frontiers | Frontiers in Quantum Science and Technology

TYPE Original Research
PUBLISHED 04 December 2025
DOI 10.3389/frqst.2025.1636042

OPEN ACCESS

EDITED BY
Jaewoo Joo,
University of Portsmouth, United Kingdom

REVIEWED BY
Saravana Prakash Thirumuruganandham,
SIT Health, Ecuador
Bikas K Chakrabarti,
Saha Institute of Nuclear Physics (SINP), India

*CORRESPONDENCE
Ruben Pariente Bassa,
scoomens@gmail.com

RECEIVED 27 May 2025
REVISED 05 August 2025
ACCEPTED 15 August 2025
PUBLISHED 04 December 2025

CITATION
Fuchs FG, Pariente Bassa R and Lien F (2025) Encodings of the weighted MAX *k*-CUT problem on qubit systems.
Front. Quantum Sci. Technol. 4:1636042.
doi: 10.3389/frqst.2025.1636042

Encodings of the weighted MAX *k*-CUT problem on qubit systems

Franz G. Fuchs[1], Ruben Pariente Bassa[1]* and Frida Lien[2]

[1]Department of Mathematics and Cybernetics, SINTEF AS, Oslo, Norway, [2]Department of Mathematics, University of Oslo, Oslo, Norway

The weighted MAX *k*-CUT problem involves partitioning a weighted undirected graph into k subsets, or colors, to maximize the sum of the weights of edges between vertices in different subsets. This problem has significant applications across multiple domains. This study explores encoding methods for MAX *k*-CUT on qubit systems by utilizing quantum approximate optimization algorithms (QAOA) and addressing the challenge of encoding integer values on quantum devices with binary variables. We examine various encoding schemes and evaluate the efficiency of these approaches. The study presents a systematic and resource-efficient method to implement the phase separation operator for the cost function of the MAX *k*-CUT problem. When encoding the problem into the full Hilbert space, we show the importance of encoding the colors in a balanced way. We also explore the option of encoding the problem into a suitable subspace by designing suitable state preparations and constrained mixers (LX- and Grover-mixer). Numerical simulations on weighted and unweighted graph instances demonstrate the effectiveness of these encoding schemes, particularly in optimizing circuit depth, approximation ratios, and computational efficiency.

KEYWORDS

quantum optimization algorithm, quantum circuit compilation, quantum approximate optimization algorithm, quantum technologies, variational quantum circuits

1 Introduction and related work

The quantum approximate optimization algorithm (Farhi et al., 2014)/quantum alternating operator ansatz (Hadfield et al., 2019) (QAOA) is a widely studied hybrid algorithm for approximately solving combinatorial optimization problems encoded into the ground state of a Hamiltonian. Given an objective function $f\colon \{0,1\}^n \to \mathbb{R}$, the problem Hamiltonian is defined as $H_P|x\rangle = f(x)|x\rangle$. Thus, the ground states of H_P correspond to the minima of the objective function. In the general framework of QAOA, a parameterized quantum state is prepared as follows:

$$|\gamma,\beta\rangle = U_M(\beta_p)U_P(\gamma_p)\cdots U_M(\beta_1)U_P(\gamma_1)|\phi_0\rangle.$$

Starting from an initial state $|\phi_0\rangle$, the algorithm alternates between applying the phase-separation operator $U_P(\gamma)$ and the mixing operator $U_M(\beta)$. The goal is to optimize the parameters $\gamma,\beta \in \mathbb{R}$ such that $\langle\gamma,\beta|H_P|\gamma,\beta\rangle$ is minimized.

Since the original proposal of QAOA, several extensions and variants have been introduced. One such extension is ADAPT-QAOA (Zhu et al., 2022), which iteratively adds only the terms that most contribute to lowering the expectation value. Another variant is R-QAOA (Bravyi et al., 2020), which progressively removes variables from the Hamiltonian until the reduced instance can be classically solvable. A third extension, WS-QAOA (Egger et al., 2021), incorporates solutions from classical algorithms to warm-

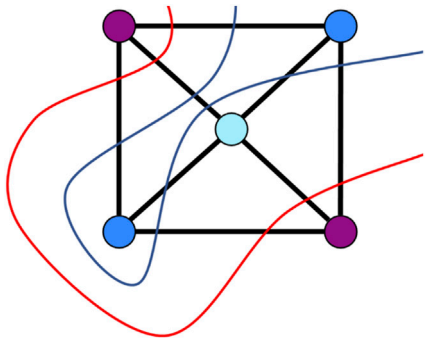


FIGURE 1
Example of an optimal solution for a MAX3-CUT problem.

start the QAOA process. When a given problem has constraints, the design of mixers is vital. For constrained problems, the design of mixers is crucial. A well-known example is the *XY*-mixer (Hadfield et al., 2017; Wang et al., 2020), which restricts evolution to "k-hot" states. GM-QAOA (Bartschi and Eidenbenz, 2020) uses Grover-like operators for all-to-all mixing of more general feasible subsets. The mixing operator of GM comprised a circuit U_S that prepares the uniform superposition of all feasible states, its inverse $U_S^\dagger$, and a multi-controlled rotational phase shift gate. Inspired by the stabilizer formalism, the LX-Mixer (Fuchs and Bassa, 2024; Fuchs et al., 2022) is a generalization of the above methods that provides a flexible framework that allows for tailored mixing within the feasible subspace.

To increase the practical relevance of QAOA, encoding problems with integer values, rather than binary variables, into qubit systems is an important area of research. This process generally introduces considerable flexibility in encoding choices, affecting not only the number of qubits and controlled operations needed but also the complexity of the optimization landscape and the quality of the approximation ratios that can be achieved. In this study, we explore various encoding schemes for the MAX k-CUT problem into qubit systems, building on and improving earlier research (Fuchs et al., 2021). This problem has numerous applications, including the placement of television commercials in program breaks, arrangement of containers on a ship with k bays, partitioning a set of items (e.g., books sold by an online shop) into k subsets, product module design, frequency assignment, scheduling, and pattern matching (Aardal et al., 2007; Ram Gaur et al., 2008). Given a weighted undirected graph $G = (V, E)$, the MAX k-CUT problem consists of finding a maximum-weight k-cut. This involves partitioning the vertices into k subsets such that the sum of the weights of the edges between vertices in different subsets is maximized (Figure 1). By assigning a label $x_i \in \{1, \ldots, k\}$ to each vertex $i \in V$ and defining $\mathbf{x} = (x_1, \ldots, x_{|V|})$, the optimization problem for MAX k-CUT can be formulated as

$$\max_{\mathbf{x}\in\{1,\ldots,k\}^n} C(\mathbf{x}), \qquad C(\mathbf{x}) = \sum_{(i,j)\in E} \begin{cases} w_{ij}, & \text{if } x_i \neq x_j \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

Here, $w_{ij} > 0$ is the weight of edge $(i, j) \in E$. The function $C(\mathbf{x})$ is commonly referred to as the "cost function." For $k = 2$, the MAX k-CUT problem becomes the well-known MAX CUT problem. The MAX k-CUT problem is NP-complete, and it has been demonstrated that no polynomial-time approximation scheme exists for $k \geq 2$ unless P = NP (Alan and Jerrum, 1997). A randomized approximation algorithm for (1) has an approximation ratio α if

$$\mathbb{E}[C(\mathbf{x})] \geq \alpha C(\mathbf{x}^*),$$

where $\mathbf{x}^*$ denotes an optimal solution.

