

Exploring Beamforming Solutions using Recursive Quantum Approximate Optimization Algorithm

Abstract—The Quantum Approximate Optimization Algorithm (QAOA) is a promising quantum algorithm for solving combinatorial optimization problems. In this work, we explore the Recursive QAOA (RQAOA) approach, which improves upon the standard QAOA by successively reducing the problem size, to optimize sensor configuration for beamforming with the objective of maximizing the Signal-to-Interference-plus-Noise Ratio (SINR). We present a series of experiments across various setups, including different numbers of sensors and a range of optimizers. Additionally, we compare the obtained SINR with the maximum achievable SINR to assess the efficiency of the algorithm.

I. INTRODUCTION

Beamforming is a spatial filter primarily designed for directional signal transmission or reception. The objective is to estimate the signal arriving from a desired direction in the presence of noise and interfering signals [15]. Recent advances in sparse array technology have significantly enhanced the performance of adaptive beamforming [16], further driving down hardware costs with a regular geometry [13].

The design of sparse arrays can be formulated as a combinatorial optimization problem since it involves selecting an optimal subset of elements from a larger set to achieve desired performance characteristics. The application of quantum computing to combinatorial optimization problems is attracting increasing research interest, especially concerning the near-term possibilities of Quantum Computing [11].

Variational quantum algorithms (VQAs), which use a classical optimizer to train a parameterized quantum circuit, have emerged as a leading strategy to address these constraints [7]. The Quantum Approximate Optimization Algorithm (QAOA) [8] is one such hybrid variational quantum-classical algorithm that solves combinatorial optimization problems. Here, a quantum state is prepared by a p -level circuit specified by $2p$ variational parameters. The parameters are then optimized by a classical optimizer, which are then fed back to the quantum circuit in a closed loop [18]. This algorithm is particularly noteworthy because even the lowest depth version cannot be efficiently simulated on any classical device, thus exhibiting “Quantum Supremacy” [9].

It is, however, known that at low depth, certain locality constraints of QAOA limit its performance. To improve the quality of approximate solutions that go beyond these limitations, a non-local variant of QAOA, namely Recursive QAOA (RQAOA) [5], was proposed. RQAOA has been studied less extensively than QAOA and is not as well understood, particularly regarding the types of instances for which it may fail to deliver high-quality solutions [12].

Recursive QAOA involves recursively reducing the problem

size by applying QAOA. This size reduction procedure removes one variable at a time using correlation rounding. Once the problem reaches a manageable size, a classical optimizer is used to solve the smaller instance. While the problem of establishing a rigorous lower bound on the approximation ratio – quality of an approximate solution compared to the optimal solution – achieved by this hybrid algorithm remains open (except for very special families of instances), RQAOA, especially at higher levels p , as the potential to yield results competitive with the best-known classical algorithms [6].

For solving the MAX-CUT problem on complete graphs with $2n$ vertices, it has been shown that the approximation ratio of the level-1 RQAOA is exactly one whereas that of the level-1 QAOA is strictly less than $1 - \frac{1}{8n^2}$, which implies that the exact solution for the MAX-CUT problem on complete graphs can be found by using the level-1 RQAOA while it can never be done by performing the level-1 QAOA [3].

RQAOA has also been applied to solve MAX-k-CUT, the problem of finding an approximate vertex k -coloring of a graph, the results of which indicate that RQAOA might have higher potential at larger levels p , which may no longer be simulable classically [6].

In this work, we simulate beamforming in the context of the RQAOA. The optimization formulation, derived by maximizing the Signal-to-Interference-plus-Noise Ratio (SINR), is based on a complete graph, which motivates the use of RQAOA. Additionally, we compare the approximation ratios for different scenarios to evaluate their effectiveness.

