Efficient Unsorted Array Search via Grover's Algorithm on Quantum Simulators

Bingying Liang
Southern Methodist University
Dallas, USA
bingyingl@smu.edu

***Abstract*—This study explores the application of Grover's Algorithm for unsorted array search within quantum simulators, contrasting its efficiency against classical linear search methods. The focus is on quantifying the algorithm's performance boost in a simulated quantum environment, leveraging the Qiskit framework. Experiments are designed to generate and search unsorted arrays of varying sizes, measuring Grover's quadratic speedup in terms of iterations and computation time. The research foresees a scalability analysis to assess the algorithm's performance as array sizes increase. The expected outcome posits Grover's Algorithm as a significant improvement over classical counterparts for larger datasets, illustrating the burgeoning impact of quantum computing on search-related problems. This paper presents a systematic approach to implementing Grover's Algorithm, offering insights into practical quantum computing applications and setting the stage for future explorations into more complex quantum algorithms.**

***Keywords*—grover's Algorithm, linear search Algorithm, quantum computing**

I. Introduction

In the era of Big Data, the ability to search through vast, unsorted databases efficiently stands as a critical challenge in the field of computer science. Even though classical algorithms have provided robust solutions for sorted data structures, searching unsorted arrays the time complexity of the best classical algorithms remains linear with respect to the size of the data set. Quantum computing brings the potential solution to this challenge. In 1982, Richard Feynman [15] presented the concept of Quantum Computation. Compared with classical computers the state of the unit is zero or one, the state of a quantum computer could be zero and one or anything in between. The unique characteristic of a quantum computer can create huge power of the parallel, which means it can pursue different paths at the same time during a single calculation unit [17]. Quantum computing presents a groundbreaking alternative with the potential to dramatically enhance search efficiency through quantum parallelism and interference.

Grover's algorithm, a quantum algorithm proposed by Lov Grover in 1996 [6]. Grover's algorithm offers a promising solution for searching unsorted databases, boasting a quadratic speedup over classical approaches by reducing the search complexity to $O(\sqrt{N})$ where N is the number of items. Despite Grover's algorithm being a significant theoretical advancement, its practical application is impeded by the current infancy of quantum hardware technology. This bottleneck has led researchers to leverage quantum simulators—classical systems designed to mimic quantum behavior—as a platform for exploring quantum algorithms. For example, the qiskit simulator platform [10] provides a controlled environment for testing, which is invaluable for understanding the practical aspects of quantum computational procedures and setting the stage for their eventual implementation on real quantum computers.

In the past few years, Grover's algorithm has shown the potential for some problems based on unstructured data. Aghaei [7] modified classical dijkstra's algorithm based on Grover's algorithm which brings a more efficient algorithm. Y. Wang and M. Perkowski [18] show that on a graph coloring problem, the ternary oracle implemented in the multi-valued Grover Algorithm yields valid solutions because of the results from [19]. Jehn-Ruey Jiang [5] has pointed out that the well-known Grover algorithm can constructed with the explicit oracle to solve the Hamiltonian cycle problem for the complete graph because the quantum circuit has a quadratic speedup over the classical unstructured search algorithm for solving the same problem.

Grover's algorithm is also known as a quantum search algorithm [4], which enables this search method to be sped up substantially. It's quite important for the unsorted array problems. During the computer network, especially the link layer, such as a sliding window protocol using selective repeat, the packets transmission information is stored in the arrays [3]. When one of packets is lost, the receiver has to use linear time to check which packet is lost. Even some classical array search algorithm like binary search [2] needs to sort the array first. So the unsorted array search is an important basement of other algorithms. And depending on the quantum parallelism, it can improve the efficient of the unsorted array search.

The structure of this paper is as follows: Section 2 expands more details about the background of classical search and Grover's Algorithm. Section 3 presents the approach to use grover's algorithm on quantum simulator to improve the efficient of unsorted array search. Section 4 shows the result and analysis of the approach. Section 5 concludes the paper.

