

Progressive quantum algorithm for maximum independent set with quantum alternating operator ansatz

Xiao-Hui Ni(倪晓慧), Ling-Xiao Li(李凌霄), Yan-Qi Song(宋燕琪), Zheng-Ping Jin(金正平), Su-Juan Qin(秦素娟), Fei Gao(高飞)

Citation: Chin. Phys. B, 2025, 34 (**7**): 070304. DOI: 10.1088/1674-1056/addd83
Journal homepage: http://cpb.iphy.ac.cn; http://iopscience.iop.org/cpb

What follows is a list of articles you may be interested in

Domain adaptation method inspired by quantum convolutional neural network

Chunhui Wu(武春辉), Junhao Pei(裴骏豪), Yihua Wu(吴逸华), Anqi Zhang(张安琪), Shengmei Zhao(赵生妹)
Chin. Phys. B, 2025, 34 (**7**): 070302. DOI: 10.1088/1674-1056/adc7ed

Distributed quantum circuit partitioning and optimization based on combined spectral clustering and search tree strategies

Zilu Chen(陈子禄), Zhijin Guan(管致锦), Shuxian Zhao(赵书娴), and Xueyun Cheng(程学云)
Chin. Phys. B, 2025, 34 (**5**): 050305. DOI: 10.1088/1674-1056/adbada

Effect of pseudo-random number on the security of quantum key distribution protocol

Xiao-Liang Yang(杨晓亮), Yu-Qing Li(李毓擎), and Hong-Wei Li(李宏伟)
Chin. Phys. B, 2025, 34 (**2**): 020301. DOI: 10.1088/1674-1056/ad99c9

Quantum color image encryption: Dual scrambling scheme based on DNA codec and quantum Arnold transform

Tao Cheng(程涛), Run-Sheng Zhao(赵润盛), Shuang Wang(王爽), Kehan Wang(王柯涵), and Hong-Yang Ma(马鸿洋)
Chin. Phys. B, 2025, 34 (**1**): 010305. DOI: 10.1088/1674-1056/ad8a4b

Variational data encoding and correlations in quantum-enhanced machine learning

Ming-Hao Wang(王明浩) and Hua Lü(吕桦)
Chin. Phys. B, 2024, 33 (**9**): 090307. DOI: 10.1088/1674-1056/ad5c3b

---

COMPUTATIONAL PROGRAMS FOR PHYSICS

Progressive quantum algorithm for maximum independent set with quantum alternating operator ansatz

Xiao-Hui Ni(倪晓慧)[1,2], Ling-Xiao Li(李凌霄)[1], Yan-Qi Song(宋燕琪)[3], Zheng-Ping Jin(金正平)[1,2], Su-Juan Qin(秦素娟)[1,2,†], and Fei Gao(高飞)[1,2,‡]

[1] *State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*
[2] *School of Cyberspace Security, Beijing University of Posts and Telecommunications, Beijing* 100876, *China*
[3] *China Academy of Information and Communications Technology, Beijing* 100191, *China*

(Received 21 April 2025; revised manuscript received 27 May 2025; accepted manuscript online 28 May 2025)

The quantum alternating operator ansatz algorithm (QAOA+) is widely used for constrained combinatorial optimization problems (CCOPs) due to its ability to construct feasible solution spaces. In this paper, we propose a progressive quantum algorithm (PQA) to reduce qubit requirements for QAOA+ in solving the maximum independent set (MIS) problem. PQA iteratively constructs a subgraph likely to include the MIS solution of the original graph and solves the problem on it to approximate the global solution. Specifically, PQA starts with a small-scale subgraph and progressively expands its graph size utilizing heuristic expansion strategies. After each expansion, PQA solves the MIS problem on the newly generated subgraph using QAOA+. In each run, PQA repeats the expansion and solving process until a predefined stopping condition is reached. Simulation results show that PQA achieves an approximation ratio of 0.95 using only 5.57% (2.17%) of the qubits and 17.59% (6.43%) of the runtime compared with directly solving the original problem with QAOA+ on Erdös–Rényi (3-regular) graphs, highlighting the efficiency and scalability of PQA.

Keywords: quantum alternating operator ansatz algorithm (QAOA+), constrained combinatorial optimization problems (CCOPs), maximum independent set (MIS), feasible space

PACS: 03.67.Ac

DOI: 10.1088/1674-1056/addd83 **CSTR:** 32038.14.CPB.addd83

1. Introduction

Quantum computing has made significant strides in recent years, showing great potential in solving complex problems such as database searching,[1] prime factorization,[2] and solving equations.[3] Despite these promising advancements, the practical implementation of quantum algorithms faces challenges due to the noise and limited qubit capacity of current noisy intermediate-scale quantum devices.[4] To fully harness the power of noisy intermediate-scale quantum devices, hybrid quantum–classical algorithms[5–9] have been proposed.

The quantum approximate optimization algorithm (QAOA)[10–14] is a class of hybrid quantum–classical algorithms widely used to solve a variety of combinatorial optimization problems (COPs).[15–20] In the context of constrained COPs (CCOPs), QAOA typically encodes problem constraints into the target Hamiltonian by introducing penalty terms[21–23] and subsequently searches for the quasi-optimal solution using a parameterized quantum circuit (PQC). However, this approach can result in infeasible solutions since QAOA explores the Hilbert space including both feasible and infeasible states, without explicit constraints on the solution space.[24]

To overcome these challenges, Hadfield *et al.* introduced the quantum alternating operator ansatz algorithm (QAOA+).[25] QAOA+ ensures that the quantum state evolves within the feasible solution space by encoding the problem constraints into the mixer Hamiltonian. Compared with QAOA, this approach reduces the search space and eliminates the risk of infeasible solutions. By leveraging these advantages, QAOA+ has been successfully applied to many CCOPs and offers significant advantages.[26–30] However, reducing the qubit requirements of QAOA+ is essential to tackle larger-scale CCOPs given the limited resources of current noisy intermediate-scale quantum devices, further expanding its applications.

The maximum independent set (MIS) problem is a significant CCOP encountered in various fields, including network design[31] and scheduling.[32] It is also equivalent to the minimum vertex cover or finding a maximum clique in the complementary graph.[33] To reduce the qubit requirements of QAOA+ for solving MIS, Tomesh *et al.* proposed a quantum local search (QLS) algorithm.[34] Unlike directly solving the MIS problem on the original graph using QAOA+ (DS-QAOA+),[35] QLS iteratively breaks the original problem into multiple sub-problems using specific strategies and respec-

†Corresponding author. E-mail: qsujuan@bupt.edu.cn
‡Corresponding author. E-mail: gaof@bupt.edu.cn

© 2025 Chinese Physical Society and IOP Publishing Ltd. All rights, including for text and data mining, AI training, and similar technologies, are reserved.

tively solves them using QAOA+. The local solutions are then combined into a global solution. While QLS reduces qubit usage by operating on smaller subgraphs, the merging complexity increases with the number of subgraphs.

In this paper, inspired by the proactively incremental learning algorithm[36] and the properties of the MIS problem, we propose a heuristic quantum algorithm termed the progressive quantum algorithm (PQA). The core idea of PQA is to iteratively construct a subgraph that is likely to contain the MIS of the original graph. The MIS problem is then solved on this subgraph to obtain an approximate solution. Specifically, PQA starts from a small and sparse subgraph and progressively enlarges it using problem-specific strategies. After each expansion, PQA applies QAOA+ to solve the MIS problem on the updated subgraph. The process of graph expansion and problem resolution is iterated by PQA until a predetermined termination criterion is satisfied. By doing so, PQA solves the MIS problem on a smaller subgraph than DS-QAOA+, reducing qubit overhead. Moreover, unlike QLS, PQA does not require merging multiple local solutions to recover the global one. Numerical results show that PQA achieves the same approximation ratio using significantly fewer qubits and shorter runtime than DS-QAOA+ and QLS, highlighting the efficiency of PQA.

This paper is structured as follows. Section 2 serves as the foundational segment, providing background knowledge on the MIS problem, QAOA, and its extension, QAOA+. Following this, in Section 3, we delineate the progressive quantum algorithm, emphasizing the construction strategies of the desired derived subgraph. Subsequently, in Section 4, we present a comprehensive set of numerical simulation experiments to showcase the quantum resource efficiency of PQA over DS-QAOA+ and QLS. Finally, in Section 5, we offer a succinct conclusion and discussion to encapsulate the key findings of our study and the next research direction.

2. Preliminaries

To help readers understand our work better, we first introduce the MIS problem in Subsection 2.1, followed by an introduction to QAOA in Subsection 2.2. Finally, we review the QAOA+ algorithm in Subsection 2.3.

