A Reduced Complexity Method of Recursive Quantum Approximate Optimization Algorithm

Youngjin Seo
School of Electrical Engineering
Korea University
Seoul, Korea
cherishiz@korea.ac.kr

Jun Heo*
School of Electrical Engineering
Korea University
Seoul, Korea
junheo@korea.ac.kr

***Abstract*—Among quantum algorithms, the quantum approximation optimization algorithm (QAOA) is an algorithm that finds approximate solutions. Recursive QAOA is proposed to overcome the limitation that QAOA has lower performance than Goemans-Williamson algorithm in the MAX-CUT problem [Physical Review Letter, 125, 260504, 2020]. In this paper, we introduce a method with lower complexity than RQAOA and compare its numerical results in the MAX-CUT problem. At the same level, the performance of the proposed method is lower than that of RQAOA, but the number of computations of quantum and classical parts is very small.**

***Index Terms*—Quantum algorithm, Quantum Approximate Optimization Algorithm**

I. Introduction

In various areas, the quantum algorithm has been studied to have more advantages than classical algorithms [1]–[6]. Recently, the usability of the variational quantum algorithm (VQA) has been discussed. The VQA is a method that does not have the disadvantage of high quantum circuit depth of existing quantum algorithms (Shor's algorithm, Grover's algorithm, and HHL algorithm). Therefore, VQA with a low quantum circuit depth that can be calculated from an incomplete quantum computer is currently being actively studied. Among the algorithms belonging to VQA, quantum approximate optimization algorithm (QAOA) [7] is a technique proposed by Farhi, Goldstone, and Gutmann. There is a MAX-CUT problem as an application of QAOA [8]. The QAOA is improved to Multi-Angle QAOA [9] and Recursive QAOA [10]. Multi-angle QAOA (MA-QAOA) is a technique that improves the performance of QAOA by increasing the degree of freedom of the angle parameter used in QAOA circuits. Recursive QAOA (RQAOA) is a technique that improves performance by deleting certain vertex of the graph and performing QAOA operations recursively.

In this paper, we propose a method to lower the high complexity arising from the recursive procedure of RQAOA. In the MAX-CUT problem, we present the performance graph of the proposed technique numerically and mention the validity of the proposed algorithm through performance comparison with RQAOA.

II. Previous works

In this section, we review Quantum Approximate Optimization Algorithm(QAOA) [7] and Recursive Quantum Approximate Optimization Algorithm(RQAOA) [10].

A. Quantum Approximate Optimization Algorithm

Quantum Approximate Optimization Algorithm(QAOA) is an algorithm to find an approximate solution. QAOA has an initial state, $|0\rangle$. A Hadamard operator (H) is applied to the initial state, $|s\rangle = H|0\rangle$. The Hadamard operator superposes the initial state as all cases of the problem.

$$|s\rangle = H^{\otimes n}|0\rangle = \sum_i \frac{1}{\sqrt{2^n}}|i\rangle \tag{1}$$

where n is the number of the qubit. QAOA has two cases of the unitary operator, Problem(H_P) and Mixing(H_M).

$$U(H_P, \gamma) = e^{-i\gamma H_P} \tag{2}$$

$$U(H_M, \beta) = e^{-i\beta H_M} \tag{3}$$

where H_P is a problem Hamiltonian composed of ZZ operator and H_M is a mixing Hamiltonian which is the sum of the pauli X operator for n qubits. The problem Hamiltonian is related to an objective function of the problem. The objective function of the bit string x is represented as $C(x)$. In MAX-CUT problem, let E be an edge set of the graph G, and let V be a vertex set of G. The objective function $C(x)$ is defined as

$$C(x) = \sum_{(j,k)\in E} C_{jk}(x) = \sum_{(j,k)\in E} \frac{1}{2}(1 - x_j x_k) \tag{4}$$

where $x = x_1 \cdots x_n$, $(x_i \in \{-1, 1\})$. If the vertex j belongs to the same set of the vertex k, $x_j x_k = 1$. Otherwise, the value of $x_j x_k$ is -1. The goal of the MAX-CUT problem is finding the cutting set maximizing the value of C. Using the objective function of the MAX-CUT problem, the problem Hamiltonian H_P is represented as follows:

$$H_P = \sum_{(j,k)\in E} \frac{1}{2}(I - Z_j Z_k) \tag{5}$$

where I is an identity operator. The final state of p-level QAOA is

$$|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle = U(H_M,\beta_p)U(H_P,\gamma_p)\cdots U(H_M,\beta_1)U(H_P,\gamma_1)\,|s\rangle \tag{6}$$

where $\boldsymbol{\gamma}=(\gamma_1,\cdots,\gamma_p)$ and $\boldsymbol{\beta}=(\beta_1,\cdots,\beta_p)$. The expectation value of C in the state $|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle$ is denoted by

$$F_p(\boldsymbol{\gamma},\boldsymbol{\beta}) = \langle\boldsymbol{\gamma},\boldsymbol{\beta}|H_P|\boldsymbol{\gamma},\boldsymbol{\beta}\rangle. \tag{7}$$

