Survey Paper on Quantum Algorithms

Anol Kurian Vadakkeparampil
UFID: 5626-8544
anolkuri.vadakke@ufl.edu
Department of Computer and Information Science and Engineering.
University of Florida, Gainesville

Abstract - Quantum Algorithms are a particular kind of algorithm that applies the notions of quantum physics to the solution of computational issues. They are designed to use the special attributes of quantum systems, such as superposition and entanglement, to address issues that are challenging or impossible to resolve with classical methods. In this paper, we present a survey of the prominent early work and current trends in quantum algorithms, along with a discussion of their challenges and limits, as well as some of the prospective future applications of the field.

Keywords – Quantum Computing, Quantum Algorithms

We present an overview of the state of the art for quantum algorithms in this work. We give a quick introduction of the key ideas and principles before introducing the fundamentals of quantum computation and quantum algorithms. We examine early research and recent developments in this area. We also go through some of the possible uses for quantum algorithms, including machine learning and cryptography. The necessity for powerful quantum computers and the difficulties in handling mistakes and noise in quantum systems are some of the major obstacles and restrictions faced by quantum algorithms. Finally, we summarize the current status of the discipline and go through some of the most important avenues for future study. Overall, our survey highlights the significant progress that has been made in the development of quantum algorithms, as well as the many challenges and opportunities that remain in this rapidly growing field.

I. INTRODUCTION

With the goal of using the special qualities of quantum systems to address challenging computational issues, quantum computing is a fast-expanding field of study. A quantum computer is a device that performs tasks that are challenging or impossible to accomplish with classical computers by utilizing the concepts of quantum physics, such as superposition and entanglement. A crucial part of quantum computing are quantum algorithms, which provide the quantum computer the guidance it needs to solve tasks.

The key ideas, principles, and fundamentals of quantum algorithms can be divided into several categories:

- Quantum mechanics: Quantum algorithms are founded on the principles of quantum mechanics, which is the fundamental theory of matter and energy at the atomic and subatomic scales. Quantum algorithms can benefit from several fundamental ideas in quantum physics, such as superposition, entanglement, and interference.
- Quantum computation: Quantum computation is the method of doing computation by utilizing quantum mechanical phenomena like superposition and entanglement. Due to the ability to manipulate quantum states in a way that is not conceivable with conventional bits, quantum computation is fundamentally different from classical computation.

- Quantum circuit model: Using a series of quantum gates connected to quantum states, the quantum circuit model may be used to express quantum algorithms. The fundamental operations that may be carried out on quantum states are represented by quantum gates, which are the building blocks of quantum circuits.
- Quantum algorithms: These are algorithms created specifically for use with quantum computers. To tackle issues that are impossible for conventional computers to handle, they make use of the concepts of quantum physics and quantum computation. Shor's method for factoring big numbers, Grover's algorithm for exploring unstructured databases, and quantum simulation algorithms for simulating quantum systems are a few examples of quantum algorithms.

The promise for exponential speed-ups over classical algorithms for specific sorts of problems has been a driving force behind the development of quantum algorithms. Shor's method, for instance, may factorize enormous numbers in polynomial time, which is thought to be impossible using conventional algorithms. Grover's search algorithm and simulation algorithms like the quantum Fourier transform are two more crucial quantum algorithms.

II. EARLY WORK

Early efforts in the subject of quantum algorithms were on creating algorithms that might make use of the special qualities of quantum systems, such as superposition and entanglement, to resolve computing issues that are challenging or impossible to resolve using classical algorithms.

Shor's approach for factorization, which has a time complexity of O(log3 N), where N is the number being factored, was one of the important early advancements [3].

Shor's method is now exponentially faster than the most well-known classical approach, making it far more effective for high values of N. Grover's search technique, which has a time complexity of O(sqrt(N)) for searching an unsorted database of size N, was another significant early invention [2]. In comparison to the traditional technique, which has an O(N) time complexity, this provides a quadratic speedup.