2.1. Review of MIS

The MIS problem is defined on an undirected graph $G = (V,E)$, where $V = \{0,1,\ldots,n-1\}$ is the set of vertices and $E \subseteq \{\{u,v\} \mid u,v \in V, u \neq v\}$ is the set of edges. A subset of vertices $V_s \subseteq V$ is called an independent set if no two vertices in V_s are adjacent, i.e., there is no edge between any pair of vertices in V_s. The MIS is the independent set with the largest cardinality among all such sets. The size of an MIS is called the independence number, denoted by $\beta(G)$. While the MIS may not be unique, the independence number is uniquely defined for a given graph, as shown in Fig. 1. For any graph G, there exists an induced subgraph $G_q = (V_q, E_q)$, with $V_q \subseteq V$ and $E_q \subseteq E$, such that the MIS of G_q is identical to that of G.[37]

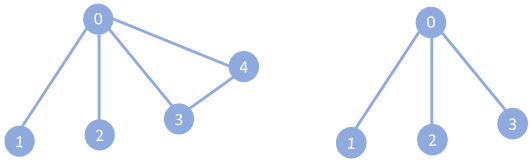


Fig. 1. Examples of MIS. (a) A target graph G whose feasible solutions are $\varnothing$, $\{0\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{4\}$, $\{1,2\}$, $\{1,3\}$, $\{1,4\}$, $\{2,3\}$, $\{2,4\}$, $\{1,2,3\}$ and $\{1,2,4\}$. The solutions of MIS are $\{1,2,3\}$ and $\{1,2,4\}$, where the MIS solutions of G are not unique, but the independence number identically equals 3. (b) An induced subgraph of G and its MIS solution is $\{1,2,3\}$. The feasible solutions of G_q are $\varnothing$, $\{0\}$, $\{1\}$, $\{2\}$, $\{3\}$, $\{1,2\}$, $\{1,3\}$, $\{2,3\}$, $\{1,2,3\}$. We respectively denote the feasible spaces including all feasible solutions of G_q and G as $S(G_q)$ and $S(G)$, where $S(G_q) \subseteq S(G)$.

In this paper, we denote the number of vertices and edges in G as n and m, respectively. To formalize the MIS problem, a binary variable x_u for each vertex u in V is introduced, where $x_u = 1$ represents that the vertex u is in the vertex subset, otherwise, $x_u = 0$. Each way of vertex partition corresponds to a unique bit string x, and there are 2^n forms for x in total, where $x = x_0 x_1 \cdots x_{n-1}$. The goal of the MIS problem is to maximize the following objective function:

$$C_1(x) = \sum_{u=0}^{n-1} x_u \tag{1}$$

when the independence constraint $\sum_{(u,v)\in E} x_u x_v = 0$ is met. This constraint can be incorporated into the objective function using the Lagrange multiplier method,[29] thus converting the constrained problem to an unconstrained problem. The obtained classical cost function is

$$\max C(x) = \sum_{u=0}^{n-1} x_u - \lambda \sum_{(u,v)\in E} x_u x_v, \tag{2}$$

where the value of λ is determined through additional hyperparameter optimization steps,[22] and $\lambda > 1$ can impose penalties on those terms that violate independence constraints,[23] making the maximal cost function value correspond to the exact MIS solution instead of other feasible or infeasible bit strings. Here, $C_{\max}$ equals the independence number $\beta(G)$.

2.2. Review of QAOA

QAOA encodes the solution of the problem into the ground state of the target Hamiltonian H_C and aims to approximate this ground state through a parameterized quantum circuit.[10] This quantum evolution starts from the ground state $|s\rangle$ of the initial Hamiltonian H_B, with a quantum circuit built using p-level QAOA ansatz, where p is the level depth, and the initial state $|s\rangle$ is required to be prepared by a constant-depth quantum circuit.

In QAOA, one level QAOA ansatz consists of two unitaries $e^{-i\gamma_i H_C}$ and $e^{-i\beta_i H_B}$, where γ_i and β_i are the parameters to be optimized in i-th level ansatz. Generally, the initial Hamiltonian is chosen as $H_B = -\sum_{u=0}^{n-1}\sigma_u^x$ whose ground state $|s\rangle = |+\rangle^{\otimes n}$ is easily prepared. Here, σ_u^x refers to the Pauli-X operator applied to the u-th qubit. The cost function $-C(x)$ is transformed into a diagonal target Hamiltonian

$$H_C = \sum_{u=0}^{n-1}\frac{\sigma_u^z - I}{2} + \lambda\sum_{(u,v)\in E}\frac{I-\sigma_u^z-\sigma_v^z+\sigma_u^z\sigma_v^z}{4} \tag{3}$$

via mapping each binary variable x_u to a quantum spin $(I-\sigma_u^z)/2$,[10] ensuring that the MIS solution corresponds to its ground state, where σ_u^z refers to applying Pauli-Z to the u-th qubit.

After alternately applying these unitaries to the initial quantum state $|s\rangle$ for p levels, the output state of the parameterized quantum circuit is

$$|\gamma_p,\beta_p\rangle = \prod_{i=1}^{p} e^{-i\beta_i H_B} e^{-i\gamma_i H_C}|s\rangle, \tag{4}$$

where $\gamma_p = (\gamma_1,\ldots,\gamma_p)$ and $\beta_p = (\beta_1,\ldots,\beta_p)$ are $2p$ variational QAOA parameters. The expectation function value of H_C in this variational quantum state is defined as

$$F(\gamma_p,\beta_p) = -\langle\gamma_p,\beta_p|H_C|\gamma_p,\beta_p\rangle, \tag{5}$$

which can be calculated by repeated measurements of the output quantum state using the computational basis. To search for the global optimal parameters

$$(\gamma_p^{\text{opt}},\beta_p^{\text{opt}}) = \arg\max_{(\gamma_p,\beta_p)} F(\gamma_p,\beta_p), \tag{6}$$

multiple optimization runs tend to be required. In each run, the search is typically done by starting with some guesses of the initial parameters. Then, the quantum computer measures the output state to compute the expected value of the output state and delivers the relevant information to the classical computer. These parameters are optimized by the classical optimizer and then delivered to the quantum computer. The process is repeated until it meets the termination condition, such as the maximal number of iterations or convergence tolerance ε.[20] The detailed procedures of QAOA in solving the MIS problem are given in Fig. 2.

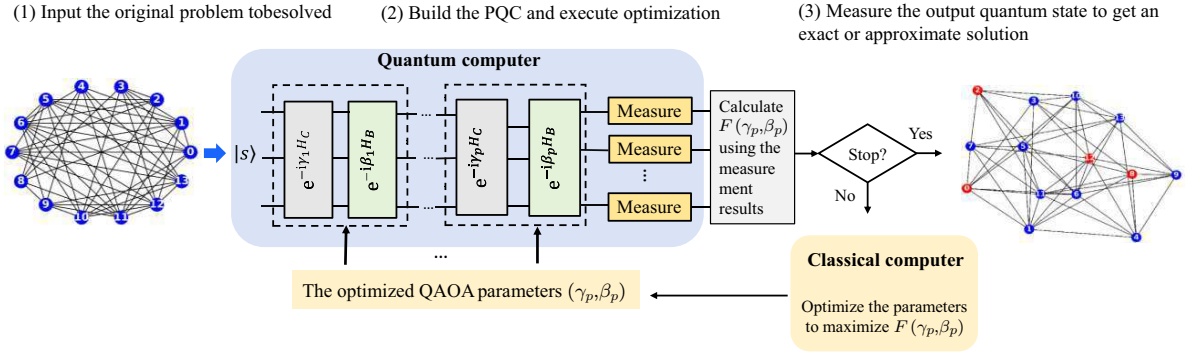


Fig. 2. A schematic of solving the MIS problem on the target graph using QAOA.

To evaluate the quality of the QAOA solution, the approximation ratio (AR) is defined as

$$r = \frac{F(\gamma_p,\beta_p)}{F_{\max}}, \tag{7}$$

where $F_{\max}$ is the negative of the ground state energy of H_C, equal to $C_{\max}$. The AR quantifies how close the output is to the exact solution, with $r\in[0,1]$ and $r=1$ indicating an exact solution.

2.3. Review of QAOA+

Unlike standard QAOA, QAOA+ incorporates problem constraints into the mixer Hamiltonian H_B, rather than penalizing constraint violations in the cost Hamiltonian. This design enables the quantum evolution to remain strictly within the feasible subspace, provided that the initial state $|s\rangle$ is a feasible solution. As a result, the constraint satisfaction is enforced throughout the entire evolution, and the optimization problem is effectively reduced to maximizing the original objective function over the feasible subspace. The target Hamiltonian is therefore defined as

$$H_C = \sum_{u=0}^{n-1}\frac{\sigma_u^z - I}{2}, \tag{8}$$

which is derived by mapping each binary variable x_u in the classical objective function $C_1(x) = \sum_{u=0}^{n-1}x_u$ to the quantum operator $(I-\sigma_u^z)/2$. As in QAOA, QAOA+ applies alternating unitaries generated by H_C and H_B to the initial quantum state $|s\rangle$. However, unlike QAOA, $|s\rangle$ is no longer required to be the ground state of H_B, and is instead chosen as a trivially preparable feasible state.

The unitary operator $e^{-i\beta_i H_B}$, derived from the mixer Hamiltonian H_B, is designed to induce transitions between feasible solutions while strictly preserving constraint satisfaction throughout the evolution.[25] For an independent vertex subset V_s, removing any vertex $v_a\in V_s$ (i.e., setting $x_{v_a}=0$) always

results in another independent set. In contrast, adding a vertex $v_a \notin V_s$ (i.e., flipping x_{v_a} from 0 to 1) preserves independence only if none of its neighbors are already in V_s. Therefore, to ensure feasibility, bit-flip operations must be conditioned on the states of neighboring vertices. Formally, the bit-flip of vertex v_a is allowed only when

$$f(v_a) = \sum_{j=1}^{d(v_a)} x_{v_{j,a}} = 0, \tag{9}$$

where $d(v_a)$ is the degree of v_a, and $v_{j,a}$ denotes its j-th neighbor.