The goal is maximizing $F_p(\boldsymbol{\gamma},\boldsymbol{\beta})$.

$$F_p(\boldsymbol{\gamma}^*,\boldsymbol{\beta}^*) = \max_{\boldsymbol{\gamma},\boldsymbol{\beta}} F_p(\boldsymbol{\gamma},\boldsymbol{\beta}). \tag{8}$$

Let R be the approximation ratio,

$$R = \frac{F_p(\boldsymbol{\gamma}^*,\boldsymbol{\beta}^*)}{C_{\max}} \tag{9}$$

where $C_{\max}$ is the maximum value of C, $C_{\max}=\max_x C(x)$.

B. Recursive Quantum Approximate Optimization Algorithm

Recursive Quantum Approximate Optimization Algorithm(RQAOA) utilizes the expectation value of the edge (j,k), $M_{j,k}$.

$$M_{j,k} = \langle\boldsymbol{\gamma}^*,\boldsymbol{\beta}^*|Z_jZ_k|\boldsymbol{\gamma}^*,\boldsymbol{\beta}^*\rangle. \tag{10}$$

After finding the largest magnitude of $M_{j,k}$ for all edges, a specific edge (u,v) is chosen, $M_{u,v} = \max_{j,k}|M_{j,k}|$. RQAOA redefine the recursive problem Hamiltonian H_n as

$$H_n = \sum_{(j,k)\in E}\frac{1}{2}(I - J_{j,k}Z_jZ_k) \tag{11}$$

where $J_{j,k}$ are arbitrary real values. For the specific edge (u,v), impose a constraint $Z_v = \operatorname{sgn}(M_{u,v})Z_u$ and substitute it into the recursive problem Hamiltonian H_n to remove the vertex v.

$$\begin{aligned} H_n =& \frac{1}{2}(mI - J_{u,v}\operatorname{sgn}(M_{u,v})) \\ &- \frac{1}{2}\sum_{(j,v),j\neq u} J_{j,v}\operatorname{sgn}(M_{u,v})Z_jZ_u \\ &- \frac{1}{2}\sum_{(j,k),k\neq v} J_{j,k}Z_jZ_k \end{aligned} \tag{12}$$

From the Hamiltonian H_n, the recursive problem Hamiltonian H_{n-1} in the next step is represented as

$$\begin{aligned} H_{n-1} =& \frac{1}{2}(m-1)I - \frac{1}{2}\sum_{(j,v),j\neq u} J_{j,v}\operatorname{sgn}(M_{u,v})Z_jZ_u \\ &- \frac{1}{2}\sum_{(j,k),k\neq v} J_{j,k}Z_jZ_k \end{aligned} \tag{13}$$

$$= \sum_{(j,k)\in E'}\frac{1}{2}(I - J'_{j,k}Z_jZ_k) \tag{14}$$

where E' is the edge set in the next step. The coefficients $J'_{j,k}$ are summarized as follows:

$$J'_{j,k} = \begin{cases} J_{j,v}\operatorname{sgn}(M_{u,v}), & \text{if } k=u \text{ and } (j,u)\notin E \\ J_{j,v}\operatorname{sgn}(M_{u,v}) + J_{j,u}, & \text{if } k=u \text{ and } (j,u)\in E \\ J_{j,k}, & \text{otherwise} \end{cases} \tag{15}$$

where the edge (j,k) for the coefficient $J'_{j,k}$ belongs to the edge set E'. Using the Hamiltonian H_{n-1}, find the expectation value for the Hamiltonian. The recursion step continues as much as n_c steps, $n_c \ll n$. After the n_c steps, the final expectation value is calculated with backtracking method.

III. Proposed Method

In this section, we describe a reduced complexity method of RQAOA. And we provide simulation results for MAX-CUT problem.

