OPEN ACCESS
IOP Publishing
Journal of Physics A: Mathematical and Theoretical
J. Phys. A: Math. Theor. **58** (2025) 485304 (22pp) https://doi.org/10.1088/1751-8121/ae20d7

Modified recursive QAOA for exact MAX-CUT solutions on bipartite graphs: closing the gap beyond QAOA limit

Eunok Bae[1,2], **Hyukjoon Kwon**[2], **V Vijendran**[3,4,5] **and Soojoon Lee**[2,6,*]

[1] Electronics and Telecommunications Research Institute, Daejeon 34129, Republic of Korea
[2] School of Computational Sciences, Korea Institute for Advanced Study (KIAS), Seoul 02455, Republic of Korea
[3] A*STAR Quantum Innovation Center (Q.InC), Agency for Science, Technology and Research (A*STAR), 2 Fusionopolis Way, Innovis #08-03, Singapore 138634, Singapore
[4] Centre for Quantum Technologies (CQT), National University of Singapore, Singapore 117543, Singapore
[5] Centre for Quantum Computation and Communication Technologies (CQC2T), Department of Quantum Science and Technology, Research School of Physics, Australian National University, Acton 2601, Australia
[6] Department of Mathematics and Research Institute for Basic Sciences, Kyung Hee University, Seoul 02447, Republic of Korea

E-mail: level@khu.ac.kr and eobae@etri.re.kr

Received 20 June 2025; revised 10 November 2025
Accepted for publication 18 November 2025
Published 28 November 2025

Abstract

Quantum approximate optimization algorithm (QAOA) is a quantum–classical hybrid algorithm proposed with the goal of approximately solving combinatorial optimization problems such as the MAX-CUT problem. It has been considered a potential candidate for achieving quantum advantage in the noisy intermediate-scale quantum era and has been extensively studied. However, the performance limitations of low-level QAOA have also been demonstrated across various instances. In this work, we first analytically prove the performance limitations of the standard level-1 QAOA in solving the MAX-CUT

* Author to whom any correspondence should be addressed.

problem on bipartite graphs. To this end, we derive an upper bound for the approximation ratio based on the average degree of bipartite graphs. Second, we demonstrate that recursive QAOA (RQAOA), which recursively reduces graph size using QAOA as a subroutine, outperforms the standard level-1 QAOA. However, the performance of RQAOA exhibits limitations as the graph size increases. Finally, we show that RQAOA with a restricted parameter regime can fully address these limitations. Surprisingly, this modified RQAOA always finds the exact maximum cut for any bipartite graphs and even for a more general graph with parity-signed weights.

Keywords: quantum approximate optimization algorithm, recursive quantum approximate optimization algorithm, MAX-CUT problem, bipartite graphs

1. Introduction

Noisy intermediate-scale quantum (NISQ) devices are characterized by their relatively small number of qubits, their susceptibility to noise, and a lack of error correction, which makes them impractical for running traditional deep quantum algorithms like Shor's algorithm [1] designed for fault-tolerant quantum systems. Variational quantum algorithms (VQAs) address these limitations through a hybrid quantum–classical framework [2–6]. In these algorithms, the quantum part prepares a parameterized quantum state and measures observables, while the classical part optimizes these parameters to either minimize or maximize an objective function, such as an energy eigenvalue or a cost function. VQAs also provide a general algorithmic paradigm that may remain relevant beyond the NISQ era, offering a flexible route to explore quantum advantage in diverse applications ranging from quantum chemistry to optimization and quantum machine learning [2–5, 7].

One of the most prominent examples of VQAs is quantum approximate optimization algorithm (QAOA) [3] designed to tackle combinatorial optimization problems (COPs). The algorithm works by alternating between applying a problem-specific operator and mixing operators in a parameterized quantum circuit. QAOA has gained significant attention for its potential to outperform classical algorithms in solving COPs like the MAX-CUT problem, where it aims to find a cut that maximizes the number of edges between two partitions of vertices of a graph [3, 5, 8, 9]. Despite its promise for near-term quantum applications, it has been established that QAOA, when applied at any constant level, exhibits limited performance in solving the MAX-CUT problem for several instances [10–14]. Several variants and extensions of QAOA have been proposed to address its limitations and enhance its performance across a broader range of problems [11, 15–19].

Recursive QAOA (RQAOA) is one of the variants of QAOA proposed to enhance the performance on complex instances and scales to larger problems by reducing the problem size [11]. The performance of QAOA can be constrained by the Z_2 symmetry of its quantum states and the geometric locality of the ansatz, which means that the cost operators only involve interactions between nearest neighbor qubits in the underlying graph [11]. To address these limitations, RQAOA reduces the problem size iteratively by fixing certain variables based on edge correlation analysis. This iterative reduction can lead to more precise solutions as the recursion progresses through the new connections between previously unlinked vertices, effectively counteracting the locality constraints of QAOA. Although RQAOA has been less extensively investigated compared to other QAOA variants, it is gaining increasing attention as a promising approach for NISQ devices [11, 20–24].

It has been shown that RQAOA outperforms the standard QAOA (X-mixer) for some problem instances. Experimental evidence demonstrates the level-1 RQAOA significantly outperforms the standard QAOA at the same level for solving the MAX-k-CUT problem (especially, when $k = 3$), in which vertices are partitioned into k disjoint subsets to maximize the number of edges across subsets [21]. Moreover, the level-1 RQAOA is even competitive with Newman's classical algorithm for general k-colorable graphs [21]. Furthermore, there have been rigorous proofs supporting this argument. It has been proved that the level-1 RQAOA achieves the optimal approximation ratio while that of any constant level-q QAOA with $q < \frac{n}{4}$ is less than $1 - \frac{1}{4q+2}$ on cycle graphs with n vertices [11]. Recently, it was analytically shown that the level-1 RQAOA achieves an approximation ratio of 1, whereas the approximation ratio of the standard level-1 QAOA is strictly upper-bounded as $1 - \frac{1}{8n^2}$ on complete graphs with $2n$ vertices [23].

Our first significant result in this work is to prove the performance limitations of the standard level-1 QAOA for solving the MAX-CUT problem on bipartite graphs in terms of the approximation ratio. Our results indicate that, intuitively, the performance of the standard level-1 QAOA tends to degrade as a bipartite graph becomes denser, that is, closer to a complete bipartite graph. In particular, as the size of the complete bipartite graph increases, the performance of the standard level-1 QAOA becomes worse approaching that of random guessing, in contrast to the case of level-1 RQAOA on complete graphs, which converge to one [23].

Unlike other provable cases such as cycle graphs [11] and complete graphs [23], as the reduction process of RQAOA progresses in bipartite graphs, the structure of the graph deteriorates, causing it to lose its bipartite structure. This issue becomes more pronounced in graphs with a large number of vertices, where the graph transforms into a weighted graph. Consequently, analytically proving the performance of RQAOA in such scenarios remains a challenging problem. Although it has not yet been proved whether RQAOA can achieve the optimal performance in solving the MAX-CUT problem on bipartite graphs, we numerically show that the level-1 RQAOA has a much better performance than the standard level-1 QAOA for this problem instance in section 5.1.

While RQAOA offers crucial improvements over the standard QAOA, it also faces several limitations as identified in recent research [22, 25] and a few works have generalized and extended RQAOA to improve its performance [22, 26–29]. From the numerical results we obtain for random weighted bipartite graphs with 128 and 256 vertices in section 5.1, RQAOA appears to fall short of achieving optimal performance. Although employing a better optimization method might lead to improved outcomes, it remains unclear whether this issue stems from the limitations of RQAOA itself in determining edge correlations or from challenges in the optimization process during QAOA iterations.

In this paper, we propose a parameter setting method to enhance the performance of RQAOA and prove that our modified RQAOA can always find the exact MAX-CUT solution for positive weighted bipartite graphs. Although similar parameter setting schemes applied to QAOA have been proposed [29–32] and they have focused on finding optimal parameters, our contribution lies in rigorously proving that modifying the parameter search space to preserve the graph structure significantly improves RQAOA performance for certain cases, regardless of whether optimal parameters are found during QAOA subroutine.

This paper is organized as follows. In section 3, we introduce the MAX-CUT problem which we aim to solve, and provide a brief overview of the quantum algorithms used to tackle this problem: QAOA and RQAOA. In section 4, we analytically prove that the standard level-1 QAOA has limited performance in solving the MAX-CUT problem on bipartite graphs, deriving an upper bound for the approximation ratio based on the average degree of the graph.

In section 5.1, we numerically show that while the level-1 RQAOA outperforms the standard level-1 QAOA in this problem instance, but it may also fail to find the exact solution when the graph size increases. We propose a modified RQAOA to improve its performance and prove that our modified RQAOA can exactly solve the MAX-CUT problem on not only positive weighted bipartite graphs but also more general weighted graphs called a graph with parity-signed weights in section 5.2. Finally, we summarize our results and discuss the potential applicability of our modified RQAOA to other instances in section 6.

2. Outline of our main results

In this section, we preview our main results. Our first main result is to prove the performance limitations of the standard level-1 QAOA in solving the MAX-CUT problem on bipartite graphs. The approximation ratio is upper bounded in terms of the average degree which is a key property representing the graph's density. The formal statement can be found in theorem 3 of section 4.

Theorem 3*. *(Informal) Let $G = (V,E)$ be a bipartite graph and α_1 be the approximation ratio of the standard level-1 QAOA for solving the MAX-CUT problem on G. Then*

$$\frac{1}{2} \leqslant \alpha_1 < \frac{1}{2} + \frac{1}{2d_{\text{ave}}},$$

where $d_{\text{ave}} = \frac{2|E|}{|V|}$ denotes the average degree of G.

This theorem implies that for a connected graph with at least three vertices, the upper bound of the approximation ratio is strictly less than one. In other words, the standard level-1 QAOA cannot achieve a solution close to the exact one for the MAX-CUT problem on a bipartite graph. Furthermore, it also shows that for a dense bipartite graph, which is nearly complete, the approximation ratio approaches 1/2 when the number of vertices is sufficiently large.