Expressing the solutions as strings of k-dits as in Equation 1 is a natural extension of the MAX (2-)CUT problem to $k > 2$. However, most quantum devices work with a two-level system, so we need to make a suitable encoding on a qubits system (Fuchs et al., 2021; Hadfield et al., 2019). *One-hot encoding* uses k bits for each vertex, where the single bit that is 1 encodes which set/color the vertex belongs to. Using this encoding requires $k|V|$ qubits. However, k not being a power of 2 results in more computational states per vertex than there are colors. The corresponding subspaces need to be dealt with in a suitable fashion. The Grover or the LX-mixer (which becomes the XY-mixer in this case) can be readily applied. However, the ratio of the dimension of the feasible states to the system size is $k/2^k$, which becomes exponentially small for increasing k. Therefore, it is advantageous to use a *binary encoding*, which does not have this problem. For a given k, we encode the information of a vertex belonging to one of the sets by $|i\rangle_{n_k}$, which requires

$$n_k := \lceil log_2(k) \rceil$$

qubits. Here, $\lceil \cdot \rceil$ means rounding up to the nearest integer. Binary encoding requires $n_k|V|$ qubits, which is exponentially fewer than required by one-hot encoding. The resulting problem Hamiltonian can be written as the sum of local terms, that is,

$$H_P = \sum_{e\in E} w_e H_e,$$

where $w_{i,j}$ is the weight of the edge between vertices i and j. Furthermore, the local term can be expressed as

$$H_e = 2\hat{H}_e - I, \ \hat{H}_e = \sum_{(i,j)\in \text{clr}} |i\rangle\langle i| \otimes |j\rangle\langle j|. \tag{2}$$

where clr is a set of index pairs (i, j) representing *equivalent colors* under an equivalence relation. From the definition, H_e has eigenvalue +1 for a state $|i\rangle|j\rangle$ if $i \sim_{\text{clr}} j$ and −1 if not. The phase separating operator e^{-itH_e} differs from $e^{-it\hat{H}_e}$ only by a scalar factor and a global phase, so it is sufficient to realize the latter.

More formally, an equivalence relation $\sim_{\text{clr}}$ is defined on $X = \{0, \ldots, 2^{n_k} - 1\}$, where clr consists of all pairs (i, j) such that $i \sim_{\text{clr}} j$. For example, if clr = $\{(0, 0), (1, 1), (2, 2), (3, 3), (2, 3), (3, 2)\}$, the equivalence relation defines the following equivalence classes: $[0] = \{0\}$, $[1] = \{1\}$, $[2] = \{2, 3\}$. For convenience, we can work with a generating set. Suppose R is a relation on the set X. The equivalence relation on X generated by a R, denoted $\bar{R}$, is then the smallest equivalence relation on X that contains R, which can be obtained by the reflexive, symmetric, and transitive closure (in that order). Hence, only the non-trivial relations must be specified, and the above example can be generated by the closure of the relation $R = \{(2, 3)\}$ which generates $\bar{R} = \{(0, 0), (1, 1), (2, 2), (3, 3), (2, 3), (3, 2)\}$. The *set of*

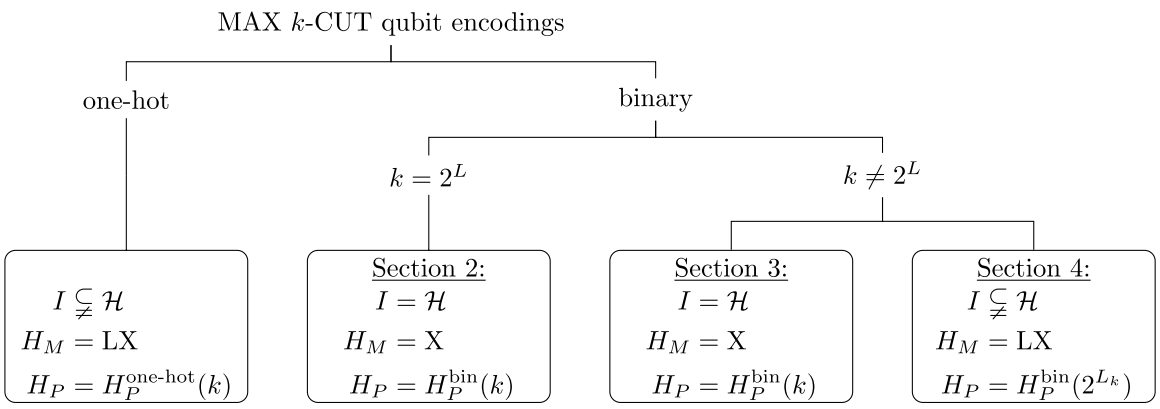


FIGURE 2
Overview of different ways to encode the MAX k-CUT-problem into a (sub-)space I of the full Hilbert space $\mathcal{H}$ with a corresponding problem Hamiltonian H_P. If I is restricted, a valid mixer (Hadfield et al., 2019) H_M needs to be constructed. For binary encoding when k is not a power of 2, we are left with two choices: use the full Hilbert space and introduce an equivalence class of k colors with an unrestricted mixer or use a restricted subspace I with a valid mixer that restricts to I.

colors in the context of MAX k-CUT is then the set of equivalence classes induced by $\sim_{\text{clr}}$, denoted

$$X/\sim_{\text{clr}} := \{[x]\colon x \in X\},$$

with each equivalence class given by

$$[c] := \{x \in X\colon c \sim_{\text{clr}} x\}.$$

We recognize that the phase separating operator—the real time evolution of $\hat{H}_e$—acts by applying a phase on a specific subsets of all computational basis states. When these subsets are sparse, such unitaries can be efficiently realized (Fuchs and Bassa, 2025): let P_B be the projector onto a subspace given by set of computational basis states B. The unitary operator e^{itP_B} can be realized by a circuit for mapping the (indexed) states of B to the first $|B|$ computational basis states, which can be achieved by $M = \prod_{j=0}^{|B|-1} T_{j,z_j}$, where $T_{i,j}$ is a transposition gate. It follows directly that

$$MP_BM^\dagger|j\rangle = MP_B|z_j\rangle = \begin{cases} M|z_j\rangle = |j\rangle, & \text{for } j < |B|, \\ \mathbf{0}, & \text{otherwise.} \end{cases}$$

Therefore, we find that $MP_BM^\dagger = \sum_{j<|B|} |j\rangle\langle j|$, from which it follows that the exponential has the form

$$e^{itP_B} = M^\dagger e^{it\sum_{j<|B|}|j\rangle\langle j|} M = M^\dagger C_{\leq|B|}Ph(t)M,$$

where $C_{\leq|B|}Ph(t)$ is a low-pass controlled phase shift gate as defined in Fuchs and Bassa (2025). For a general subspace, the overall circuit requires $\mathcal{O}(n|B|)$ CX gates and $\mathcal{O}(n|B| + \log(|B|)\log(\frac{1}{\epsilon}))$ T gates.

