Complement Grover’s Search Algorithm: An Amplitude Suppression Implementation

1st Andrew Vlasic  
Deloitte Consulting, LLP  
Chicago, United States  
avlasic@deloitte.com

2nd Salvatore Certo  
Deloitte Consulting, LLP  
Charlotte, United States  
scerto@deloitte.com

3rd Anh Pham  
Deloitte Consulting, LLP  
Atlanta, United States  
anhdpham@deloitte.com

Abstract—Grover’s search algorithm was a groundbreaking advancement in quantum algorithms, providing a quadratic speed-up of querying for items. Since the creation of this algorithm it has been utilized in various ways, including in preparing specific states for a general circuit. However, as the number of desired items increases, so does the gate complexity of the sub-process that conducts the query. To counter this complexity, an extension of Grover’s search algorithm is derived to suppress the amplitude of the queried items. To display the efficacy of this extension, the algorithm is implemented as an oracle that searches for desirable states. One may observe that as the number of desirable states increase, the gate complexity of the oracle increases. Depending on the complexity of the oracle query, it may be more efficient (from a circuit perspective) to invert this step by flipping the state of the ancilla qubit instead of acting on it. The increase in complexity is found within the oracle query that searches for the desirable states. One may observe that as the number of desirable states increase, the gate complexity of the oracle increases. Depending on the complexity of the oracle query, it may be more efficient (from a circuit requirements and gate depth perspective) to invert this step and construct an oracle to add a negative phase to all states, thereby decreasing complexity. This paper describes in detail how one may construct such an oracle.

Index Terms—Fundamental Quantum Algorithm, Grover’s Algorithm, QAOA

I. INTRODUCTION

Grover’s search algorithm [1] is a pivotal quantum routine that leads to quadratic speedups in unstructured search, counting, and amplitude estimation procedures. Since the inception of this algorithm, it has been utilized as a form of state preparation, an essential subroutine. If a substantial subset of states are required for the circuit then there is potential for the depth complexity of the Grover subroutine to substantially increase, potentially eroding or eliminating the quadratic advantage it provides.

This increase in complexity is found within the oracle query that searches for the desirable states. One may observe that as the number of desirable states increase, the gate complexity of the oracle increases. Depending on the complexity of the oracle query, it may be more efficient (from a circuit requirements and gate depth perspective) to invert this step and construct an oracle to add a negative phase to all states, subsequently only adding a positive phase to the undesirable solutions. Ergo, this oracle would search for fewer states, thereby decreasing complexity. This paper describes in detail how one may construct such an oracle.

II. DIFFERENT VIEW OF GROVER’S SEARCH ALGORITHM

Grover’s search algorithm marks items in an unstructured data set by increasing the amplitude of the desirable items relative to the undesirable. Hence, when the circuit is measured, there is a higher probability of finding the desirable items.

Hence, Grover’s search algorithm is a two step process, identify if the state is the desired state, then increase the amplitudes of the desired states. To identify if a state is desired, we denote this operator as the oracle operator $O$ where $O : |x\rangle|q\rangle \rightarrow |x\rangle|q \oplus f(x)\rangle$, for $\oplus$ the modulo 2 operator and $f(x) = 1$ if $x$ is the desired state and 0 otherwise. One way to implement the oracle is by utilizing a Hadamard state so that the evolution of individual qubits can be considered as an evolution of a Householder matrix and $H^\otimes n \left( 2|0\rangle\langle 0^\otimes n - I_{2^n} \right) H^\otimes n = 2|\psi\rangle\langle \psi| - I_{2^n}$. While the diffuser is fairly optimal and scales well with the dimension of the data, there is potential to decrease the gate complexity of the oracle. A decrease in gate complexity may come from changing the focus of the oracle from desired states to undesirable states. Do note that the oracle is not required to explicitly set controls states, but know how to check for the states, see Huang et al. [2], for an example.

To obtain the goal of adding the negative phase to the desirable states by acting on the undesirable states, the ancilla qubit is put in the state $|0\rangle$ and flipped to $|1\rangle$ in the oracle. If the state in the register is in $S^c$, then this value is flipped back to $|0\rangle$. Finally, applying the Pauli-Z operator gives the respective phase. Hence, denoting the complement oracle as $O_{\tilde{S}}$, we have $O_{\tilde{S}} |0\rangle = (-1)^{f(x)} |x\rangle f(x))$.