Our second main result is to propose a modified RQAOA with a restricted parameter regime and rigorously prove that this modified RQAOA can achieve the approximation ratio of one in solving the MAX-CUT problem on a graph with parity-signed weights which is a generalization of bipartite graphs as shown in section 5.2.

Theorem 8. *Our modified RQAOA can exactly solve the MAX-CUT problem on a graph with parity-signed weights.*

3. Premilinaries

3.1. MAX-CUT problem

Let $G = (V,E)$ be a (undirected) graph with the set of vertices V and the set of edges $E = \{(i,j) : i,j \in V\}$. The MAX-CUT problem is a fundamental COP that aims to find two disjoint subsets of V that maximize the number of edges crossing between the two sets. The MAX-CUT problem can be formulated by maximizing the cost function

$$C(z) = \frac{1}{2} \sum_{(i,j)\in E} (1 - z_i z_j)$$

for $\mathbf{z} = (z_1, z_2, \ldots, z_n) \in \{-1,1\}^n$.

3.2. QAOA

QAOA can be viewed as a discrete version of adiabatic quantum computing [3]. The design of QAOA involves constructing a parameterized quantum circuit that alternates between applying the problem Hamiltonian H_C (which encodes the optimization problem) and the mixing Hamiltonian H_B (which ensures broad exploration of the solution space). Here, we only focus on the MAX-CUT problem which can be converted to the following problem Hamiltonian

$$H_C = \frac{1}{2} \sum_{(i,j) \in E} (I - Z_i Z_j),$$

where Z_i is the Pauli operator Z acting on the ith qubit. The level-q QAOA, denoted by QAOA$_q$, can be described as the following algorithm.

Algorithm 1 (QAOA$_q$ [3]). The QAOA$_q$ is as follows.

(i) Prepare the initial state $|+\rangle^{\otimes n}$.
(ii) Generate a variational wave function

$$|\psi_q(\boldsymbol{\beta}, \boldsymbol{\gamma})\rangle = \prod_{j=1}^{q} e^{-i\beta_j H_B} e^{-i\gamma_j H_C} |+\rangle^{\otimes n},$$

where $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_q)$, $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_q)$, where $H_B = \sum_{i=1}^{n} X_i$ is a mixing Hamiltonian and X_i is the Pauli operator X acting on the ith qubit.
(iii) Compute the expectation value

$$F_q(\boldsymbol{\beta}, \boldsymbol{\gamma}) = \langle \psi_q(\boldsymbol{\beta}, \boldsymbol{\gamma}) | H_C | \psi_q(\boldsymbol{\beta}, \boldsymbol{\gamma}) \rangle = \langle H_C \rangle$$

by performing the measurement in the computational basis.
(iv) Apply a classical optimization algorithm to find the optimal parameters

$$(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*) = \mathrm{argmax}_{\boldsymbol{\beta}, \boldsymbol{\gamma}} F_q(\boldsymbol{\beta}, \boldsymbol{\gamma}).$$

The approximation ratio α_q of QAOA$_q$ is defined as

$$\alpha_q = \frac{F_q(\boldsymbol{\beta}^*, \boldsymbol{\gamma}^*)}{C_{\max}},$$

where $C_{\max} = \max_{z \in \{-1,1\}^n} C(z)$.

3.3. Recursive QAOA (RQAOA)

In this section, we provide a brief overview of RQAOA which was introduced to address the limitations of the standard QAOA by incorporating a recursive problem reduction strategy [11]. RQAOA iteratively reduces the problem size by fixing specific variables based on edge correlation analysis. As the recursion proceeds, this iterative reduction can yield more accurate solutions by establishing new connections between previously unconnected vertices, thereby counteracting the locality constraints of QAOA. RQAOA was originally introduced on Ising-like Hamiltonians given by

$$H_n = \sum_{(i,j) \in E} J_{i,j} Z_i Z_j$$

defined on a graph $G_n = (V,E)$ with $|V| = n$, where $J_{i,j} \in \mathbb{R}$ denotes the strength of the coupling interaction between vertices i and j. Since the weighted MAX-CUT cost Hamiltonian H_C of QAOA can be converted to an Ising-like model as

$$H_C = \frac{1}{2}\sum_{(i,j)\in E} w_{ij}(I - Z_i Z_j) = \frac{1}{2}\sum_{(i,j)\in E} w_{ij} - \frac{1}{2}\sum_{(i,j)\in E} w_{ij} Z_i Z_j,$$

it is natural to study RQAOA directly in the weighted MAX-CUT setting. The goal of RQAOA is to approximate

$$\max_{\boldsymbol{x}\in\{0,1\}^n} \langle \boldsymbol{x}|H_n|\boldsymbol{x}\rangle.$$

Here, $|\boldsymbol{x}\rangle = |x_1,\ldots,x_n\rangle$ and $Z|x_i\rangle = z_i|x_i\rangle$ with $z_i = (-1)^{x_i}$ for each $i = 1,\ldots,n$. The level-q RQAOA, denoted by RQAOA$_q$, can be described as the following algorithm.

Algorithm 2 (RQAOA$_q$ [11]). The level-q RQAOA is as follows.

(i) Perform the standard QAOA$_q$ to find the optimal parameters $(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*)$ to maximize $F_q(\boldsymbol{\beta},\boldsymbol{\gamma})$.

(ii) Compute the edge expectation values
$$M_{ij} = \langle \psi_q(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*)|Z_i Z_j|\psi_q(\boldsymbol{\beta}^*,\boldsymbol{\gamma}^*)\rangle$$
for every edges $(i,j)\in E$.

(iii) Pick the edge $(k,l) = \mathrm{argmax}_{(i,j)\in E} M_{ij}$

(iv) By imposing the constraint $Z_k = \mathrm{sgn}(M_{kl})Z_l$, replace it with H_n to obtain

$$H_n' = \mathrm{sgn}(M_{kl})\left[\sum_{(i,k)\in E} J_{ik} Z_i Z_l\right] + \sum_{i,j\neq k} J_{ij} Z_i Z_j.$$

(v) Call the QAOA recursively to maximize the expected value of a new Ising Hamiltonian H_{n-1} depending on $n-1$ variables:
$$H_{n-1} = \sum_{(i,l)\in E_0'} J_{ij}' Z_i Z_l + \sum_{(i,j)\in E_1'} J_{ij}' Z_i Z_j,$$
where
$$E_0' = \{(i,l) : (i,k)\in E\}, E_1' = \{(i,j) : i,j\neq k\},$$
and
$$J_{ij}' = \begin{cases} \mathrm{sgn}(M_{kl}) J_{ik} & \text{if } (i,l)\in E_0' \\ J_{ij} & \text{if } (i,j)\in E_1'. \end{cases}$$

(vi) The recursion stops when the number of variables reaches some threshold value $n_c \ll n$, and find $\boldsymbol{x}^* = \mathrm{argmax}_{\boldsymbol{x}\in\{0,1\}^{n_c}} \langle \boldsymbol{x}|H_{n_c}|\boldsymbol{x}\rangle$ by a classical brute-force search.

(vii) Reconstruct the original (approximate) solution $\tilde{\boldsymbol{x}} \in \{0,1\}^n$ from $\boldsymbol{x}^*$ using the constraints.

4. Limitation of QAOA$_1$ on bipartite graphs

In this section, we analytically prove that the standard level-1 QAOA has limited performance in solving the MAX-CUT problem on bipartite graphs. We derive the upper bound of the approximation ratio in terms of the average degree of a given graph.

The analytical form for the expectation value of the standard level-1 QAOA has been known in [8]. For all $(i,j) \in E$,

$$\begin{aligned}\langle C_{ij}\rangle &= \langle\psi_1(\beta,\gamma)|\frac{1}{2}(I - Z_iZ_j)|\psi_1(\beta,\gamma)\rangle \\ &= \frac{1}{2} - \frac{1}{4}\sin^2(2\beta)\cos^{d_i+d_j-2(n_{ij}+1)}(1-\cos^{n_{ij}}(2\gamma)) \\ &\quad + \frac{1}{4}\sin(4\beta)\sin\gamma\left(\cos^{d_i-1}\gamma + \cos^{d_j-1}\gamma\right), \qquad (1)\end{aligned}$$

where d_i is the degree of the vertex i and n_{ij} is the number of triangles with the edge (i,j). Since bipartite graphs are triangle-free, equation (1) becomes simpler. In this case, we can rewrite the expectation value of the MAX-CUT Hamiltonian of the level-1 QAOA in terms of the vertex degrees instead of the edges as follows.

$$\langle H_C\rangle = \sum_{(i,j)\in E}\left[\frac{1}{2} + \frac{1}{4}\sin(4\beta)\sin\gamma\left(\cos^{d_i-1}\gamma + \cos^{d_j-1}\gamma\right)\right] \qquad (2)$$

$$= \frac{1}{2}|E| + \frac{1}{4}\sin(4\beta)\sin\gamma\sum_{d=1}^{d_{\max}} d|D_d|\cos^{d-1}\gamma, \qquad (3)$$

where D_d is the set of vertices with degree d. For bipartite graphs, we know the optimal cut is the number of all edges. Thus, the approximation ratio is

$$\alpha_1 = \max_{\beta,\gamma}\langle H_C\rangle / C_{\max} \qquad (4)$$

$$= \max_{\gamma}\left[\frac{1}{2} + \frac{1}{2}\sum_{d=1}^{d_{\max}}\frac{d|D_d|}{2|E|}\sin\gamma\cos^{d-1}\gamma\right] \qquad (5)$$

$$= \max_{\gamma}\left[\frac{1}{2} + \frac{1}{2}\sum_{d=1}^{d_{\max}}\frac{d|D_d|}{2|E|}f_d(\gamma)\right], \qquad (6)$$

where $f_d(\gamma) = \sin\gamma\cos^{d-1}\gamma$. Then we can get a bound of α_1 by obtaining the upper bound of $f_d(\gamma)$ for each d. Although this bound is obtained quite naturally, but it falls short in capturing the properties of a given bipartite graph. Therefore, by proposing a new bound that utilizes the average degree d_{ave}, which reflects one of the important properties of a graph—its density—we can intuitively observe how the density of bipartite graphs impacts the performance of the level-1 QAOA for solving the MAX-CUT problem.

