Complexity of graph-state preparation by Clifford circuits

Soh Kumabe[1], Ryuhei Mori[2], and Yusei Yoshimura[3]

[1]CyberAgent, Japan
[2]Graduate School of Mathematics, Nagoya University, Japan
[3]School of Computing, The Institute of Science Tokyo, Japan

In this work, we study the complexity of graph-state preparation in a general model of quantum algorithms that allows measurements in the computational basis, single-qubit Clifford operations, and two-qubit Clifford operations. We define the *CZ-complexity* of a graph state $|G\rangle$ as the minimum number of two-qubit Clifford operations required to generate $|G\rangle$ from $|0\rangle^{\otimes(n+s)}$ for some $s \geq 0$. Equivalently, every optimal algorithm can be taken to use only controlled-Z (CZ) gates as its two-qubit Clifford operations. We then give a combinatorial characterization of graph-state transformations. Specifically, $|G\rangle$ can be generated from another graph state $|H\rangle$ by an algorithm of CZ-complexity at most t if and only if G can be obtained from H by vertex deletions, local complementations and at most t *elementary edge-complementations*. Here, an elementary edge-complementation toggles either a single edge, all edges between one vertex and the neighborhood of another, or all edges between the neighborhoods of two non-adjacent vertices. Using this characterization, we relate CZ-complexity to rank-width. For any graph G with n vertices and rank-width r, the CZ-complexity is $O(rn)$, and if G is connected then it is at least $n + r - 2$. We also show that these bounds are close to optimal. Finally, for interval graphs and circle graphs, whose rank-width is unbounded, we present preparation algorithms with CZ-complexity $O(n)$ and $O(n \log n)$, respectively.

1 Introduction

The set of stabilizer states is a large class of quantum states used in quantum information processing. The set of graph states is a subclass of stabilizer states represented by an undirected graph. Any stabilizer state can be transformed into a graph state using local Clifford operations [1]. Hence, it is important to develop efficient quantum algorithms for preparing graph states since any stabilizer state can be transformed from some graph state by local Clifford operations. Graph states have been studied from several perspectives, including their multipartite entanglement and entanglement purification [2, 3], their role as resource states for measurement-based quantum computation [4, 5, 6, 7, 8], graph-theoretic descriptions of their transformation under local Clifford operations and related single-qubit operations [1, 9, 10, 11, 12], and optimal preparation problems [13, 14]. In this work, we focus on the last aspect and study graph-state preparation complexity from the viewpoint of the number of two-qubit Clifford operations.

In general, any stabilizer state can be generated from $|0\rangle^{\otimes n}$ by an n-qubit Clifford operation. Any n-qubit Clifford operation is generated by single-qubit and two-qubit Clifford operations. For example, the commonly used generators of the Clifford group are the S, H and CZ gates. In this work, we study the complexity of graph-state preparation. Although there are several implementations of quantum computing, e.g., superconductors, trapped-ion, etc., we are interested

Soh Kumabe: kumabe_soh@cyberagent.co.jp
Ryuhei Mori: mori@math.nagoya-u.ac.jp

in complexity measures that capture universal difficulties in graph-state preparation. The implementation of two-qubit operations is generally challenging. Therefore, we define the complexity of graph states as follows.

Definition 1 (The CZ-complexity of graph states)**.** Let $\mathcal{A}$ be a quantum algorithm that consists of the following operations.

1. Measurements of qubits in the computational basis.
2. Single-qubit Clifford operations.
3. Two-qubit Clifford operations.

Here, we assume that measured qubits are discarded in quantum algorithms. After the measurements of some qubits, the algorithm may choose subsequent operations depending on the measurement outcomes. The CZ-*complexity of* $\mathcal{A}$ is defined as the maximum number (with respect to the measurement outcome) of two-qubit Clifford operations in $\mathcal{A}$. The CZ-*complexity* $\Gamma_{\mathrm{CZ}}(|G\rangle)$ *of graph state* $|G\rangle$ is defined as the minimum CZ-complexity of $\mathcal{A}$ that generates the n-qubit graph state $|G\rangle$ from $|0\rangle^{\otimes(n+s)}$ with probability 1 for some $s \geq 0$.

