1 Introduction

Quantum computing has meant a paradigm shift in computer science, introducing a new era of computational power and possibilities. By leveraging the principles of quantum mechanics, quantum computing offers capabilities far beyond those of classical computers. Although quantum hardware is still evolving, Noisy Intermediate-Scale Quantum (NISQ) [1] computers have already begun to unveil the potential of quantum computing, igniting widespread interest in implementing quantum algorithms that promise significant speedups over classical counterparts. In particular, the speedup of solving NP-complete problems is an area that quantum computing can tackle. In this paper, we have focused on the Hamiltonian Cycle Problem.

This problem is one of Karp's 21 seminal NP-complete problems, it involves determining whether a given graph contains a Hamiltonian cycle—a cycle that visits every vertex exactly once. The problem is significant because of its similarities with the TSP, and its applications span various domains, including logistics and network optimization, where identifying optimal cycles is crucial AGREGAR CITAS. Traditional solution methods range from brute force approaches to sophisticated algorithms like dynamic programming and Monte Carlo simulations. In this paper, we propose utilizing Grover's algorithm to tackle the HC problem, offering the potential for quadratic speedup.

The structure of this paper is as follows. An explanation of Grover's algorithm is given in Section II. The proposed methodology is given in Section III. Section IV explores the existing solutions for the problem and compares the theoretical results.

2 Background

2.1 Grover's Algorithm

The search algorithm developed by Lov K. Grover in 1997 [2] is a milestone in the development of quantum algorithms. It solves the problem of finding a tagged element (or M elements) in a disordered set of N elements using $k_{Gr} = \lfloor \frac{\pi}{4}\sqrt{\frac{N}{M}} \rfloor$ times a system, called oracle, which is able to identify this element. It presents a quadratic improvement in order relative to the classical brute-force search algorithm, which requires in the worst-case scenario N system calls. Grover's algorithm does not converge with 100% probability to the desired state, it converges with an arbitrarily high probability that depends on the relationship between solutions and the size of the search space.

As Grover proposed later, the algorithm can be used to speed up the solution of NP-complete problems [3]. Since, its application has been explored with some problems such as Clique problem [4, 5], k-coloring [6, 7] and SAT [8, 9]. Nielsen and Chuang predicted the solution of the HC problem using Grover[10].

The original problem presented by Grover [2] is as follows. Let there be a set of $N = 2^n$ quantum states in a Hilbert space ($\mathcal{H}_n$), where n is the number of qubits, and an unknown canonical basis state marked between them (solution to the problem). Given a system, an operator, called an oracle that identifies the marked element, the objective is to find this marked element with high probability, using as few queries to the oracle as possible.

Let $|t\rangle$ be the marked element of the base, and $|s\rangle$ the uniform superposition of all basis states. An oracle operator $O = I_d - 2\,|t\rangle\,\langle t|$ is applied and changes the phase of the marked state. A diffusion operator $D = I_d - 2\,|s\rangle\,\langle s|$ is applied and amplifies the amplitude of the marked state. Applying the Grover operator $(G = DO)$ k_{Gr} times maximizes the probability of measuring $|t\rangle$.
A more detailed and extended explanation for when the set has more than one solution can be found in reference [10].

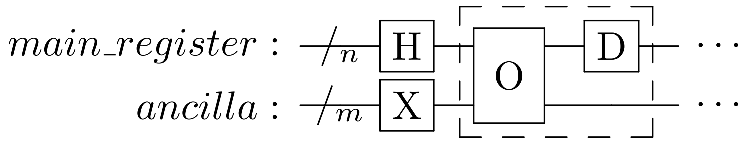


Figure 1: Grover's algorithm circuit, the Grover operator $G = DO$ is applied k_{Gr} times.

2.2 Hamiltonian Cycle Problem

Explicamos qué es el Hamiltonian Cycle Problem. Explicamos cuáles son los principales usos de resolver el HCP. Consultamos las heurísticas actuales de resolución del HCP.