Mapping to a quantum circuit, for any vertex u, a Pauli-X operation is applied on the u-th qubit (i.e., this qubit is flipped) if and only if all quantum states of adjacent vertices of u are $|0\rangle$ (i.e., $f(u) = 0$). Otherwise, a Pauli-I operation is applied on the u-th qubit. For the node u, its corresponding partial mixer Hamiltonian B_u can be described as

$$\begin{aligned} B_u = & \sum_{f(u)=0} |x_{v_{1,u}} \cdots x_{v_{d(u),u}}\rangle\langle x_{v_{1,u}} \cdots x_{v_{d(u),u}}| \otimes \sigma_u^x \\ & + \sum_{f(u)>0} |x_{v_{1,u}} \cdots x_{v_{d(u),u}}\rangle\langle x_{v_{1,u}} \cdots x_{v_{d(u),u}}| \otimes I_u. \end{aligned} \tag{10}$$

The complete mixer Hamiltonian is $H_{\mathrm{B}} = \sum_{u=0}^{n-1} B_u$.[25] Specifically, the quantum state resulting from applying $\mathrm{e}^{-\mathrm{i}\beta_i B_u}$ to the single feasible quantum state $|x_{\mathrm{o}}\rangle$ is

$$\mathrm{e}^{-\mathrm{i}\beta_i B_u}|x_{\mathrm{o}}\rangle = \begin{cases} R_x(2\beta_i)|x_{\mathrm{o}}\rangle, & f(u) = 0, \\ |x_{\mathrm{o}}\rangle, & f(u) > 0, \end{cases} \tag{11}$$

where the detailed result of $R_x(2\beta_i)|x_{\mathrm{o}}\rangle$ is $\cos 2\beta_i|x_{\mathrm{o}}\rangle - \mathrm{i}\sin 2\beta_i|x_0 \cdots \bar{x_u} \cdots x_{n-1}\rangle$, and $\bar{x_u}$ denotes that the state of the u-th qubit is converted from $|0\rangle$ to $|1\rangle$ (or ($|1\rangle$ to $|0\rangle$)), implementing the addition or removal of node u without generating infeasible quantum states. The circuit implementation of $\mathrm{e}^{-\mathrm{i}\beta_i B_u}$ is shown in Fig. 3 when the node has $d(u) \geq 1$ adjacent vertices. Additionally, Table 1 is provided to illustrate clearly the differences between using QAOA and QAOA+ to solve the MIS problem.

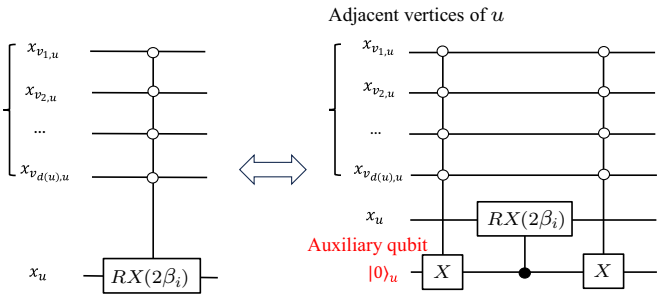


Fig. 3. The implementation of $\mathrm{e}^{-\mathrm{i}\beta_i B_u}$ when $d(u) \geq 1$. The multi-qubit controlled R_x gate can be decomposed into two multi-qubit controlled gates and a single-qubit controlled R_x gate, utilizing an auxiliary qubit.[38] The auxiliary qubit is initialized as $|0\rangle$ and is flipped to $|1\rangle$ only when the quantum states of all adjacent vertices are $|0\rangle$ (i.e., $f(u) = 0$). Once this condition is met, an R_x gate is applied on the u-th qubit. After the operation, the state of the auxiliary qubit can be reset from $|1\rangle$ to $|0\rangle$ using the right multi-qubit controlled gate operation. Otherwise, the auxiliary qubit is always $|0\rangle$ when $f(u) > 0$, and there is an identity operation applied to the u-th qubit.

Table 1. The differences between QAOA and QAOA+ when solving the MIS problem.

Manipulation of constraint	Add it into H_{C} by Lagrange multiplier	Encode it into H_{B}
$\|s\rangle$	$\|+\rangle^{\otimes n}$	feasible states
Mixer Hamiltonian H_{B}	$H_{\mathrm{B}} = -\sum_{u=0}^{n-1} \sigma_u^x$	$H_{\mathrm{B}} = \sum_{u=0}^{n-1} B_u$
Target Hamiltonian H_{C}	$H_{\mathrm{C}} = \sum_{u=0}^{n-1} \frac{\sigma_u^z - I}{2} + \lambda \sum_{(u,v)\in E} \frac{I - \sigma_u^z - \sigma_v^z + \sigma_u^z \sigma_v^z}{4}$	$H_{\mathrm{C}} = \sum_{u=0}^{n-1} \frac{\sigma_u^z - I}{2}$
$\|s\rangle$ is the ground state of H_{B}?	yes	no requirement
Hilbert space	including feasible and infeasible states	feasible states

3. Progressive quantum algotirhm

Any arbitrary graph has at least one induced subgraph that preserves the same MIS solution as the original graph.[37] Inspired by this property, PQA attempts to construct a subgraph likely to contain the MIS solution of G and solve it using QAOA+ to approximate the global solution. Specifically, PQA starts from a small initial subgraph and iteratively expands it using heuristic problem-tailored strategies. After each expansion, PQA solves the MIS problem on the current subgraph using QAOA+. This process repeats until a predefined stopping condition is met. In PQA, the expansion aims to increase the likelihood that the current subgraph contains the MIS of the original graph. Meanwhile, solving each subgraph using QAOA+ provides an estimate based on the expectation value, helping assess whether the MIS has likely been included. The following subsections describe the expansion strategies and stopping conditions in detail.

3.1. The subgraph construction strategies

Notably, the original graph may yield multiple derived subgraphs that vary in size but share the same MIS solution. In solving the MIS problem, QAOA+ implements the mixer unitary $\mathrm{e}^{-\mathrm{i}\beta_i H_{\mathrm{B}}}$ with multi-qubit controlled R_x gates, where the target qubit represents the flipped vertex and the control qubits correspond to its neighbors.[25,34] The implementation overhead of multi-qubit controlled R_x gates increases polynomially with the number of control qubits.[38,39] To reduce qubit usage and gate overhead, an ideal subgraph should preserve the MIS of the original graph while remaining small and sparse. A smaller subgraph reduces the qubit requirement, and sparsity lowers the cost of multi-qubit controlled gate implementations.

To construct such an ideal subgraph, PQA starts from a

sparse and small-sized initial subgraph G_0 and gradually expands it. In each expansion step, the node with the lowest closeness is selected to preserve subgraph sparsity. The closeness is defined as

$$C(v_c) = \sum_{v_c \in V_c, v_q \in V_q} E(v_c, v_q), \tag{12}$$

where $V_c = V - V_q$ is the vertex candidate set, and V_q is the vertex set of the current induced subgraph G_q. If the nodes v_c and v_q are adjacent (i.e., $(v_c, v_q) \in E$), $E(v_c, v_q) = 1$. Otherwise, $E(v_c, v_q) = 0$. A small value of $C(v_c)$ indicates a minimal change in sparsity. By starting with a small subgraph and gradually expanding its size, PQA can effectively control the growth of qubits and minimize unnecessary resource usage. The construction of G_0 involves the selection of the initial node and graph expansion. The detailed strategies are as follows. Additionally, an example is provided in Appendix A to help readers understand the process of graph expansion.

Initial node selection To ensure the initial subgraph G_0 remains sparse, the node with the minimum degree in the target graph G is chosen as the first node. If multiple nodes share the minimum degree, one is randomly selected.

Subgraph expansion To preserve sparsity during expansion, the candidate node v_c with the smallest $C(v_c)$ is selected. If multiple nodes have the same minimum $C(v_c)$, a secondary evaluation is performed. Each candidate is hypothetically added to the current subgraph, and its updated $C(v_c)$ is computed. The node leading to the lowest updated value is selected, and ties are resolved by random choice. During expansion, new edges may be introduced. Specifically, if the added node v_{new} is adjacent to any node v_q in the current subgraph, the edge (v_q, v_{new}) is included.

