A Benchmarking Study of Grover's Algorithm for Solving Boolean SAT with Quantum Circuits

Jirapas U. Jipipob
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand
6670037521@student.chula.ac.th

Prabhas Chongstitvatana
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand
prabhas.c@chula.ac.th

Abstract*— This study explores how well Grover's Algorithm performs in solving the Boolean Satisfiability Problem (SAT) using quantum circuits. The algorithm is implemented with IBM's Qiskit framework and compared to classical brute-force methods. Experiments focus on 3-SAT, 4-SAT, and 5-SAT problems, using quantum simulators and IBM quantum hardware. The results show that Grover's Algorithm is more efficient, offering a theoretical quadratic speedup over classical methods. However, practical issues like limited qubit availability, hardware noise, and optimization challenges impact its current performance. The data highlights the potential for quantum computing to scale and solve NP-complete problems. This research shows how quantum computing can improve problem-solving and lays the groundwork for future studies on more complex SAT problems and advanced quantum hardware.

***Keywords*—quantum circuits, quantum noise, quantum simulators, computational efficiency, classical brute-force methods**

I. Introduction

Quantum computing is transforming how we solve complex problems, offering capabilities far beyond classical systems. This paper examines Grover's Algorithm, a quantum search algorithm known for its quadratic speedup, and its application to the Boolean Satisfiability Problem (SAT). SAT, a fundamental NP-complete problem, is critical in optimization, logic, and computational theory. Classical methods like brute force often struggle with the computational demands of SAT as problem size increases.

Using IBM's Qiskit framework, this study implements Grover's Algorithm to solve 3-SAT, 4-SAT, and 5-SAT problems, benchmarking its performance against classical brute-force methods. Experiments are conducted on quantum simulators and IBM hardware to evaluate computational efficiency and scalability. Despite current hardware limitations, the results highlight quantum computing's potential for solving NP-complete problems, marking a significant step toward practical quantum algorithm applications.

II. Background Knowledge

This section is divided into six parts. First, it will cover the Boolean Satisfiability Problem (SAT) and its complexity. Second, it will discuss classical approaches to SAT, focusing on brute-force methods and their limitations. The third section will introduce Grover's Algorithm and its potential speedup for solving SAT problems with quantum computing. Fourth, it will explore quantum computing and Qiskit, detailing the principles of quantum computation and the use of Qiskit for simulating quantum algorithms. Fifth, it will examine quantum entanglement and its role in quantum processes. Finally, the sixth section will discuss NP-Complete problems and their implications for SAT.

A. Boolean Satisfiability Problem (SAT)

First, The Satisfiability (SAT) problem is one of the most critical issues in computational theory, particularly in logic, constraint satisfaction, and various areas such as VLSI design and machine learning. SAT is the task of determining whether there exists an assignment of truth values to variables that makes a given Boolean formula satisfiable. This formula is typically expressed in Conjunctive Normal Form (CNF), where clauses are composed of literals (variables or their negations) connected by logical OR, and the overall formula is a conjunction (AND) of these clauses. SAT is central to a family of NP-complete problems, meaning that while a solution can be verified quickly, finding the solution itself may require non-deterministic polynomial time in the worst case. It has broad applications, including optimization, artificial intelligence, and circuit design. Traditional methods for solving SAT focus on treating it as a decision problem, employing algorithms such as resolution-based techniques. However, more recent approaches have transformed the SAT problem into an optimization problem, where the objective is to minimize the number of unsatisfied clauses. This allows the use of iterative optimization techniques, including local search methods, which have shown increased efficiency for specific classes of SAT formulas compared to classical approaches [1].

B. Classical Approaches to SAT

First, The Classically, SAT problems are often solved using algorithms like the brute-force method, which explores every possible combination of truth values, making it highly inefficient for large problems. More sophisticated algorithms, like the Davis-Putnam-Logemann-Loveland (DPLL) algorithm [2] and its variations, attempt to reduce the search space by recursively simplifying the problem. However, even these optimized algorithms struggle with the computational demand

979-8-3315-2223-0/25/$31.00 ©2025 IEEE

as the problem size increases. SAT solvers are widely used, but they have limitations when dealing with large, complex instances due to the exponential time complexity associated with NP-complete problems.