Theorem 3 (Bipartite graph). *Let $G = (V,E)$ be a bipartite graph and let α_1 be the approximation ratio of the level-1 QAOA for solving the MAX-CUT problem on G. Then*

$$\alpha_1 \leqslant \frac{1}{2} + \sum_{d=1}^{d_{\max}}\frac{d|D_d|}{2|E|}\left[\frac{1}{2\sqrt{d}}\left(\frac{\sqrt{d-1}}{\sqrt{d}}\right)^{d-1}\right] \leqslant \frac{1}{2} + \frac{1}{2\sqrt{e}}\left(\frac{1}{\sqrt{d_{\text{ave}}}} + \frac{\sqrt{e}-1}{d_{\text{ave}}}\right),$$

where d_{ave} denotes the average vertex degree of G which can be defined as $\frac{\sum_{v\in V} d_v}{|V|}$, or equivalently, $\frac{2|E|}{|V|}$. The second upper bound is clearly less than that of the informal version in section 2.

Proof. In order to get the upper bound of α_1, let us first find the optimal value of $f_d(\gamma)$ as follows. Since $f_1(\gamma) = \sin\gamma$, $f_1(\gamma) \leqslant 1$. Observe that $f_d(\gamma) = \sin\gamma\cos^{d-1}\gamma$ and $f_d'(\gamma) =$

$\cos^{d-2}\gamma(d\cos^2\gamma-(d-1))$ for each $d\geqslant 2$. Then we can find the optimal γ^* such that $\cos\gamma^* = (1-\frac{1}{d})^{1/2}$. Thus, we have

$$f_d(\gamma)\leqslant f_d(\gamma^*)=\frac{1}{\sqrt{d}}\left(\frac{\sqrt{d-1}}{\sqrt{d}}\right)^{d-1},$$

and naturally get the upper bound of the approximation ratio as follows

$$\begin{aligned}\alpha_1 &\leq \frac{1}{2}+\sum_{d=1}^{d_{\max}}\frac{d|D_d|}{2|E|}\left[\frac{1}{2\sqrt{d}}\left(\frac{\sqrt{d-1}}{\sqrt{d}}\right)^{d-1}\right]\\ &\leq \frac{1}{2}+\frac{1}{2}\sum_{d=1}^{d_{\max}}\frac{d|D_d|}{2|E|}\left[\sqrt{\frac{1}{ed}}+\frac{1}{d}\left(1-\frac{1}{\sqrt{e}}\right)\right]\\ &\leq \frac{1}{2}+\frac{1}{2}\sqrt{\sum_{d=1}^{d_{\max}}\frac{|D_d|}{2e|E|}}+\sum_{d=1}^{d_{\max}}\frac{|D_d|}{2|E|}\left(1-\frac{1}{\sqrt{e}}\right)\\ &\leq \frac{1}{2}+\frac{1}{2\sqrt{e}}\left(\frac{1}{\sqrt{d_{\mathrm{ave}}}}+\frac{\sqrt{e}-1}{d_{\mathrm{ave}}}\right),\end{aligned}$$

where d_{ave} denotes the average vertex degree of G which can be defined as $\frac{\sum_{v\in V}d_v}{|V|}$, or equivalently, $\frac{2|E|}{|V|}$, or equivalently, $\frac{2|E|}{\sum_d |D_d|}$. To show that the second inequality holds, we can prove that the following inequalities hold separately

$$\frac{1}{\sqrt{d}}\left(\frac{\sqrt{d-1}}{\sqrt{d}}\right)^{d-1}\leqslant\frac{1}{\sqrt{ed+1-e}}\quad\text{and}\quad\frac{1}{\sqrt{ed+1-e}}\leqslant\frac{1}{\sqrt{ed}}\left(\sqrt{d}+\sqrt{e}-1\right)$$

by showing that

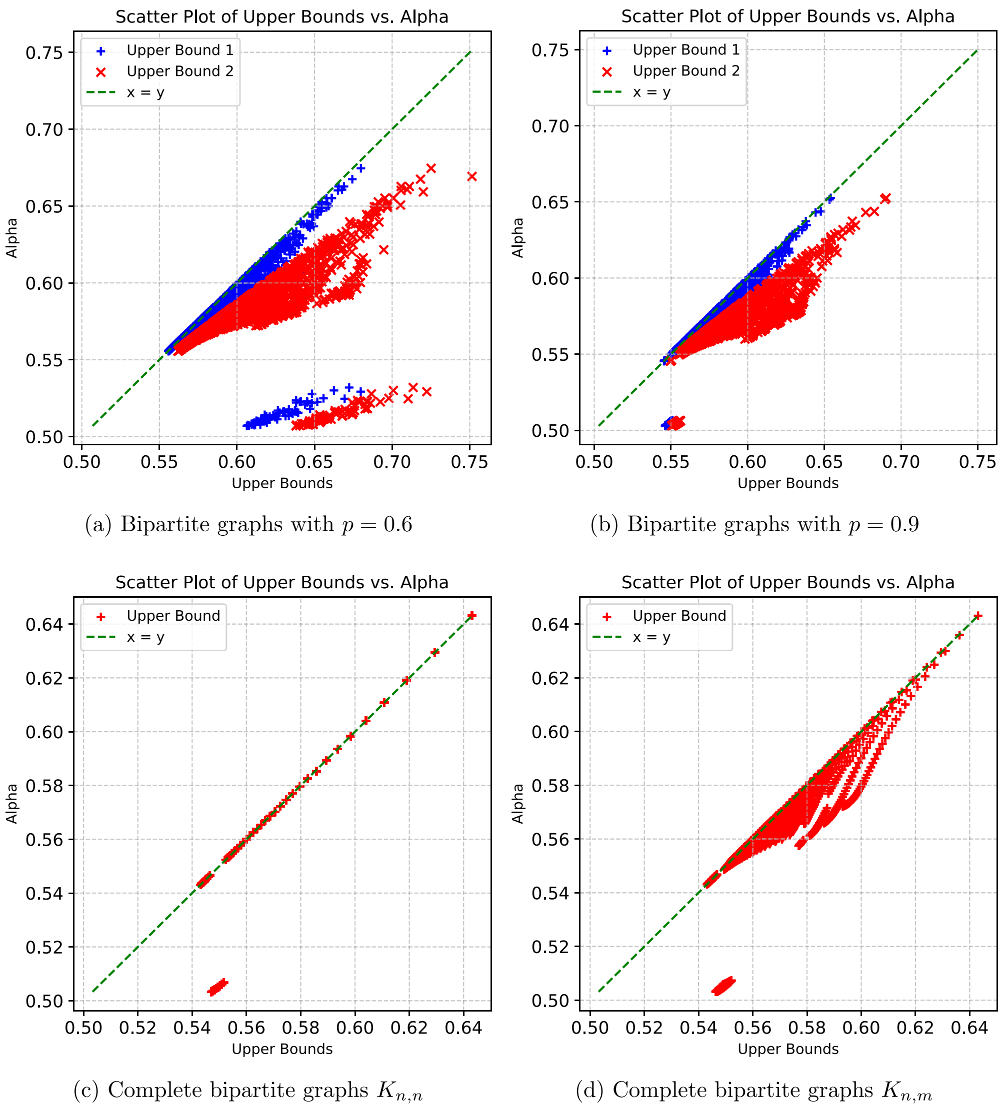
$$\ln\left[\frac{1}{\sqrt{d}}\left(\frac{\sqrt{d-1}}{\sqrt{d}}\right)^{d-1}\right]\leqslant\ln\left(\frac{1}{\sqrt{ed+1-e}}\right)\quad\text{and}\quad ed^2\leqslant(ed+1-e)\left(\sqrt{d}+\sqrt{e}-1\right)^2.$$

It follows from the concavity of the function

$$g(x):=\frac{1}{\sqrt{e}}\sqrt{x}+\left(1-\frac{1}{\sqrt{e}}\right)x$$

that the third inequality holds. □

We now turn to numerical results to validate the trend suggested by theorem 3. Figure 1 presents α_1 alongside our bounds for random bipartite graphs parameterized by the edge probability p so that the number of edges is $|E|=nmp$. Thus, the graph instances in (b) have more edges compared to (a), that is, they have a larger d_{ave} value, representing density, and the upper bounds become tighter compared to (a) as shown in figure 1. Although the upper bounds we have obtained in theorem 3 are not tight, they become tighter as a bipartite graph becomes denser which means that d_{ave} increases. In particular, we can get a tight bound for the complete bipartite graphs. Moreover, the intuitive conclusion that can be drawn from the following inequality is that as the size of the complete bipartite graph increases, the performance of the standard level-1 QAOA deteriorates, approaching that of random guessing.



(a) Bipartite graphs with $p = 0.6$

(b) Bipartite graphs with $p = 0.9$

(c) Complete bipartite graphs $K_{n,n}$

(d) Complete bipartite graphs $K_{n,m}$

Figure 1. Plots illustrating the tightness of the bounds in theorem 3 and corollary 4. In (a) and (b), bipartite graphs with two disjoint sets of n, m vertices. Here, n, m were randomly selected from 5 to 51 for $p = 0.6$ and $p = 0.9$, respectively, where p represents the edge probability. The standard level-1 QAOA approximation ratio α_1 was compared with two upper bounds provided in theorem 3. Compared to (a), the graph instances in (b) are denser, and the bounds are tighter. In (c) and (d), α_1 is compared with the bound provided in corollary 4 for complete bipartite graphs $K_{n,n}$ and $K_{n,m}$, respectively. In this case, it is confirmed that the bounds are tight. The points plotted below the boundary appear to have achieved relatively low approximation ratios compared to the upper bounds due to numerical optimization issues. The upper bounds are obtained from the values for the optimal parameter γ^*, and we observed that the approximation ratio drops as shown in the points below when the numerical optimization of the standard level-1 QAOA fails.