3 Proposed Algorithm for Hamiltonian Cycle Problem

In order to solve a NP-complete problem with Grover's algorithm, it is necessary to translate a decision problem into a search problem. First, every possible cycle is encoded into a binary code. Let $G = (V, E)$ be the input graph with N vertices and e edges. $V = (v_1, ..., v_N)$ For every vertex, $n = \lceil \log_2 N \rceil$ are used to indicate the position it occupies in the cycle.

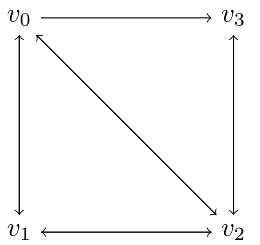


Figure 2: Example Graph

For example the bitstring $11, 00, 01, 10$ would be translated to $3 - 0 - 1 - 2$. This means v_0 is on the position 3, v_1 in on the position 0 and so on. The resulting cycle is (v_1, v_2, v_3, v_0). The search space includes every possible permutation of the positions of the vertices in the hamiltonian cycle. Therefore, its size is $2^{N \log n} = N^N$. The solutions of the HC problem satisfy two conditions:

1. Each vertex should have a different position.
2. Vertices with consecutive positions have to be connected by an edge. (The last and first positions are considered adjacent)

Bitstring	Integer representation	Validity	Cycle
00,01,10,01	0-1-2-1	Violates constraint 1	None
01,11,10,00	1-3-2-0	Violates constraint 2	(v_3, v_0, v_2, v_1)
00,11,10,01	0-3-2-1	Yes	(v_0, v_3, v_2, v_1)

Table 1: Some examples of the possible solution encodings.

An algorithm to solve the HCP of a directed graph with $N = 2^n$ vertices is presented. The main contribution of this paper is the oracle O that flips the phase of the states that satisfy both constraints. Later, possible adaptations of the algorithm for arbitrary graphs are proposed.

3.1 Circuit initialization

The position index register $|\psi\rangle$, which covers the search space, has $m = N \log N$ qubits. Applying O requires the addition of two registers, one for each constraint. An ancilla register $|\theta\rangle$ of $k = C_2^N$ qubits, and a second ancilla register $|\zeta\rangle$ with l qubits. l is the number of edges of the complementary graph.
Firstly, all position index qubits should be set in superposition of all states in the search space. This is done by applying the Hadamard Gate to each qubit in the main register. The qubits in the ancillas are set to $|1\rangle$ with the NOT gate. The resulting state is expressed as:

$$|\psi_0\rangle \otimes |\theta_0\rangle \otimes |\zeta_0\rangle$$

where:

$$|\psi_0\rangle = \frac{1}{\sqrt{2^m}} \sum_{i=0}^{m} |i\rangle$$

$$|\theta_0\rangle = |1\rangle^{\otimes k}$$

$$|\zeta_0\rangle = |1\rangle^{\otimes l}$$

3.2 Grover's Operator

The Grover operator is composed of two parts, the oracle O and the diffusion operator D. The oracle is the main contribution of this paper, and it is divided into three blocks: Positional Exclusivity, Missing Edge Detector, and a multi-controlled phase shift. The diffusion operator is the same as in the original Grover's algorithm.

Positional Exclusivity Block

To satisfy the first constraint, the position of each vertex is compared with the position of every other vertex. The comparison is made by a comparator circuit defined in [6]. The circuit, composed of NOT, CNOT and Toffoli/MCT gates, is applied to two registers of the same size, a and b, and stores the result in an ancilla qubit o.

$$\text{Comparator}(a, b, o) = \begin{cases} o \oplus 1, & \text{if } a = b; \\ o, & \text{if } a \neq b. \end{cases}$$

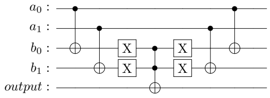


Figure 3: Two bits Comparator circuit

The comparator circuit is applied to the C_2^N combinations of position indexes, and each result is stored in a different qubit of $|\theta\rangle$ If any two vertices are assigned the same position, at least one qubit in $|\theta\rangle$ will be in state $|0\rangle$.