In a physical realization of a quantum computer, the use of additional working qubits may be costly. In such a setting, the number of working qubits should also be included in the cost of a preparation algorithm. Furthermore, real quantum computers have hardware-specific connectivity constraints, so that CZ gates can be applied only to restricted pairs of qubits. In this work, however, we ignore such hardware-dependent issues, and count only the number of CZ operations. This viewpoint allows us to study the cost of graph-state preparation in a clean, architecture-independent manner. Motivated by applications of graph states in several settings, including measurement-based quantum computation, our goal is to understand which graph states can be prepared using only a small number of CZ operations and how this preparation cost is governed by graph structure. Our results give upper bounds, lower bounds, and structural characterizations for CZ-complexity of preparing such resource states.

There is a simple quantum algorithm generating an n-qubit graph state $|G\rangle$. First, $|+\rangle^{\otimes n}$ is prepared, which corresponds to the graph state for the empty graph of n vertices. Here, each qubit corresponds to each vertex of G. Then, for each edge of G, the controlled-Z (CZ) operation is applied for the corresponding pair of qubits. This quantum algorithm generates $|G\rangle$ with m two-qubit Clifford operations where m is the number of edges in G.

Van den Nest, Dehaene, and De Moor proved that two graph states $|G\rangle$ and $|H\rangle$ can be transformed to each other by local Clifford operations if and only if G and H can be transformed to each other by some combinatorial graph transformations, called local complementations [1]. The local complementation at a vertex v is an operation that complements edges for all pairs in neighborhoods of v. The local complementation can reduce the number of edges in graphs in some cases. For example, the complete graph K_n can be transformed to the star graph S_{n-1} by a local complementation. This implies that $|K_n\rangle$ is generated from $|S_{n-1}\rangle$ by local Clifford operations. Since S_{n-1} has $n-1$ edges, $|S_{n-1}\rangle$ can be generated with $n-1$ two-qubit Clifford operations. Hence, the CZ-complexity of $|K_n\rangle$ is at most $n-1$ (Indeed, this is a lower bound for any connected graph as well). This example shows that local Clifford operations are useful for reducing the number of two-qubit Clifford operations. As another important example, Figure 1 shows that the cycle graph C_4 of size four can be transformed to the path graph P_4 of size four by the local complementations. Note that we do not allow the use of local non-Clifford operations in Definition 1 although they are generally stronger than local Clifford operations [15, 12].

Based on the above observations, we can consider a general strategy for designing low CZ-complexity quantum algorithms. To generate $|G\rangle$, find a graph H that has the minimum number of edges among all graphs that can be transformed to G by local complementations. Then, $|H\rangle$ can be generated with m two-qubit operations where m is the number of edges of H. Finally, $|G\rangle$ can be generated from $|H\rangle$ by a local Clifford operation. The CZ-complexities of the algorithms based on this strategy were calculated for all graph states up to 12 qubits in [13]. Recently, algorithms for minimizing the number of edges by local complementations were provided in [14]. The algorithm based on simulated annealing efficiently finds a graph with approximately minimum number of

Figure 1: C_4 and P_4 are local-complement equivalent. When the local complementations are applied for the filled vertices, C_4 is transformed to P_4.

Figure 2: Graphs that have the minimum number of edges among all graphs obtained by local complementations, and include cycles of size three or four.

edges up to 100 qubits. Another algorithm exactly finds a graph with minimum number of edges up to 16 qubits. In quantum algorithms of this type, the CZ operations are applied for $|+\rangle^{\otimes n}$ first, and then a local Clifford operation is applied.

In fact, this simple strategy is not necessarily optimal. The two graphs in Figure 2 each have six edges, which is the minimum number possible under local-complementations [11]. However, these graphs can be generated with five CZ operations since cycles of size three and four can be generated with two and three CZ operations, respectively.[1] The class of quantum algorithms in Definition 1 is more general than quantum algorithms from the above strategy since any two-qubit Clifford operations rather than the CZ operation are allowed, and single-qubit Clifford operations and two-qubit Clifford operations can be applied in arbitrary order. We first observe that, without loss of generality, we can assume that all two-qubit Clifford operations in graph-state preparation algorithms are CZ operations, thus justifying the terminology of CZ-complexity introduced in Definition 1.

Proposition 2. *For any graph G, there exists a quantum algorithm $\mathcal{A}$ generating $|G\rangle$ of the type in Definition 1 with CZ-complexity $\Gamma_{\mathrm{CZ}}(|G\rangle)$ where all two-qubit Clifford operations in $\mathcal{A}$ are the* CZ *operations.*

Proposition 2 is proved in Appendix B. For a graph state $|G\rangle$, a CZ operation corresponds to toggling a single edge of G. However, in the framework of quantum algorithms in Definition 1, CZ operations can be performed on a stabilizer state which may not be a graph state. We prove a combinatorial characterization of the graph state transformation with CZ-complexity at most t.