Corollary 4 (Complete bipartite graph). *Let $K_{n,m}$ be a complete bipartite graph that has two partitioned subsets V_1 and V_2 of vertices with $|V_1| = n$ and $|V_2| = m$.*

- *For $n = m \geqslant 2$, $\alpha_1 \leqslant \frac{1}{2} + \frac{1}{2\sqrt{n-1}}\left(1-\frac{1}{n}\right)^{n/2}$.*
- *For $n \neq m \geqslant 2$, $\alpha_1 \leqslant \frac{1}{2} + \frac{1}{4\sqrt{n-1}}\left(1-\frac{1}{n}\right)^{n/2} + \frac{1}{4\sqrt{m-1}}\left(1-\frac{1}{m}\right)^{m/2}$.*

Proof. For the case of $K_{n,n}$ with $n \geqslant 2$, all vertices has degree n and so, $|D_n| = 2n$. Then $\alpha_1 = \max_\gamma\left(\frac{1}{2} + \frac{1}{2}f_n(\gamma)\right)$. Observe that $f_n(\gamma) = 2\sin\gamma\cos^{n-1}\gamma$ and $f_n'(\gamma) = 2\cos^{n-2}\gamma(n\cos^2\gamma - (n-1))$. Then we can find the optimal γ^* such that $\cos\gamma^* = (1-\frac{1}{n})^{1/2}$ and $f_n(\gamma^*) = \frac{2}{\sqrt{n}}\left(\frac{\sqrt{n-1}}{\sqrt{n}}\right)^{n-1}$. Thus,

$$\alpha_1 \leqslant \frac{1}{2} + \frac{1}{2}f_n(\gamma^*) = \frac{1}{2} + \frac{1}{2\sqrt{n-1}}\left(1-\frac{1}{n}\right)^{n/2}.$$

Similarly, for the case of $K_{n,m}$ with $n \neq m \geqslant 2$, n vertices have degree m and m vertices have degree n, that is, $|D_n| = m$ and $|D_m| = n$. So, $\alpha_1 = \max_\gamma\left[\frac{1}{2} + \frac{1}{4}(f_n(\gamma) + f_m(\gamma))\right]$ and thus, we get

$$\begin{aligned}\alpha_1 &\leq \frac{1}{2} + \frac{1}{4}(f_n(\gamma^*) + f_m(\gamma^*)) \\ &= \frac{1}{2} + \frac{1}{4\sqrt{n-1}}\left(1-\frac{1}{n}\right)^{n/2} + \frac{1}{4\sqrt{m-1}}\left(1-\frac{1}{m}\right)^{m/2}.\end{aligned}$$

□

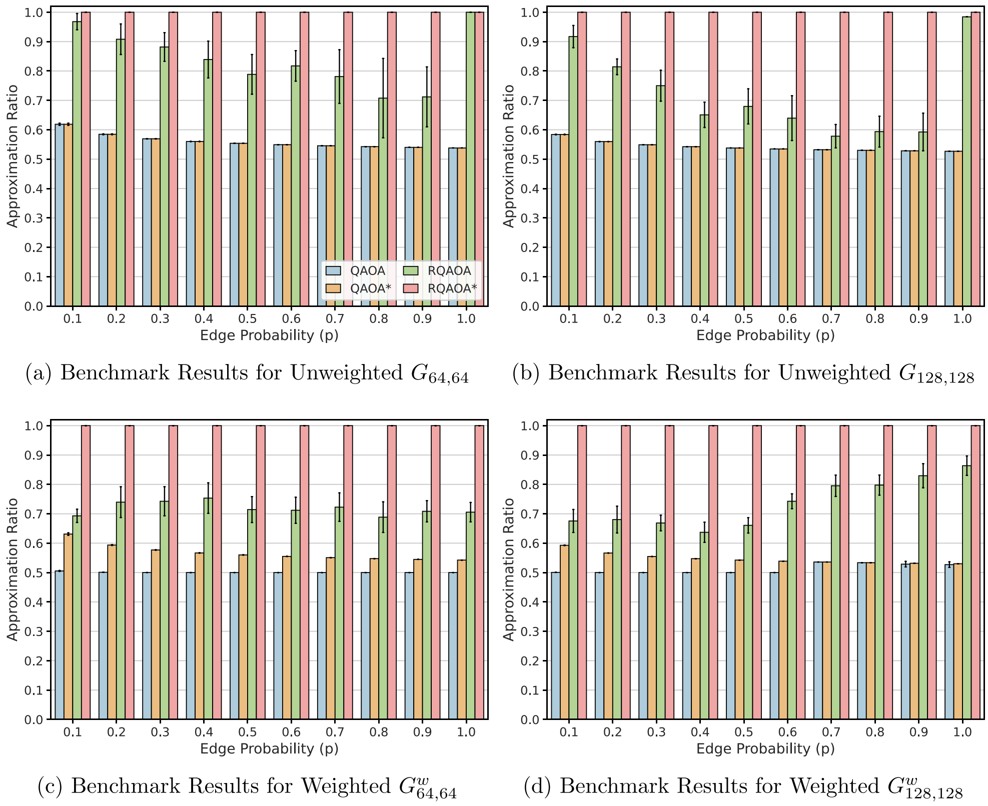
Remark that the upper bounds derived above are based on optimized parameters. This result implies that, even with optimal parameters, the performance of the standard level-1 QAOA remains fundamentally limited. However, the performance of the level-1 RQAOA can be influenced by parameter optimization. In the following section, we numerically demonstrate that while the level-1 RQAOA outperforms the standard QAOA at the same level, it may still fail to find the exact MAX-CUT solution on weighted bipartite graphs. This suggests that the performance of RQAOA could be affected by the parameter optimization inherent in the QAOA subroutine. To address these challenges, we introduce a modified RQAOA with a parameter-setting strategy and prove that it can exactly solve the problem on a graph with parity-signed weights, even if global optimality is not guaranteed at each QAOA iteration.

5. Improving performance using RQAOA$_1$

5.1. RQAOA$_1$ performance: superior to QAOA$_1$

In this section, we numerically demonstrate a clear separation in performance between the standard level-1 QAOA and the level-1 RQAOA when solving the MAX-CUT problem on unweighted and weighted bipartite graphs with two disjoint vertex sets of equal size n, denoted by $G_{n,n}$ and $G^w_{n,n}$, respectively, as shown in figure 2.

To compute the expectation values of level-1 QAOA, we employed the analytical form from [19], which applies to both unweighted and weighted graphs. The parameters (β,γ) were optimized using a line search over $\gamma \in [0,\pi]$ with 20 samples, followed by gradient descent, fixing $\beta^* = \pi/8$, which maximizes the expectation value in equation (2). For triangle-free graphs, this value of β^* is analytically optimal. However, because bipartiteness—and therefore



(a) Benchmark Results for Unweighted $G_{64,64}$

(b) Benchmark Results for Unweighted $G_{128,128}$

(c) Benchmark Results for Weighted $G^w_{64,64}$

(d) Benchmark Results for Weighted $G^w_{128,128}$

Figure 2. Benchmark results comparing level-1 QAOA, QAOA* (restricted-parameter variant), RQAOA, and RQAOA* (algorithm 7) on unweighted bipartite graphs $G_{64,64}$ and $G_{128,128}$, and weighted bipartite graphs $G^w_{64,64}$ and $G^w_{128,128}$, for edge probabilities $p \in \{0.1, 0.2, \ldots, 1.0\}$. Each point represents the mean over ten random instances, except for the unweighted graphs at $p = 1.0$, where only a single instance exists. Weighted graphs reuse the unweighted adjacencies, with edge weights drawn from a Gaussian distribution $\mathcal{N}(50, 25)$ and rounded to integers. QAOA uses a 20 point line search over $\gamma \in [0, \pi]$ followed by gradient descent with $\beta^* = \pi/8$; QAOA* restricts $0 < |\gamma| \leq \pi/(2w^*)$, where $w^* = \max_{(i,j)\in E} |w_{ij}|$. RQAOA employs a 20×20 grid over $(\gamma, \beta) \in [0, \pi]^2$ followed by gradient descent, while RQAOA* uses the same grid within the corresponding restricted domain. Across all 382 instances, RQAOA* attains an approximation ratio of 1, recovering the exact Max-Cut solutions. In contrast, RQAOA shows non-monotonic behavior: for unweighted graphs it is near-optimal at the extremes $p = 0.1$ and $p = 1.0$ but degrades for intermediate values of p; for weighted $G^w_{64,64}$ it hovers around approximation ratio ≈ 0.7 with mild p-dependence, while for weighted $G^w_{128,128}$ it is ≈ 0.65 for $p \leq 0.5$ and rises toward ≈ 0.85 for $p \geq 0.6$. Comparing QAOA and QAOA*, their performance is nearly identical on unweighted graphs, while on weighted graphs QAOA* outperforms QAOA on most instances across p, indicating that restricting γ by w^* chiefly benefits instances with heterogeneous weights.

triangle-freeness—may not be preserved under the contractions performed in RQAOA, β^* cannot be fixed in advance. Consequently, the level-1 RQAOA was optimized using a coarse grid search over $(\gamma, \beta) \in [0, \pi]^2$ with 20×20 points, followed by gradient descent on the best-found configuration.