In the early stages of quantum computing, simulation methods like the quantum Fourier transform were also created. The dynamics of quantum systems, such as the movement of particles in a potential well, are simulated using these techniques. For big systems, they are more effective than conventional simulation methods because of their O(log N) time complexity [1].

- Deutsch's algorithm (1985): One of the earliest quantum algorithms, Deutsch's algorithm showed the power of quantum computing by resolving a straightforward issue that could not be resolved by classical computers. The "Deutsch issue," which requires figuring out if a function's output is odd or even, was solved by the method using the superposition principle.
- Simon's algorithm (1994): The "Simon problem," which entails locating a hidden function that fulfills specific requirements, may be solved using Simon's algorithm, a quantum algorithm. The program solves the issue more quickly than traditional algorithms by using the notion of entanglement (Simon, 1994).
- Shor's algorithm (1997): This quantum algorithm breaks huge integers down into their prime components. When factoring high values of N, its time complexity is O(log3 N), which makes it substantially quicker than traditional procedures [3].

- Grover's algorithm (1996): Grover's method (1996) is a quantum algorithm for looking up information in an unsorted database. Its temporal complexity is O(sqrt(N)), where N is the size of the database, and it outperforms conventional methods by a factor of four [2].
- The quantum Fourier transform (1994): The quantum Fourier transform (1994) is a quantum method used to model the dynamics of quantum systems. For big systems, it is more effective than traditional simulation techniques because of its O(log N) time complexity [1].

Overall, these early advancements showed that quantum algorithms have the potential to be more effective than conventional algorithms in solving computational problems. They also emphasized the problems in dealing with mistakes and noise in quantum systems, as well as the constraints of quantum computing, such as the requirement for large-scale quantum computers. Despite these difficulties, there is still a lot of research and development being done in the field of quantum algorithms.

III. PROMINENT WORK IN CURRENT TIMES

- Variational quantum algorithms: A type of algorithms known as variational quantum algorithms uses a parametrized quantum circuit to determine the function's global minimum. They have been used to a variety of issues, including as machine learning and quantum chemistry simulations [9],[13].
- Quantum machine learning algorithms: Quantum systems are a sort of algorithm that may be used to conduct machine learning operations like classification and clustering. They have been demonstrated to potentially outperform traditional algorithms in some situations [5].
- Quantum algorithms for sampling: A family of algorithms known as quantum algorithms for sampling makes use of quantum systems to sample more effectively from a probability distribution than traditional algorithms. They are used in areas like statistical physics and machine learning [4].
- Quantum algorithms for optimization: In contrast to conventional algorithms, quantum algorithms for optimization employ quantum systems to locate the global minimum of a function. Quantum annealing techniques and variational quantum algorithms are two examples [6],[9].
- Development of hybrid classical-quantum algorithms: The usage of hybrid classical-quantum algorithms, which combine the benefits of classical and quantum algorithms, is being investigated by several academics. These algorithms frequently perform better than only classical or quantum algorithms [7].
- Focus on the development of algorithms that are robust to noise and errors: Managing mistakes and noise in quantum systems is one of the key difficulties in the science of quantum computing. The creation of algorithms that are resistant to various causes of mistake is thus receiving more attention [8].
- Research on the applications of quantum algorithms in a range of domains: The potential use of quantum algorithms in industries like banking, chemistry, and machine learning is gaining more and more attention. This study examines the possible advantages of applying quantum algorithms in certain fields as well as the difficulties and limitations of doing so [10],[9],[5].

Collectively, these patterns show how the area of quantum algorithms is still evolving and progressing. They also imply that in the upcoming years, quantum algorithms are expected to take on a greater significance in a variety of fields.

IV. INNER WORKINGS OF PROMINENT ALGORITHM

Shor's algorithm for Factorization:

Time complexity: O(log^3 N)

Reasoning: The time complexity of Shor's algorithm is dominated by the computation of the period r using the quantum Fourier transform, which has a time complexity of O(log^2 N) [3]. The other steps of the algorithm, such as computing the factors g and h, have a time complexity of O(log N), giving a total time complexity of O(log^3 N).