II. Background

A. Limitations of classical search

Classical search, also know as linear search or sequential search, which is an elementary search algorithm. The input is a unsorted unsorted. The algorithm compares the value one by one until a match is found. For the time complexity of linear search is $O(n)$ where n is the size of the array [2]. The best case of linear search is when the target value is at the first position of the array. The worst case of linear search is when the target value is at the end position of the array or the value is not in the array, which means the algorithm need to traverses the entire array to search the value. The linear search algorithm is in the following:

Algorithm 1 Classical Unsorted Search (Linear Search)

Input: Array $A[x_0, x_1, ..., x_{n-1}]$, Value to find v
Output: Index of w in Array A or *null* (if w is not in A)
Initialisation :
$index \leftarrow$ *null*
Search Process :
for $i = 0$ to $n - 1$ **do**
  if $(A[i] = w)$ **then**
    $index \leftarrow i$
    break Value found, exit loop
  end if
end for
return $index$

B. Grover's Algorithm

The Grover's algorithm [6] is a quantum algorithm proposed by Grover in 1996 to solve the unstructured search problem with a high probability. Suppose in $N = 2^n$ do the search. The algorithm is summarized in the following:

1) Initialize a quantum system of $n + 1$ qubits with state $|0\rangle$.
2) Apply H gate to the first n qubits, and XH to the last qubit.
3) Repeat the below steps in Grover iteration $G \approx \left\lceil \frac{\pi\sqrt{2^n}}{4} \right\rceil$ times:
   a) Apply an Oracle's operation

$$f(x) = \begin{cases} 0, & \text{if } x \neq w \\ 1, & \text{if } x = w \end{cases} \tag{1}$$

   Oracle

$$U_f = (-1)^{f(x)} |i\rangle \tag{2}$$

   which is also called black box. It flips the amplitude of the desired state.
   b) Apply the diffusion operator

$$U_s = H^{\otimes n}(2|0\rangle\langle 0| - I)H^{\otimes n} \tag{3}$$
$$= 2|s\rangle\langle s| - I \tag{4}$$

   where $|s\rangle$ is the equally weighted superposition. The operator performs an inversion by the mean, which apples on all amplitudes.
4) Measure the first n qubits.

Algorithm 2 Grover's Algorithm

Input: A black-box oracle O that marks the winner state $|w\rangle$, number of elements N
Output: The winner state $|w\rangle$
Initialisation :
Prepare a uniform superposition of all states, $\frac{1}{\sqrt{N}}\sum_{x=0}^{N-1}|x\rangle$
Oracle and Amplification :
for $k = 1$ to $approx \left\lceil \frac{\pi\sqrt{2^n}}{4} \right\rceil$ times **do**
  Apply the oracle G to mark the winner state $|w\rangle$
  Apply the Grover diffusion operator U_s for amplitude amplification
end for
Measurement :
Measure the quantum state to obtain the winner state $|w\rangle$
return The winner state $|w\rangle$ or *null* if not found

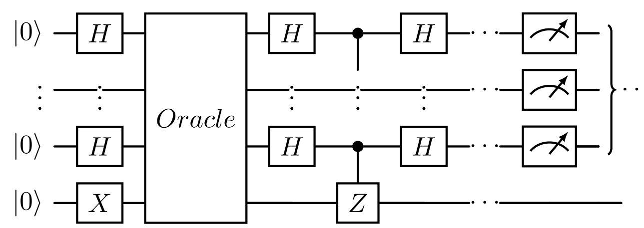

Fig. 1. Grover's Algorithm Circuit

Fig. 1 shows the circuit diagram [14] for Grover's algorithm and the pseudocode of algorithm [4] [14] is in the Algorithm 2.

For unsorted array search, Grover's Algorithm is particularly relevant as it does not necessitate any preliminary sorting or structuring of data. The algorithm effectively squares the search speed, which is profound for large datasets where classical algorithms falter. Grover's Algorithm also benefits from the intrinsic properties of quantum mechanics such as superposition and entanglement, which enables the quantum computer to evaluate multiple states simultaneously, further contributing to its search efficiency.