Theorem 3. *For any graphs G and H, $|G\rangle$ can be generated from $|H\rangle$ with CZ-complexity at most t if and only if G can be obtained from H by vertex deletions, local complementations and at most t elementary edge-complementations, which are defined in Definition 24.*

This combinatorial characterization can be regarded as a generalization of the combinatorial characterization for $t = 0$ [1, 10]. Theorem 3 is derived from the graphical description of CZ operations on stabilizer states [9].

It is known that the asymptotic CZ-complexity is $\Theta(n^2/\log n)$ [16]. For better upper bounds, we consider graphs with bounded rank-width. The rank-width is a complexity measure of graphs introduced by Oum and Seymour [17, 18]. We derived the upper and lower bounds of the CZ-complexity using the rank-width.

[1] Interestingly, there exist graphs that have the minimum number of edges among all graphs obtained by local complementations, and do not contain cycles of size three or four, but can be generated more efficiently, e.g., No.131 in [11].

Theorem 4. *Let G be a graph with n vertices and rank-width at most $r \geq 1$. If r is odd, for any $n \geq \frac{13r^2-6r-3}{4r}$,*

$$\Gamma_{\mathrm{CZ}}(|G\rangle) \leq \frac{5r^2-1}{4r}n - \frac{221r^4-180r^3+10r^2+36r+9}{96r^2}.$$

If r is even, for any $n \geq \frac{13r-6}{4}$,

$$\Gamma_{\mathrm{CZ}}(|G\rangle) \leq \frac{5r}{4}n - \frac{221r^2-180r+100}{96}.$$

For the case $r = 1$, this upper bound is optimal for connected graphs.

Corollary 5. *For any graph G with n vertices and rank-width 1,*

$$\Gamma_{\mathrm{CZ}}(|G\rangle) \leq n - 1.$$

Since $n-1$ is a trivial lower bound on the CZ-complexity for any connected graph G, Corollary 5 implies that $\Gamma_{\mathrm{CZ}}(|G\rangle) = n-1$ for any connected graph G with rank-width 1. The upper bound $O(rn)$ is nearly optimal, as demonstrated by the following lemma derived from information-theoretic counting arguments.

Lemma 6. *There exists a positive constant c such that for any positive integer n and $r \leq \lceil n/3 \rceil$, there exists a graph G with n vertices and rank-width at most r such that $\Gamma_{\mathrm{CZ}}(|G\rangle) \geq crn/\log n$.*

The lower bound derived from the rank-width is formulated as follows.

Theorem 7. *For any connected graph G with $n \geq 2$ vertices and rank-width at least r, $\Gamma_{\mathrm{CZ}}(|G\rangle) \geq n + r - 2$.*

The cycle graph C_n with n vertices and n edges has rank-width two when $n \geq 5$. Theorem 7 implies that $\Gamma_{\mathrm{CZ}}(|C_n\rangle) = n$ when $n \geq 5$. This is in contrast to the fact $\Gamma_{\mathrm{CZ}}(|C_n\rangle) = n-1$ for $n = 3, 4$. From the analysis of the rank-width of random sparse graphs [19], the following lemma is obtained.

Lemma 8. *There exists a constant $c \geq 1$ such that for any r' there exist n' and $r \geq r'$ such that for any $n \geq n'$, there exists a connected graph G with n vertices and rank-width r satisfying $\Gamma_{\mathrm{CZ}}(|G\rangle) \leq n - 1 + c(r-1)$.*

This lemma implies that the lower bound $\Gamma_{\mathrm{CZ}}(|G\rangle) - (n-1) \geq r - 1$ for a connected graph G obtained from Theorem 7 is optimal up to a constant factor.

It is natural to ask whether the rank-width is an appropriate choice of the complexity measure of graphs for bounding the CZ-complexity of graph states. A similar upper bound to Theorem 4 may be obtained from other complexity measures satisfying some properties required for the proof of Theorem 4. On the other hand, for the proof of Theorem 7, the invariance under the local complementations is required for the complexity measure. Deriving lower bounds for the CZ-complexity from other graph complexity measures is challenging due to this highly restrictive requirement, despite the upper and lower bounds in Theorems 4 and 7 not being tight.