When the subspace is given by the span of a Pauli X-orbit as defined in Fuchs and Bassa (205), the overall circuit requires only $\mathcal{O}(n\log(|B|))$ CX gates and $\mathcal{O}(n + \log(\frac{1}{\epsilon}))$ T gates. Here, ϵ is a given tolerance for approximating rotational gates. Even though span (B) might not be a Pauli X-orbit subspace, we can always decompose into a direct sum of Pauli X-orbit subspaces and apply the construction individually on the subspaces of the decomposition. We will therefore here aim to divide the subspaces such that they are Pauli X-orbit subspaces if it leads to an efficient circuit.

An overview of possible encoding techniques is provided in Figure 2. When k is not a power of 2, we must deal with the fact that we have non-trivial equivalence classes clr. There are two ways to deal with this. The first is to directly implement the phase separating Hamiltonian on the full Hilbert space (Fuchs et al., 2021) or restrict the evolution onto the relevant subspace. The main contributions of this study are for the cases when k is not a power of 2. We introduce:

- A systematic way to implement the phase separating Hamiltonian, resulting in a reduction of the complexity of the resulting circuit using the theorems from Fuchs and Bassa (2025).
- A method to directly work with a feasible subspace with a suitable state preparation and constrained mixer in Section 4.
- More balanced sets of equivalent colors, which help improve the resulting cost landscape, in Section 3.

We also employ the Grover mixer to all cases, indicating that the resulting landscape might be favorable. Throughout this study, the notation C^lU means an l–controlled U gate. For example, a Toffoli gate is C^2X, and $CPh = CPh(t)$ is the controlled phase shift gate, where we mostly omit the parameter. We will make use of the following notation for *multi-controlled unitary (MCU)* gates

$$C^{\mathbf{c}}_{\mathbf{b}}U_{\mathbf{t}} = (I_{\mathbf{c}} - |\mathbf{b}\rangle\langle\mathbf{b}|_{\mathbf{c}}) \otimes I_{\mathbf{t}} + |\mathbf{b}\rangle\langle\mathbf{b}|_{\mathbf{c}} \otimes U_{\mathbf{t}}, \tag{3}$$

where $\mathbf{c}, \mathbf{t}$ is index control and target qubits, and $\mathbf{b}$ specifies the control condition.

2 Binary encodings when k is a power of 2

When $k = 2^{n_k}$, the equivalence relation for the problem Hamiltonian in Equation 2 becomes trivial—that is, $\text{clr} = \overline{\{\}} = \{(0,0), (1,1), \ldots, (k-1, k-1)\}$—and hence

TABLE 1 Approximation ratios achieved for the case when $k = 2^{n_k}$.

Graph	Mixer	$k = 2$	$k = 4$	$k = 8$
Erdős–Rényi	X	0.87	0.89	0.97
	Grover$^{\square}$	0.87	0.90	0.98
	Grover	0.79	0.83	0.93
Barabási–Albert	X	0.80	0.87	0.96
	Grover$^{\square}$	0.80	0.88	0.96
	Grover	0.74	0.82	0.92

$$\hat{H}_e = \sum_{i=0}^{k-1} |i\rangle\langle i| \otimes |i\rangle\langle i|. \tag{4}$$

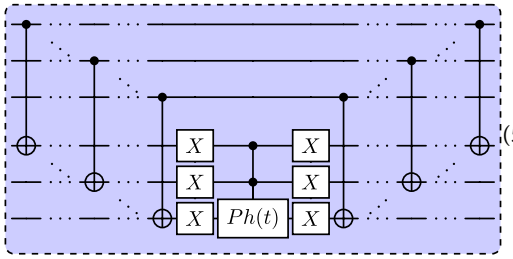
By definition, $\hat{H}_e$ is a projection operator P_B of rank k onto the subspace spanned by $B = \{|j\rangle \otimes |j\rangle, 0 \le j < k\}$, consisting of all states with identical basis vectors in both registers. We notice that the set B is Pauli X-orbit generated as defined in Fuchs and Bassa (2025) since it can be written as

$$B = G_{n_k} |0\rangle^{\otimes 2n_k},$$

that is, the orbit of the computational basis state $|0\rangle^{\otimes 2n_k}$ under the group action of $G_{n_k} = \langle \{X_i \otimes X_i\}_{i=0}^{n_k-1} \rangle$. Here, $\langle S \rangle$ denotes the subgroup generated by a subset S of the Pauli group. To construct the phase separating operator for our concrete case, we follow the proof of Theorem 3 in Fuchs and Bassa (2025) which recursively gives us the following permutation matrix in each step, $M_j = C_1^j X_{j+n_k}$, which leaves the all-zero state unchanged and maps the state $|j\rangle \otimes |j\rangle$ to $|j\rangle|0\rangle$. Overall, this leads to

$$M P_B M^\dagger = I^{n_k} \otimes |0\rangle\langle 0|^{n_k}, \quad M = \prod_{i=0}^{n_k-1} C_1^i X_{i+n_k},$$

resulting in the following circuit:

$e^{it\hat{H}_e} =$ 

(5)

The number of controlled gates required for realizing the mixer and phase separating operators are as follows.

- Realizing $e^{-it\hat{H}_e}$ through Equation 5 requires $2n_k$ CX gates and 1 $C^{n_k-1}Ph$ gate. Alternatively, we can also do a direct decomposition of $\hat{H}_e$ in the Pauli basis and perform Pauli evolution gates with this. For $k = 2, 4, 8$, the first approach needs $2CX$, ($4CX$ and 1 CPh) and ($6CX$ and $1C^2Ph$), whereas the Pauli evolution needs $2CX$, $10CX$, and $34CX$ per gate, respectively. Thus, it is better for these cases to realize the phase separation with the circuit given in Equation 5 and not through Pauli evolution.
- The X mixer does not require any controlled gates, the Grover mixer needs one $C^{|V|n_k-1}P$ gate, and the Grover$^{\square^V}$ needs $|V|$ $C^{n_k-1}P$ gates, where the box product is defined in Equation 11.

We numerically test the performance of the resulting QAOA algorithm on the same instances as in Fuchs et al. (2021), given in Supplementary Appendix A. Table 1 shows that the approximation ratios achieved are very good for all cases. An overview of the energy/approximation ratio landscapes are provided in Supplementary Appendix Table A3.

3 Binary encodings for $k \neq 2^{n_k}$ using the full Hilbert space

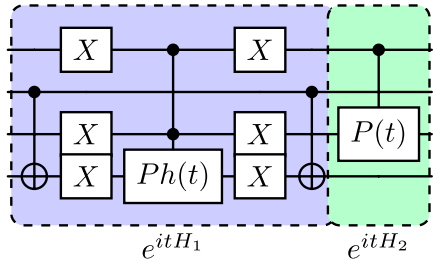
When k is not a power of 2, the equivalence relation for the problem Hamiltonian in Equation 2 becomes non-trivial. One way to deal with this is what we will refer to as $\le k$-encoding, where we encode the color k into all states with $|i\rangle$ for $k \le i \le 2^{n_k} - 1$. This gives us

$$\mathrm{clr}^k_{<k} = \overline{\{(k-1, k), (k, k+1), \ldots, (2^{n_k} - 2, 2^{n_k} - 1)\}}.$$

Of course, in general there are many ways to define k colors through an equivalence relation clr, which influences the cardinality of the equivalence classes. There is, however, always a way to define it such that the maximum cardinality of all equivalence classes is at most 2. In the following, we denote such a relation as $\mathrm{clr}^k_{\mathrm{bal}}$. This also influences the rank of the Hamiltonian $\hat{H}_e$. In the following, we make a concrete construction of the Hamiltonians for $k \in \{3, 5, 6, 7\}$.