II. MVDR BEAMFORMER

Consider N omnidirectional antennas arranged in uniform linear array (ULA) with inter-element spacing of $p_n d$, $p_n \in \mathbb{N}$, with $d = \lambda/2$ and λ being the operating wavelength. Consider this array receiving signals, with a desired signal arriving from direction θ_d and q other interfering signals from various directions $\{\phi_1, \phi_2, \dots, \phi_q\}$. The corresponding steering vectors are given by,

$$\mathbf{u}_a = [e^{jk p_1 d \cos \theta_d} \quad e^{jk p_2 d \cos \theta_d} \quad \dots \quad e^{jk p_N d \cos \theta_d}]^T \quad (1)$$

$$\mathbf{v}_i = [e^{jk p_1 d \cos \phi_i} \quad e^{jk p_2 d \cos \phi_i} \quad \dots \quad e^{jk p_N d \cos \phi_i}]^T \quad (2)$$

where, $i = 1, 2, \dots, q$ and the wavenumber is denoted by $k = 2\pi/\lambda$. The received signal at time instant t is given by,

$$\mathbf{x}(t) = \mathbf{u}_a s(t) + \mathbf{V} \mathbf{v}(t) + \mathbf{n}(t)$$

where $\mathbf{V} = [\mathbf{v}_1 \quad \mathbf{v}_2 \quad \dots \quad \mathbf{v}_q]$ is the interference array manifold matrix with a full column rank, $s(t) \in \mathbb{C}$ denotes

the desired signal waveform with power σ_s^2 , $\mathbf{v}(t) \in \mathbb{C}^q$ is the vector of q different statistically independent interfering signals with the covariance matrix $\mathbf{R}_v$ and $\mathbf{n}(t) \in \mathbb{C}^N$ is the received noise vector with power σ_n^2 . Noise is assumed to be Gaussian in nature.

III. FORMULATION : OPTIMIZING SINR

In the context of optimizing sensor array configurations, maximizing SINR allows for improved detection and processing capabilities.

To optimize sensor configuration for maximizing the signal-to-interference-plus-noise ratio (SINR), the optimal configuration of K sensors selected from N uniformly spaced locations can be determined for a scenario involving a single interfering source by formulating the optimization problem as:

$$\begin{aligned} \max \quad & \text{SINR} \\ \text{such that} \quad & \mathbf{1}^T \mathbf{z} = K \end{aligned} \quad (3)$$

The SINR of MVDR beamformer is given by:

$$\text{SINR} = \sigma_s^2 \mathbf{u}_a^H \mathbf{R}_n^{-1} \mathbf{u}_a \quad (4)$$

where $\mathbf{R}_n = \mathbf{V} \mathbf{R}_v \mathbf{V}^H + \sigma_n^2 \mathbf{I}$.

The constraint being $\mathbf{1}^T \mathbf{z} = K$, refers to the number of active sensors.

We restrict ourselves to one interfering signal.

For $q = 1$, $\mathbf{R}_v = \sigma_v^2$, i.e. variance of the signal $\mathbf{v}(t)$. We also utilize Woodbury Matrix Identity [17] to simplify the expression.

Thus, maximizing SINR would be equivalent to:

$$\max \frac{\sigma_s^2}{\sigma_n^2} \left[K - \frac{\sigma_v^2 \mathbf{u}_a^H \text{diag}(\mathbf{z}) \mathbf{V} \mathbf{V}^H \text{diag}(\mathbf{z}) \mathbf{u}_a}{1 + \frac{\sigma_v^2}{\sigma_n^2} K} \right] \quad (5)$$

Thus, $\max \text{SINR}$ becomes equivalent to :

$$\min \mathbf{u}_a^H \text{diag}(\mathbf{z}) \mathbf{V} \mathbf{V}^H \text{diag}(\mathbf{z}) \mathbf{u}_a \quad (6)$$

Formulation as QUBO : : To derive the QUBO formulation, we rewrite equation (6) as:

$$\sum_{i=1}^N \sum_{j=1}^N z_i z_j (u_{a_i}^* v_i v_j^* u_{a_j}) \quad (7)$$

We use the Lagrange multiplier to make it unconstrained problem. Incorporating the constraint as quadratic penalization factor in equation (7), we get the optimization formulation as:

$$\sum_{i=1}^N \sum_{j=1}^N z_i z_j (u_{a_i}^* v_i v_j^* u_{a_j} + \mu) - 2K\mu \sum_{i=1}^N z_i + \text{constant} \quad (8)$$

To derive Ising Hamiltonian from Equation (8), we substitute $z_p = \frac{1-Z_p}{2}$ where Z_p is the Pauli-Z operator acting on qubit p , which is then used in the optimization algorithm. We can ignore the constant term in the above Hamiltonian since it does not have any effect on the relative energy differences among quantum states.