Finally, we present efficient quantum algorithms for special classes of graphs with unbounded rank-widths.

Theorem 9. *For any interval graph G with n vertices, $\Gamma_{\mathrm{CZ}}(|G\rangle) \leq 2n - 2$. For any circle graph G with n vertices, $\Gamma_{\mathrm{CZ}}(|G\rangle) \leq 1.262 \cdot (n-1)\log_2(n+1)$.*

Recently, Davies and Jena proved that for any circle graph G with n vertices and the chromatic number k, $\Gamma_{\mathrm{CZ}}(|G\rangle) \leq 2n\lceil \log_2 k \rceil$ [20, 21]. The class of circle graphs plays a key role within vertex-minor theory, analogous to the role of planar graphs within graph minor theory [22]. Davies and Jena concurrently and independently introduced the notion of CZ-distance which is almost the same as the CZ-complexity [20, 21]. Compared to the CZ-complexity, the CZ-distance has two restrictions: (1) the use of working qubits that are measured and discarded is not allowed, and (2) CZ-operations may be applied only to graph states, and not to general stabilizer states. The

second restriction incurs an overhead of at most a factor of two since only one of the three types of operations appearing in elementary edge-complementations in Definition 24 is allowed. We do not know how large the overhead is when the use of working qubits is forbidden. Note that all graph-state preparation algorithms in this paper do not require working qubits. Indeed, we do not have any example in which the use of working qubits helps to reduce the number of CZ operations. However, we prefer to define the CZ-complexity so as to allow the use of working qubits since the lower bounds in this paper hold even in this setting, and since the CZ-complexity has the monotonicity property, i.e., $\Gamma_{\mathrm{CZ}}(|H\rangle) \leq \Gamma_{\mathrm{CZ}}(|G\rangle)$ for any graph G and any vertex-minor H of G.

1.1 Related work

The graph state was introduced in [2] and has been extensively studied [4, 3, 1, 9, 12].

Before the introduction of graph states, Bouchet introduced and studied *the isotropic system*, which is technically very similar to the stabilizer states [23, 24, 25, 26, 27, 28, 29]. Important notions in graph states, e.g., the local complementation, the pivoting, the vertex-minor, etc., have been studied in the literature. As an important result, Bouchet presented an $O(n^4)$-time algorithm to decide whether two graphs can be transformed to each other by the local complementations [27]. The notion of the local complementation was introduced by Kotzig in the context of Eulerian cycle of 4-regular graphs [30]. Fon-Der-Flaass also showed some properties of the local complementations [31].

Oum and Seymour introduced the rank-width [17, 18]. The rank-width is based on the cut-rank function, which was introduced and called *the connectivity function* by Bouchet [32, 28]. Although rank-width was originally introduced for approximating clique-width, it possesses its own interesting features and has since been studied extensively [33, 34, 18, 19, 35, 36, 37, 38, 39].

Høyer, Mhalla, and Perdrix introduced a graph-theoretical measure, *local minimum degree*, which is the minimum of the minimum degree up to the local complementations [40]. The local minimum degree has been studied in [41, 42, 43]. Indeed, the local minimum degree is relevant to our algorithms in this paper.

Høyer, Mhalla, and Perdrix also proved that any graph state can be prepared with a constant-depth Clifford circuit [40].

1.2 Organization

The remainder of the paper is organized as follows. Section 2 introduces notation and terminology for stabilizer states and the Clifford group. Section 3 reviews graph states and the combinatorial characterizations of relevant local Clifford operations and measurements from previous work. Section 4 presents the combinatorial characterization of two-qubit Clifford operations and proves Theorem 3. Section 5 discusses rank-width and its known properties, and also provides the proofs for Lemmas 6 and 8. Finally, Sections 6, 7, and 8 are dedicated to proving Theorems 7, 4, and 9, respectively.

2 Stabilizer states and Clifford group

2.1 Stabilizer states

Definition 10 (Pauli group)**.** The Pauli matrices are defined as

$$I_2 := \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \qquad X := \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad Y := \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \qquad Z := \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

The Pauli group $\mathcal{G}_n$ is defined as

$$\mathcal{G}_n := \{\alpha h_1 \otimes h_2 \otimes \cdots \otimes h_n \mid \alpha \in \{\pm 1, \pm i\}, \quad h_j \in \{I_2, X, Y, Z\}\ \forall j \in \{1, 2, \ldots, n\}\}.$$

Definition 11 (Stabilizer state)**.** Let S be a subset of $\mathcal{G}_n$ of size n satisfying

1. $\forall s \in S, \quad s^2 = I_2^{\otimes n}$.

