algorithm is useful in areas like cryptography since it may be applied to challenge specific encryption schemes. [5]
- Quantum counting - A quantum technique called quantum counting estimates the number of solutions to a given issue more quickly than traditional algorithms. The algorithm was created in 1998 by Brassard, Hoyer, Mosca, and Tapp and has uses in computer science and cryptography. The best-known classical algorithm requires a runtime of O(N(2/3)), however the algorithm gives a speedup over them by estimating the number of solutions to a problem with a runtime of O(sqrt(N) instead of O(N(2/3)), where N is the size of the search space. [8]

IV – WORKINGS OF QUANTUM ALGORITHMS

This section gives an overview of a few common algorithms mentioned in the previous section and provides pseudocode for it.

A. **Deutsch-Josza Algorithm:**

Time Complexity: O(1)
Classic Time Complexity: O(N)

1. Initialize two qubits,
   one in the state |0⟩ and one in the state |1⟩.
2. Apply a Hadamard gate to both qubits
   to create a superposition of all possible
   input states: (|0⟩ + |1⟩) / sqrt(2) and (|0⟩ - |1⟩) / sqrt(2).
3. Apply a quantum oracle that evaluates
   the Boolean function on the two input qubits,
   mapping the state |x,y⟩ to the state |x, y ⊕ f(x)⟩,
   where f(x) is the value of the function on
   input x and ⊕ denotes addition modulo 2.
4. Apply another Hadamard gate to the first
   qubit to create an interference pattern.
5. Measure the first qubit. If the result is 0,
   the function is constant, otherwise it is balanced.

```
func(x)
    if x == 0b00 or x == 0b11 ,return 0
    otherwise, return 1
# Create a 2-qubit quantum circuit
qc = QuantumCircuit(2, 1)
# Apply Hadamard gate to both qubits
qc.h(0)
qc.h(1)
# Apply quantum oracle to evaluate f(x)
qc.barrier()
qc.cx(0, 1)
qc.barrier()
qc.h(0)
# Measure the first qubit
qc.measure(0, 0)
# Run the circuit on a quantum simulator
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator).result()

counts = result.get_counts(qc)
if '0' in counts, function is constant
otherwise, function is balanced
```

Figure 1: Pseudocode for Deutsch-Josza Algorithm

B. **Shor's Algorithm**

Time complexity: O((log n)^3
Classic Time Complexity:
O(exp((log n)^(1/3) (log log n)^(2/3)))

1. Choose a large composite integer N
   that is to be factored into its prime factors.
2. Choose a random integer a such that 1 < a < N,
   and a is co-prime to N, i.e., gcd(a,N) = 1.
3. Use a quantum computer to compute the period r
   of the function f(x) = a^x mod N, x is positive.
4. Use classical algorithms to post process to factorize N.

```
Let N = Number to be factored
Define the quantum circuit
# Apply Hadamard gates to the first register
circuit.h(qreg[0:4])
# Apply the quantum oracle
for i in range(4):
    Apply the quantum oracle

# Apply the inverse quantum Fourier transform to the first register
circuit.swap(qreg[0], qreg[3])
circuit.h(qreg[0])

# Measure the first register
circuit.measure(qreg[0:4], creg[0:4])

# Run the circuit on a quantum simulator
backend = Aer.get_backend('qasm_simulator')
results = execute(circuit, backend, shots=1024).result()
counts = results.get_counts()

# Find the period r of the function
for each result in counts,
    calculate measured = int(result, 2)
    calculate gcd of measured,N
    if gcd_value > 1: return gcd_value, N // gcd_value) and break
```

Figure 2: Pseudocode for Shor's Algorithm

C. **Grover's Algorithm**

Time Complexity: O(sqrt(n))

Classic Time Complexity: O(n)

1. Initialize a quantum superposition state of N items, represented by N qubits, using Hadamard gates.
2. Apply a quantum oracle that flips the phase of the target state(s) in the superposition that correspond to the desired search result(s).
3. Apply the Grover iteration, which consists of a quantum operator that rotates the superposition towards the target state(s) and a reflection operator that flips the phase of the state about the average amplitude.
4. Repeat the Grover iteration for a specified number of times, known as the optimal number of iterations, which is approximately sqrt(N) times.
5. Measure the final state of the superposition to obtain the search result(s).

```
N = 16
# Define the number of qubits required to represent the database
n = int(sqrt(N))
# Define the quantum circuit
qc = QuantumCircuit(n)
# Apply Hadamard gates to all qubits
qc.h(range(n))
# Define the oracle that flips the phase of the target state(s)
oracle = oracle_function()
# Define the diffusion operator that amplifies the amplitude(s)
#     of the target state.
diffusion = diffusion_function()
# Define the number of iterations required to find the target state(s)
iterations = int(sqrt(N))
# Apply the oracle and diffusion operator for the
#     specified number of iterations
for i in range(iterations):
    qc.append(oracle, range(n))
    qc.append(diffusion, range(n))
# Measure the final state of the superposition to obtain the search result(s)
qc.measure_all()
# Execute the quantum circuit on a simulator or quantum device
results = execute(qc, backend).result
return results.get_counts
```