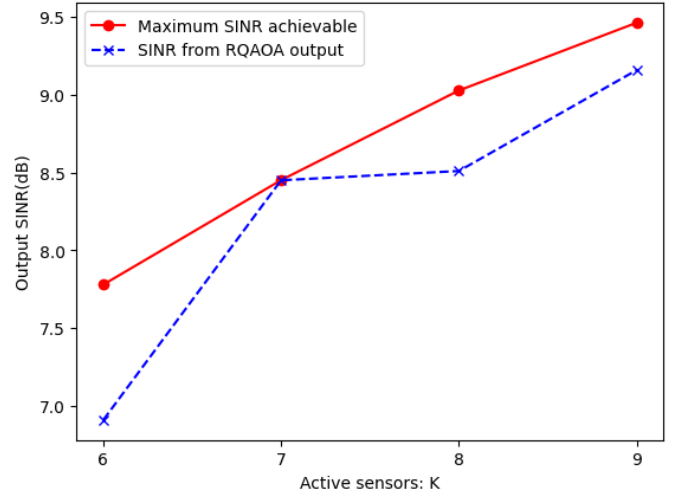


Fig. 1: SINR plot showing the maximum SINR achievable and the SINR corresponding to the RQAOA output for varying number of active sensors K . Total sensors N is taken as 10, with θ_d set at 23° with SNR 0dB, ϕ_i set at 50° with INR 5dB.

IV. RQAOA WITH BEAMFORMING

RQAOA, a non-local variant of QAOA, iteratively reduces the size of the problem. It is designed to address the limitations caused by the geometric locality of the ansatz [4]. In simpler terms, RQAOA improves performance by progressively simplifying the problem through deleting certain vertex of the graph and performing QAOA operations recursively [14].

We quantify the performance of the algorithm on a given problem by its approximation ratio which is defined as the ratio between expected value of cost function on the optimal sensor configuration produced by the RQAOA algorithm and the maximum achievable SINR, which can be formulated as:

$$\frac{\max_{\beta, \gamma} \langle \psi_p | C | \psi_p \rangle}{\max_x C(x)}$$

where the level- p QAOA circuit is parametrized by $\gamma_1, \dots, \gamma_p$ and $\beta_1, \dots, \beta_p$; $C(x)$ is the objective function to be maximized. In this work, objective function to be optimized is the SINR formulation derived in Equation 8. A higher approximation ratio signifies better performance in achieving near-optimal sensor configurations for the given problem.

The RQAOA proceeds by successively reducing the size of the problem, eliminating a single variable in each step by a procedure called correlation rounding [6]. We begin with the Ising Hamiltonian obtained from Equation 8 be denoted as H_N , which is used in the original QAOA to find the optimal state $|\psi_p(\beta^*, \gamma^*)\rangle$ which maximize the expectation value: $\langle \psi_p(\beta, \gamma) | H_N | \psi_p(\beta, \gamma) \rangle$.

It is important to note that Equation 8 can be interpreted as a complete weighted graph on N vertices where each node is connected to every other node. This represents a fully interconnected system in which every pair (i, j) forms a