3.2. Progressive quantum optimization

After constructing the initial subgraph G_0, PQA applies p-level QAOA+ ansatz with randomly initialized parameters to solve the MIS problem on G_0. After optimization, PQA expands the graph according to problem-tailored strategies and applies a new p-level QAOA+ ansatz to the updated subgraph. At this stage, the optimized parameters from the previous subgraph are reused and further optimized, referred to as "parameter transfer".[40–43] After each optimization, PQA obtains an expectation function value $F_{G_i,p}$ about the subgraph G_i, and PQA terminates if the values satisfy $\Delta_1 \le \xi$ and $\Delta_2 \le \xi$ or if the size matches that of the target graph in the worst case, where

$$\Delta_1 = |F_{G_{i+1},p} - F_{G_i,p}|, \quad \Delta_2 = |F_{G_i,p} - F_{G_{i-1},p}|. \tag{13}$$

Here, ξ is the tolerance for function values, and $F_{G_i,p}$ ($F_{G_{i-1},p}$) represents the function value obtained using p-level QAOA+ ansatz to solve the MIS problem on the current (previous) subgraph G_i (G_{i-1}). We explain the rationale for using $\Delta_1 \le \xi$ and $\Delta_2 \le \xi$ as stopping conditions. According to the properties of the MIS problem, expanding the subgraph does not reduce its independence number, i.e., $\beta(G_i) \le \beta(G_{i+1})$, and thus $F_{\max,G_i} \le F_{\max,G_{i+1}}$. If the initial subgraph already includes

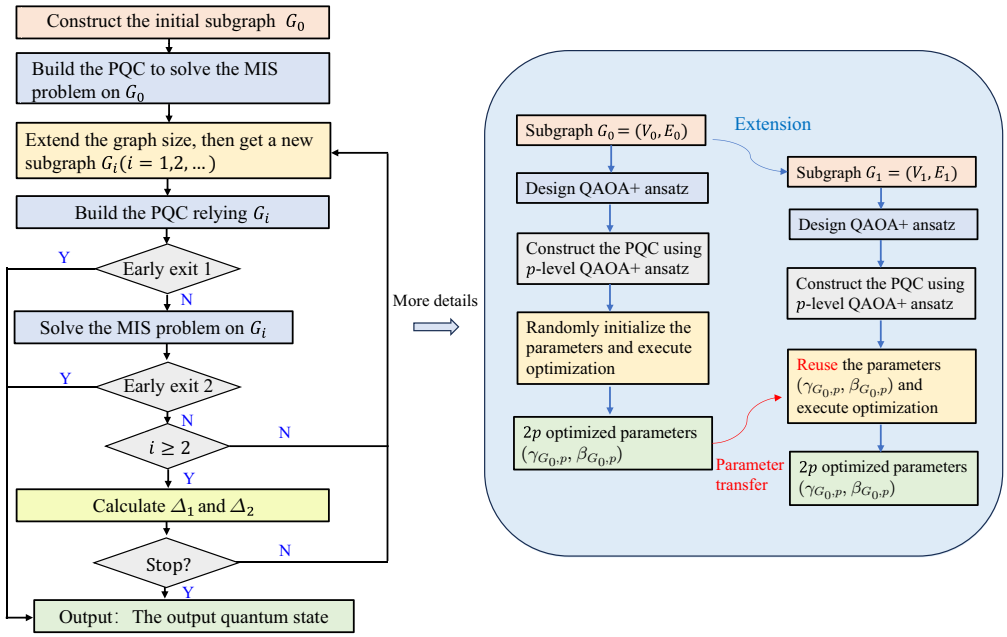


Fig. 4. A procedure schematic for PQA. PQA starts with an initial subgraph and solves the MIS problem using p-level QAOA+ ansatz, where the parameters of parameterized quantum circuit are randomly initialized. Then, PQA progressively enlarges the graph and solves the MIS problem on the updated subgraph, retraining the parameterized quantum circuit with pre-trained parameters. Notably, PQA incorporates two early exit mechanisms to terminate ineffective optimization, conserving quantum resources. The first mechanism assesses whether the current optimization process should proceed. The second mechanism evaluates whether further graph expansion is necessary. If the first mechanism is triggered, the parameterized quantum circuit outputs the quantum state obtained from the previous subgraph G_{i-1}. The output state that corresponds to the maximal $F_{G_j,p}$ ($j = 0, 1, \ldots, i$) is given when the second mechanism is triggered. Otherwise, the quantum state is obtained from the current subgraph G_i when the stop condition is met.

the MIS solution, $F_{\max,G_i}$ tends to stabilize during expansion. Otherwise, a temporary plateau may occur due to sub-optimal node selection, where further expansion brings no immediate improvement. Although continued growth may enhance the result, it also increases quantum resource consumption. Hence, blindly enlarging the subgraph is inefficient. To balance performance and resource usage, PQA monitors the expectation values and stops if no notable improvement is observed across three consecutive optimization rounds.

Two early exit mechanisms During actual optimization, the expectation function value may fluctuate as the subgraph grows. This can prevent PQA from meeting the stopping condition, causing it to run unnecessarily even if the ideal subgraph has already been found. To avoid unnecessary overhead, we introduce two early exit mechanisms based on the quality of initial parameters and optimization outcomes. After each graph expansion, PQA first decides whether to proceed with the optimization round by assessing the quality of the reused (initial) parameters. PQA often converges to quasi-optimal solutions within a few iterations when initialized with high-quality parameters. Otherwise, it may require many iterations to converge or may become trapped in low-quality local minima. Therefore, PQA stops the current optimization run if the expected function value using the initial parameters is significantly lower than that of the previous subgraph. If the initial parameters are acceptable, PQA proceeds with the optimization and applies a second early exit check afterward. If the resulting function value is substantially lower than the previous maxima, the optimization is trapped in a poor local solution. In this case, PQA terminates and returns the best function value and corresponding output state obtained in the current run. Otherwise, PQA continues to execute. The full procedure of PQA is illustrated in Fig. 4 for clarity.

4. Numerical simulations

We evaluate algorithm performance on three types of graphs, including Erdös–Rényi (ER) graphs with edge probabilities 0.4 and 0.5, and 3-regular graphs. Each type includes 20 randomly generated instances with $n = 14$ vertices. For PQA, the function value tolerance is set to $\xi = 0.1$, and the initial state is $|0\rangle$ for each optimization round. The first early stop mechanism is triggered if the initial expectation value is less than half of the expectation value obtained in the previous optimization round. The second mechanism is activated if the expectation value of the current optimization round is smaller than that of the previous optimization round by at least 1. These criteria identify unpromising optimization rounds and terminate them early to reduce computation time. For DS-QAOA+ and QLS, they also start from $|0\rangle$ in each run. All algorithms terminate the outer optimization when the convergence tolerance satisfies $\varepsilon \leq 0.001$.[20] To ensure an accurate calculation of the approximation ratio, we obtain the value of $C_{\max}$, which equals the independence number $\beta(G)$, through the exhaustive search for each graph instance used in our numerical simulations.

4.1. The AR obtained by various algorithms with the same number of optimization runs and level depths

To compare the performance of various quantum algorithms, we present the optimal AR (OAR) and average AR (AAR) achieved by each algorithm on different graph types with 14 vertices, under the same number of runs and level depths. Specifically, the level depth ranges from one to five, and each algorithm is executed with 100 optimization runs at each level depth. Each graph type yields 20 independent OAR and AAR values, and the mean OAR (AAR) is computed across these 20 instances. In addition, to establish a baseline for comparison, we include the AR obtained by the hill-climbing algorithm,[44] a classical local search algorithm with wide applications, and a classical counterpart to PQA. The classical PQA variant follows the same procedures as PQA, but QAOA+ is replaced with hill climbing to solve each subproblem. Besides, when solving each subproblem using the hill-climbing algorithm, the initial solution is inherited from the approximate solution obtained from the previous subgraph, except for the first subgraph, where an initial feasible solution is randomly generated.

The OAR is defined as the maximum approximation ratio obtained over multiple optimization runs, providing a measure of the highest-quality solution. Meanwhile, the AAR represents their average, reflecting solution stability. As shown in Fig. 5, OAR increases with the level depth, indicating that the obtained solutions are closer to the global optimal solution. This trend arises from deeper ansatz enhancing the expressive power of the quantum circuit, allowing it to represent a broader range of quantum states and search for high-quality solutions. Notably, the results also show that PQA achieves a higher OAR and AAR compared with other quantum algorithms under identical settings. This suggests that PQA not only excels in finding high-quality solutions but also maintains a more stable performance. Such stability is crucial for practical applications as it reduces the risk of performance fluctuations, making PQA a more robust choice. The specific improvements are as follows.

On ER graphs with an edge probability of 0.4, the OAR achieved by PQA is approximately 8.240% and 13.668% higher than those achieved by QLS and DS-QAOA+, respectively. The AAR achieved by PQA is approximately 3.766% and 4.416% higher than those achieved by QLS and DS-QAOA+, respectively.

On ER graphs with an edge probability of 0.5, the OAR achieved by PQA is approximately 8.128% and 14.422%

higher than those achieved by QLS and DS-QAOA+, respectively. The AAR achieved by PQA is approximately 6.048% and 6.658% higher than those achieved by QLS and DS-QAOA+, respectively.

On 3-regular graphs, the OAR obtained by PQA is approximately 8.486% and 9.438% higher than those achieved by QLS and DS-QAOA+, respectively. The AAR obtained by PQA is approximately 2.588% and 1.618% higher than those achieved by QLS and DS-QAOA+, respectively.

Compared with directly applying the hill-climbing algorithm to solve the original problem, the classical counterpart to PQA significantly improves the solution stability by progressively refining the obtained solution on each subgraph. This demonstrates the effectiveness of our method in incrementally addressing the problem. The results also show that the AR obtained by PQA can be higher than the classical counterpart to PQA, benefiting from quantum superposition and quantum parallelism. Specifically, the classical local search algorithm requires multiple verification and correction steps to ensure solution feasibility.[34] Moreover, its feasible solution space is limited by the sequential, local nature of its search, as it updates only one solution at a time. This limitation is mitigated in PQA, as it constructs a larger feasible solution space through quantum superposition and elaborate circuit designs. By leveraging quantum parallelism, QAOA+ evaluates multiple feasible solutions in parallel, offering an advantage over classical heuristics for solving MIS on each subgraph. In Appendix B, we provide the AR obtained by various algorithms versus the graph size to show how their performance scales with increasing problem size.