As shown in figure 2, RQAOA consistently outperforms QAOA across all tested instances, establishing a clear separation in their performance. On unweighted bipartite graphs, QAOA achieves an average approximation ratio ≈ 0.55, while RQAOA yields strictly higher values for every instance. For weighted graphs, both algorithms show reduced performance, yet RQAOA maintains a distinct advantage. Despite this improvement, RQAOA does not attain the exact Max-Cut solution and exhibits a non-monotonic dependence on the edge probability p, performing well at the extremes $p = 0.1$ and $p = 1.0$ but degrading at intermediate densities.

Bipartite graphs (both unweighted and weighted) are well-structured, so the exact solution can in principle be easily determined. Nevertheless, our numerical results show that both the standard level-1 QAOA and RQAOA have limitations in solving the MAX-CUT problem even for such structured instances. This observation motivates a modified RQAOA with a novel parameter-scaling strategy, introduced in the following subsection, which also gives rise to restricted-parameter variants of both algorithms—denoted QAOA* and RQAOA* in figure 2

These restricted versions normalize the cost-Hamiltonian angles by the maximum edge weight w^*, effectively bounding the evolution parameter γ, thereby mitigating over-rotation and improving convergence. As a result, RQAOA* achieves an approximation ratio of 1 across all tested instances, recovering the exact Max-Cut solutions, while QAOA* performs comparably to QAOA on unweighted graphs but outperforms it on weighted instances, particularly on sparse instances. This parameter-scaling rule is a special case of the general result presented in [29]. The following subsection formally introduces this modified formulation and provides an analytical justification for its exactness on bipartite graphs.

5.2. *Modified RQAOA achieves the approximation ratio of 1*

In this subsection, we introduce a modified RQAOA with a tuned parameter regime that recovers exact solutions for the MAX-CUT problem on bipartite graphs and even more general cases. Let us first define a (undirected) weighted graph called a *graph with parity-signed weights* as follows. (In graph theory, a parity-signed graph, unlike our case, assigns a positive sign to edges between vertices with the same parity and a negative sign to edges between vertices with opposite parity.)

Definition 5. Let $G = (V, E, w)$ be a weighted graph with a vertex set V, an edge set E, and a weight function $w : E \to \mathbb{R} - \{0\}$. Suppose that there is a function $\sigma : V \to \{0, 1\}$. Then V can be partitioned into two subsets $V_0 = \{v \in V : \sigma(v) = 0\}$ and $V_1 = \{v \in V : \sigma(v) = 1\}$. We call $G = (V, E, w, \sigma)$ as a graph with parity-signed weights if

$$\mathrm{sgn}(w(e)) = (-1)^{\sigma(i)+\sigma(j)+1}$$

for all edge $e = (i, j) \in E$.

Remark 6. (i) The edge set E of a graph with parity-signed weights $G = (V, E, w, \sigma)$ can be written as $E = E_0^- \bigcup E_1^- \bigcup E_2^+$, where $E_0^- \subseteq V_0 \times V_0$, $E_1^- \subseteq V_1 \times V_1$, and $E_2^+ \subseteq V_0 \times V_1$. By the definition of G, $w(e_0), w(e_1) < 0$ and $w(e_2) > 0$ for $e_0 \in E_0^-, e_1 \in E_1^-$ and $e_2 \in E_2^+$.

(ii) Every positive weighted bipartite graph can be considered as a graph with parity-signed weights because it is the case when $E_0^- = E_1^- = \emptyset$.

We propose a modified RQAOA equipped with a novel parameter setting method and prove that it can get the exact MAX-CUT solution on a graph with parity-signed weights $G = (V, E, w, \sigma)$.

Algorithm 7 (Modified RQAOA$_1$). We only modify the first step of RQAOA$_1$ and the rest part of our algorithm is the same as the original RQAOA$_1$.

(i) Apply the level-1 QAOA to find the optimal parameters (β^*, γ^*) to maximize $F_1(\beta, \gamma)$ in the restricted domain where $0 < |\gamma| \leqslant \frac{\pi}{2w^*}$ and $0 < \beta < \frac{\pi}{4}$ with $w^* = \max_{(i,j)\in E} |w(ij)|$.

Theorem 8. *Our algorithm 7 can exactly solve the MAX-CUT problem on a graph with parity-signed weights. Hence, it can always get the exact MAX-CUT solution on positive weighted bipartite graphs.*

Remark that one can modify the standard level-1 QAOA by applying the same strategy. However, theorem 3 implies that any solution obtained from the standard level-1 QAOA, even when optimized to its global parameter optimum, cannot be close to the exact MAX-CUT solution for bipartite graphs which are special cases of graphs with parity-signed weights. By contrast, theorem 8 shows us that our modified RQAOA that uses the standard level-1 QAOA with a tuned parameter regime as a subroutine attains exact solutions. Hence, the two main theorems tell us that when using only the standard QAOA, we should increase its level to find a good solution, while our modified level-1 RQAOA is enough to get the exact solution.

Now, we provide the proof of theorem 8 in the following subsection.

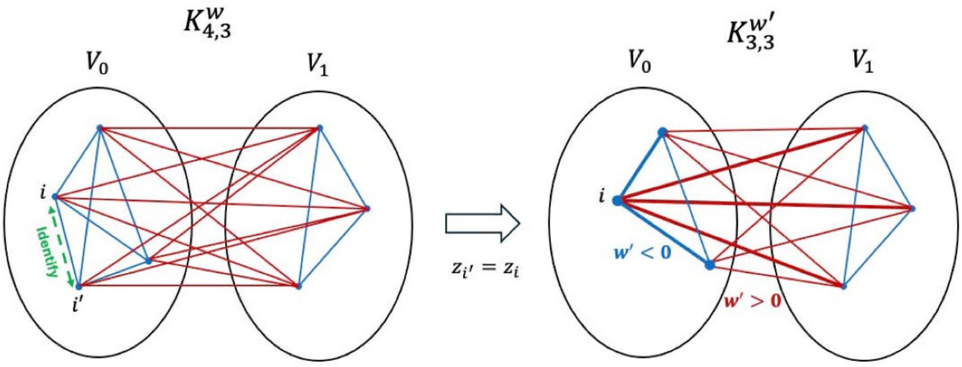
5.2.1. Proof of theorem 8. In this section, we prove that applying our parameter setting method can exactly solve the MAX-CUT problem on a graph with parity-signed weights in the following steps. Lemma 9 states that the MAX-CUT cost function value and the weight structure of G are preserved after identifying two connected vertices with the same or opposite sign when they belong to the same or different parts of the vertex set, respectively. Lemma 10 yields that our modified RQAOA assigns edge correlation as described in lemma 9. Consequently, combining lemmas 9 and 10, we can prove that our modified RQAOA can always find the exact MAX-CUT solution for a graph with parity-signed weights.

Lemma 9. *If we eliminate an edge e by identifying the vertices of e such that the variables of the vertices with the same or opposite sign when* $e \in E_0^- \cup E_1^-$ *or* $e \in E_2^+$*, respectively, the value of the MAX-CUT cost function on* $G = (V, E, w, \sigma)$ *and the weight structure of reduced graphs remain preserved.*

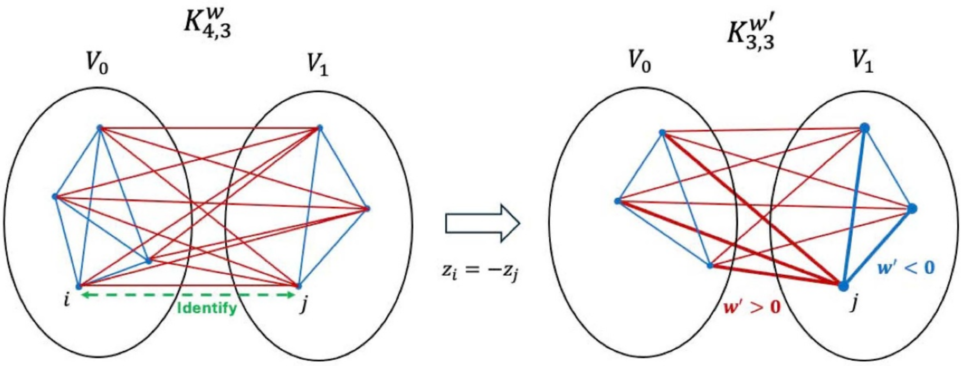
We can intuitively see that the weight structure of a graph with parity-signed weights is preserved after identifying two connected vertices for both cases as described in figure 3. Moreover, we can also figure out that the maximum value of the MAX-CUT cost function $C^w(z)$ which is defined in equation (A.1), does not change. Note that the maximum value of $C^w(z)$ on $K_{4,3}^w$ is the sum of all positive weights, that is, the sum of all red edge weights. In Case 1, the positive weights of the removed edges are maintained by merging with the edge weights of the red bold lines. Thus, the maximum value of $C^w(z)$ remains to be unchanged. See appendix A for the detailed proof.

Lemma 10. *If* $0 < |\gamma^*| \leqslant \frac{\pi}{2w^*}$ *with* $w^* = \max_{(i,j)\in E} |w(ij)|$ *and* $0 < \beta^* < \frac{\pi}{4}$ *in the QAOA*$_1$ *subroutine for solving the MAX-CUT problem on a graph with parity-signed weights* $G = (V, E, w, \sigma)$*, the edge expectation values are always positive or negative when the edges belong to* $E_0^- \cup E_1^-$ *or* E_2^+*, respectively.*

We also put the detailed proof of lemma 10 in appendix B. Now, we can prove theorem 8 as follows.



(a) Case 1: Identify two vertices i and i' in V_0 with the same sign



(b) Case 2: Identify a vertex i in V_0 and a vertex j in V_1 with opposite signs

Figure 3. A schematic diagram showing how the MAX-CUT cost function on $K^{w}_{4,3}$ and the weight structure of the graph change after one iteration of our modified RQAOA for the case when we identify (a) two connected vertices in the same partition with the same sign and (b) in the different partitions with different signs; the red/blue edges indicate the edges with the positive/negative weights, respectively. The bold lines represent the edges whose weights changed after identifying two vertices. After one iteration, the reduced graph remains to have the same weight structure as shown in the right graph for both cases.