Pseudo Code:

1. Choose an integer N to be factorized
2. Choose an integer a that is relatively prime to N
3. Compute the period r of the function x^a mod N using the quantum Fourier transform
4. If r is even, compute g = gcd(a^(r/2) - 1, N) and h = gcd(a^(r/2) + 1, N)
5. If r is odd, compute g = gcd(a^((r-1)/2) - 1, N) and h = gcd(a^((r-1)/2) + 1, N)
6. Output the factors g and h

```
function shors_algorithm(number):
    # Find a factor of the given number
    prime_factors = []
    while number is not 1:
        # Choose a random integer between 1 and the number
        random_integer = randint(1, number)

        # Use the gcd function to find the greatest common divisor
        # of the random integer and the number
        gcd = greatest_common_divisor(random_integer, number)

        # If the gcd is not 1, then we have found a factor of the number
        if gcd != 1:
            # Add the factor to the list of prime factors
            prime_factors.append(gcd)
            # Divide the number by the factor
            number = number / gcd

    # Return the list of prime factors
    return prime_factors
```

Figure 1: Pseudocode Shor's Algorithm

```
function grovers_algorithm(oracle, input_size):
    # Initialize the state of the quantum system to the superposition state
    state = initialize_state(input_size)

    # Apply the oracle function to the state of the quantum system
    state = oracle(state)

    # Apply the "inversion about the mean" operation to the state of the quantum
system
    state = inversion_about_mean(state)

    # Measure the state of the quantum system
    result = measure(state)

    # Return the result of the measurement
    return result
```

Figure 2: Pseudocode Grover's Algoritm

- **Grover's algorithm for Search:**

Time complexity: O(sqrt(N))

Reasoning: The algorithm has a time complexity of O(sqrt(N)) due to the number of iterations required to find the target element [2].

Pseudo Code:

1. Choose a search space of size N and a target element t
2. Initialize a quantum register of size log(N) qubits
3. Apply the Grover iteration to the quantum register until a measurement yields t
4. Output the index of the measured element.

The Quantum Fourier transform:

Time complexity: O(log N)

Reasoning: The algorithm has a time complexity of O(log N) due to the number of qubits required to represent the input state and the number of gates required to apply the quantum Fourier transform [1].

Pseudocode:

1. Choose a quantum state |x> in the computational basis
2. Apply the quantum Fourier transform to |x> to obtain the state |y> in the Fourier basis
3. Measure |y> in the computational basis to obtain the value y
4. Output y

```
function quantum_fourier_transform(state):
    # Initialize an empty list to store the output state
    output_state = []

    # Loop over all the qubits in the state
    for i in range(state.num_qubits):
        # Initialize the output qubit to the zero state
        output_qubit = 0

        # Loop over all the basis states of the input state
        for j in range(2 ** state.num_qubits):
            # Compute the complex exponent
            exponent = -2 * pi * j * i / (2 ** state.num_qubits)

            # Compute the complex coefficient
            coefficient = cis(exponent)

            # Add the contribution of the current basis state to the output
qubit
            output_qubit += state[j] * coefficient

        # Add the output qubit to the output state
        output_state.append(output_qubit)

    # Return the output state
    return output_state
```

Figure 3: Pseudocode Quantum Fourier Transform

V. APPLICATION AND FUTURE

Quantum machine learning: This refers to the use of quantum computing to machine learning tasks like data clustering and neural network training. E. Farhi and colleagues created the first quantum machine learning algorithms in 2014. [13].

Quantum optimization: This is a reference to using quantum computers to resolve optimization issues like the traveling salesperson problem and the knapsack problem. In 2009, A. W. Harrow and associates created the first quantum optimization algorithms [14].