C. Grover's Algorithm

Grover's Algorithm, introduced by Lov Grover in 1996 [3], is a quantum search algorithm that offers a quadratic speedup over classical search algorithms. It is designed to search through an unsorted database or solution space of size N in O(√N) time, making it a powerful tool for solving problems like SAT. While classical brute-force methods for SAT operate in exponential time (O(2^n) for n variables), Grover's Algorithm can reduce the time complexity to O(√ 2^n), offering a significant performance advantage, particularly for large problem instances. Grover's Algorithm is particularly suited for structured search problems and demonstrates the potential of quantum computing to solve NP-complete problems more efficiently than classical methods.

D. Quantum Computing and Qiskit

Quantum computing leverages the principles of superposition and entanglement to perform operations on quantum bits (qubits), which can represent both 0 and 1 simultaneously. This allows quantum computers to process vast amounts of data in parallel, which classical computers cannot achieve [4]. Qiskit is an open-source quantum computing framework developed by IBM [5], which provides tools for designing, simulating, and running quantum circuits. Qiskit's quantum simulators enable the testing of quantum algorithms, such as Grover's, in a controlled environment before deploying them on actual quantum hardware. The framework is essential for experimenting with quantum algorithms and comparing their performance to classical methods.

E. Quantum Entanglement

Quantum entanglement, a foundational concept in quantum mechanics, refers to the non-classical correlations between subsystems of a compound quantum system. Initially recognized by Einstein, Podolsky, and Rosen, it took over 70 years to be fully appreciated as a tangible resource with profound implications for quantum processes. Entanglement plays a central role in quantum cryptography, teleportation, and dense coding, offering potential for advances in quantum communication and computation. Despite its complexity and environmental fragility, entanglement is robust in theoretical frameworks, with tools like Bell inequalities and entanglement witnesses used for its detection and characterization. The irreversibility of entanglement manipulations, particularly in the context of bound entanglement, highlights its unique role in quantum communication and computation [6].

F. NP-Complete

NP-Complete problems are a pivotal class within computational complexity theory [7]. A problem is in NP if, given a solution, it can be verified in polynomial time. Any problem in NP can be transformed into an NP-Complete problem in polynomial time. A problem P is NP-Complete if it is in NP and every problem Q in NP can be reduced to P using a polynomial-time transformation. This indicates that if any NP-Complete problem can be solved in polynomial time, all NP problems can also be solved in polynomial time, effectively establishing that $P = NP$ [8].

III. Related Work

We review the foundational and contemporary works relevant to the study of satisfiability problems (SAT) and the application of quantum computing, particularly Grover's algorithm. It examines the classical complexity of SAT, the emergence of quantum computing, and the challenges and advancements in applying quantum algorithms to solve SAT problems.

A. Satisfiability Problem

The In the field of computational complexity, the Satisfiability Problem (SAT) plays a central role due to its wide applicability in logic, optimization, artificial intelligence, and computational theory. The work of Thomas J. Schaefer in "The Complexity of Satisfiability Problems" [9] made a significant contribution to this area by classifying a broad range of SAT problems and demonstrating that each problem in this infinite class is either polynomial-time solvable or NP-complete, with no intermediate complexity classes.

Schaefer's paper specifically explores the distinction between SAT instances with clauses restricted to two literals, which are efficiently solvable in polynomial time, and SAT instances with three literals per clause, which are proven to be NP-complete. This establishes a foundational result in that SAT problems with larger clause sizes are generally more computationally challenging and belong to the NP-complete class, for which no efficient (polynomial-time) solution is known unless P equals NP. Schaefer's work broadens this understanding by presenting a classification theorem that applies to an infinite family of SAT problems, determining whether a given SAT problem is in the polynomial-time solvable class or NP-complete, depending on the specific structure and constraints of the propositional formulas.

Additionally, Schaefer extends this analysis to quantified SAT problems, a more complex version involving quantifiers, and demonstrates that these problems are either solvable in polynomial time or require exponential space. The results from this paper serve as a framework for identifying new NP-complete problems, as well as polynomial-time problems, offering a deeper understanding of computational complexity in logic-based problem solving. Schaefer's classification theorem has been a critical tool in the study of satisfiability problems and has inspired extensive research in both theoretical and practical aspects of computational complexity.