3.1 The case $k = 3$

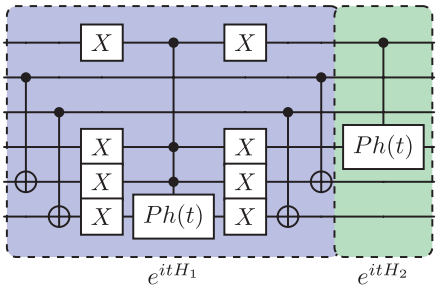
For $k = 3$, $\mathrm{clr}^3_{\mathrm{bal}} = \mathrm{clr}^3_{<3} = \overline{\{(2, 3)\}}$, which means that the three colors are given by $[0] = \{0\}$, $[1] = \{1\}$, $[2] = \{2, 3\}$. The subspace of the projector is given by $B = \{|0000\rangle, |0101\rangle, |1010\rangle, |1111\rangle, |1011\rangle, |1110\rangle\}$. We can realize the phase separating operator using the low-pass controlled phase shift gate $C_{\le 6}Ph$ in the permuted basis, as described in Fuchs and Bassa (2025). However, to reduce the circuit complexity, we can instead divide the subspace in two Pauli X-orbit generated subspaces, $B_1 = \langle X_1 X_3 \rangle |0000\rangle$ and $B_2 = \langle X_1, X_3 \rangle |1010\rangle$, which results in the circuit

$e^{itH_{\mathrm{clr}^3_{<3}}} =$ 

3.2 The case $k = 5$

For $k = 5$, $\mathrm{clr}^5_{<5} = \overline{\{(4, 5, 6, 7)\}}$, which means that the five colors are given by $[0] = \{0\}$, $[1] = \{1\}$, $[2] = \{2\}$, $[3] = \{3\}$, $[4] = \{4, 5, 6, 7\}$.

We can divide B into two Pauli X-orbit generated subspaces. The first subspace containing four states is related to the first four colors $|000\rangle, |001\rangle, |010\rangle |011\rangle$, forming the subset $B_1 = \langle X_1X_4, X_2X_5\rangle|000000\rangle$. The second subspace contains all states related to the last degenerated color $|100\rangle, |101\rangle, |110\rangle, |111\rangle$, forming $B_2 = \langle X_1, X_2, X_4, X_5\rangle|100100\rangle$. The resulting circuit is given by

$e^{itH_{\mathrm{clr}^5_{<5}}} =$  (6)

which realizes the phase separating unitary with one single- and one triple-controlled phase shift gate.

Looking at the case for $k = 3$ and $k = 5$, it becomes clear that the structure of the circuit can be generalized for $\mathrm{clr}_{<k}$ with $k = 2^l + 1$. The feasible subspace can always be divided into two Pauli X-orbit generated subspaces, $B_1 = \langle X_1X_{n_k+1}, \ldots, X_lX_{n_k+l}\rangle|0\cdots0\rangle$, related to the first $k-1$ color states characterized by bitstrings $x = yy$, where the first digit of y is always 0. The second subspace related to the degenerate color is given by $B_2 = \langle X_1, \ldots, X_{n_k-1}, X_{n_k+1}, \ldots, X_{2n_k-1}\rangle|10\ldots0\rangle|10\cdots0\rangle$. From this, we can conclude that the resulting circuit is a generalization of Equation 6, where e^{itH_1} consists of the M_{GHZ} circuit where the CX between the first and n_k-th qubit is omitted and a phase shift gate the with target on the last qubit controlling the first and last n_k (without the last) qubits. The circuit for e^{itH_2} is a phase shift gate with control of the first and target on the $n_k + 1$-th qubit.

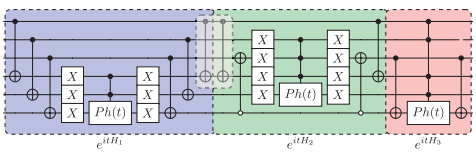
A second possibility is to encode the problem so that the cardinality of the equivalence classes is minimized. This can, for instance, be achieved by $\mathrm{clr}^5_{\mathrm{bal}} = \overline{\{(0,1),(4,5),(6,7)\}}$. In this case, the five colors are given by $[0] = \{0,1\}$, $[2] = \{2\}$, $[3] = \{3\}$, $[4] = \{4,5\}$, $[6] = \{6,7\}$ and the according Pauli X-orbit generated subspaces are

$$B_1 = \langle X_0X_3, X_1X_4, X_2X_5\rangle|000000\rangle, \quad (7)$$

$$B_2 = \langle X_0X_3, X_2X_5\rangle|000001\rangle, \quad (8)$$

$$B_3 = \langle X_2X_5\rangle|111110\rangle. \quad (9)$$

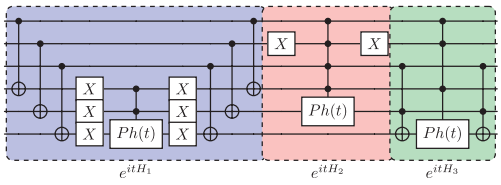
The resulting circuit for the phase separating operator is

$e^{itH_{\mathrm{clr}^5_{\mathrm{bal}}}} =$  (10)

where the gates in the gray box cancel to the identity.

3.3 The case $k = 6$

For $k = 6$, we have that $\mathrm{clr}^6_{<6} = \overline{\{(5,6,7)\}}$, meaning that the six colors are given by $[0] = \{0\}$, $[1] = \{1\}$, $[2] = \{2\}$, $[3] = \{3\}$, $[4] = \{4\}$, $[5] = \{5,6,7\}$. We can divide B into the following two Pauli X-orbit generated subspaces $B_1 = \langle X_0X_3, X_1X_4, X_2X_5\rangle|000000\rangle, B_2 = \langle X_2X_5\rangle|110111\rangle$, together with $B_3 = \{|110101\rangle, |110111\rangle, |111101\rangle, |111110\rangle\}$, which cannot be generated by a Pauli X-orbit. The resulting circuit is given by

$e^{itH_{\mathrm{clr}^6_{<6}}} =$ 

The circuit for the real time evolution of H_2 contains a Toffoli gate for realizing the basis change since the subspace is not Pauli X-orbit generated. In particular, the circuit for the second term H_2 can be derived considering that if we exchange the state $|111110\rangle$ and $|111111\rangle$ using a transposition gate (in this case a 5-controlled not-gate C^5X), the new set $\widetilde{B_2} = \langle X_2, X_4\rangle|110101\rangle$ becomes Pauli X-orbit generated. Since all the states in the subset $B_2 \cup \widetilde{B_2}$ have the bit value 1 in position 0,1,3,5 we can instead replace the use of C^5X by a Toffoli gate $CCX_{2,4\to5}$.