C. The evolution of quantum simulators

Quantum simulators have played a pivotal role in the development and testing of quantum algorithms. The progression from rudimentary simulators capable of handling only a few qubits to today's sophisticated systems marks a significant milestone in quantum computing. In the past few years, scores of frameworks, tools and platforms are emerged , improvement of currently available facilities would exploit the research activities in the quantum research community. Paramita Basak Upama [13] did a survey on these frameworks, which mentions the IBM's 5 qubits gate-level quantum processor on the web that allows the users to apply to get access to it. Quantum simulators like IBM's Qiskit simulator platforms that support the execution of quantum programs within a controlled environment.

Qiskit [9] is a framework developed by IBM, which allows users to create quantum programs and run them either on a simulation environment or IBM's real quantum devices. The normal quantum program in Qiskit usually is built in two steps: First, users should create a quantum circuit which required qubits and classical bits. The purpose of the classical bits is to store the results of measurements that will be taken later. Secondly, various gates and measurements need to be added to the circuit [10]. This paper will use Qiskit to implement algorithms.

III. Approach

A. Theoretical Analysis

1) Qubit Mappiing: From Maurice Clerc 's [8] and N. Khammassi [11], can use mathematical ways to find some relation between classical bits and quantum qubits, which is also called qubit mapping. For example, the array is $\{1, 0, 2, 3\}$. Then it also be written in binary sequence, like $\{01, 00, 10, 11\}$. In binary sequence, it will be quite a simple mapping from bits to qubits. In this way, the array actually is very convenient for mapping to the qubits and also convenient for Grover's algorithm's application.

The unsorted array search problem, in the realm of classical computing, entails scanning each element of the array until the desired value is located, typically requiring linear time complexity. Due to the mapping, this problem is transformed by Grover's Algorithm, which uses the principles of superposition and interference to search all items simultaneously, theoretically yielding a quadratic speedup with a square root time complexity. The algorithm's unique approach to processing information encapsulates the potential of quantum computing to outpace classical methods in specific computational tasks.

2) Theoretical Quantum Speedup: In the above example, in the classical machine, when searching "3" in the array, it has to run the code for 4 times. However, in the quantum machine, it can search the whole array of elements at the same time. Therefore, it just needs to run one time. With the growth of input numbers, the speedup of quantum machines is more clear. Due to the special characteristic of Grover's Algorithm, which is predicated on the quantum mechanical phenomenon of amplitude amplification, it provides a significant speedup over classical search algorithms. While a classical search would require $O(N)$ operations to find an element in an unsorted array of N items, Grover's algorithm can locate the item in approximately $O(\sqrt{N})$ operations. This quadratic speedup is especially beneficial as the size of the data set increases, where classical algorithms become impractical.

B. Resources

This approach leverages open-source quantum programming frameworks for the implementations. The primary platform will be IBM's Qiskit [9], given its compatibility with IBM's quantum machines. Access to high-performance classical computers will be essential for comparison benchmarks [12]. Compared with C-style languages, Python is easier for beginners, Qiskit supports Python, which means Qiskit may be more appropriate for beginners. [16]

The resources required for this approach include access to quantum simulators (either local or cloud-based), a development environment with Qiskit installed, and computational tools for data analysis and visualization. Additionally, sufficient computational power is necessary to handle the simulation of larger quantum circuits as the array size grows.

C. Experimental Design

Using IBM quantum simulators, which can implement Grover's Algorithm with different numbers of qubits [1]. The results will be compared with classical algorithms to observe any advantage in terms of speed or resource utilization.

1) Objective: The primary objective of the proposed experiments is to demonstrate the efficiency and effectiveness of Grover's Algorithm in searching unsorted arrays compared to classical search algorithms. The aim is to quantify the speedup offered by Grover's Algorithm when executed on quantum simulators.