B. The Significance of Quantum Computing

The field of quantum computing has emerged as a revolutionary area of research, blending concepts from classical information theory, computer science, and quantum physics. In a paper, Quantum Computing [10], Andrew Steane highlights the pivotal role quantum computing plays in reshaping our understanding of computation and the natural world, particularly by integrating the concept of quantum information into the computational domain.

Steane begins by positioning quantum computing within the broader framework of information theory, tracing its roots back to mid-20th century developments in classical information theory and computer science. Classical theories of computation, such as Turing machines and Shannon's information theory, provide the foundation for understanding quantum computing's departure from classical systems. The difference is most notably captured in the Einstein, Podolsky, and Rosen (EPR) experiment and the EPR-Bell correlations [11], which distinguish quantum from classical physics. Quantum entanglement, as Steane notes, is a core component in these divergences, enabling new forms of computation and information transfer.

A central theme in Steane's review is the quantum bit (qubit), which serves as the fundamental unit of quantum information. Unlike classical bits, which are binary and exist as 0 or 1, qubits can exist in superpositions of states. This principle underlies the significant.

C. *Exploring Grover's Algorithm*

In recent years, quantum computing has emerged as a transformative field with the potential to outperform classical algorithms in various applications, particularly in data processing and optimization. One notable quantum algorithm is Grover's algorithm, which offers a quadratic speedup for unstructured search problems compared to classical counterparts. Grover ' s algorithm operates on a quantum superposition of states, allowing it to search through N items in approximately $\sqrt{N}$ queries, a significant advantage over the O(N) time complexity required by classical search algorithms.

The theoretical underpinnings of Grover's algorithm demonstrate its applicability across various domains, including cryptography, database searching, and optimization problems. As the data landscape expands, with datasets reaching and surpassing petabytes, the demand for efficient algorithms like Grover's is increasingly pressing. This need is particularly pronounced in areas such as machine learning and artificial intelligence, where rapid data processing is crucial. Recent studies have focused on implementing Grover's algorithm on real quantum hardware, such as IBM's quantum computers. Mandviwalla et al. explore the practical application of Grover's algorithm through multiple implementations on IBM Q devices, providing empirical results that reflect the capabilities and limitations of current quantum technology. Their research highlights the challenges of achieving theoretical accuracy in practical applications, where factors such as qubit coherence and error rates play critical roles [12].

D. *Grover's Algorithm's Impact on SAT Problems*

The satisfiability problem, particularly in its NP-complete form, poses significant challenges in classical computing. Specifically, determining whether a given Boolean formula in conjunctive normal form (CNF) is satisfiable involves searching through an exponential number of potential variable assignments. Grover's algorithm can be employed to enhance the efficiency of this search process, offering a promising approach to tackle the SAT problem.

Cheng and Tao (2024) investigate the application of Grover's algorithm specifically for 3-SAT, a variant of SAT where each clause contains exactly three literals. They highlight the limitations imposed by current quantum technology, particularly the number of stable qubits available for practical implementations. The authors point out that the performance of Grover's algorithm is directly linked to the size of the oracle used and the number of repeated calls to it. This creates a constraint on the number of qubits that can be effectively employed without sacrificing performance [13].

E. *Applications of Grover's Algorithm Beyond SAT Problems*

One notable application of Grover's algorithm is in database search and pattern matching. Tezuka et al. (2024) proposes a novel approach that implements Grover's algorithm for image pattern matching, demonstrating its potential to enhance data retrieval processes significantly. The authors utilize an approximate amplitude encoding method in a shallow quantum circuit, enabling efficient data loading and amplitude amplification. This adaptation addresses the challenges previously faced in realizing the original motivations behind Grover's algorithm in practical settings [14].

The algorithm operates by encoding the data in a quantum state that resembles the query, followed by an amplitude amplification process independent of the target data index. This approach not only highlights the algorithm's capability in traditional database search scenarios but also showcases its application in more complex contexts such as image processing, where pattern recognition and matching are crucial.