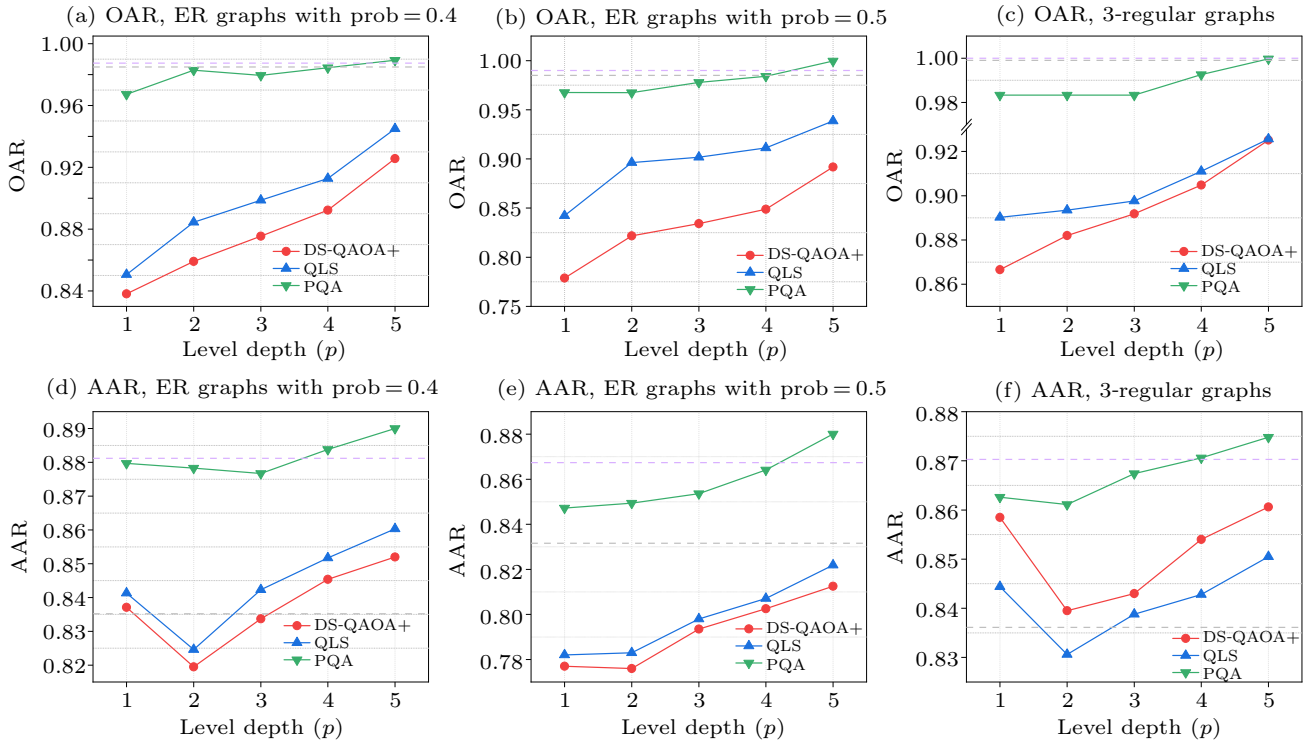


Fig. 5. The AR achieved by various algorithms. The gray dotted line corresponds to the AR obtained by the hill climbing algorithm, and the purple dotted line corresponds to the AR obtained by the classical counterpart of PQA. In each subplot, the curves depict the AR achieved by different algorithms across various graph types with increasing level depth p.

4.2. Quantum resource consumption of different algorithms at the same AR

To evaluate algorithmic efficiency, we compare the total resources required to achieve a given AR. We select several key metrics, including the number of iterations, the number of qubits, the number of multi-qubit controlled R_x gates, and the algorithm runtime. In Appendix C, we give the methods of calculating the total resources for each algorithm.

Figure 6 shows the total resources consumed by different algorithms to reach a given AR across various graph types. As expected, the overall consumption increases with the target AR, since achieving higher AR requires more accurate solutions, which in turn demand more optimization runs or deeper circuits. However, DS-QAOA+ and PQA occasionally show counterintuitive trends. For instance, DS-QAOA+ consumes more qubits and multi-qubit controlled R_x gates at $r = 0.9$ than at $r = 0.95$. This occurs because it uses fewer layers but more runs at $r = 0.9$, while requiring deeper level depths but fewer runs at $r = 0.95$. Shallow circuits have limited expressive power and must compensate through more runs, whereas deeper circuits can explore a broader solution space and achieve better results in fewer runs. Nonetheless, increased level depth raises gate overhead, which may limit scalability on current noisy intermediate-scale quantum devices. In our calculation, total cost equals the number of runs times the average cost per run. Although deeper circuits use

more gates, this increase is relatively small compared with the reduction in run count. Since qubit usage per run remains constant, total resource consumption can be higher at $r = 0.9$. The above analysis also suits PQA.

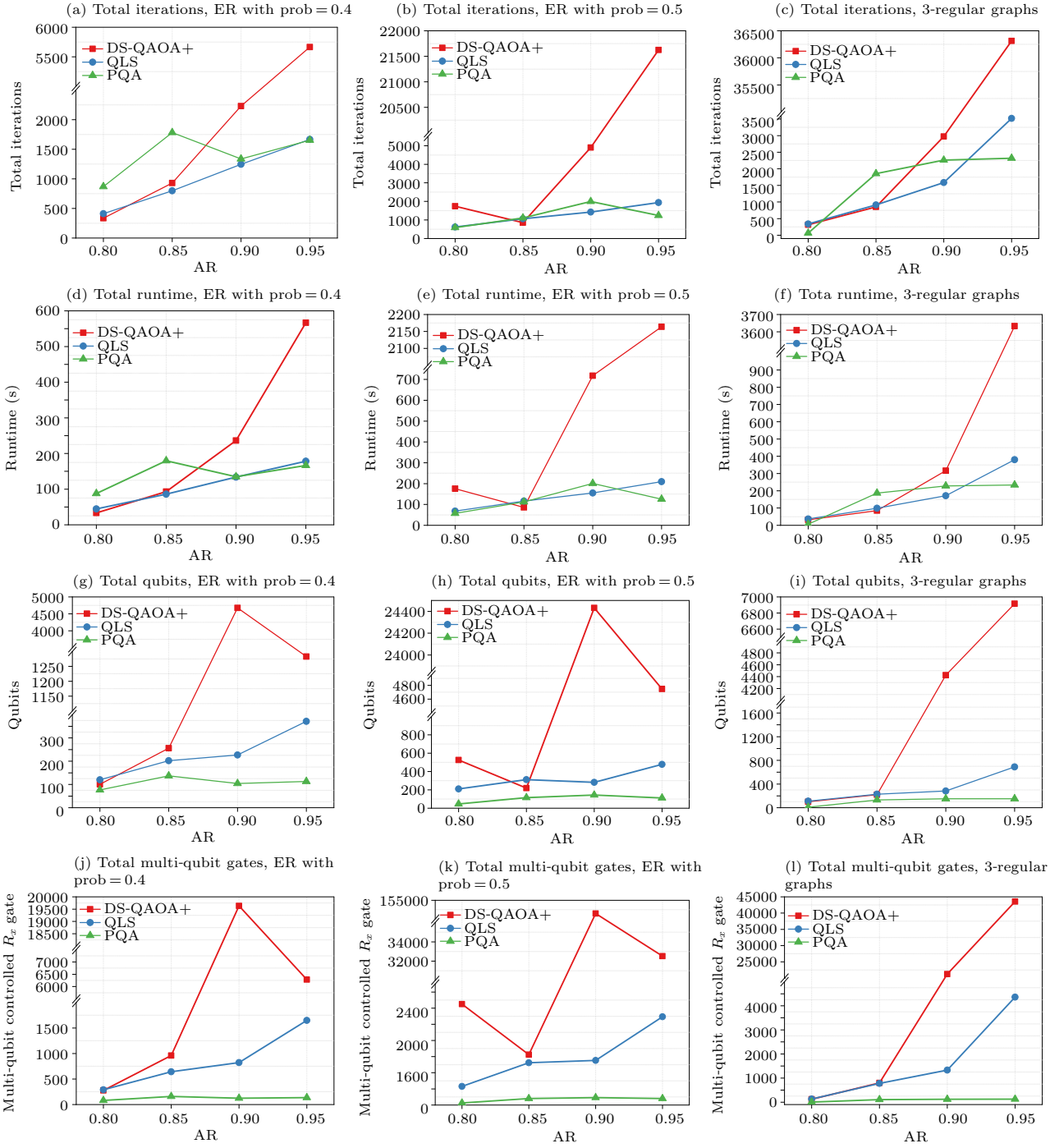


Fig. 6. The total resource consumption of various algorithms on different graph types.