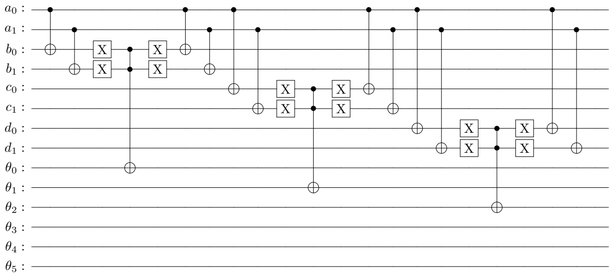


Figure 4: First three comparisons in the Positional Exclusivity block

Missing Edge Detector Block

Another subcircuit is introduced in this part, called the Plus One circuit. This circuit takes state $|i\rangle$ to state $|(i+1) \mod 2^p\rangle$. Given a register q with p qubits, a MCT is applied with the $p-1$ least significant qubits as control, and the most significant qubit as the target. This process is repeated taking with the $p-1$ least significant bits, so that every step the MCT has one less control. The inverse circuit, named Minus One circuit, takes state $|i\rangle$ to state $|(i-1) \mod 2^p\rangle$.

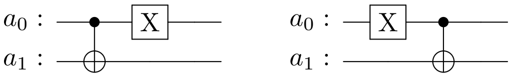


Figure 5: 2 qubits Plus One circuit and Minus One Circuit

The missing edges set is defined as the set of edges of the complementary graph. In the example, the set of missing edges is $\{(1,3),(3,1),(3,0)\}$. To

satisfy the second constraint, the vertices on adjacent positions should not have a missing edge between them.
The Plus One circuit is used to take the vertex to the next position, and then the Comparator circuit is used on the vertexes of the missing edge. If the next position of the vertex is occupied by a vertex that has a missing edge between them, the Comparator will flip an ancilla qubit.

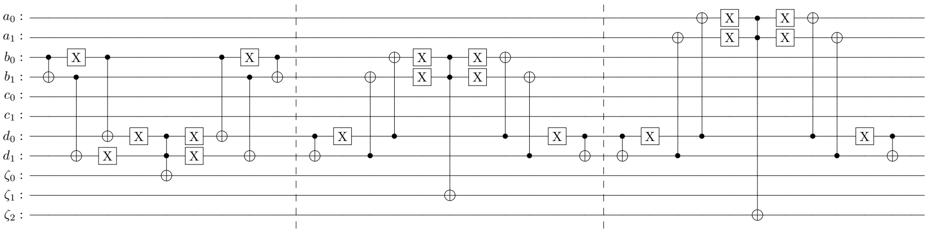
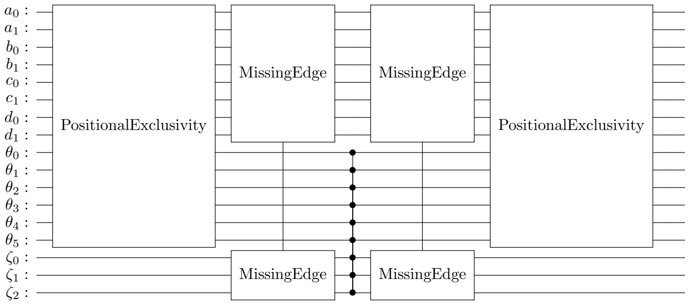


Figure 6: Missing Edge block of the example above

Complete Oracle Structure

The complete oracle consists of the Positional Exclusivity Block, a Missing Edge Block, and a multi-controlled phase shift, acting in all the ancilla qubits. Then, the Positional Exclusivity Block and the Missing Edge Block are repeated to return the ancillas to their original state.

Figure 7: Oracle



Diffusion Operator

The diffusion operator is applied on the position_index register.

4 Ancilla Reutilization

This approach employs a distinct ancilla register for each constraint enforcement. It is possible to reuse the ancilla, which helps minimize the required qubit count. Moreover, it reduces the size of the maximum MCT gate needed. However, this strategy involves a tradeoff between qubit quantity, circuit length,

and gate utilization. This tradeoff is studied in [9].
A second oracle that employs a single ancilla register is proposed.