Quantum chemistry: This refers to simulating the behavior of chemical systems, such as molecules and materials, using quantum computers. T. K. Kim and colleagues created the first quantum chemistry algorithms in 2010 [15].

VI. CHALLENGES

- The need for large-scale quantum computers: For many quantum algorithms to significantly outperform classical algorithms in terms of speed, a high number of qubits (the quantum counterpart of classical bits) are necessary. Nevertheless, creating large-scale quantum computers is a difficult technical task, and existing quantum computers only contain a few hundred qubits at best [12].
- The challenges of dealing with errors and noise in quantum systems: Due to the sensitivity of quantum systems to noise and mistakes, quantum algorithms may perform poorly. A significant problem in the subject is creating algorithms that are resistant to various types of mistakes [8].

- The difficulty of implementing quantum algorithms in practice: Since precise control of quantum systems is needed, putting quantum algorithms into practice can be challenging. It can be difficult to do this in practice, and study on it is ongoing [10].
- The lack of practical applications for quantum algorithms: Although several quantum algorithms have been created, they currently have little real-world uses. This results from the fact that many quantum algorithms are only much quicker than conventional algorithms for large-scale problems that are unfeasible to tackle with existing technology, as well as the difficulties noted above [11].

These difficulties highlight the substantial obstacles preventing the general deployment of quantum algorithms. Despite these difficulties, several academics are striving to find solutions, and research and development in quantum algorithms are still quite active.

VII. CONCLUSION

In conclusion, the research of quantum algorithms is expanding quickly and has enormous potential for resolving challenging computing issues. Significant progress has been achieved in the creation of quantum algorithms, and several academics are attempting to create new algorithms and enhance the performance of current ones, despite the numerous obstacles and restrictions that still exist. The future of quantum algorithms is promising, and it is probable that the next several years will bring forth a lot of fascinating advances.

REFERENCES:

[1]. Coppersmith, D. (1994). An approximate Fourier transform useful in quantum factoring. IBM Journal of Research and Development, 38(4), 653-656.

[2]. Grover, L. K. (1996). A fast quantum mechanical algorithm for database search. Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC), 212-219.

[3]. Shor, P. W. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. SIAM Journal on Computing, 26(5), 1484-1509.

[4]. Aaronson, S., & Ambainis, A. (2005). Quantum search of spatial regions. Theory of Computing, 1(1), 47-79.

[5]. Biamonte, J., et al. (2017). Quantum machine learning. Nature, 549(7671), 195-202.

[6]. Das, S., & Chakrabarti, B. K. (2008). Quantum annealing and related optimization methods. Reviews of Modern Physics, 80(3), 1061-1081.

[7]. Bravyi, S., et al. (2018). Quantum algorithms for solving linear systems of equations. Nature, 561(7723), 354-361.

[8]. Endo, S., et al. (2018). Practical quantum error correction with only two extra qubits. Nature, 559(7715), 516-519.

[9]. Peruzzo, A., et al. (2014). A variational eigenvalue solver on a quantum processor. Nature Communications, 5, 4213.

[10]. Rebentrost, P., et al. (2018). Quantum computational finance: Monte Carlo pricing of financial derivatives. Physical Review Letters, 120(12), 120502.

[11]. Nielsen, M. A., & Chuang, I. L. (2010). Quantum Computation and Quantum Information: 10th Anniversary Edition. Cambridge University Press.

[12]. Preskill, J. (2012). Quantum computing in the NISQ era and beyond. Quantum, 2, 79.

[13]. Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2014). Quantum computation by adiabatic evolution. arXiv preprint arXiv:1411.4028.

[14]. Harrow, A. W., Hassidim, A., & Lloyd, S. (2009). Quantum algorithm for linear systems of equations. Physical Review Letters, 103(15), 150502.

[15]. Kim, T. K., Wei, K. M., Narang, P., & Aspuru-Guzik, A. (2010). Quantum algorithm for electronic structure calculations on a quantum computer. Nature, 467(7315), 195-198.