In order to encode the problem in a balanced way, we can use $\mathrm{clr}^6_{\mathrm{bal}} = \overline{\{(0,1),(4,5)\}}$. Since $\mathrm{clr}^6_{\mathrm{bal}} \cup \overline{\{(6,7)\}} = \mathrm{clr}^5_{\mathrm{bal}}$, we can divide B into the subspace B_1 and B_2 shown in Equation 7; the resulting circuit is shown in Equation 10, but the circuit for H_3 is omitted.

3.4 The case $k = 7$

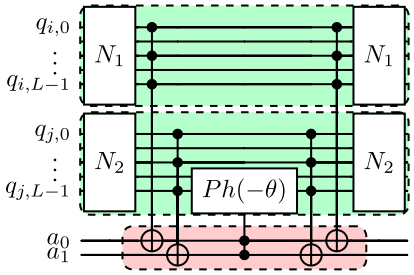
For $k = 7$, $\mathrm{clr}^7_{\mathrm{bal}} = \mathrm{clr}^7_{<7} = \overline{\{(6,7)\}}$, which means that the seven colors are given by $[0] = \{0\}, \ldots, [5] = \{5\}, [6] = \{6,7\}$. Since $\mathrm{clr}^7_{\mathrm{bal}} \cup \overline{\{(0,1),(4,5)\}} = \mathrm{clr}^5_{\mathrm{bal}}$, we can divide B into the subspace B_1 and B_3 shown in Equation 7, where the resulting circuit is shown in Equation 10, but the circuit for H_2 is omitted. Notice here too that the two adjacent C^2X_5 gates between the circuits for H_1 and H_3 cancel.

3.5 Resource analysis

When k is not a power of 2, the method presented in [9, Section 3.2.2] is not optimal in terms of resource efficiency; for each edge of the graph, it requires the circuit shown in Equation 5 and $a(a-1)$ times the circuit shown in Figure 3b, where $a = 2^{n_k} - (k-1)$. Each of these individual circuits uses four $C^{n_k}X$-gates, and one C^2P gate and requires the use of two ancillary qubits. Concretely, for each edge, $2, 12, 6, 2, 56, \cdots$ of these circuits are needed for $k = 3, 5, 6, 7, 9, \cdots$, respectively, with the cost shown in Figure 3a. The method derived from the theorem in Fuchs and Bassa (2025) on the other hand, requires much fewer entangling gates and does not require ancilla qubits. For a single edge, the size of the relevant

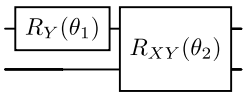
k	[9, Section 3.2.2]	this work
3	$1CPh, 2(4C^2X, 1C^2Ph), 4CX$	$1CPh, 1C^2Ph, 2CX$
5	$1C^2Ph, 12(4C^3X, 1C^2PhP), 6CX$	$1CPh, 1C^3Ph, 4CX$
6	$1C^2Ph, 6(4C^3X, 1C^2PhP), 6CX$	$1C^2Ph, 1C^3Ph, 8CX$
7	$1C^2Ph, 2(4C^3X, 1C^2PhP), 6CX$	$1C^2Ph, 1C^3Ph, 2C^2X, 6CX$

(a) Number of controlled gates to realize the phase shift gate for one edge of the graph.

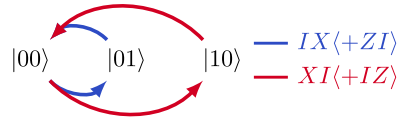


(b) Phase shift gate required in previous work [9].

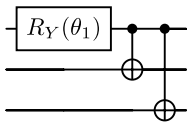
FIGURE 3
Resource comparison shows that the circuits for realizing the phase separating operator U_P as proposed here require fewer resources in terms of entangling gates as Fuchs et al. (2021). Another advantage of the proposed method is that it does not require ancilla qubits. Whereas we realize the diagonal operator directly, Fuchs et al. (2021) utilized the circuit shown in **(b)** between each edge (i,j). For each element $\{y_1, y_2\} \in \mathrm{clr}^k \setminus \cup_{i,j}\{i,i\}$, one such circuit is necessary with $N_{y_i} = X^{bin(y_i)}$. **(a)** Number of controlled gates to realize the phase shift gate for one edge of the graph.



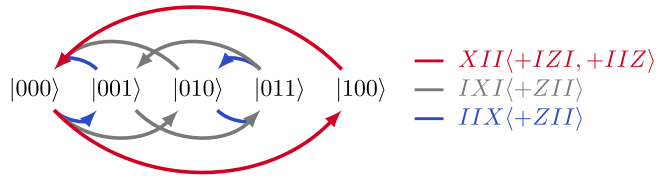
(a) $|\Phi_0^3\rangle$, $t_1 = 2\arccos\left(1/\sqrt{3}\right), t_2 = \pi/2$.



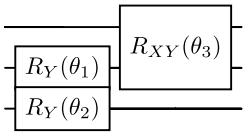
(b) One possible valid mixer for B_3, using 4 CX gates.



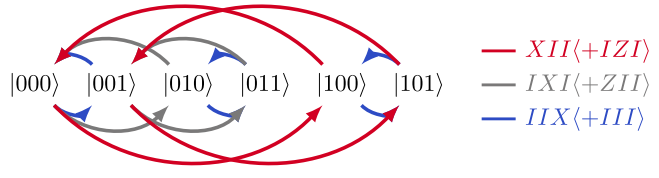
(c) $|\Phi_0^5\rangle$, $t_1 = 2\arcsin\left(1/\sqrt{5}\right)$.



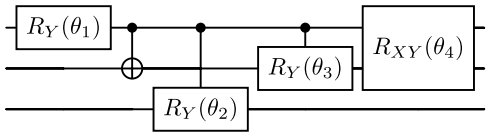
(d) One possible valid mixer for B_5, using 12 CX gates.



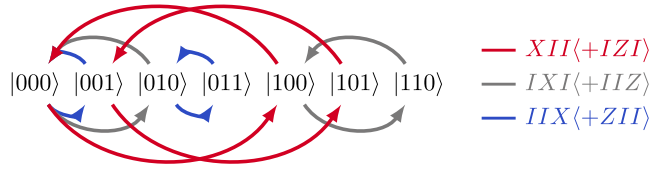
(e) $|\Phi_0^6\rangle$, $t_2 = 2\arccos\left(1/\sqrt{3}\right), t_1 = t_3 = \pi/2$.



(f) One possible valid mixer for B_6, using 4 CX gates.



(g) $|\Phi_0^7\rangle$, $t_1 = 2\arcsin\left(1/\sqrt{7}\right), t_2 = t_4 = \pi/2, t_3 = 2\arccos\left(1/\sqrt{3}\right)$.



(h) One possible valid mixer for B_7, using 6 CX gates.

FIGURE 4
Initial state preparation and mixer for $k = 3$ **(a)** and **(b)**, $k = 5$ **(c)** and **(d)**, $k = 6$ **(e)** and **(f)**, $k = 7$ **(g)** and **(h)**, when restricting to a suitably chosen subspace. We use the notation $R_P(t) = \exp(-i\frac{t}{2}p)$, $P \in \{X, Y, Z\}$, and $R_{XY}(t) = \exp(-i\frac{t}{4}(X \otimes X + Y \otimes Y))$.

subspace B scales as $\mathcal{O}(k^2)$ in the encoding using $\mathrm{clr}^k_{\mathrm{bal}}$ and as $\mathcal{O}(k^4)$ in the encoding using $\mathrm{clr}^k_{<k}$. This quadratic scaling in the latter case is due to the degeneracy of the highest color level, which leads to $(2^{2n} + 1 - k)^2$ valid computational basis states. From the theorem in Fuchs and Bassa (2025), the overall gate complexity for the two encoding strategies is as follows.