F. *Heuristic Method Limitations*

The paper "Computers and Intractability" by Garey and Johnson (1979) highlights key limitations of heuristic methods like WalkSAT and DOLL when applied to combinatorial optimization problems such as SAT. These heuristics, often used to find approximate or locally optimal solutions, do not guarantee finding the global optimum or even all solutions. WalkSAT, for example, relies on a probabilistic approach with random flips to explore the solution space, but it is prone to getting stuck in local minima, limiting its effectiveness. As these heuristics typically search only a small portion of the solution space, they may miss valid solutions or underperform on specific problem instances. In contrast, Grover's algorithm offers a significant advantage by providing a quantum speedup that allows it to systematically explore the entire solution space, with a high probability of success, and guarantees the discovery of the correct solution if run enough times. This fundamental difference—heuristics offering approximate solutions versus Grover's exact search capability—makes heuristics less suitable for direct comparison with Grover's algorithm, which is specifically designed for exhaustive, exact search [15].

G. *Classical Brute Force vs Classic Heuristics*

One notable application of quantum computing in solving SAT problems is the use of Grover's algorithm, which provides a quadratic speedup over classical brute force methods. While classical heuristics like WalkSAT and DPLL may perform well in practice, they are limited by their inability to guarantee a solution and potential performance bottlenecks as problem size increases. Montanaro (2016) discusses how Grover's algorithm offers a more scalable solution than classical methods, especially for larger instances, and suggests that benchmarking quantum algorithms against brute force is crucial because it provides a guaranteed solution, unlike heuristic methods [16].

H. Quantum Supremacy

One notable application of quantum computing in solving Arute et al. (2019) demonstrated quantum supremacy by utilizing a noisy quantum processor to solve a specific random circuit sampling task, outperforming classical methods. This achievement marked a significant milestone in quantum computing, illustrating that quantum systems, despite being imperfect and subject to noise, can still provide a computational advantage for certain problems. While the quantum processor used in this experiment was not error-corrected, the results emphasize the potential of quantum computing to surpass classical methods in specific contexts, even with current noisy hardware. The experiment highlights the importance of continuing to explore the capabilities of quantum systems in their present noisy state, as it offers valuable insights into the future development of quantum computing [17].

IV. Experimental Methodology

We outline the experimental methodology employed to evaluate the performance of Grover's algorithm on the satisfiability problems (3-SAT, 4-SAT, and 5-SAT). Utilizing Qiskit, we conduct experiments on both a quantum simulator and actual quantum hardware, when feasible. This allows us to compare the efficiency of Grover's algorithm against classical brute-force methods in solving these SAT problems.

A. Overview of the Experimental Setup

For the quantum simulations, we employed Qiskit, an open-source quantum computing framework, running on Google Colab with the Qiskit Aer Simulator as the backend. This environment allowed us to efficiently prototype and test Grover's algorithm across different SAT problem instances. The flexibility of Qiskit enables the creation of quantum circuits tailored to the specific requirements of each problem, facilitating quick iterations and optimizations.

To evaluate the performance of Grover's algorithm on actual quantum hardware, we used IBM's quantum processor, specifically the IBM Quantum System One, codenamed “Sherbrooke”. This platform provides access to real quantum devices, allowing us to compare results obtained from simulations with those executed on a physical quantum computer. The Sherbrooke device offers a limited number of qubits, which is crucial for the implementation of Grover's algorithm, particularly for larger SAT problems.

For the classical brute-force approach, we implemented the SAT solving algorithm using Python in Google Colab. This setup enabled us to efficiently explore all possible combinations of variable assignments to determine the satisfiability of each SAT instance. By running the classical algorithm alongside the quantum implementations, we aimed to draw direct comparisons between their performance metrics.

The experiments were designed to collect data on execution time and success rates for each algorithm across varying problem sizes. This comprehensive approach enables us to analyze the impact of problem size on computational speed and to predict the advantages of quantum computing as the number of qubits increases.

B. Implementation of Grover’s Algorithm

To begin, we express each SAT instance in a human readable format known as the Boolean string format. This format represents the SAT problem as a conjunction (AND) of disjunctions (OR), where each clause consists of literals, either as variables or their negations. Once the SAT problem is expressed in this format, we translate it into a quantum circuit using Qiskit's PhaseOracle function. The PhaseOracle generates a corresponding quantum oracle that marks the satisfying assignments (solutions) of the SAT problem.