Proof of theorem 8. The weighted MAX-CUT cost function on $G = (V, E, w, \sigma)$, where $V = V_0 \cup V_1$ with $|V_0| = n$ and $|V_1| = m$ and $E = E_0^- \cup E_1^- \cup E_2^+$, can be written as

$$C^{w}_{n,m}(z) = \frac{1}{2}\left(\sum_{e_2 \in E_2^+} w(e_2)^+ + \sum_{e_0 \in E_0^-} w(e_0)^- + \sum_{e_1 \in E_1^-} w(e_1)^- \right.$$
$$\left. - \sum_{e_2 \in E_2^+} w(e_2)^+ z_{e_2} - \sum_{e_0 \in E_0^-} w(e_0)^- z_{e_0} - \sum_{e_1 \in E_1^-} w(e_1)^- z_{e_1}\right),$$

where $w(e)^+$ and $w(e)^-$ indicate the positivity and negativity of the edge weight of the edge e, respectively. It follows from lemmas 9 and 10 that we can exactly calculate the new MAX-CUT cost function after k iterations. Assume that there are l eliminated edges with the correlated end vertices and $k-l$ eliminated edges with the anti-correlated end vertices during k iterations. Let $\mathcal{Z}$ be the subset of $\{-1,1\}^{n+m}$ satisfying those k constraints. Then the maximum value of the new MAX-CUT cost function we can obtain by imposing the k constraints is

$$\begin{aligned}\max_{z\in\mathcal{Z}} C^w_{n,m}(z) &= \sum_{e\in E^{\text{elm}}} w(e) - \sum_{e\in E^{\text{cre}}} w(e) + \max_{z'} C_{n-k_1,m-k_2}(z') \\ &= \sum_{e\in E^{\text{elm}}} w(e) - \sum_{e\in E^{\text{cre}}} w(e) \\ &\quad + \sum_{e_2\in E_2^+} w(e_2)^+ - \sum_{e\in E^{\text{elm}}} w(e) + \sum_{e\in E^{\text{cre}}} w(e) \\ &= \max_{z\in\{-1,1\}^{n+m}} C^w_{n,m}(z),\end{aligned}$$

where E^{elm} and E^{cre} are the set of all eliminated and created edges in the edge set E_2^+ during k iterations, respectively, and $k = k_1 + k_2$. Thus, by finding the exact MAX-CUT solution of the final reduced graph $G' = (V', E', w', \sigma)$, where $V' = V'_0 \cup V'_1$ with $|V'_0| = n - k_1$ and $|V'_1| = m - k_2$, the MAX-CUT cost function value of the initial graph will recover perfectly using the k constraints. This completes the proof. □

6. Conclusion

We have shown the upper bound of the approximation ratio of the level-1 QAOA in solving the MAX-CUT problem on bipartite graphs. Our findings indicate that the performance of the standard level-1 QAOA generally deteriorates as a bipartite graph becomes denser and as the size of a complete bipartite graph grows.

In addition, we have numerically shown that although RQAOA outperforms the standard QAOA at the same level in solving the MAX-CUT problem on weighted bipartite graphs, there is still room for further performance improvement. To get a better performance of RQAOA, we have proposed a modified RQAOA with a parameter setting strategy and have rigorously proved that it can exactly solve the MAX-CUT problem on a graph with parity-signed weights which is a generalization of a positive weighted bipartite graph.

The standard level-1 QAOA itself has limitations in parameter optimization. Although increasing the QAOA depth or employing alternative mixing Hamiltonians may improve performance, the central point of our work is that we have presented a method which attains exact solutions for certain instances even when restricted to use the standard level-1 QAOA as a subroutine, and have rigorously proved the effectiveness of the method. In fact, since our modified RQAOA even restricts the optimization region in the QAOA subroutine, it may not always yield true optimal parameters at each iteration. Nevertheless, our algorithm demonstrates better performance than the original level-1 RQAOA in solving the MAX-CUT problem on positive weighted bipartite graphs. What aspects of our modified RQAOA lead to its performance improvement? Unfortunately, we have not yet obtained a complete answer to this question from this work. Hence, rigorously analyzing it would be an important area for future research.

We have proved that our strategy restricting the optimization region yields the optimal performance of RQAOA in solving the MAX-CUT problem for graphs with parity-signed weights. Our results support that good performance of RQAOA does not require good optimization of QAOA, although refining the grid search area more finely in the classical optimization of the QAOA subroutine to obtain a better optimal parameter is expected to improve RQAOA's performance. In addition, in the case of well-structured graphs like parity-signed weighted graphs, preserving the graph structure across iterations appears to enhance the RQAOA performance. Therefore, for future works, it would be interesting to study if our strategy could fit into more general instances such as triangle-free graphs or triangle-free d-regular graphs since almost all triangle-free graphs are bipartite [33].

Data availability statement

The data that support the findings of this study are openly available at the following URL/DOI: https://github.com/vijeycreative/Modified_RQAOA.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) of Korea grant funded by the Ministry of Science and ICT (MSIT) (Grant No. NRF-2022R1C1C2006396). S L acknowledges support from the NRF grants funded by the MSIT (No. RS-2022-NR068791 and No. RS-2024-00432214), and Creation of the Quantum Information Science R&D Ecosystem (No. RS-2023-NR068116) through the NRF funded by the MSIT. H K is supported by the KIAS Individual Grant No. CG085302 at Korea Institute for Advanced Study. E B and H K acknowledge support from the NRF of Korea (Grant No. RS-2023-NR119931). V V is supported by the A*STAR Start Up Fund (KIMR220701aIMRSEF), Q.InC Strategic Research and Translational Thrust, and the National Quantum Scholarship Scheme.

Appendix A. Proof of lemma 9

In this section, we provide the detailed calculations in the proof of lemma 9 which states that the reduced graph remains to preserve the weight structure of a graph with parity-signed weights. Moreover, the MAX-CUT cost function will not change after identifying two connected vertices with the same or opposite sign when they have the same or opposite parity, respectively. The MAX-CUT cost function $C^w(z)$ on a weighted graph $G = (V, E, w)$ can be formulated as

$$C^w(z) = \frac{1}{2}\sum_{e\in E} w(e)(1 - z_e), \tag{A.1}$$

where $z_e = z_i z_j$ for $e = (i,j) \in E$ with $z_i \in \{-1, 1\}$ for all $i \in V$, and $w(e)$ is the edge weight of the edge e.

Let us denote the cost function of the weighted MAX-CUT problem over a graph with parity-signed weights $G_{n,m} = (V_0 \cup V_1, E_0^- \cup E_1^- \cup E_2^+, w, \sigma)$ with $|V_0| = n$ and $|V_1| = m$ by

$$C^w_{n,m}(z) = \frac{1}{2}\left(\sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') - \sum_{k\in V_0}\sum_{l\in V_1} w(kl)\, z_k z_l - \sum_{k\neq k'\in V_0} w(kk')\, z_k z_{k'} - \sum_{l\neq l'\in V_1} w(ll')\, z_l z_{l'}\right), \tag{A.2}$$

where $w(kl)$ denotes the edge weight of the edge (k,l).

Case 1) Assume that we pick $e^* \in E_0^- \cup E_1^-$. Without loss of generality, let $e^* = (i,i') \in E_0^-$. We can rewrite the cost function as

$$C^w_{n,m}(z) = \frac{1}{2}\left(\sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') - \sum_{k\in V_0-\{i'\}}\sum_{l\in V_1} w(kl) z_k z_l - \sum_{l\in V_1} w(i'l) z_{i'} z_l - \sum_{k\neq k'\in V_0-\{i'\}} w(kk') z_k z_{k'} - \sum_{k\in V_0-\{i'\}} w(ki') z_k z_{i'} - \sum_{l\neq l'\in V_1} w(ll') z_l z_{l'}\right). \tag{A.3}$$

We impose the constraint $z_{i'} = z_i$ on the cost function $C^w_{n,m}(z)$ to get the new cost function

$$\begin{aligned} C^w_{n,m}(z) &= \frac{1}{2}\left(\sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') - \sum_{k\in V_0-\{i'\}}\sum_{l\in V_1} w(kl)\, z_k z_l - \sum_{l\in V_1} w(i'l)\, z_i z_l - \sum_{k\neq k'\in V_0-\{i'\}} w(kk')\, z_k z_{k'} - \sum_{k\in V_0-\{i'\}} w(ki')\, z_k z_i - \sum_{l\neq l'\in V_1} w(ll')\, z_l z_{l'}\right) \\ &= \frac{1}{2}\left(\sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') - \sum_{k\in V_0-\{i'\}}\sum_{l\in V_1} w(kl) z_k z_l - \sum_{l\in V_1} w(i'l) z_i z_l - \sum_{k\neq k'\in V_0-\{i'\}} w(kk') z_k z_{k'} - \sum_{k\in V_0-\{i,i'\}} w(ki') z_k z_i - w(ii') - \sum_{l\neq l'\in V_1} w(ll') z_l z_{l'}\right) \end{aligned} \tag{A.4}$$

$$
\begin{aligned}
&= \frac{1}{2}\left(\sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll')\right) \\
&\quad - \frac{1}{2}\left(\underbrace{\sum_{k\in V_0-\{i'\}}\sum_{l\in V_1} w(kl) + \sum_{l\in V_1} w(i'l)}_{=\sum_{k\in V_0}\sum_{l\in V_1} w(kl)}\right. \\
&\quad \left. + \underbrace{\sum_{k\neq k'\in V_0-\{i'\}} w(kk') + \sum_{k\in V_0-\{i,i'\}} w(ki') + w(ii')}_{=\sum_{k\neq k'\in V_0} w(kk')} + \sum_{l\neq l'\in V_1} w(ll')\right) \\
&\quad + C^w_{n-1,m}(z') \\
&= C^w_{n-1,m}(z')\,. \qquad \text{(A.5)}
\end{aligned}
$$