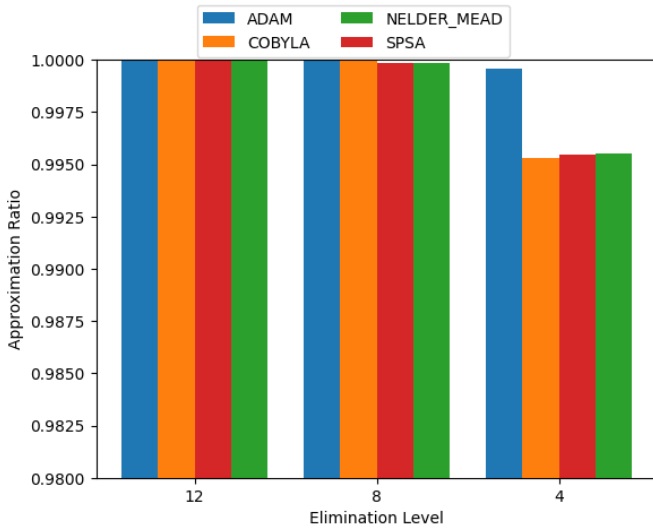


Fig. 2: Histogram plots for total sensors $N = 20$, active sensors $K = 15$, $\theta_d = 48^\circ$ with SNR $0dB$, $\phi_i = 95^\circ$ with INR $5dB$, indicating the approximation ratio achieved while varying the threshold number of variables for different choice of optimizers used as a part of QAOA which is used recursively in RQAOA. The classical optimizer used after the threshold number of variables is reached is Numpy Minimum Eigen Solver for all cases.

weighted edge.

We compute expectation value M_{ij} corresponding to every edge (i, j) of the optimal state $|\psi_p(\beta^*, \gamma^*)\rangle$:

$$M_{ij} = \langle \psi_p(\beta^*, \gamma^*) | Z_i Z_j | \psi_p(\beta^*, \gamma^*) \rangle$$

and choose an edge (k, l) which maximizes the magnitude of the expectation value $M_{i,j}$.

In order to eliminate the variable Z_k we impose the constraint $Z_k = \text{sgn}(M_{kl})Z_l$ and replace it in the Hamiltonian H_N to define a new Ising Hamiltonian on $(N - 1)$ terms as H_{N-1} . RQAOA is recursively called to maximize H_{N-1} .

This recursive procedure of variable elimination stops when the number of variables reach a threshold value which is further solved by a classical algorithm. The original (approximate) solution on N sensors is then obtained using the constraints using backtracking method.

By construction, the maximum energy of the new Hamiltonian coincides with the maximum energy of the initial Hamiltonian over the subset of assignments satisfying the constraint. The elimination step is motivated by the rounding of fractional correlations in relaxations of linear programs [10]. It is also noteworthy, that RQAOA was first applied to the Z_2 -symmetric Max-Cut problem, which exclusively comprised two-body terms [10]. In this work, we also include single-body terms along with two-body terms.

V. RESULTS AND SIMULATIONS

The objective of this study was to optimize the sensor configuration using the RQAOA for various categories and

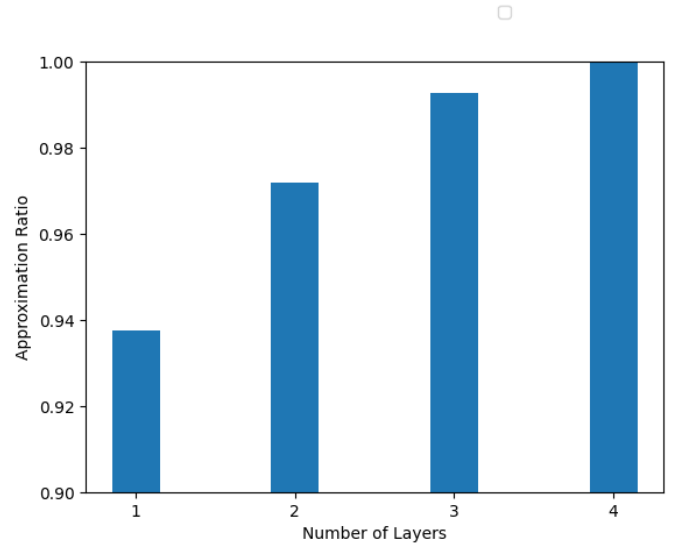


Fig. 3: Histogram plot portraying the approximation ratio obtained for different number of QAOA layers for total sensors $N = 20$, active sensors $K = 15$, θ_d set at 48° with SNR $0dB$, ϕ_i set at 95° with INR $5dB$ with the threshold number of variable set to 1. The classical optimizer used as a part of QAOA is COBYLA optimizer, while the classical optimizer used after the threshold is reached is Numpy Minimum Eigen Solver.

configurations. The formulation obtained in Equation 8 acts as the Hamiltonian which represents the objective function to be optimized.