Next, Grover’s algorithm is applied to search for a satisfying solution. A register of qubits is prepared, with each qubit corresponding to one of the variables in the SAT problem. A Hadamard gate is then applied to each qubit to create an equal superposition of all possible states. This allows the algorithm to explore all potential variable assignments simultaneously, thus leveraging quantum parallelism.

Before executing the quantum circuit, it must be optimized and transpiled for the target quantum hardware. The Qiskit transpiler is used to map the high-level circuit onto the hardware, minimizing gate errors and decoherence [18]. In our implementation, we set the optimization level to 3 to achieve the highest degree of optimization for both the simulator and quantum hardware. This step ensures that the circuit layout is optimized, reducing gate count and improving performance, which is crucial for obtaining accurate results, especially when using real quantum hardware with limited qubit coherence time.

C. Experimental Procedure

This section outlines the procedure for conducting experiments to solve 3-SAT, 4-SAT, and 5-SAT problems using both quantum and classical methods. The goal is to compare the performance of Grover's algorithm implemented in Qiskit on a quantum simulator and actual quantum hardware against classical brute-force algorithms implemented in Python.

The first set of experiments is conducted using the Qiskit framework to implement Grover's algorithm on a quantum simulator as well as on IBM’s Sherbrooke quantum hardware. To compare the quantum results with classical methods, a brute-force algorithm is implemented in Python. The brute-force algorithm systematically checks every possible combination of variable assignments to find the solution that satisfies the SAT problem.

After gathering the experimental results from both quantum and classical approaches, the data will be compared in terms of execution time and accuracy. The time taken to execute Grover's algorithm on the Qiskit simulator and quantum hardware will be plotted against the size of the SAT problem (i.e., number of variables and clauses). The performance of the quantum simulator and quantum hardware will also be compared. Similarly, the time taken by the brute-force method in Python will be plotted against the size of the SAT problem.

For the quantum experiments, the success rate of Grover's algorithm (i.e., how often the correct solution is found) will also be recorded and compared against theoretical expectations. The execution times for both quantum and classical approaches will be plotted on a graph to visualize the difference in scaling as the SAT problem size increases. This chart will help illustrate how quantum methods scale with increasing problem size in comparison to classical brute-force methods.

D. Data Collection and Visualization

Data collection was performed systematically during the experiments, ensuring that key metrics were recorded for each test case. For the quantum implementations, we gathered data on execution times from both the Qiskit simulator and the IBM Sherbrooke quantum processor. Key metrics included execution time, which is the total time taken for the algorithm to complete, measured in milliseconds, and success rate, which is the percentage of successful runs that resulted in the correct output for the given SAT instance.

To facilitate the analysis of the data collected, we utilized graphical representations. Various types of visualizations were employed, including bar charts to compare the execution times of Grover's algorithm against the classical method for each problem size, and line graphs to illustrate trends in execution time as the size of the SAT problem increases, highlighting the potential benefits of quantum speedup. These visualizations aided in understanding how the performance of Grover's algorithm scales with problem size and the implication of increased qubit counts on computational efficiency. By analyzing these results, we aimed to derive conclusions regarding the potential of quantum computing to outperform classical methods in solving NP-complete problems.

E. Limitations and Challenge

While the experiments conducted in this paper offer valuable insights into the performance of Grover's algorithm for solving SAT problems, there are several key limitations and challenges that affect the scope and accuracy of the results.

One of the most significant limitations of this study is the restricted number of qubits available on current quantum hardware. As Grover’s algorithm requires an increasing number of qubits to represent larger SAT problems (such as 6-SAT or higher), the available quantum computers, such as IBM Sherbrooke, do not have enough qubits to experiment with high-complexity SAT instances. This limitation prevents us from exploring higher-order k-SAT problems and fully testing the scalability of Grover’s algorithm on such problems.

Quantum computers available to the public, such as those provided by IBM, often have limited access times. The restricted usage time for the quantum hardware means that only a limited number of SAT problem instances could be run and tested, making it difficult to conduct large-scale experiments or repeat experiments extensively to improve accuracy. Additionally, the current error rates in quantum hardware can lead to less reliable results, further contributing to the challenges of performing high-precision experiments.