From the numerical simulation results in Fig. 6, we also observe that for smaller target AR (e.g., 0.8 and 0.85), DS-QAOA+ and QLS typically achieve the given AR with fewer iterations and shorter runtime than PQA. As shown in Fig. C1, for AR values of 0.8 and 0.85, the total number of optimization runs consumed by DS-QAOA+, QLS, and PQA is similar. However, since PQA performs more optimization rounds per run, it results in a higher total iteration consumption. Moreover, runtime scales with the number of iterations,[45] and more iterations result in longer runtime. While the number of iterations is important, the number of qubits and gates required per run more directly determines the problem size that can be supported by resource-constrained quantum devices. Excessive qubit and gate demands per iteration may limit the ability of hardware to solve larger-scale problems efficiently. Therefore, sacrificing some iterations to reduce qubit and gate consumption is a justified trade-off, as it allows the algorithm to tackle larger problems within the constraints of current quan-

tum hardware. However, this does not imply that PQA always requires the most iterations for any AR. Numerical results show that as the target AR increases, the efficiency of PQA becomes more evident. This is mainly because PQA can achieve the desired AR with fewer runs than DS-QAOA+ and QLS, significantly reducing overall resource consumption. By minimizing the quantum resources required while achieving high performance, PQA may be increasingly promising. Specifically, for each graph type, the resource consumption of PQA compared with DS-QAOA+ and QLS for a given AR of 0.95 is as follows.

On the ER graphs with an edge probability of 0.4, the number of iterations consumed by PQA is approximately 29.17% and 99.16% of those consumed by DS-QAOA+ and QLS, respectively. The amount of runtime consumed by PQA is approximately 29.38% and 93.48% of those consumed by DS-QAOA+ and QLS, respectively. The number of qubits consumed by PQA is approximately 8.79% and 30.36% of those consumed by DS-QAOA+ and QLS, respectively. The number of multi-qubit controlled gates consumed is 2.16% and 8.24% of those consumed by DS-QAOA+ and QLS, respectively.

On the ER graphs with an edge probability of 0.5, the number of iterations consumed by PQA is approximately 5.70% and 64.00% of those consumed by DS-QAOA+ and QLS, respectively. The amount of runtime consumed by PQA is approximately 5.80% and 59.95% of those consumed by DS-QAOA+ and QLS, respectively. The number of qubits consumed by PQA is approximately 2.34% and 23.18% of those consumed by DS-QAOA+ and QLS, respectively. The number of multi-qubit controlled gates consumed is 0.49% and 7.36% of those consumed by DS-QAOA+ and QLS, respectively.

On the 3-regular graphs, the number of iterations consumed by PQA is approximately 6.39% and 65.92% of those consumed by DS-QAOA+ and QLS, respectively. The amount of runtime consumed by PQA is approximately 6.43% and 61.49% of those consumed by DS-QAOA+ and QLS, respectively. The number of qubits consumed by PQA is approximately 2.17% and 21.76% of those consumed by DS-QAOA+ and QLS, respectively. The number of multi-qubit controlled gates consumed is 0.29% and 2.88% of those consumed by DS-QAOA+ and QLS, respectively.

4.3. The AR obtained by PQA versus the number of available qubits

Actual noisy intermediate-scale quantum devices are typically equipped with only a limited number of usable qubits. Therefore, to further evaluate the practicality of PQA under resource-constrained conditions, we investigate how its solution quality varies with the number of available qubits. In the experiments, we set the level depth to 5 and ran PQA for 100 optimization runs on each graph instance. In each run, PQA terminates either by triggering an early-exit mechanism, reaching convergence of the objective function (i.e., $\Delta_1 \leq \xi$ and $\Delta_2 \leq \xi$), or reaching the allowed qubit budget. Figure 7 illustrates how the optimal approximation ratio (r_{opt}) and the average approximation ratio (r_{avg}) obtained by PQA vary with the number of available qubits. To intuitively depict the deviation between the obtained and exact solutions, we use $1-r$ as the vertical axis. Such a representation makes the convergence behavior more visually interpretable, especially when the approximation ratio approaches 1.

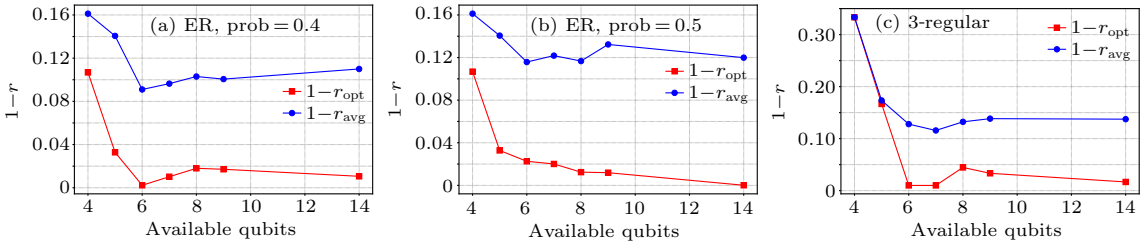


Fig. 7. The approximation ratio obtained by PQA versus the number of available qubits, evaluated on ER graphs with edge probabilities of 0.4 and 0.5, and 3-regular graphs with 14 vertices.

The numerical results show that the solution quality obtained by PQA improves significantly as the number of available qubits increases, as evidenced by a rapid decrease in the error $1-r$. When the qubit count exceeds a certain threshold, the performance improvement gradually levels off. This trend can be explained by the construction mechanism of PQA. Specifically, PQA progressively expands the current subgraph to increase the likelihood that it contains the MIS solution of the original graph. When the number of available qubits is limited, PQA is often forced to terminate due to resource constraints. In this case, the constructed subgraph is structurally incomplete and fails to capture the critical components of the MIS solution, leading to limited solution quality. As the available resources increase, PQA is able to construct larger subgraphs that better cover the core components of the MIS solution, resulting in a significant improvement in solution quality. Furthermore, once the available resources exceed a certain threshold, PQA typically terminates due to

the early-exit mechanisms or convergence criteria as defined in Equation (13), rather than resource exhaustion. At this point, further increases in qubit resources have little effect on overall performance. These results quantitatively reflect the trade-off between solution quality and available resources, and highlight that the performance gain of PQA gradually diminishes once the number of available qubits exceeds a certain threshold. Additionally, they demonstrate that PQA can maintain strong performance even under constrained qubit budgets, making it a practical and scalable choice for current noisy intermediate-scale quantum devices.

5. Conclusion and outlook

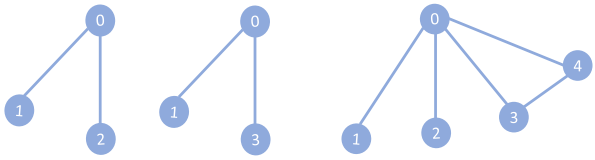
We propose a heuristic quantum algorithm termed PQA to reduce the number of qubits in the application of QAOA+ when solving the MIS problem. PQA harnesses the superposition and parallelism of the quantum computer to solve the MIS problem on each subgraph. It collaborates with the classical computer, which handles tasks such as obtaining target graph information and adjusting parameters. This design distributes the computational load and improves flexibility. Numerical results show that PQA consistently outperforms DS-QAOA+ and QLS in terms of both OAR and AAR under the same number of optimization runs and level depths. Moreover, PQA achieves the same solution quality with significantly fewer qubits and multi-qubit controlled R_x gates, making it more suitable for noisy intermediate-scale quantum devices with limited connectivity and high gate noise.

To maintain fine-grained control over subgraph growth, PQA starts with a small subgraph and incrementally expands it by adding one node at a time during each expansion step. However, this incremental approach may lead to more optimization rounds, potentially increasing the overall computational overheads. To address this limitation, future research should explore more effective graph construction strategies, taking into account the graph structure to improve PQA performance. One possible approach is to introduce multiple vertices in each expansion step. However, further research is required to evaluate its impact on the effectiveness and stability of PQA. Another promising direction is to incorporate machine learning techniques to optimize the selection of initial graph size and dynamically refine the graph expansion strategy based on feedback obtained during the optimization process. Finally, while this paper focuses on MIS, the PQA framework can be extended to other CCOPs, such as maximum weighted independent set, facility location, and knapsack problems. Designing problem-specific expansion heuristics and evaluating their performance in various instances is a promising direction.

Appendix A: An example of the subgraph expansion

Denote the current subgraph as $G_i = (V_i, E_i)$. For the first subgraph in Fig. A1, $V_i = \{0,1,2\}$ and $E_i = \{(0,1),(0,2)\}$. The candidate vertex set $V_c = V - V_i = \{3,4\}$. Our goal is to select the vertex from V_c that has the least connection with the current subgraph to be added to G_i. Specifically, if node 3 is added to G_i, $C(v_c = 3) = 1$. Similarly, if node 4 is added, $C(v_c = 4) = 1$. Since both nodes have the same connection, we proceed to the next round of selection. When node 3 is added to V_i, the edge $(0,3)$ is included in E_i, leaving $V_c = \{4\}$ and $C(v_c = 4) = 2$. Similarly, if node 4 were added, $C(v_c = 3) = 2$. After evaluating both nodes, since they equally satisfy the selection criteria, one can be randomly added to V_i. If node 3 (or 4) is added, the edge $(0,3)$ (or $(0,4)$) is included in E_i. The resulting subgraph becomes is $G_{i+1} = \{\{0,1,2,3\},\{(0,1),(0,2),(0,3)\}\}$ (or $G_{i+1} = \{\{0,1,2,4\},\{(0,1),(0,2),(0,4)\}\}$).