We used IBMQ Qsikit program for the simulations. Figure 1 illustrates the maximum achievable SINR as well as the SINR corresponding to the RQAOA output, with respect to different values of the number of active sensors K with fixed total sensors $N = 10$, $\theta_d = 23^\circ$ with SNR $0dB$, $\phi_i = 50^\circ$ with INR $5dB$. The SINR obtained from the RQAOA output is consistently close to the maximum, indicating that RQAOA performs well in optimizing SINR.

Figure 2 presents a multi-bar plot comparing the performance of different optimizers for total sensors $N = 20$, active sensors $K = 15$, θ_d set at 48° with SNR $0dB$, ϕ_i set at 95° with INR $5dB$. The optimizers [1] ADAM(Adaptive Moment Estimation), COBYLA(Constrained Optimization BY Linear Approximation), NELDER-MEAD and SPSA(Simultaneous Perturbation Stochastic Approximation) are used within the QAOA framework which is applied recursively until the threshold number of variables are reached. The horizontal axis shows the elimination level, which is minimum number of variables to apply the recursive scheme. Once this threshold is reached, a classical optimizer is used to solve the problem. The classical optimizer we used is Numpy Minimum Eigen Solver [2] once the threshold is reached. It can be observed that as the threshold decreases, indicating that the recursive scheme is applied more times, the approximation ratio decreases for all optimizers.

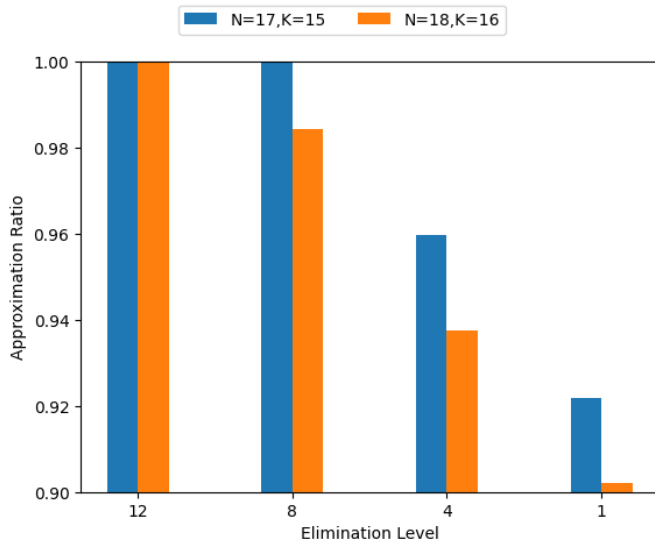


Fig. 4: Histogram plot showing the approximation ratio obtained for two problems of different problem size against the elimination level. The two configurations are : total sensors $N = 17$, active sensors $K = 15$ and total sensors $N = 18$, active sensors $K = 16$, with θ_d set at 48° , SNR $0dB$, ϕ_i set at 95° with INR $5dB$ for both cases

Figure 3 shows a histogram plot of the number of layers of the QAOA algorithm, which is used recursively versus the approximation ratio for total sensors $N = 15$, active sensors $K = 12$, θ_d set at 48° with SNR $0dB$, ϕ_i set at 95° with INR $5dB$. The threshold number of variables is fixed to 1, while the optimizer used within the QAOA framework is COBYLA. The classical optimizer used after the recursion is over is Numpy Minimum Eigen Solver [2]. It is evident that a smaller number of QAOA layers was unable to achieve a high approximation ratio.

Figure 4 shows the histogram plots for two different problem size. The approximation ratio is plotted against the threshold number of variables or the elimination level for total sensors $N = 17$, active sensors $K = 15$ and total sensors $N = 18$, active sensors $K = 16$ with θ_d set at 48° with SNR $0dB$, ϕ_i set at 95° with INR $5dB$ for both cases. The optimizer used as a part of QAOA is COBYLA, while the classical optimizer used after the recursion is over is Numpy Minimum Eigen Solver [2]. It is evident that as the problem size increases, the approximation ratio decreases for the same threshold number of variables.