4.1 Circuit description

The position_index register $|\psi\rangle$ and $|\theta\rangle$ remain the same, but the $|\zeta\rangle$ register is reduced to a single qubit, initializated in the $|0\rangle$ state.
It is important for this formulation that the maximum amount of qubits needed for the missing edge block is less than C_2^N. The maximum number of edges a directed graph can have is $2C_2^N$. However, if both edges (u, v) and (v, u) are missing, the same ancilla can store the result of the comparison. This is because a vertex can not have the previous and the next position of another vertex in any cycle of more than two elements. Therefore, the constraints are exclusionary, and a single qubit can be used as a witness for both.
The proposed oracle follows these steps: it initiates with the ME block and applies an MCT controlled by the $|\theta\rangle$ with $|\zeta\rangle$ as the target, another ME block is used to return the ancilla register to its initial state. Subsequently, the PE block is executed. The ancilla registers will be in state $|1\rangle$ for valid solutions, therefore, a phase shift is applied on $|\zeta\rangle$ controlled by $|\theta\rangle$. The inverse circuit is applied in order to restore the ancilla to its initial state.
Fig. 8 shows the diagram of the complete oracle O. The first generic run of the oracle will be analyzed. The solution space is divided into four groups. The A states that represent a valid hamiltonian cycle α,the B states that only satisfy the missing edge constraint β, the C states that only satisfy the positional exclusivity constraint γ, and the rest of the D states δ.

$$|\rho_0\rangle = \frac{1}{2^m}\left(\sum_{i=1}^{A}|\alpha_i\rangle + \sum_{i=1}^{B}|\beta_i\rangle + \sum_{i=1}^{C}|\gamma_i\rangle + \sum_{i=1}^{D}|\delta_i\rangle\right) \otimes |1\rangle^{\otimes k} \otimes |0\rangle$$

In $|\rho_1\rangle$, the ancilla of the states C and D have at least one flipped qubit. The string of k qubits that are not all in state $|1\rangle$ will be represented with a $|l\rangle$.

$$|\rho_1\rangle = \frac{1}{2^m}\left(\sum_{i=1}^{A}|\alpha_i\rangle|1\rangle^{\otimes k} + \sum_{i=1}^{B}|\beta_i\rangle|1\rangle^{\otimes k} + \sum_{i=1}^{C}|\gamma_i\rangle|l_{\gamma_i}\rangle + \sum_{i=1}^{D}|\delta_i\rangle|l_{\delta_i}\rangle\right) \otimes |0\rangle$$

The $|\zeta\rangle$ state is only flipped for the states that have the $|\theta\rangle$ ancilla in state $|1\rangle^{\otimes k}$.

$$|\rho_2\rangle = \frac{1}{2^m}\left(\sum_{i=1}^{A}|\alpha_i\rangle|1\rangle^{\otimes k}|1\rangle + \sum_{i=1}^{B}|\beta_i\rangle|1\rangle^{\otimes k}|1\rangle + \sum_{i=1}^{C}|\gamma_i\rangle|l_{\gamma_i}\rangle|0\rangle + \sum_{i=1}^{D}|\delta_i\rangle|l_{\delta_i}\rangle|0\rangle\right)$$

The second missing edge block returns the ancilla to its initial state.

$$|\rho_3\rangle = \frac{1}{2^m}\left(\sum_{i=1}^{A}|\alpha_i\rangle|1\rangle^{\otimes k}|1\rangle + \sum_{i=1}^{B}|\beta_i\rangle|l_{\beta_i}\rangle|1\rangle + \sum_{i=1}^{C}|\gamma_i\rangle|1\rangle^{\otimes k}|0\rangle + \sum_{i=1}^{D}|\delta_i\rangle|1\rangle^{\otimes k}|0\rangle\right)$$