For the second subgraph in Fig. A1, $V_i = \{0,1,3\}$ and $E_i = \{(0,1),(0,3)\}$. The candidate vertex set $V_c = \{2,4\}$, with $C(v_c = 2) = 1$ and $C(v_c = 4) = 2$. We choose the node with the smallest $C(v_c)$ value from the candidate vertex set. Thus, node 2 is the next node to be added to V_i, and the edge $(0,2)$ is added to E_i. The newly constructed subgraph is $G_{i+1} = \{\{0,1,2,3\},\{(0,1),(0,2),(0,3)\}\}$.



(a) Subgraph (b) Target graph

Fig. A1. A target graph and its induced subgraphs.

Appendix B: The obtained AR by various algorithms versus the graph size

To evaluate algorithm performance as graph size increases, particularly to assess the scalability of the proposed subgraph expansion strategy in PQA, we conducted comparative experiments to measure the solution quality achieved by various algorithms. Specifically, we considered graphs of size 8, 10, 12, and 14. For each size, 20 ER graphs (with an edge probability of 0.5) and 20 3-regular graphs were randomly generated. All quantum algorithms used a fixed level depth of 5, and each was executed for 100 optimization runs per graph instance. For each algorithm, we recorded the optimal and average approximation ratios across the 100 runs for every graph instance. Then, for each algorithm, we computed the mean optimal and average approximation ratios under the given graph types.

Figure B1 presents how the optimal and average approximation ratios of each algorithm change with graph size on the given graph types. In most cases, we observe a downward trend in both the optimal and average approximation ratios as the graph size increases. This aligns with expectations since larger graphs typically exhibit greater structural complexity, enlarging the solution space and making the algorithms more prone to getting trapped in local optima. However, the optimal and average approximation ratios achieved by PQA are consistently higher than those of DS-QAOA+ and QLS at the same graph size, and the performance gap becomes more pronounced as the graph size increases, demonstrating that PQA not only achieves higher-quality solutions but also maintains more stable performance. Meanwhile, we also observe that although CLS and classical PQA exhibit similar optimal approximation ratios, the average approximation ratios of classical PQA are consistently higher. This demonstrates that our progressive subgraph expansion and solving strategy improves average performance across multiple runs, likely by reducing the risk of low-quality solutions and enhancing robustness. All in all, the trends exhibited by both PQA and classical PQA compared with DS-QAOA+ and CLS indicate that the subgraph construction mechanism remains effective even as the graph size increases, as it consistently identifies high-quality substructures that lead to superior solutions.

Table B1 reports the relative change in average approximation ratios of five algorithms across three consecutive graph size transitions ($8 \to 10$, $10 \to 12$, and $12 \to 14$), evaluated on the ER and 3-regular graphs discussed above. Each entry denotes the percentage change in average performance between two adjacent graph sizes for a given algorithm. Negative values indicate performance degradation, with larger magnitudes reflecting more substantial declines. As shown in the table, all algorithms exhibit some degree of performance degradation as the graph size increases. This reflects the increased optimization difficulty and structural complexity of larger graph instances for both quantum and classical methods. However, the rate and consistency of degradation vary considerably among the algorithms. Notably, PQA exhibits a more stable and gradually flattening degradation trend as graph size increases, as evidenced by its monotonically decreasing performance drops of -4.480%, -4.055%, and -0.650% across the three transitions. In contrast, CLS and QAOA+ experience larger or more irregular drops, particularly in the $10 \to 12$ transition. In addition, classical PQA exhibits a stable degradation trend closely aligned with PQA, with corresponding drops of -4.460%, -4.290%, and -2.360%. This consistency supports the effectiveness of the underlying subgraph construction and solving strategy. Moreover, the smaller performance drop of PQA in larger graphs (e.g., only -0.650% in the $12 \to 14$ transition) suggests that the quantum enhancement helps further stabilize performance degradation. In summary, PQA exhibits a slow and stable performance decline with increasing graph size, suggesting that its effectiveness is likely to extend to even larger problem instances. These results highlight its potential as a scalable and practical approach for solving large-scale combinatorial optimization problems on noisy intermediate-scale quantum devices.

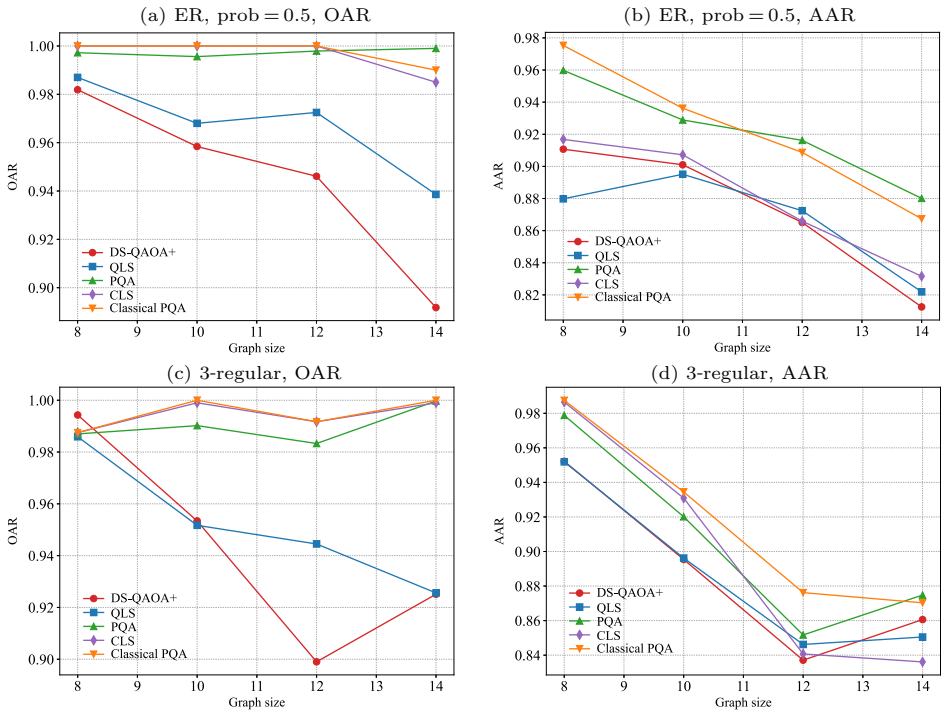


Fig. B1. The approximation ratios obtained by various algorithms versus graph size on ER graphs with an edge probability of 0.5 and 3-regular graphs.

Table B1. Impact of increasing graph size on the average approximation ratios of various algorithms.

Graph size changes	PQA	QLS	QAOA+	CLS	Classical PQA
$8 \to 10$	−4.480%	−2.020%	−3.325%	−3.260%	−4.460%
$10 \to 12$	−4.055%	−3.635%	−4.710%	−6.575%	−4.290%
$12 \to 14$	−0.650%	−2.310%	−1.455%	−1.945%	−2.360%

Appendix C: The methods of calculating total resources

When the algorithm reaches a given AR at a certain level depth, the total resource consumption is calculated as the product of the total number of runs required to achieve the given AR under this level depth and the average resource consumption per run. Therefore, the key to this calculation lies in determining the required level depths, the total number of runs, and the resource consumption per run. To determine at which level depth the algorithm can achieve the given AR, we vary the level depth from 1 to n and perform multiple optimization runs at each depth. Specifically, we conduct 100 optimization runs at each level depth p for $p \leq 6$, and 2^p optimization runs when $p > 6$. After completing all simulations, we first determine the shallowest level depths at which the algorithm achieves the given AR, based on the optimal approximate ratio obtained from multiple optimization runs at each depth. Next, we calculate the total number of runs and average resource consumption at this level depth. The total number of runs required to achieve the given AR is estimated using the following method. First, we calculate the number of runs Q where the resulting AR exceeds the given threshold across multiple optimization runs. Next, we calculate the ratio of Q to the total number of runs at level depth p, which represents the probability of exceeding the threshold. The reciprocal of this probability provides the total number of runs needed for the algorithm to achieve the given AR. We then calculate the total resource consumption for each algorithm to achieve the given AR on each graph. Next, we average the total number of runs, iterations, qubits, multi-qubit controlled R_x gates, and runtime for each graph type, and use these values to represent the resources required by each algorithm to achieve the given AR on that graph type.