- Encoding using $\mathrm{clr}^k_{\mathrm{bal}}$: $\mathcal{O}(|E|\,k^2\log_2(k))$ CNOTs and $\mathcal{O}(|E|k^2\log_2(k)) + 2\log_2(k)\log_2(1/\epsilon)$ single-qubit rotations.
- Encoding using $\mathrm{clr}^k_{<k}$: $\mathcal{O}(|E|k^4\log_2(k))$ CNOTs and $\mathcal{O}(|E|k^4\log_2(k)) + 4\log_2(k)\log_2(1/\epsilon)$ single-qubit rotations.

Here, we have explicitly constructed circuits for $k = 3, 5, 6, 7, 9, \ldots$ using a decomposition similar to that in Fuchs and Bassa (2025), but with a key optimization: we partitioned subspace $|B|$ into ad hoc sums of Pauli X-orbits. This additional structure further reduces the implementation cost. See Supplementary Appendix Table A3a for a comparison.

4 Binary encodings for $k \neq 2^{n_k}$ using subspaces

When k is not a power of 2, there is an alternative to encoding the problem in the full Hilbert space as presented in Section 3. Instead, the evolution of the QAOA can be restricted to a suitably chosen subspace spanned by a set B consisting of k computational basis states. Consequently, we need a way to prepare an initial state in a k-dimensional subspace for each vertex of the graph and a mixer constrained to this subspace. We observe that

$$H_P^{\mathrm{bin}}(2^{n_k})\big|_{\mathrm{span}\,(B)} = H_P^{\mathrm{bin}}(k),$$

meaning we can use the circuit for the power-of-2 case. Overall, this approach shifts complexity from the phase separating Hamiltonian to the mixer.

4.1 Initial state

For the MAX k-CUT problem, one possibility is to define the subspace for each vertex through $B_k = \{|i\rangle \mid 0 \le i \le k-1\}$. Since the feasible bit-strings admit a Cartesian product structure, the feasible subspace becomes a tensor product of the form

$$B_k^1 \otimes \cdots \otimes B_k^{|V|}.$$

An initial state can be prepared as the uniform superposition of all feasible basis states—

$$|\Phi_0^{<k}\rangle^{\otimes|V|}, \text{ where } |\Phi_0^{<k}\rangle = \frac{1}{\sqrt{k}}\sum_{i=0}^{k-1}|i\rangle.$$

This can, in general, be efficiently prepared with a gate complexity and circuit depth of only $\mathcal{O}(\log_2(k))$ (Shukla and Vedula, 2024). One way to realize the initial state for $k \in 3, 5, 6, 7$ is shown in Figures 4a–h.

4.2 Constrained mixer

Given a feasible subspace $\mathrm{span}\,(B) \subset \mathcal{H}$, a mixer U_M is then called valid (Hadfield et al., 2019) if it *preserves the feasible subspace*, that is,

$$U_M(\beta)|v\rangle \in \mathrm{span}\,(B), \quad \forall |v\rangle \in \mathrm{span}\,(B), \forall \beta \in \mathbb{R},$$

and if it *provides transitions between all pairs of feasible states*, where for each pair of computational basis states $|x\rangle, |y\rangle \in B$ there exist $\beta^\star \in \mathbb{R}$ and $r \in \mathbb{N} \cup \{0\}$, such that

$$|\langle x|U_M^r(\beta^\star)|y\rangle| > 0.$$

Apart from the Grover mixer, we consider here constraint-preserving mixers in the stabilizer form (Fuchs and Bassa, 2024), in which case the Hamiltonian can be written as

$$H_M = \sum_\alpha c_\alpha \bar{X}^\alpha \prod_{V_\alpha},$$

where $\alpha \in \{0,1\}^n\backslash\{0\}^n$ is a multi-index, $\bar{X}^\alpha = X_1^{\alpha_1}\cdots X_n^{\alpha_n} \in \{I, X\}^n\backslash\{I\}^n$, and $\prod_{V_\alpha}$ is a projection onto V_α. Furthermore, V_α is a subspace fulfilling $|x\rangle \in V_\alpha \Rightarrow \bar{X}^\alpha|x\rangle \in V_\alpha$. The free parameter $c_\alpha \in \{0, 1\}$ must be chosen such that we have a valid mixer. Let $\{S_1, \ldots, S_l\}$ be a minimal generating set of the stabilizer group $S = \langle S_1, \ldots, S_l\rangle$ defining the code subspace $C(S)$. We can choose the projection operator to be of the form

$$\prod_{C(S)} = \frac{1}{2^k}\sum_{S \in \langle S_1,\ldots,S_k\rangle} S.$$

For more details, see Fuchs and Bassa (2024). In the following, we will use the notation

$$\bar{X}\langle S_1, \ldots, S_k\rangle := \frac{1}{2^k}\sum_{S \in \langle S_1,\ldots,S_k\rangle} \bar{X}S.$$

Given the independence of the different color constraints for each vertex of the problem, we can then define the following LX-mixer Hamiltonian

$$H_M = G^{\square|V|} = G \square G \cdots \square G,$$

where the Cartesian or Box product is defined as

$$G \square H = G \otimes \mathbb{I} + \mathbb{I} \otimes H. \tag{11}$$

It can be shown that if G is a valid mixer, H_M is constructed in this way (Fuchs and Bassa, 2024). Valid mixers with optimal cost in terms of CX gates are shown in Figure 4 for $k \in \{3, 5, 6, 7\}$. We note that this choice is not unique and that other mixers are possible as well.

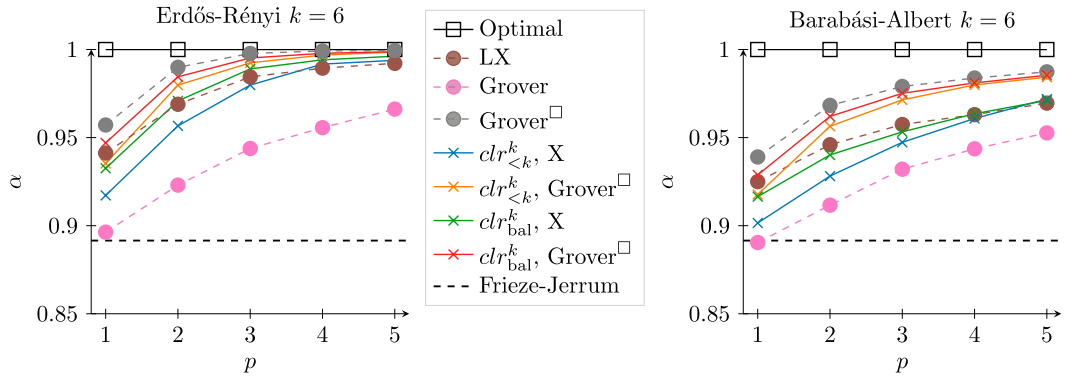
5 Simulations

We perform numerical simulations for an unweighted Erdős–Rényi graph and a weighted Barabási–Albert graph with ten vertices (Supplementary Appendix A). Since the case where k is a power of 2 is presented in Section 2, we will focus on evaluating the methods presented in Sections 3 and 4. Using an ideal simulator, we compare the results for the binary encoding with the standard X-mixer and different choices of equivalence relations to encode the colors, and we compare the subspace encoding with different choices of mixers. The resulting energy landscapes are provided in Supplementary Appendix A. As expected, the landscapes are periodic in β when using the X- and Grover-mixer, whereas the landscape associated with the LX-mixer is generally not. The resulting approximation ratios are shown in Table 2. Overall, the subspace-encoding using the Grover$^\square$ mixer achieves the highest values for these two graphs, although with only a small distance to