Here, $C^w_{n-1,m}(z')$ is the MAX-CUT cost function of the reduced graph $G_{n-1,m} = (V_0 - \{i'\} \cup V_1, E', w', \sigma)$ which can be formulated as

$$
C^w_{n-1,m}(z') = \frac{1}{2}\sum_{e\in E'} w'(e)(1 - z'_e),
$$

where $E' = E - \{e^*\}, z' = (z_1,\ldots,\hat{z_{i'}},\ldots,z_{n+m}) \in \{-1,1\}^{n+m-1}$ is the vector $z \in \{-1,1\}^{n+m}$ with the component $z_{i'}$ excluded, and

$$
w'(e) = \begin{cases} w(e) & \text{if} i \notin e \in E, \\ w(ki) + w(ki') & \text{if} e = (k,i) \in E_0^- \text{for} k \in V_0 - \{i,i'\}, \\ w(il) + w(i'l) & \text{if} e = (i,l) \in E_2^+ \text{for} l \in V_1. \end{cases}
$$

Thus, the value of the MAX-CUT cost function does not change after identifying two correlated vertices with the same sign. Since $w(ki)^- + w(ki')^-$ is less than zero and $w(il)^+ + w(i'l)^+$ is greater than zero, the reduced graph $G_{n-1,m}$ is also a graph with parity-signed weights as shown in figure 3.

Case 2) Assume that we pick $e^* \in E_2^+$. Without loss of generality, let $e^* = (i,j) \in E_2^+$. Similarly, we can rewrite the MAX-CUT cost function as

$$
\begin{aligned}
C^w_{n,m}(z) = \frac{1}{2}\Bigg(&\sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') \\
&- \sum_{k\in V_0-\{i\}}\sum_{l\in V_1} w(kl) z_k z_l - \sum_{l\in V_1} w(il) z_i z_l - \sum_{k\neq k'\in V_0-\{i\}} w(kk') z_k z_{k'} \\
&- \sum_{k\in V_0-\{i\}} w(ki) z_k z_i - \sum_{l\neq l'\in V_1} w(ll') z_l z_{l'}\Bigg)\,. \qquad \text{(A.6)}
\end{aligned}
$$

We impose the constraint $z_i = -z_j$ on the cost function $C^w_{n,m}(z)$ to get the new cost function

$$
\begin{aligned}
C^w_{n,m}(z) = \frac{1}{2}\Bigg(& \sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') \\
& - \sum_{k\in V_0-\{i\}}\sum_{l\in V_1} w(kl)z_k z_l - \sum_{l\in V_1} w(il)(-z_j)z_l - \sum_{k\neq k'\in V_0-\{i\}} w(kk')z_k z_{k'} \\
& - \sum_{k\in V_0-\{i\}} w(ki)z_k(-z_j) - \sum_{l\neq l'\in V_1} w(ll')z_l z_{l'} \Bigg) \qquad \text{(A.7)}\\
= \frac{1}{2}\Bigg(& \sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') - \sum_{k\in V_0-\{i\}}\sum_{l\in V_1} w(kl)z_k z_l \\
& - \sum_{l\in V_1-\{j\}} (-w(il))z_j z_l - (-w(ij)) - \sum_{k\neq k'\in V_0-\{i\}} w(kk')z_k z_{k'} \\
& - \sum_{k\in V_0-\{i\}} (-w(ki))z_k z_j - \sum_{l\neq l'\in V_1} w(ll')z_l z_{l'} \Bigg) \qquad \text{(A.8)}\\
= \frac{1}{2}\Bigg(& \sum_{k\in V_0}\sum_{l\in V_1} w(kl) + \sum_{k\neq k'\in V_0} w(kk') + \sum_{l\neq l'\in V_1} w(ll') \Bigg) \\
& - \frac{1}{2}\Bigg(\underbrace{\sum_{k\in V_0-\{i\}}\sum_{l\in V_1} w(kl) + \sum_{l\in V_1-\{j\}} (-w(il)) + (-w(ij))}_{=\sum_{k\in V_0}\sum_{l\in V_1} w(kl) - 2\sum_{l\in V_1} w(il)} \\
& + \underbrace{\sum_{k\neq k'\in V_0-\{i\}} w(kk') + \sum_{k\in V_0-\{i\}} (-w(ki))}_{=\sum_{k\neq k'\in V_0} w(kk') - 2\sum_{k\in V_0-\{i\}} w(ki)} + \sum_{l\neq l'\in V_1} w(ll') \Bigg) \\
& + C^w_{n-1,m}(z') \qquad \text{(A.9)}\\
= & \sum_{l\in V_1} w(il) + \sum_{k\in V_0-\{i\}} w(ki) + C^w_{n-1,m}(z'). \qquad \text{(A.10)}
\end{aligned}
$$

Here, $C^w_{n-1,m}(z')$ is the MAX-CUT cost function of the reduced graph $G_{n-1,m} = (V_0 - \{i\} \cup V_1, E', w', \sigma)$ which can be formulated as

$$C^w_{n-1,m}(z') = \frac{1}{2}\sum_{e\in E'} w'(e)(1 - z'_e),$$

where $E' = E - \{e^*\}$, $z' = (z_1, \ldots, \hat{z_i}, \ldots, z_{n+m}) \in \{-1,1\}^{n+m-1}$ is the vector $z \in \{-1,1\}^{n+m}$ with the component z_i excluded, and

$$w'(e) = \begin{cases} w(e) & \text{if} i \notin e \in E, \\ w(kj) - w(ki) & \text{if} e = (k,j) \in E_2^+ \text{for} k \in V_0 - \{i\}, \\ -w(il) + w(jl) & \text{if} e = (j,l) \in E_1^- \text{for} l \in V_1 - \{j\}. \end{cases}$$

Note that the sum of edge weights $\sum_{l \in V_1} w(il) + \sum_{k \in V_0 - \{i\}} w(ki)$ in the new cost function ${C^w_{n,m}}'(z)$ exactly corresponds to the total sum of edge weights gained or lost in the edge set E_2^+ when the edge (i,j) is removed. Therefore, we have also proved that the maximum value of the cost function will not change after identifying two anti-correlated vertices with opposite sign. Moreoever, since $w(kj)^+ - w(ki)^- > 0$ and $-w(il)^+ + w(jl)^- < 0$, the reduced graph $G_{n-1,m}$ is also a graph with parity-signed weights as shown in figure 3.

Appendix B. Proof of lemma 10

In this section, we provide the detailed proof of lemma 10. For simplicity, we will denote the edge weight $w(ij)$ as w_{ij} for an edge $e = (i,j)$ in this section.

Proof. We can use the analytic form of the edge expectation values in the level-1 QAOA subroutine for weighted graphs [19]. Let us denote $\mathcal{N}_i$ and $\mathcal{T}_{ij}$ as the set of vertices that are adjacent to a vertex i and form a triangle with the edge (i,j), respectively. Then the edge expectation value of the edge (i,j) is

$$\begin{aligned} \langle Z_i Z_j \rangle = & -\frac{1}{2} \sin(4\beta) \sin(w_{ij}\gamma) \left(\prod_{k \neq j \in \mathcal{N}_i} \cos(w_{ki}\gamma) + \prod_{l \neq i \in \mathcal{N}_j} \cos(w_{lj}\gamma) \right) \\ & -\frac{1}{2} S \sin^2(2\beta) \prod_{\substack{k \neq j \in \mathcal{N}_i \\ k \notin \mathcal{T}_{ij}}} \cos(w_{ki}\gamma) \prod_{\substack{l \neq j \in \mathcal{N}_j \\ l \notin \mathcal{T}_{ij}}} \cos(w_{lj}\gamma), \end{aligned}$$

where

$$S = \prod_{t \in \mathcal{T}_{ij}} \cos((w_{it} + w_{tj})\gamma) - \prod_{t \in \mathcal{T}_{ij}} \cos((w_{it} - w_{tj})\gamma).$$

Case 1) Assume that we pick $e^* = (i^*, j^*) \in E_0^- \cup E_1^-$ to be eliminated. Without loss of generality, let $e^* \in E_0^-$. Let $E_{i^*} = \{(i^*, k) \in E : k \neq j^*\}$ be the set of edges connected to the vertex i^* excluding the edge e^*, and $F_{i^*} = \{(i^*, k) \in E : k \in \mathcal{T}_{i^* j^*}\}$ be the set of edges connected to the vertex i^*, together with the edge e^*, form triangles. Now, we can rewrite the edge expectation value in terms of the edges with these notations as

$$\langle Z_{i^*} Z_{j^*} \rangle = -\frac{1}{2} S_1 \sin(4\beta) \sin\left(w_{e^*}^- \gamma\right) - \frac{1}{2} S_2 S_3 \sin^2(2\beta),$$

where

$$S_1 = \prod_{e_0 \in E_{i^*} \cap E_0^-} \cos\left(w_{e_0}^- \gamma\right) \prod_{e_2 \in E_{i^*} \cap E_2^+} \cos\left(w_{e_2}^+ \gamma\right) + \prod_{e_0' \in E_{j^*} \cap E_0^-} \cos\left(w_{e_0'}^- \gamma\right) \prod_{e_2' \in E_{j^*} \cap E_2^+} \cos\left(w_{e_2'}^+ \gamma\right),$$

$$S_2 = \prod_{\substack{e_0 \in E_{i^*} \cap E_0^- \\ e_0 \notin F_{i^*}}} \cos\left(w_{e_0}^- \gamma\right) \prod_{\substack{e_2 \in E_{i^*} \cap E_2^+ \\ e_2 \notin F_{i^*}}} \cos\left(w_{e_2}^+ \gamma\right) \prod_{\substack{e_0' \in E_{j^*} \cap E_0^- \\ e_0' \notin F_{j^*}}} \cos\left(w_{e_0'}^- \gamma\right) \prod_{\substack{e_2' \in E_{j^*} \cap E_2^+ \\ e_2' \notin F_{j^*}}} \cos\left(w_{e_2'}^+ \gamma\right),$$