With the complement oracle, the standard diffuser operator (see Figure 1b) requires adjustment since the operator is expecting a superposition state of either $|+\rangle$ of $|-\rangle$ in the ancilla register. Therefore, a Pauli-X gate is added, followed by a Hadamard gate. Since the ancilla register in the complement
Fig. 1: (a) This circuit displays an amplitude suppression of Grover’s search algorithm for the states |000⟩ and |111⟩ where the oracle marks all the states when the values of the qubits are all the same. (b) This is the classical implementation searching for all of the six desirable states.

The oracle requires the qubit to not be in superposition, the inverse of these gates are included after the general qubit Toffoli gate; see Figure 1.a for an example.

One may see the final output of the Grover algorithm in the register is the same with either the complement oracle \( O_S \) with the adjusted diffuser or the standard algorithm with the \( O \) oracle. Therefore, the complexity analysis for Grover’s algorithm remains the same; see [3] for further detail.

III. APPLICATION TO THE QUANTUM APPROXIMATE OPTIMIZATION ALGORITHM

An application of this algorithm is to prepare states for the Quantum Approximate Optimization Algorithm (QAOA) [4] to assist in finding the lowest energy within the system. QAOA requires an ansatz for both the mixing and problem Hamiltonians, and typically requires the application of a classical optimizer to derive the parameters that yield the optimal or near-optimal solution.

To show the efficacy the Complement Grover’s algorithm in Figure 1.a is implemented as an initial state preparation procedure before the typical QAOA ansatz layers are applied. The algorithm in Figure 1 can be viewed as a type of warm starting, where the infeasible states of the problem are suppressed. For a basis of comparison, the same traveling salesman problem is applied to the original QAOA. The circuits are written in Qiskit [5] where the QAOA package is utilized. For a typical travelling salesman problem, the infeasible states would be ones where the salesman stays at the same node, or travels an incompletely cycle of the full tour. For instance, if there are 3 cities that need to be visited in 3 time steps then |100⟩ \( \rightarrow \) |010⟩ \( \rightarrow \) |001⟩ is a feasible solution and |100⟩ \( \rightarrow \) |010⟩ \( \rightarrow \) |100⟩ is infeasible. Using our modified Grover algorithm, we removed these states from the initial state preparation.

While there have been advancements in optimization techniques in universal computing, these methods are considerably different. A particularly close technique is described in Bärtschi and Eidenbenz [6] where the authors derive a QAOA that implements the mixer operator as the Hamiltonian of the equal superposition of feasible states. The creativity of the algorithm is balanced by the gate complexity and still requires the derivation of an operator that prepares the equal superposition of feasible states.

The results are given in Figure 2 where QAOA using complement Grover’s algorithm out-performs QAOA alone. The reason for the performance boost is that the amplitude suppression Grover’s algorithm decreases the amplitudes of the infeasible states, allowing for initial higher probabilities of feasible states within the initial state preparation, where solutions are found faster.

IV. CONCLUSION

In the case where desirable states \( S \) outnumber undesirable states \( S' \), the algorithm presented above represents an efficient way to structure an oracle. There is potential improvement with a different derivation of the diffuser.

REFERENCES

[1] L. Grover, “A fast quantum mechanical algorithm for database search,” in 28th Annual ACM Symposium on the Theory of Computing (STOC), (May 1996), 1996.

[2] H.-L. Huang, X.-L. Wang, P. P. Rohde, Y.-H. Luo, Y.-W. Zhao, C. Liu, L. Li, N.-L. Liu, C.-Y. Lu, and J.-W. Pan, “Demonstration of topological data analysis on a quantum processor,” Optica, vol. 5, no. 2, pp. 193–198, 2018.

[3] M. A. Nielsen and I. Chuang, “Quantum computation and quantum information,” 2002.

[4] E. Farhi, J. Goldstone, and S. Gutmann, “A quantum approximate optimization algorithm,” arXiv preprint arXiv:1411.4028, 2014.

[5] G. Aleksandrowicz, T. Alexander, P. Barkoutsos, L. Bello, Y. Ben-Haim, D. Bucher, F. J. Cabrera-Hernández, J. Carballo-Franquis, A. Chen, C.-F. Chen et al., “Qiskit: An open-source framework for quantum computing,” Accessed on: Mar, vol. 16, 2019.

[6] A. Bärtschi and S. Eidenbenz, “Grover mixers for qaoa: Shifting complexity from mixer design to state preparation,” in 2020 IEEE International Conference on Quantum Computing and Engineering (QCE). IEEE, 2020, pp. 72–82.