Due to noise and decoherence in quantum systems, the results of the quantum experiments may not always be accurate or consistent with theoretical expectations [19]. While Grover’s algorithm is designed to provide quadratic speedup, this advantage can be diminished on real quantum hardware by issues like gate errors and readout errors. As a result, the outcomes presented in this paper are more indicative of the potential future performance of quantum computers rather than an exact measure of their current capabilities.

The study focuses on 3-SAT, 4-SAT, and 5-SAT problems, which are relatively small instances due to the limitations of both classical and quantum computing resources. As a result, the experimental data may not fully represent how Grover’s algorithm would perform on larger, real-world SAT problems. While the results provide insight into how increasing problem size affects execution time, the conclusions drawn here are more predictive and hypothetical for future experiments with more powerful quantum hardware.

Given the limitations mentioned above, this paper emphasizes the collection of data to make predictions about the future of quantum computing rather than delivering concrete, definitive results. The experiments conducted serve as a preliminary foundation for future research. The predictions made in this paper are based on extrapolations of the data collected from small-scale experiments, and more comprehensive studies are required once quantum computers with more qubits and greater stability become available.

V. RESULT

We collected results from the benchmark experiment. The experiment involved running 15 SAT problems on classical brute force, the Qiskit simulator, and the IBM Sherbrooke. These problems can be separated into three groups: 3-SAT, 4SAT, and 5-SAT, each consisting of five problems. The results of each group were averaged across its five problems.

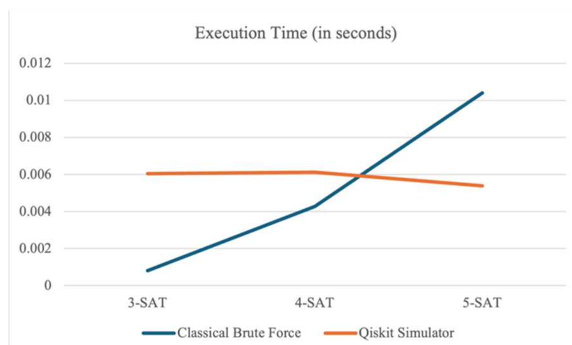

Fig. 1. Execution time for classical brute force and the Qiskit simulator

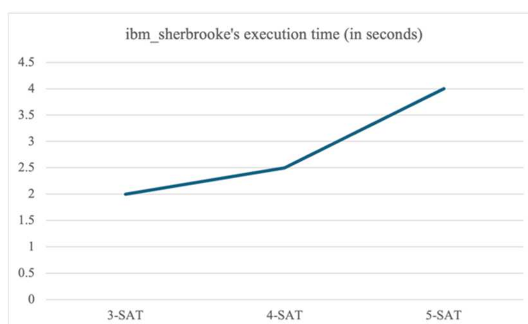

Fig. 2. Execution time for ibm_sherbrooke

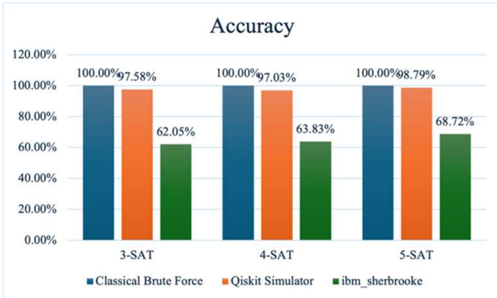

Fig. 3. Accuracy

It is important to note the following disclaimers regarding the accuracy of the data collected. The experiments were conducted over a short period, resulting in a limited number of problem instances for testing. For the quantum algorithm, we used a fixed number of shots (1024) with a custom parameter setup that may not be optimized. The execution times reported for the real quantum hardware may not be entirely accurate, as we utilized time data from the built-in metrics function, which displays times in seconds without floating-point precision. Additionally, a fixed optimization level (3) was used for both the simulator and actual hardware, which may affect the performance results.

VI. Conclusion

The results show that classical brute force is the most efficient approach, achieving the fastest execution times and perfect accuracy across all problem sizes. The Qiskit simulator, while slower than classical brute force, demonstrated near-perfect accuracy, making it a viable option for quantum simulations, though its execution times are not yet competitive for small problem sizes. On the other hand, IBM Sherbrooke quantum hardware exhibited significantly slower execution times and lower accuracy compared to both classical methods and the simulator, suggesting that quantum hardware, as of now, is not yet suitable for solving SAT problems of these sizes effectively.