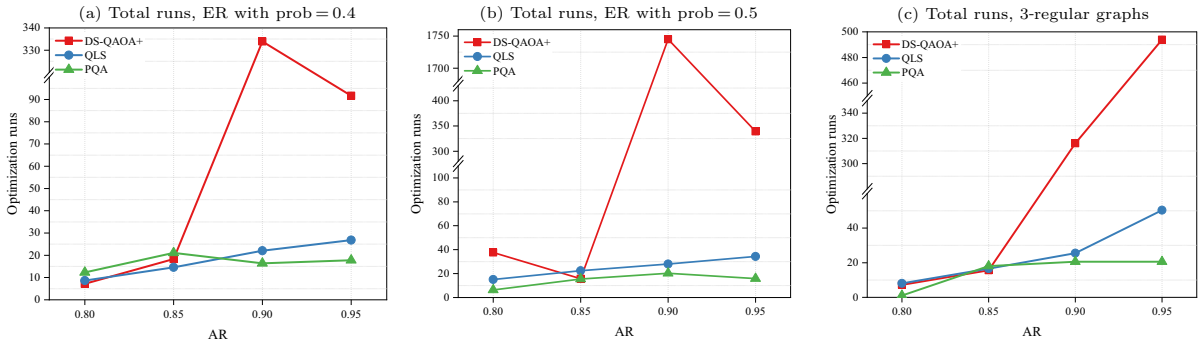


Fig. C1. The total optimization runs of various algorithms when they achieve the given AR on different graph types.

Runtime measures the overall execution efficiency, including both quantum and classical computation time. For QLS and PQA, merging the global solution and graph expansion both include edge checking, while DS-QAOA+ excludes them. Although a single edge-check is fast, we include classical time in all methods to ensure fair comparison.

The classical time T_c accounts for edge-check operations during graph construction or global solution update. If each check takes t_j and J checks are performed in a run, then

$$T_c = J \times t_j. \tag{C1}$$

The quantum running time T_q includes the time required for initial state preparation, quantum state measurements, and quantum gate evolution across multiple iterations.[45] In each iteration, M repeated measurements are needed to calculate the average expectation value of the objective function. The time consumed in a single repetition is

$$t_q = t_p + t_M + (\text{circuit depth}) \times t_G, \tag{C2}$$

where t_p is the initial state preparation time, t_M is the time for a single measurement, t_G is the time for a single quantum gate evolution, and circuit depth is the depth of the parameterized quantum circuit. The time for a single iteration is

$$T_{ITR} = t_q \times M, \tag{C3}$$

and the quantum time consumed in a single optimization run is

$$T_q = T_{ITR} \times \text{ITRs}, \tag{C4}$$

where ITRs are the number of iterations required for the objective function to converge in this run. Therefore, the time consumed in a single run for QLS and PQA is

$$T_{PQA} = T_{q,\,PQA} + T_{c,\,PQA}, \tag{C5}$$

$$T_{QLS} = T_{q,\,QLS} + T_{c,\,QLS}. \tag{C6}$$

This study explores the runtime required by different algorithms to achieve the same AR across various graph types. The calculations employ the same realistic assumptions as Ref. [45], namely, $t_p + t_M$ is one millisecond, and t_G is 10

nanoseconds. The tests are conducted on a server equipped with an Intel (R) Xeon (R) Sliver 4316 CPU running at 2.3 GHz and 251 GB of RAM, where t_j is determined to be 0.31226 millisecond.

Program availability

The related source code is openly available in Science Data Bank at https://doi.org/10.57760/sciencedb.25335 and at https://github.com/nixiaohui303/Progressive-Quantum-Algorithm, which are freely available to academic users.

Acknowledgements

We thank the Tianyan Quantum Computing Program. This work is supported by the National Natural Science Foundation of China (Grant Nos. 62371069, 62372048, and 62272056) and BUPT Excellent Ph.D. Students Foundation (Grant No. CX2023123).

References

[1] Grover L K 1997 *Phys. Rev. Lett.* **79** 325
[2] Shor P W 1999 *SIAM Rev.* **41** 303
[3] Harrow A W, Hassidim A and Lloyd S 2009 *Phys. Rev. Lett.* **103** 150502
[4] Preskill J 2018 *Quantum* **2** 79
[5] Grimsley H R, Economou S E, Barnes E and Mayhall N J 2019 *Nat. Commun.* **10** 3007
[6] Tang H L, Shkolnikov V O, Barron G S, Grimsley H R, Mayhall N J, Barnes E and Economou S E 2021 *PRX Quantum* **2** 020310
[7] Du Y X, Huang T, You S, Hsieh M H and Tao D C 2022 *npj Quantum Inf.* **8** 62
[8] Wu S Y, Song Y Q, Li R Z, Qin S J, Wen Q Y and Gao F 2025 *Adv. Quantum Technol.* 2400484
[9] Song Y Q, Wu Y S, Wu S Y, Li D D, Wen Q Y, Qin S J and Gao F 2024 *Sci. China Phys. Mech. Astron.* **67** 250311
[10] Farhi E, Goldstone J and Gutmann S 2014 arXiv:1411.4028 [quant-ph]
[11] Bravyi S, Kliesch A, Koenig R and Tang E 2020 *Phys. Rev. Lett.* **125** 260505
[12] Streif M and Leib M 2020 *Quantum Sci. Technol.* **5** 034008
[13] Zhu L H, 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
[14] Zhou Z Q, Du Y X, Tian X M and Tao D C 2023 *Phys. Rev. Appl.* **19** 024027
[15] Cook J, Eidenbenz S and Bärtschi A 2020 *Proceedings of the International Conference on Quantum Computing and Engineering*, October 12–16, 2020, Denver, CO, USA, pp. 83–92
[16] Vikstål P, Grönkvist M, Svensson M, Andersson M, Johansson G and Ferrini G 2020 *Phys. Rev. Appl.* **14** 034009
[17] Zhou L, Wang S T, Choi S, Pichler H and Lukin M D 2020 *Phys. Rev. X* **10** 021067
[18] Brandhofer S, Braun D, Dehn V, Hellstern G, Hüls M, Ji Y J, Polian I, Bhatia A S and Wellens T 2022 *Quantum Inf. Process.* **22** 25
[19] Tomesh T, Saleem Z H, Perlin M A, Gokhale P, Suchara M and Martonosi M 2023 *Proceedings of the International Conference on Quantum Computing and Engineering*, September 17–22, 2023, Bellevue, WA, USA, pp. 1–12
[20] Ni X H, Cai B B, Liu H L, Qin S J, Gao F and Wen Q Y 2024 *Adv. Quantum Technol.* **7** 2300419
[21] Zhang Y J, Mu X D, Liu X W, Wang X Y, Zhang X, Li K, Wu T Y, Zhao D and Dong C 2022 *Appl. Soft Comput.* **118** 108554
[22] Finžgar J R, Kerschbaumer A, Schuetz M J A, Mendl C B and Katzgraber H G 2024 *PRX Quantum* **5** 020327
[23] Lucas A 2014 *Front. Phys.* **2** 5
[24] Ruan Y, Yuan Z Q, Xue X L and Liu Z H 2023 *Inf. Sci.* **619** 98
[25] Hadfield S, Wang Z H, O Gorman B, Rieffel E G, Venturelli D and Biswas R 2019 *Algorithms* **12** 34
[26] Hadfield S, Wang Z H, Rieffel E G, O'Gorman B, Venturelli D and Biswas R 2017 *Proceedings of the Second International Workshop on Post Moores Era Supercomputing*, November 12–17, 2017, Denver, CO, USA pp. 15–21
[27] Wang Z H, Rubin N C, Dominy J M and Rieffel E G 2020 *Phys. Rev. A* **101** 012320
[28] He Z C, Shaydulin R, Chakrabarti S, Herman D, Li C H, Sun Y and Pistoia M 2023 *npj Quantum Inf.* **9** 121
[29] Brady L T and Hadfield S 2024 *Phys. Rev. A* **110** 052435
[30] Wang S S, Liu H L, Song Y Q, Gao F, Qin S J and Wen Q Y 2023 *Physica A* **626** 129089
[31] Moscibroda T and Wattenhofer R 2005 *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing*, July 17–20, 2005, Las Vegas, NV, USA, pp. 148–157
[32] Eddy D and Kochenderfer M J 2021 *J. Spacecr. Rockets* **58** 1416
[33] Hochbaum D S 1997 *Approximation Algorithms for NP-Hard Problems* (Boston: PWS Publishing) pp. 94–143
[34] Tomesh T, Saleem Z H and Suchara M 2022 *Quantum* **6** 781
[35] Saleem Z H 2020 *Int. J. Quantum Inf.* **18** 2050011
[36] Li L X, Li J, Song Y Q, Qin S J, Wen Q Y and Gao F 2024 *Sci. China Phys. Mech. Astron.* **68** 210313
[37] Bondy J A and Murty U S R 1976 *Graph Theory with Applications* (London: Macmillan)
[38] Vale R, Azevedo T M D, Araújo I C S, Araujo I F and da Silva A J 2024 *Trans. Comput.-Aided Des. Integr. Circuits Syst.* **43** 802
[39] Tomesh T, Allen N, Dilley D and Saleem Z H 2024 *Quantum* **8** 1493
[40] Lyngfelt I and García-Álvarez L 2025 *Phys. Rev. A* **111** 022418
[41] Shaydulin R, Lotshaw P C, Larson J, Ostrowski J and Humble T S 2023 *ACM Trans. Quantum Comput.* **4** 19
[42] Wurtz J and Lykov D 2021 *Phys. Rev. A* **104** 052419
[43] Brandao F G S L, Broughton M, Farhi E, Gutmann S and Neven H 2018 arXiv:1812.04170 [quant-ph]
[44] Chinnasamy S, Ramachandran M, Amudha M and Ramu K 2022 *Recent Trends Manag. Commer.* **3** 1
[45] Guerreschi G G and Matsuura A Y 2019 *Sci. Rep.* **9** 6903