2) Experimental Setup: The experiments will be conducted on a quantum simulator provided by the Qiskit framework. Given the current limitations of actual quantum hardware, simulators offer the most accessible and reliable way to test and validate quantum algorithms. The quantum simulator will emulate a noiseless quantum environment, allowing me to focus on the algorithm's ideal performance.

3) Resource Estimation: The resources required for these experiments include access to a standard computing system with sufficient processing power to run the Qiskit simulator, as well as the Qiskit software package itself. The time required to set up the experiment, write the necessary scripts, and conduct the simulations is estimated to be around 40-60 hours, spread over several weeks.

4) Procedure:

- Array Generation: Generating unsorted arrays of varying sizes, and mapping the classical bits to the qubits.
- Classical Benchmark: Perform a linear search [2] on the unsorted arrays using a classical algorithm to establish a performance benchmark. Each search will be executed multiple times to ensure statistical significance.
- Grover's Algorithm Implementation [1]: Implement Grover's Algorithm in Qiskit, ensuring the oracle is correctly constructed to mark the desired element in the superposition of states. The Fig 2 shows the basic circuit from qiskit.
- Quantum Search Execution: Run Grover's Algorithm on the quantum simulator with the generated arrays, recording the number of iterations required and the success rate of the search.
- Metrics: The performance metrics will include the number of iterations to success, the success rate, and the

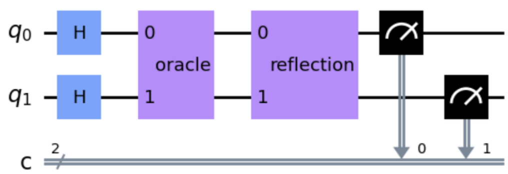

Fig. 2. 2-qubits Grover's Algorithm ciruit

computational time for both classical and quantum approaches.

- Scalability Analysis: Assess the scalability of Grover's Algorithm by increasing the size of the unsorted arrays and evaluating the growth of the resources and time required.
- Data Analysis and Comparison: Analyze the collected data to compare the performance of Grover's Algorithm against classical benchmarks. Special attention will be given to the quantum speedup ratio as a function of array size.
- Error Analysis: Although the simulator provides an idealized environment, we will also consider theoretical error sources that could affect the algorithm's performance on actual quantum hardware.

5) Deliverables: The final deliverables will include a detailed report documenting the experiment design, methodology, raw data, data analysis, findings, and a comprehensive discussion of the implications of Grover's Algorithm for unsorted array search problems.

IV. Expected Results

The expected result is that Grover's Algorithm will exhibit a quadratic speedup in the search of an unsorted array compared to classical search algorithms. The experimental results are expected to yield a clear trend that demonstrates the efficiency of Grover's Algorithm as the size of the search space increases. And showing the potential of combining classical algorithms.

V. Summary

This proposal examines the application of Grover's Algorithm for unsorted array search on quantum simulators, aiming to validate its theoretical quadratic speedup over classical search methods. Utilizing the Qiskit platform, the project will simulate searches across arrays of increasing sizes, focusing on iteration count and computation time to quantify performance enhancements. Preliminary simulations will establish baseline efficiencies, crucial for refining the approach for larger datasets. Expected outcomes suggest that Grover's Algorithm will significantly outperform classical algorithms, particularly for large-scale searches. Future work will consider quantum noise impacts and extend investigations into more complex search domains, furthering the practical understanding of quantum computing's potential.

References

[1] A. Mandviwalla, K. Ohshiro, and B. Ji, "Implementing Grover's Algorithm on the IBM Quantum Computers," in 2018 IEEE International Conference on Big Data (Big Data), Seattle, WA, USA: IEEE, Dec. 2018, pp. 2531–2537. doi: 10.1109/BigData.2018.8622457.

[2] A. E. Jacob, N. Ashodariya, and A. Dhongade, "Hybrid search algorithm: Combined linear and binary search algorithm," in 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS), Chennai: IEEE, Aug. 2017, pp. 1543–1547. doi: 10.1109/ICECDS.2017.8389704.

[3] A. S. Tanenbaum, N. Feamster, and D. Wetherall, Computer networks, Sixth edition, Global edition. Harlow, United Kingdom: Pearson, 2021.