TABLE 2 Comparison of different encoding schemes when k is not a power of 2. The X-mixer uses the encoding of the full Hilbert space presented in Section 3, whereas the LX- and Grover mixers encode the problem in a subspace as described in Section 4. Approximation ratios achieved for the graphs shown in Supplementary Appendix A for depth p = 1. The light gray color indicates methods using subspaces as described in Section 4, whereas the rest indicates methods using the full Hilbert space as described in Section 3.

Graph	Mixer	$\|\phi_0\rangle$	clr	$k=3$	$k=5$	$k=6$	$k=7$
Erdős–Rényi	X	$\|+\rangle$	$\mathrm{clr}^k_{<k}$	0.80	0.86	0.92	0.95
	Grover$^\square$	$\|+\rangle$	$\mathrm{clr}^k_{<k}$	0.81	0.90	0.94	0.96
	X	$\|+\rangle$	$\mathrm{clr}^k_{\mathrm{bal}}$	—	0.90	0.94	—
	Grover$^\square$	$\|+\rangle$	$\mathrm{clr}^k_{\mathrm{bal}}$	—	0.93	0.95	—
	LX	$\|\Phi_0^{<k}\rangle$	-	0.80	0.91	0.94	0.94
	Grover$^\square$	$\|\Phi_0^{<k}\rangle$	-	0.83	0.94	0.96	0.97
	Grover	$\|\Phi_0^{<k}\rangle$	-	0.75	0.87	0.90	0.92
Barabási–Albert	X	$\|+\rangle$	$\mathrm{clr}^k_{<k}$	0.79	0.84	0.90	0.93
	Grover$^\square$	$\|+\rangle$	$\mathrm{clr}^k_{<k}$	0.79	0.88	0.92	0.95
	X	$\|+\rangle$	$\mathrm{clr}^k_{\mathrm{bal}}$	—	0.88	0.92	—
	Grover$^\square$	$\|+\rangle$	$\mathrm{clr}^k_{\mathrm{bal}}$	—	0.91	0.93	—
	LX	$\|\Phi_0^{<k}\rangle$	—	0.79	0.89	0.93	0.93
	Grover$^\square$	$\|\Phi_0^{<k}\rangle$	—	0.82	0.92	0.94	0.95
	Grover	$\|\Phi_0^{<k}\rangle$	—	0.75	0.86	0.89	0.91

TABLE 3 Approximation ratio for $k = 6$ for the different methods as the depth p is increased, employing the interpolation-based heuristic described in (Zhou et al., 2020) and compared with the Frieze–Jerrum semidefinite programming relaxation (Alan and Jerrum, 1997). The subspace methods constrained to subspaces use dashed lines with circles as makers, and the methods using the full Hilbert space use solid lines with ´´x" as markers.



the LX-mixer. We can also see that balancing the bin sizes for the different colors has a very positive effect on the approximation ratio when using the full Hilbert space and the X-mixer. The effect of increasing the depth is shown in Table 3, which is exemplified for $k = 6$. Note that the behavior for $k = 3, 5, 6$, and 7 were very similar. Overall, the Grover$^\square$ achieved the best convergence for these two graphs, irrespective of whether the full Hilbert space or the subspace method was used.

We note, however, that the averaged approximation ratio achieved should be carefully considered with respect to the required circuit depth. For encoding in the full Hilbert space, we can use the X-mixer, but the phase-separation operator requires a deeper circuit. For encoding into a subspace, we can use the phase-separation operator for the power-of-2 case (leading to a more shallow circuit), but the circuit for the constrained preserving mixer is deeper. An indication of the cost is given in Table 4, where we used

TABLE 4 Cost in terms of CX gates when using the qiskit compiler. Methods from previous research is marked with light gray background: The line with $H_P^{\text{old}}(k)$ is the Hamiltonian presented in Fuchs et al. (2021), and the $H_P^{\text{Pauli}}(k)$ decomposes the Hamiltonian in the Pauli basis. Terms related to the phase-separating Hamiltonian scale with the number of edges and terms related to the mixer scale with the number of vertices. Notice that by encoding the problem in a specific subspace rather than the entire Hilbert space, the complexity is shifted away from the phase-separating Hamiltonian and instead transferred to the mixer.

Mixer	Initial state	Cost Hamiltonian	$k = 3$	$k = 5$	$k = 6$	$k = 7$
X	$\|+\rangle$	$H_P^{<k,\text{old}}(k)$	$70\|E\|$	$782\|E\|$	$398\|E\|$	$142\|E\|$
X	$\|+\rangle$	$H_P^{<k,\text{Pauli}}(k)$	$18\|E\|$	$66\|E\|$	$258\|E\|$	$130\|E\|$
X	$\|+\rangle$	$H_P^{<k}(k)$	$12\|E\|$	$26\|E\|$	$90\|E\|$	$58\|E\|$
X	$\|+\rangle$	$H_P^{\text{bal,old}}(k)$	—	$398\|E\|$	$270\|E\|$	-
X	$\|+\rangle$	$H_P^{\text{bal,Pauli}}(k)$	—	$130\|E\|$	$66\|E\|$	-
X	$\|+\rangle$	$H_P^{\text{bal}}(k)$	—	$82\|E\|$	$36\|E\|$	-
LX	$\|\Phi_0^{<k}\rangle$	$H_P(2^{n_k})$	$(2+4)\|V\|+6\|E\|$	$(2+12)\|V\|+14\|E\|$	$(2+4)\|V\|+14\|E\|$	$(5+6)\|V\|+14\|E\|$
Grover$^{\square}$	$\|\Phi_0^{<k}\rangle$	$H_P(2^{n_k})$	$(2+6)\|V\|+6\|E\|$	$(2+12)\|V\|+14\|E\|$	$(2+12)\|V\|+14\|E\|$	$(5+18)\|V\|+14\|E\|$

the compiler from qiskit, assuming full connectivity. Note that for a sparse graph $|E| = \mathcal{O}(|V|)$ and for a fully connected graph $|E| = \Theta(|V|^2)$. Therefore, for graphs with sufficient connectivity, the subspace encoding uses fewer resources than the encoding into the full Hilbert space.