VI. CONCLUSION

We utilized Recursive QAOA for optimal sensor configuration and compared the resulting approximation ratios across different array configurations, optimizer choices, numbers of QAOA layers, and threshold variables. Since the objective function represents a fully connected graph, RQAOA was the natural choice for this study. Not much research has been conducted on RQAOA, particularly on the types of problems

it can solve. We hope this study will encourage further exploration of RQAOA in solving optimization problems.

REFERENCES

- [1] "<https://docs.quantum.ibm.com/api/qiskit/0.28/qiskit.algorithms.optimizers>".
- [2] "<https://docs.quantum.ibm.com/api/qiskit/0.25/qiskit.algorithms.NumPyMinimumEigensolver>".
- [3] Eunok Bae and Soojoon Lee. Recursive qaoa outperforms the original qaoa for the max-cut problem on complete graphs. *Quantum Information Processing*, 2024.
- [4] Kostas Blekos, Dean Brand, Andrea Ceschini, Chiao-Hui Chou, Rui-Hao Li, Komal Pandya, and Alessandro Summer. A review on quantum approximate optimization algorithm and its variants. *Physics Reports*, 2024.
- [5] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Obstacles to variational quantum optimization from symmetry protection. *Physical Review Letters*, 2020.
- [6] Sergey Bravyi, Alexander Kliesch, Robert Koenig, and Eugene Tang. Hybrid quantum-classical algorithms for approximate graph coloring. *Quantum*, 2022.
- [7] M. Cerezo, Andrew Arrasmith, Ryan Babbush, Simon C. Benjamin, Suguru Endo, Keisuke Fujii, Jarrod R. McClean, Kosuke Mitarai, Xiao Yuan, Lukasz Cincio, and Patrick J. Coles. *Variational quantum algorithms*, 2021.
- [8] E. Farhi, J. Goldstone, and S. Gutmann. A Quantum Approximate Optimization Algorithm. *arXiv preprint arXiv:1411.4028*, 2014.
- [9] Edward Farhi and Aram W Harrow. Quantum supremacy through the quantum approximate optimization algorithm. *arXiv:1602.07674*.
- [10] Jernej Rudi Finžgar, Aron Kerschbaumer, Martin J.A. Schuetz, Christian B. Mendl, and Helmut G. Katzgraber. Quantum-informed recursive optimization algorithms. *PRX Quantum*, May 2024.
- [11] Felix Gemeinhardt, Antonio Garmendia, Manuel Wimmer, Benjamin Weder, and Frank Leymann. Quantum combinatorial optimization in the nisq era: A systematic mapping study. 56, 2023.
- [12] Yash J. Patel, Sofiene Jerbi, Thomas Bäck, and Vedran Dunjko. Reinforcement learning assisted recursive qaoa. *EPJ Quantum Technology*, 2024.
- [13] Robin Rajamäki, Sundeep Prabhakar Chepuri, and Visa Koivunen. Hybrid beamforming for active sensing using sparse arrays. *IEEE Transactions on Signal Processing*, 68, 2020.
- [14] Youngjin Seo and Jun Heo. A reduced complexity method of recursive quantum approximate optimization algorithm. In *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, 2023.
- [15] B. D. V. Veen and K.M. Buckley. Beamforming: a versatile approach to spatial filtering. *IEEE ASSP Magazine*, 5, 1988.
- [16] X. Wang and M. Amin. Design of Optimum Sparse Array for Robust MVDR Beamforming against DOA mismatch. *Proceedings of the 2017 IEEE 7th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, 2017.
- [17] Wikipedia contributors. Woodbury matrix identity — Wikipedia, the free encyclopedia, 2024.
- [18] Leo Zhou, Sheng-Tao Wang, Soonwon Choi, Hannes Pichler, and Mikhail D. Lukin. Quantum approximate optimization algorithm: Performance, mechanism, and implementation on near-term devices. *Physical Review X*, 10, 2020.