The positional exclusivity block changes the ancilla of states $|\beta\rangle$ and $|\delta\rangle$.

$$|\rho_4\rangle = \frac{1}{2^m}\left(\sum_{i=1}^{A}|\alpha_i\rangle|1\rangle^{\otimes k}|1\rangle + \sum_{i=1}^{B}|\beta_i\rangle|l_{\beta_i}\rangle|1\rangle + \sum_{i=1}^{C}|\gamma_i\rangle|1\rangle^{\otimes k}|0\rangle + \sum_{i=1}^{D}|\delta_i\rangle|l_{\delta_{2i}}\rangle|0\rangle\right)$$

In state $|\rho_5\rangle$, the $|\alpha\rangle$ states are flipped.

$$|\rho_5\rangle = \frac{1}{2^m}\left(\sum_{i=1}^{A}|\alpha_i\rangle|1\rangle^{\otimes k}|1\rangle + \sum_{i=1}^{B}|\beta_i\rangle|l_{\beta_i}\rangle|1\rangle + \sum_{i=1}^{C}|\gamma_i\rangle|1\rangle^{\otimes k}|0\rangle + \sum_{i=1}^{D}|\delta_i\rangle|l_{\delta_{2i}}\rangle|0\rangle\right)$$

The next blocks return the ancilla registers $|\theta\rangle$, $|\zeta\rangle$, and $|\psi\rangle$ to their initial states.

$$|\rho_8\rangle = \frac{1}{2^m}\left(-\sum_{i=1}^{A}|\alpha_i\rangle + \sum_{i=1}^{B}|\beta_i\rangle + \sum_{i=1}^{C}|\gamma_i\rangle + \sum_{i=1}^{D}|\delta_i\rangle\right)\otimes|1\rangle^{\otimes k}\otimes|0\rangle$$

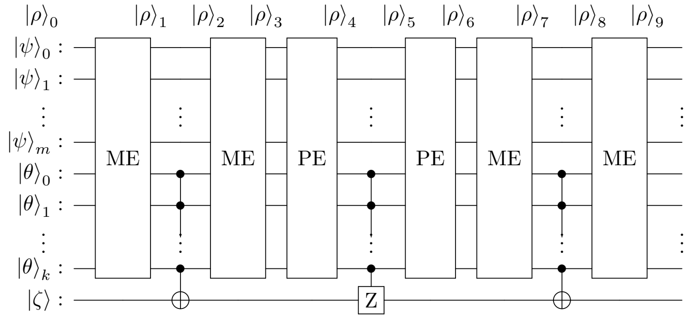


Figure 8: Oracle with ancilla reutilization

5 Generalization for an arbitrary graph

Adapting the oracle for a graph with an arbitrary number of vertices faces two challenges. Firstly, the superposition of states introduces invalid positions, where it is nonsensical for a vertex to occupy, for instance, the 15th position in a cycle of 10 elements. To address this issue, a strategy akin to the one employed in [6] can be applied, introducing an invalid position block.

The second challenge is associated with the Plus One circuit. This circuit efficiently transitions a state from $|i\rangle$ to $|(i+1) \mod 2p\rangle$. However, constructing a circuit to transition a state from $|i\rangle$ to $|(i+1) \mod l\rangle$ for an arbitrary l is more intricate. A modular adder is proposed in [11], using additional ancilla registers.

The alternative approach followed is to seek a transformation that transforms graphs with an arbitrary number of vertices into graphs with a number of vertices that is a power of two. This transformation should maintain the existing Hamiltonian cycles without introducing new ones.