6 Conclusion

This study presents advances in encoding the MAX k-CUT problem into qubit systems, with a focus on cases where k is not a power of 2. We present new ways of encoding that significantly reduce the resource requirements compared to previous approaches. In our implementation, we compared two different encodings using the full Hilbert space. Empirically, we observed that the balanced encoding consistently produced more compact circuits and achieved higher approximation ratios across all tested circuit depths and for both graph instances. We believe that this improvement is linked to the structure of the cost Hamiltonian, specifically the reduced degeneracy induced by the balanced encoding, which may lead to a smoother optimization landscape. However, we acknowledge that we do not currently have a theoretical guarantee that the balanced encoding results in fewer barren plateaus than the $\leq k$ encoding. While the empirical results suggest an improvement in trainability, a rigorous analysis of the gradient landscape remains an open direction for future work. Balanced color sets and constrained mixers demonstrate notable improvements in optimization landscapes and approximation ratios.

While our method is designed for digital quantum architectures, extending it to quantum annealers is possible in principle but practically nontrivial. Quantum annealing hardware is restricted to 2-local Ising Hamiltonians, whereas our encoding yields k-local PUBO terms for $k \geq 3$. Mapping these higher-order terms to 2-local form would require ancillary qubits and penalty constraints, introducing significant overhead and embedding complexity (Nagies et al., 2025) or the use of hybrid techniques as exposed in (Wurtz et al., 2024). In the future, we plan to do a careful analysis of which ansatz is preferable when k is not a power of 2, using a suite of graph instances. The analysis should take into account the overall circuit depth, the achieved approximation ratio for increasing depth size, as well as if the landscape has barren plateaus or many "bad" local minima.

Data availability statement

The raw data supporting the conclusions of this article will be made available by the authors, without undue reservation.

Author contributions

FF: Writing – review and editing, Investigation, Software, Conceptualization, Writing – original draft, Supervision, Visualization, Data curation, Project administration, Formal Analysis, Methodology. RP: Conceptualization, Investigation, Formal Analysis, Writing – review and editing, Methodology, Writing – original draft. FL: Writing – review and editing, Software, Data curation.

Funding

The authors declare that financial support was received for the research and/or publication of this article. We would like to thank for funding of the work by the Research Council of Norway through project number 332023.

Acknowledgements

The authors wish to thank Terje Nilsen at Kongsberg Discovery for the access to an H100 GPU for the simulations.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Generative AI statement

The authors declare that no Generative AI was used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

Supplementary material

The Supplementary Material for this article can be found online at: https://www.frontiersin.org/articles/10.3389/frqst.2025.1636042/full#supplementary-material

References

Aardal, K. I., van Hoesel, S. P. M., Koster, A. M. C. A., Mannino, C., and Sassano, A. (2007). Models and solution techniques for frequency assignment problems. *Ann. Operations Res.* 153 (1), 79–129. doi:10.1007/s10479-007-0178-0

Alan, F., and Jerrum, M. (1997). Improved approximation algorithms for maxk-cut and max bisection. *Algorithmica* 18 (1), 67–81. doi:10.1007/BF02523688

Bartschi, A., and Eidenbenz, S. (2020). "Grover mixers for qaoa: shifting complexity from mixer design to state preparation," in 2020 IEEE International Conference on Quantum Computing and Engineering (QCE) (IEEE), 72–82. doi:10.1109/qce49297.2020.00020

Bravyi, S., Kliesch, A., Koenig, R., and Tang, E. (2020). Obstacles to variational quantum optimization from symmetry protection. *Phys. Rev. Lett.* 125 (26), 260505. doi:10.1103/PhysRevLett.125.260505

Egger, D. J., Mareček, J., and Woerner, S. (2021). Warm-starting quantum optimization. *Quantum* 5, 479. doi:10.22331/q-2021-06-17-479

Farhi, E., Goldstone, J., and Gutmann, S. (2014). A quantum approximate optimization algorithm. *arXiv Prepr. arXiv:1411.4028*. doi:10.48550/arXiv.1411.4028

Fuchs, F. G., and Bassa, R. P. (2024). Lx-mixers for qaoa: optimal mixers restricted to subspaces and the stabilizer formalism. *Quantum* 8, 1535–11. doi:10.22331/q-2024-11-25-1535

Fuchs, F. G., and Bassa, R. P. (2025). Compact circuits for constrained quantum evolutions of sparse operators. *arXiv*. doi:10.48550/arXiv.2504.09133

Fuchs, F. G., Kolden, H. Ø., Henrik Aase, N., and Sartor, G. (2021). Efficient encoding of the weighted MAX $$k$$-CUT on a quantum computer using QAOA. *SN Comput. Sci.* 2 (2), 89. doi:10.1007/s42979-020-00437-z

Fuchs, F. G., Olsen Lye, K., Nilsen, H. M., Stasik, A. J., and Sartor, G. (2022). Constraint preserving mixers for the quantum approximate optimization algorithm. *Algorithms* 15 (6), 202. doi:10.3390/a15060202

Hadfield, S., Wang, Z., Rieffel, E. G., O'Gorman, B., Venturelli, D., and Biswas, R. (2017). "Quantum approximate optimization with hard and soft constraints," in *Proceedings of the second international workshop on post moores era supercomputing*, 15–21. doi:10.1145/3149526.3149530

Hadfield, S., Wang, Z., O'Gorman, B., Rieffel, E. G., Venturelli, D., and Biswas, R. (2019). From the quantum approximate optimization algorithm to a quantum alternating operator ansatz. *Algorithms* 12 (2), 34. doi:10.3390/a12020034

Nagies, S., Geier, K. T., Akram, J., Bantounas, D., Johanning, M., and Hauke, P. (2025). Boosting quantum annealing performance through direct polynomial unconstrained binary optimization. *Quantum Sci. Technol.* 10 (3), 035008. doi:10.1088/2058-9565/adcae6

Ram Gaur, D., Krishnamurti, R., and Kohli, R. (2008). The capacitated max k-cut problem. *Math. Program.* 115 (1), 65–72. doi:10.1007/s10107-007-0139-z

Shukla, A., and Vedula, P. (2024). An efficient quantum algorithm for preparation of uniform quantum superposition states. *Quantum Inf. Process.* 23 (2), 38–1332. doi:10.1007/s11128-024-04258-4

Wang, Z., Rubin, N. C., Dominy, J. M., and Rieffel, E. G. (2020). XY mixers: analytical and numerical results for the quantum alternating operator ansatz. *Phys. Rev. A* 101 (1), 012320. doi:10.1103/physreva.101.012320

Wurtz, J., Sack, S. H., and Wang, S.-T. (2024). Solving non-native combinatorial optimization problems using hybrid quantum-classical algorithms. *IEEE Trans. Quantum Eng.* 5, 1–14. doi:10.1109/tqe.2024.3443660

Zhou, L., Wang, S.-T., Choi, S., Pichler, H., and Lukin, M. D. (2020). Quantum approximate optimization algorithm: performance, mechanism, and implementation on near-term devices. *Phys. Rev. X* 10 (2), 021067. doi:10.1103/physrevx.10.021067

Zhu, L., Tang, Ho L., Barron, G. S., Calderon-Vargas, F. A., Mayhall, N. J., Barnes, E., et al. (2022). Adaptive quantum approximate optimization algorithm for solving combinatorial problems on a quantum computer. *Phys. Rev. Res.* 4 (3), 033029. doi:10.1103/PhysRevResearch.4.033029