A. Reduced Complexity Method of Recursive Quantum Approximate Optimization Algorithm

In RQAOA, the recursive problem Hamiltonian H_n is updated by eliminating the specific vertex. Since this procedure is executed by serial, the RQAOA can have a high total complexity. In order to mitigate this disadvantage, we propose a reduced complexity method of RQAOA having slightly lower performance.

First, we consider the recursion size to be n_r. We choose the n_r edges with large expectation values. The n_r nodes in the chosen edges are eliminated with the same method in RQAOA. We repeat the elimination as much as n_c steps. The number of total removed nodes is $n_r \times n_c$. The recursive problem Hamiltonian H_{n,n_r} is represented as

$$H_{n,n_r} = \sum_{(j,k)\in E}\frac{1}{2}(I - J_{j,k}Z_jZ_k). \tag{16}$$

For simplicity, we consider that $n_r = 2$. We choose the specific two edges, $(u_1,v_1),(u_2,v_2)$. In some cases, the same nodes might be picked. If $u_1=u_2$ or $u_1=v_2$, we choose the other edge with the next large expectation value. We assume that $u_1\neq v_1\neq u_2\neq v_2$. For the same method of RQAOA, we impose two constraints $Z_{v_1} = \operatorname{sgn}(M_{u_1,v_1})Z_{u_1}$, $Z_{v_2} = \operatorname{sgn}(M_{u_2,v_2})Z_{u_2}$. We substitute these into the recursive problem Hamiltonian $H_{n,2}$ to eliminate the nodes v_1 and v_2. If edge (v_1,v_2) exists, it is replaced by edge (u_1,u_2). $Z_{v_1}Z_{v_2} = \operatorname{sgn}(M_{u_1,v_1})\operatorname{sgn}(M_{u_2,v_2})Z_{u_1}Z_{u_2}$. Otherwise, the Hamiltonian $H_{n,2}$ is represented by

$$\begin{aligned} H_{n,2} =& \frac{1}{2}(mI - J_{u_1,v_1}\operatorname{sgn}(M_{u_1,v_1}) - J_{u_2,v_2}\operatorname{sgn}(M_{u_2,v_2})) \\ &- \frac{1}{2}\sum_{(j,v_1),j\neq u_1} J_{j,v_1}\operatorname{sgn}(M_{u_1,v_1})Z_jZ_{u_1} \\ &- \frac{1}{2}\sum_{(j,v_2),j\neq u_2} J_{j,v_2}\operatorname{sgn}(M_{u_2,v_2})Z_jZ_{u_2} \\ &- \frac{1}{2}\sum_{(j,k),k\neq v_1,v_2} J_{j,k}Z_jZ_k. \end{aligned} \tag{17}$$

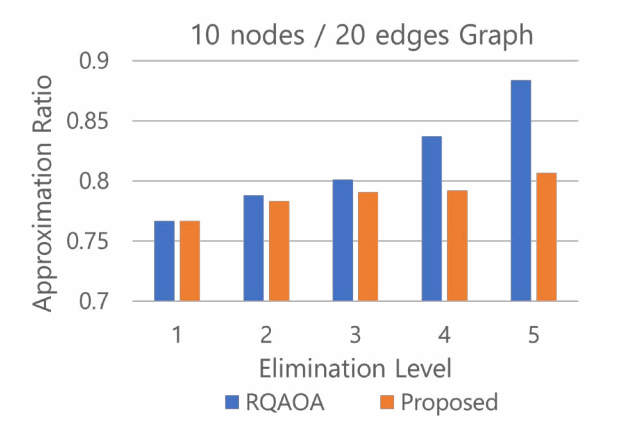

Fig. 1. Simulation results of 10 nodes / 20 edges random graph. The simulation is executed in 2 cases which are RQAOA and the proposed algorithm. The elimination step of RQAOA n_c is considered from 1 to 5. The elimination level of the proposed algorithm is also from 1 to 5. The value of the parameter n_c is assumed by 2 in the proposed algorithm. From the 1 to 3 level, the performance of the proposed algorithm is close to RQAOA. The proposed algorithm is executed by one-time quantum and classical computation.