$$S_3 = \prod_{\substack{f_0 \in F_{i^*} \cap E_0^- \\ f_2 \in F_{i^*} \cap E_2^+}} \cos\left(\left(w_{f_0}^- + w_{\tilde{f}_0}^-\right)\gamma\right) \cos\left(\left(w_{f_2}^+ + w_{\tilde{f}_2}^+\right)\gamma\right) - \prod_{\substack{f_0 \in F_{i^*} \cap E_0^- \\ f_2 \in F_{i^*} \cap E_2^+}} \cos\left(\left(w_{f_0}^- - w_{\tilde{f}_0}^-\right)\gamma\right) \cos\left(\left(w_{f_2}^+ - w_{\tilde{f}_2}^+\right)\gamma\right),$$

and $\tilde{f}$ represents the third edge of the triangle formed by the edges e^* and f. We denote the negative and positive edge weight of the edge e by w_e^- and w_e^+, respectively. We can easily see that if $0 < |\gamma| \leqslant \frac{\pi}{2w_{ij}^*}$ with $w_{ij}^* = \max_{(i,j)\in E} |w_{ij}|$, S_1 and S_2 are always positive while S_3 is negative because the cosine function is decreasing on that region. Thus, the edge expectation values of the edges belonging to E_0^- (and E_1^-) are always positive.

Case 2) Assume that we pick $e^* = (i^*, j^*) \in E_2^+$ to be eliminated. We can apply a similar argument to the above case. The main differences will be the sign of $\sin(w_{e^*}^+\gamma)$ and S_3. Let us rewrite the edge expectation value for this case as

$$\langle Z_{i^*} Z_{j^*} \rangle = -\frac{1}{2} S_1' \sin(4\beta) \sin\left(w_{e^*}^+\gamma\right) - \frac{1}{2} S_2' S_3' \sin^2(2\beta),$$

$$S_1' = \prod_{e_0 \in E_{i^*} \cap E_0^-} \cos\left(w_{e_0}^-\gamma\right) \prod_{e_2 \in E_{i^*} \cap E_2^+} \cos\left(w_{e_2}^+\gamma\right) + \prod_{e_1 \in E_{j^*} \cap E_1^-} \cos\left(w_{e_1}^-\gamma\right) \prod_{e_2' \in E_{j^*} \cap E_2^+} \cos\left(w_{e_2'}^+\gamma\right),$$

$$S_2' = \prod_{\substack{e_0 \in E_{i^*} \cap E_0^- \\ e_0 \notin F_{i^*}}} \cos\left(w_{e_0}^-\gamma\right) \prod_{\substack{e_2 \in E_{i^*} \cap E_2^+ \\ e_2 \notin F_{i^*}}} \cos\left(w_{e_2}^+\gamma\right) \prod_{\substack{e_1 \in E_{j^*} \cap E_1^- \\ e_1 \notin F_{j^*}}} \cos\left(w_{e_2}^-\gamma\right) \prod_{\substack{e_2' \in E_{j^*} \cap E_2^+ \\ e_2' \notin F_{j^*}}} \cos\left(w_{e_2'}^+\gamma\right),$$

$$S_3' = \prod_{\substack{f_0 \in F_{i^*} \cap E_0^- \\ f_2 \in F_{i^*} \cap E_2^+}} \cos\left(\left(w_{f_0}^- + w_{\tilde{f}_0}^+\right)\gamma\right) \cos\left(\left(w_{f_2}^+ + w_{\tilde{f}_2}^-\right)\gamma\right) - \prod_{\substack{f_1 \in F_{i^*} \cap E_1^- \\ f_2 \in F_{i^*} \cap E_2^+}} \cos\left(\left(w_{f_1}^- - w_{\tilde{f}_1}^+\right)\gamma\right) \cos\left(\left(w_{f_2}^+ - w_{\tilde{f}_2}^-\right)\gamma\right).$$

Suppose that $0 < |\gamma| \leqslant \frac{\pi}{2w^*}$ with $w^* = \max_{(i,j)\in E} w_{ij}$. Similar to Case 1), S_1' and S_2' are always positive. However, in contrast to Case 1), $w_{e^*} > 0$ so that $\sin(w_{e^*}\gamma)$ has the opposite sign, and S_3' is positive in this case because the cosine function is decreasing on that region together with the fact that

$$|\left(w_{f_0}^- + w_{\tilde{f}_0}^+\right)\gamma| \leqslant |\left(w_{f_1}^- - w_{\tilde{f}_1}^+\right)\gamma| \text{ and } |\left(w_{f_2}^+ + w_{\tilde{f}_2}^-\right)\gamma| \leqslant |\left(w_{f_2}^+ - w_{\tilde{f}_2}^-\right)\gamma|.$$

Therefore, the edge expectation values of the edges belonging to E_2^+ are always negative. □

ORCID iDs

Eunok Bae 0000-0001-7531-0992
Hyukjoon Kwon 0000-0001-5520-0905
V Vijendran 0000-0003-3398-1821
Soojoon Lee 0000-0003-2925-1017

References

[1] Shor P M 1997 *SIAM J. Comput.* **26** 1484–509
[2] Peruzzo A, McClean J, Shadbolt P, Yung M H, Zhou X Q, Love P J, Aspuru-Guzik A and O'Brienc J L 2014 *Nat. Commun.* **5** 4213
[3] Farhi E, Goldstone J and Gutmann S A Quantum approximate optimization algorithm (arXiv:1411.4028)
[4] Moll N, Barkoutsos P, Bishop L S, Chow J M, Cross A, Egger D J, Filipp S, Fuhrer A, Gambetta J M and Ganzhorn M 2018 *Quantum Sci. Technol.* **3** 030503
[5] Cerezo M *et al* 2021 *Nat. Rev. Phys.* **3** 625–44
[6] Tilly J *et al* 2022 *Phys. Rep.* **986** 1–128
[7] Schuld M and Killoran N 2019 *Phys. Rev. Lett.* **122** 040504
[8] Wang Z, Hadfield S, Jiang Z and Rieffel E G 2018 *Phys. Rev. A* **97** 022304
[9] Zhou L, Wang S T, Choi S, Pichler H and Lukin M D 2020 *Phys. Rev. X* **10** 021067
[10] Hastings M B Classical and quantum bounded depth approximation algorithms (arXiv:1905.07047)
[11] Bravyi S, Kliesch A, König R and Tang E 2020 *Phys. Rev. Lett.* **125** 260504
[12] Farhi E, Goldstone J and Gutmann S The quantum approximate optimization algorithm needs to see the whole graph: a typical case (arXiv:2004.09002)
[13] Marwaha K 2021 *Quantum* **5** 437
[14] Barak B and Marwaha K Classical algorithms and quantum limitations for maximum cut on high-girth graphs (arXiv:2106.05900)
[15] Hadfield S, Wang Z, O'Gorman B, Rieffel E G, Venturelli D and Biswas R 2019 *Algorithms* **12** 34
[16] Egger D J, Marecek J and Woerner S 2021 *Quantum* **5** 479
[17] Herrman R, Lotshaw P C, Ostrowski J, Humble T S and Siopsis G 2022 *Sci. Rep.* **12** 6781
[18] Zhu L, Tang H L, Barron G S, Calderon-Vargas F A, Mayhall N J, Barnes E and Economou S E 2022 *Phys. Rev. Res.* **4** 033029
[19] Vijendran V, Das A, Koh D E, Assad S M and Lam P K 2024 *Quantum Sci. Technol.* **9** 025010
[20] Bravyi S, Gosset D, Grier D and Schaeffer L Classical algorithms for forrelation (arXiv:2102.06963)
[21] Bravyi S, Kliesch A, König R and Tang E 2022 *Quantum* **6** 678
[22] Patel Y J, Jerbi S, Bäck T and Dunjko V 2024 *EPJ Quantum Technol.* **11** 6
[23] Bae E and Lee S 2024 *Quantum Inf. Process.* **23** 78
[24] Fischer V, Passek M, Wagner F, Finžgar J R, Palackal L and Mendl C B The role of quantum and classical correlations in shrinking algorithms for optimization (arXiv:2404.17242)
[25] Vijendran V, Koh D E, Lam P K and Assad S M Classical and quantum heuristics for the binary paint shop problem (arXiv:2509.15294)
[26] Finžgar J R, Kerschbaumer A, Schuetz M J, Mendl C B and Katzgraber H G 2024 *PRX Quantum* **5** 020327
[27] Brady L T and Hadfield S Iterative quantum algorithms for maximum independent set: a tale of low-depth quantum algorithms (arXiv:2309.13110)
[28] Wagner F, Nüßlein J and Liers F Enhancing quantum algorithms for quadratic unconstrained binary optimization via integer programming (arXiv:2302.05493)
[29] Vijendran V, Koh D E, Bae E, Kwon H, Lam P K and Assad S M Near-optimal parameter tuning of level-1 QAOA for Ising models (arXiv:2501.16419v2)
[30] Sureshbabu S H, Herman D, Shaydulin R, Basso J, Chakrabarti S, Sun Y and Pistoia M 2024 *Quantum* **8** 1231
[31] Shaydulin R, Lotshaw P C, Larson J, Ostrowski J and Humble T S 2023 *ACM Trans. Quantum Comput.* **4** 1–15
[32] Lyngfelt I and García-Álvarez L Symmetry-informed transferability of optimal parameters in the quantum approximate optimization algorithm (arXiv:2407.04496v2)
[33] Promela H J, Schickingerb T and Stegerb A 2002 *Discrete Math.* **257** 531–40