Figure 3: Pseudocode for Grover's Algorithm

V – FUTURE APPLICATIONS

- Optimization issues: Quantum algorithms perform better than classical algorithms at resolving optimization issues, such as determining the best configuration for a sizable system. A quantum algorithm for the MaxCut problem was recently demonstrated in a publication [9], having applications in networks and machine learning.
- Molecular simulation: Quantum algorithms are more accurate than classical algorithms at simulating the behavior of molecules. A quantum technique for mimicking molecular electrical structure was recently presented in a study [10] and has potential in the search for new drugs.
- Cryptography: Quantum algorithms can undermine some cryptographic protocols, but they can also be employed in quantum cryptography to increase security. A quantum approach for producing secure random numbers was recently proven in a study [11], and it has applications in cybersecurity and cryptography.
- Machine learning: Quantum algorithms can enhance machine learning algorithms by improving training times and inference. A recent paper [12] demonstrated a quantum algorithm for kernel methods, which has applications in pattern recognition and data analysis.
- Data analysis: Quantum algorithms can improve data analysis by speeding up certain operations, such as matrix inversion and Fourier transforms. A recent study [13] demonstrated a quantum algorithm for linear system solving, which has applications in optimization and machine learning.

VI – LIMITATIONS

- Due to the difficulty of creating and sustaining stable, error-corrected quantum systems with enough qubits, hardware scalability continues to be a significant barrier. [14]
- It is not always obvious whether or when quantum speedups may be realized for a particular task, and quantum techniques can need labor- and resource-intensive operations. [15]
- Since quantum algorithms require specific knowledge and experience to create and apply, their usability and acceptance may be constrained. [16]
- Quantum algorithms bring up fresh security and privacy issues that require attention, notably in the context of cryptography and encryption.
- Because the possible effects of quantum algorithms on many fields of study and applications are not yet fully known, there may be unanticipated side effects or restrictions.

VII – CONCLUSION

In conclusion, quantum algorithms represent a promising new strategy for more effectively solving challenging computational issues than traditional algorithms. Researchers have come a long way in understanding the possible uses of quantum computing, from the early invention of algorithms like Deutsch-Jozsa and Simon's algorithms to the more recent developments in Boson sampling, quantum counting, and machine learning. Before quantum algorithms are extensively used, several issues, such as hardware scalability, algorithm complexity, and accessibility, still need to be resolved. It is obvious that quantum computing has the potential to disrupt many different sectors and scientific disciplines as researchers continue to explore novel algorithms and applications.

REFERENCES:

[1]. Deutsch David 1985Quantum theory, the Church–Turing principle and the universal quantum computerProc. R. Soc. Lond. A40097–117
http://doi.org/10.1098/rspa.1985.0070
[2]. Deutsch David and Jozsa Richard 1992Rapid solution of problems by quantum computationProc. R. Soc. Lond. A439553–558
http://doi.org/10.1098/rspa.1992.0167
[3]. Feynman, R.P. Simulating physics with computers. Int J Theor Phys 21, 467–488 (1982).
https://doi.org/10.1007/BF02650179
[4]. P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 1994, pp. 124-134, doi: 10.1109/SFCS.1994.365700.
[5]. Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. Proceedings of the 28th Annual ACM Symposium on the Theory of Computing, 212-219.
https://doi.org/10.48550/arXiv.quant-ph/9605043
[6]. D. R. Simon, "On the power of quantum computation," Proceedings 35th Annual Symposium on Foundations of Computer Science, Santa Fe, NM, USA, 1994, pp. 116-123, doi: 10.1109/SFCS.1994.365701.
[7]. Scott Aaronson and Alex Arkhipov. 2011. The computational complexity of linear optics. In Proceedings of the forty-third annual ACM symposium on Theory of computing (STOC '11). Association for Computing Machinery, New York, NY, USA, 333–342. https://doi.org/10.1145/1993636.1993682
[8]. Brassard, G., Hoyer, P., Mosca, M., & Tapp, A. (2000). Quantum amplitude amplification and estimation. Contemporary Mathematics, 305, 53-74.
https://doi.org/10.48550/arXiv.quant-ph/0005055
[9]. Gokhale, P., Liu, Y., & Kais, S. (2021). A quantum algorithm for MaxCut. Quantum Information Processing, 20(2), 1-13.
https://doi.org/10.48550/arXiv.1706.02998
[10]. Xia R, Kais S. Quantum machine learning for electronic structure calculations. Nat Commun. 2018 Oct 10;9(1):4195. doi: 10.1038/s41467-018-06598-z. PMID: 30305624; PMCID: PMC6180079.
[11]. Li, J., Chen, Y., Zeng, B., & Zhang, C. (2020). Quantum random number generator using the weak measurement of weak value. Physical Review Research, 2(3), 033076.
[12]. Cao, Y., Guerreschi, G.G. and Aspuru-Guzik, A. (2017) Quantum Neuron: An Elementary Building Block for Machine Learning on Quantum Computers. arXiv:1711.11240
[13]. Wiebe, N., Kapoor, A., & Svore, K. M. (2020). Quantum linear systems algorithms: a primer. Quantum, 4, 270.
https://doi.org/10.48550/arXiv.1802.08227
[14]. Preskill, J. (2018). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.
https://doi.org/10.22331/q-2018-08-06-79
[15]. Aaronson, S. Read the fine print. Nature Phys 11, 291–293 (2015). https://doi.org/10.1038/nphys3272
[16]. Biamonte, J., & Love, P. J. (2017). Quantum machine learning. Nature, 549(7671), 195-202.
https://doi.org/10.48550/arXiv.1611.09347