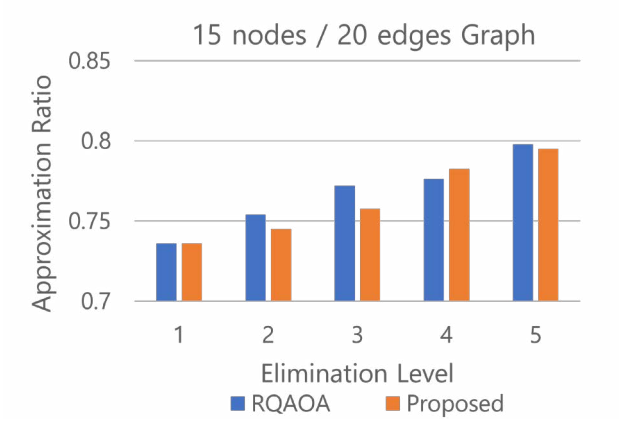

Fig. 2. Simulation results of 15 nodes / 20 edges random graph. The elimination step of RQAOA n_c is considered from 1 to 5. The elimination level of the proposed algorithm is also from 1 to 5. The value of the parameter n_c is assumed by 2 in the proposed algorithm. In level 4, the performance of the proposed algorithm is higher than that of RQAOA.

B. Simulation Results

In this section, we provide the simulation results for MAX-CUT problem. We used IBMQ qiskit program. The graph models are generated by 'nx.gnm_random_graph' module in networkx library. We used the COBYLA(Constrained Optimization By Linear Approximation) algorithm to find the optimized angles. Figure 1 shows the simulation results of 10 nodes / 20 edges graph model. The horizontal axis denotes the elimination level, where the value of n_c in RQAOA and the value of n_r in the proposed algorithm. The elimination step of the proposed algorithm n_c is considered by 2. The vertical axis is the approximation ratio(A.R.) which is defined by in Section II-A. The maximum A.R. value is 1. In the 1,2, and 3 levels, the proposed algorithm has a similar performance to RQAOA. Since the proposed algorithm is performed by one-time computation, using the proposed algorithm is more economical than RQAOA in level 3.

We provide the simulation results of the other graph model (15 nodes / 20 edges) in Fig. 2. In this graph model, the proposed algorithm has a high performance in level 4.

IV. Conclusion

In this paper, we presented the lower complexity method of RQAOA. We introduced the recursive problem Hamiltonian H_{n,n_r} eliminating n_r nodes in one step. By removing the n_r nodes in one step, the performance increases and complexity decreases. In addition, we described the simulation results for MAX-CUT problem. The complexity of the proposed method is lower than RQAOA. However, the proposed method has slightly lower performance than RQAOA. We compared the proposed algorithm with RQAOA in only simulation. A more valid comparison needs to consider the formula of the two algorithms.

Acknowledgment

"This work was supported by Institute for Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) [No. 2020-0-00014, A Technology Development of Quantum OS for Fault-tolerant Logical Qubit Computing Environment] and the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2023-2021-0-01810) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation) and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MIST) (No. 2019R1A2C2010061)"

References

[1] D. Deutsch, "Quantum theory, the church-turing principle and the universal quantum computer," in *Proc. R. Soc. Lond. A*, vol. 400, no. 1818, pp. 97-117, 1985.

[2] D. Deutsch and R. Jozsa, "Rapid solution of problems by quantum computation," *Proc. R. Soc. Lond. A*, vol. 439, no. 1907, pp. 553-558, 1992.

[3] D. R. Simon, "On the power of quantum computation," *Proceedings 35th Annual Symposium on Foundations of Computer Science*, 1994, pp. 124-134, doi: 10.1109/SFCS.1994.365700.

[4] P. W. Shor, "Algorithms for quantum computation: Discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, 1994, pp. 124-134, doi: 10.1109/SFCS.1994.365700.

[5] L. K. Grover, "A fast quantum mechanical algorithm for database search," in *Proc. 28th Annu. ACM Symp. Theory Comput. (STOC)*, 1996, pp. 212-219.

[6] A. W. Harrow, A. Hassidim, and S. Lloyd, "Quantum algorithm for linear systems of equations," *Phys. Rev. Lett.*, vol. 103, no. 15, Oct. 2009, Art. no. 150502.

[7] E. Farhi, J. Goldstone, and S. Gutmann, "A Quantum Approximate Optimization Algorithm," arXiv:1411.4028, 2014.

[8] Wang, Z., Hadfield, S., Jiang, Z. & Rieffel, E. G., "Quantum approximate optimization algorithm for MaxCut: A fermionic view," *Phys. Rev. A*, **97**(2), 022304, 2018.

[9] R. Herrman *et al.*, "Multi-angle quantum approximate optimization algorithm," *Scientific Reports*, vol. 12, no. 1, pp. 1-10, 2022.

[10] S. Bravyi, A. Kliesch, R. Koenig, and E. Tang, "Obstacles to Variational Quantum Optimization from Symmetry Protection," *Phys. Rev. Lett.* **125**, 260504, 2020.