[4] I. L. C. Michael A. Nielsen, Quantum Computation And Quantum Information, 10th Anniversary Edition. Cambridge University Press, 2010.

[5] J.-R. Jiang, "Quantum Circuit Based on Grover Algorithm to Solve Hamiltonian Cycle Problem," in 2022 IEEE 4th Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan: IEEE, Oct. 2022, pp. 364–367. doi: 10.1109/ECICE55674.2022.10042919.

[6] L. K. Grover, "A fast quantum mechanical algorithm for database search," In Proceedings of the twenty-eighth annual ACM symposium on Theory of computing, pp. 212-219, 1996

[7] M. R. S. Aghaei, Z. A. Zukarnain, A. Mamat, and H. Zainuddin, "A Hybrid Algorithm for the Shortest-Path Problem in the Graph," in 2008 International Conference on Advanced Computer Theory and Engineering, Phuket, Thailand: IEEE, Dec. 2008, pp. 251–255. doi: 10.1109/ICACTE.2008.137.

[8] M. Clerc, "Graph colouring: a polynomial complexity quantum algorithm," 2023, doi: 10.13140/RG.2.2.36154.77760.

[9] M. S. A. et al., "Qiskit: An open-source framework for quantum computing," 2021.

[10] M. Kashif and S. Al-Kuwari, "Qiskit As a Simulation Platform for Measurement-based Quantum Computation," in 2022 IEEE 19th International Conference on Software Architecture Companion (ICSA-C), Honolulu, HI, USA: IEEE, Mar. 2022, pp. 152–159. doi: 10.1109/ICSA-C54293.2022.00037.

[11] N. Khammassi, I. Ashraf, X. Fu, C. G. Almudever, and K. Bertels, "QX: A high-performance quantum computer simulation platform," in Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017, Lausanne, Switzerland: IEEE, Mar. 2017, pp. 464–469. doi: 10.23919/DATE.2017.7927034.

[12] "New IBM, UC Berkeley paper shows path toward useful quantum," IBM Research Blog. Accessed: Nov. 06, 2023. [Online]. Available: https://research.ibm.com/blog/utility-toward-useful-quantum

[13] P. B. Upama et al., "Evolution of Quantum Computing: A Systematic Survey on the Use of Quantum Computing Tools," in 2022 IEEE 46th Annual Computers, Software, and Applications Conference (COMPSAC), Los Alamitos, CA, USA: IEEE, Jun. 2022, pp. 520–529. doi: 10.1109/COMPSAC54236.2022.00096.

[14] R. H. Preston, "Applying Grover's Algorithm to Hash Functions: A Software Perspective," IEEE Trans. Quantum Eng., vol. 3, pp. 1–10, 2022, doi: 10.1109/TQE.2022.3233526.

[15] R. P. Feynman, "Simulating physics with computers," Int. J. Theor. Phys., vol. 21, no. 6–7, pp. 467–488, 1982.

[16] R. LaRose, "Overview and Comparison of Gate Level Quantum Software Platforms," Quantum, vol. 3, p. 130, Mar. 2019, doi: 10.22331/q-2019-03-25-130.

[17] S. B. Ramezani, A. Sommers, H. K. Manchukonda, S. Rahimi, and A. Amirlatifi, "Machine Learning Algorithms in Quantum Computing: A Survey," in 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, United Kingdom: IEEE, Jul. 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207714.

[18] Y. Wang and M. Perkowski, "Improved Complexity of Quantum Oracles for Ternary Grover Algorithm for Graph Coloring," in 2011 41st IEEE International Symposium on Multiple-Valued Logic, Tuusula, Finland: IEEE, May 2011, pp. 294–301. doi: 10.1109/ISMVL.2011.42.

[19] Y. Fan, "Applications of Multi-Valued Quantum Algorithms," in 37th International Symposium on Multiple-Valued Logic (ISMVL'07), May 2007, pp. 12–12. doi: 10.1109/ISMVL.2007.3.