These findings highlight the current dominance of classical methods in terms of both execution time and accuracy, with quantum simulators offering a promising but slower alternative. The results also emphasize the need for further advancements in quantum hardware and algorithms to make quantum computing competitive for SAT problem-solving tasks.

Future work should focus on optimizing quantum hardware and developing more efficient quantum algorithms to improve both the execution time and accuracy for larger and more complex problem instances. Grover's Algorithm, while theoretically offering a quadratic speedup, faces significant scalability challenges as the clause size increases beyond problems like 3-SAT. As we approach 5-SAT and beyond, the size of the search space grows exponentially, demanding more qubits, deeper quantum circuits, and greater coherence times than current hardware can support. Furthermore, encoding large SAT instances into quantum oracles becomes increasingly complex, and error rates compound with circuit depth, further diminishing practical performance. Addressing these issues will be critical to realizing the full potential of Grover's Algorithm for real-world applications. Research into error mitigation, more compact oracle representations, and hybrid quantum-classical strategies may be key to overcoming these scalability limitations soon.

Acknowledgment

This research is supported by Chulalongkorn University

References

1. J. Gu, "Local search for satisfiability (SAT) problem," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 4, pp. 1108–1129, 1993.
2. R. Nieuwenhuis, A. Oliveras, and C. Tinelli, "Solving SAT and SAT modulo theories: From an abstract Davis--Putnam--Logemann--Loveland procedure to DPLL(T)," *Journal of the ACM (JACM)*, vol. 53, no. 6, pp. 937–977, 2006.
3. L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the twenty-eighth annual ACM symposium on Theory of computing*, 1996, pp. 212–219.
4. J. Gruska, *Quantum Computing*. London, U.K.: McGraw-Hill, 1999.
5. R. Wille, R. Van Meter, and Y. Naveh, "IBM's Qiskit tool chain: Working with and developing for real quantum computers," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 2019, pp. 1234–1240.
6. R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki, "Quantum entanglement," *Reviews of modern physics*, vol. 81, no. 2, pp. 865–942, 2009
7. D.-Z. Du and K.-I. Ko, *Theory of computational complexity*. John Wiley & Sons, 2011.
8. C. H. Papadimitriou, "Computational complexity," in *Encyclopedia of computer science*, 2003, pp. 260–265.
9. T. J. Schaefer, "The complexity of satisfiability problems," in *Proceedings of the 10th Annual ACM Symposium on Theory of Computing*, 1978, pp. 216–226.
10. A. Steane, "Quantum computing," *Reports on Progress in Physics*, vol. 61, no. 2, pp. 117–173, 1998.
11. R. B. Griffiths, "EPR, Bell, and quantum locality," *American Journal of Physics*, vol. 79, no. 9, pp. 954–965, 2011.
12. 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, 2018, pp. 2531–2537.
13. S.-T. Cheng and M.-H. Tao, "Quantum cooperative search algorithm for 3-SAT," *Journal of Computer and System Sciences*, vol. 73, no. 1, pp. 123–136, 2007.
14. H. Tezuka, K. Nakaji, T. Satoh, and N. Yamamoto, "Grover search revisited: Application to image pattern matching," *Physical Review A*, vol. 105, no. 3, p. 032440, 2022.
15. M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: W. H. Freeman, 1979.
16. A. Montanaro, "Quantum algorithms: an overview," npj Quantum Information, vol. 2, no. 1, pp. 1-8, 2016.
17. F. Arute *et al*., "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019.
18. N. Dilillo, E. Giusto, E. Dri, B. Baheri, Q. Guan, B. Montrucchio, and P. Rech, "Understanding the Effect of Transpilation in the Reliability of Quantum Circuits," in *2023 IEEE International Conference on Quantum Computing and Engineering (QCE)*, 2023, vol. 2: IEEE, pp. 232-235.
19. S. Resch and U. R. Karpuzcu, "Benchmarking quantum computers and the impact of quantum noise," *ACM Computing Surveys (CSUR)*, vol. 54, no. 7, pp. 1–35, 202