The selected transformation operates as follows: for an arbitrary vertex, it is split into two separate connected vertices. The first vertex retains all the edges that entered the original vertex, while the second vertex preserves all the edges that exited the original vertex. Mathematically, let $G = (V, E)$ represent the original graph, where $V = \{v_1, v_2, ..., v_n\}$ is the set of vertices and E is the set of edges. The transformation $T : (V, E) \to (V', E')$ transforms vertex v_n in V into two vertices v'_n and v'_{n+1} in V'. E' is defined as follows:

$$\begin{aligned} E' = \{(v'_j, v'_i) | (v_j, v_i) \in E, \text{ where } i \neq n \text{ and } j \neq n\} \\ \cup\{(v'_j, v'_n) | (v_j, v_n) \in E\} \\ \cup\{(v'_{n+1}, v'_j) | (v_n, v_j) \in E\} \end{aligned}$$

In case it is necessary to add more than one vertex, a clique is added between the newly formed vertices, so as to minimize the number of missing edges, and thus the length of the oracle.

6 Discussion

A main challenge in the implementation of the proposed algorithm is determining k_{Gr} when the number of solutions is not known. [10] propose an algorithm that combines the Grover operator with the phase estimation technique to estimate the number of solutions. This algorithm requires the implementation of a Quantum Fourier Transform, which is a complex circuit.

Another approach is to choose a fixed k_{Gr} and run the algorithm multiple times. The probability of finding the solution increases with the number of runs. If there is no solution for the problem, the solution distribution will be uniform, with no solution having a greater probability than the others. If there are solutions, the valid states will be amplified.

This algorithm is quadratic in the number of qubits. The first formulation uses at most $N \log N + 2N(N-1)/2$ qubits, while the second formulation uses a fixed number of qubits $N \log N + N(N-1)/2 + 1$. Both formulations require $O(N^2)$ gates for the oracle, that should be applied $k_{Gr} \approx \sqrt{N^N}$ times. Therefore, the order of the algorithm is $O(N^2 2^{N \log N})$.

Asymptotically, the quantum algorithm requires the square root of the number of operation a brute force classical algorithm requires.

Some examples of the algorithm running on a simulator are shown in Fig. 9. The algorithm was implemented in Qiskit [12], and the results were obtained by running the algorithm on a classical simulator. The algorithm was run with $N = 4$ vertices.

References

[1] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, Aug. 2018.

[2] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proceedings of the Twenty-Eighth Annual ACM Symposium on Theory of Computing*, STOC '96, (New York, NY, USA), p. 212–219, Association for Computing Machinery, 1996.

[3] N. J. Cerf, L. K. Grover, and C. P. Williams, "Nested quantum search and structured problems," *Phys. Rev. A*, vol. 61, p. 032303, Feb 2000.

[4] A. Bhaduri, A. Saha, B. Saha, and A. Chakrabarti, "Circuit design for clique problem and its implementation on quantum computer," *IET Quantum Communication*, vol. 3, p. 30–49, Dec. 2021.

[5] A. Bhaduri, A. Saha, B. Saha, and A. Chakrabarti, "Robust quantum circuit for clique problem with intermediate qudits," *Nano Communication Networks*, vol. 37, p. 100466, Sept. 2023.

[6] A. Saha, D. Saha, and A. Chakrabarti, "Circuit design for k-coloring problem and its implementation on near-term quantum devices," in *2020 IEEE International Symposium on Smart Electronic Systems (iSES) (Formerly iNiS)*, pp. 17–22, 2020.

[7] S. Mukherjee, "A grover search-based algorithm for the list coloring problem," *IEEE Transactions on Quantum Engineering*, vol. 3, pp. 1–8, 2022.

[8] W.-L. Yang, H. Wei, F. Zhou, W.-L. Chang, and M. Feng, "Solution to the satisfiability problem using a complete grover search with trapped ions," *Journal of Physics B: Atomic, Molecular and Optical Physics*, vol. 42, p. 145503, jun 2009.

[9] S.-W. Lin, T.-F. Wang, Y.-R. Chen, Z. Hou, D. Sanán, and Y. S. Teo, "A parallel and distributed quantum sat solver based on entanglement and quantum teleportation," 2023.

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

[11] M. Roetteler, M. Naehrig, K. M. Svore, and K. Lauter, "Quantum resource estimates for computing elliptic curve discrete logarithms," 2017.

[12] Qiskit contributors, "Qiskit: An open-source framework for quantum computing," 2023.