2. $\forall s, t \in S, \quad st = ts.$

3. For $T \subseteq S, \quad \prod_{s \in T} s = \pm I_2^{\otimes n}$ only when $T = \varnothing$.

Then, there exists a one-dimensional subspace $\left\{|\psi\rangle \in \mathbb{C}^{2^n} \mid s|\psi\rangle = |\psi\rangle \ \forall s \in S\right\}$. Here, the quantum state in the subspace is called a *stabilizer state that is stabilized by the stabilizer* $\langle S \rangle$. Furthermore, S is called *the stabilizer generator*.

From the condition $s^2 = I_2^{\otimes n}$ for stabilizer generators, a stabilizer $\langle S \rangle$ must be a subset of

$$\mathcal{G}_n^{\pm} := \left\{\alpha h_1 \otimes h_2 \otimes \cdots \otimes h_n \mid \alpha \in \{\pm 1\}, \quad h_j \in \{I_2, X, Y, Z\} \ \forall j \in \{1, 2, \ldots, n\}\right\}.$$

If $S = \{s_1, \ldots, s_n\} \subseteq \mathcal{G}_n^{\pm}$ is a stabilizer generator, then for any $v \in \{0,1\}^n$, $\{(-1)^{v_1} s_1, \ldots, (-1)^{v_n} s_n\} \subseteq \mathcal{G}_n^{\pm}$ is also a stabilizer generator for some stabilizer state. Stabilizers without the sign information are represented by $n \times 2n$ binary matrices.

Definition 12 (The binary representation of Pauli operators [44, 1])**.** There is an isomorphism between $\mathcal{G}_n / \langle i I_2^{\otimes n} \rangle$ and $(\mathbb{Z}/2\mathbb{Z})^{2n}$. For $n = 1$, we define an isomorphism as

$$\sigma_{00} := I_2 \longleftrightarrow \begin{bmatrix} 0 & 0 \end{bmatrix}, \quad \sigma_{10} := X \longleftrightarrow \begin{bmatrix} 1 & 0 \end{bmatrix}, \quad \sigma_{11} := Y \longleftrightarrow \begin{bmatrix} 1 & 1 \end{bmatrix}, \quad \sigma_{01} := Z \longleftrightarrow \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

For general n, we define an isomorphism as

$$\sigma_{x_1 z_1} \otimes \sigma_{x_2 z_2} \otimes \cdots \otimes \sigma_{x_n z_n} \longleftrightarrow \begin{bmatrix} x_1 & \cdots & x_n & z_1 & \cdots & z_n \end{bmatrix}.$$

For $a \in (\mathbb{Z}/2\mathbb{Z})^{2n}$, the corresponding operator in $\mathcal{G}_n / \langle i I_2^{\otimes n} \rangle$ is denoted by $\sigma_a := \sigma_{a_1 a_{n+1}} \otimes \cdots \otimes \sigma_{a_n a_{2n}}$.

For row vectors $a, b \in (\mathbb{Z}/2\mathbb{Z})^{2n}$, σ_a and σ_b commute if and only if

$$a \begin{bmatrix} O_n & I_n \\ I_n & O_n \end{bmatrix} b^T = 0$$

where O_n and I_n are the $n \times n$ zero matrix and identity matrix on the field $\mathbb{F}_2$. For a set $A = \{a_1, \ldots, a_n\} \subseteq (\mathbb{Z}/2\mathbb{Z})^{2n}$ of size n, a set of Pauli operators $\{\sigma_a \mid a \in A\}$ is a stabilizer generator if and only if

$$G \begin{bmatrix} O_n & I_n \\ I_n & O_n \end{bmatrix} G^T = O_n \qquad \text{and} \qquad \text{the rows of } G \text{ are linearly independent}$$

where $G \in \mathbb{F}_2^{n \times 2n}$ denotes a matrix whose j-th row is a_j [1]. Here, G is called a *generating matrix* of the stabilizer state.

In this study, stabilizer states with the same generating matrix are not distinguished since they are equivalent up to a Pauli operator.

Fact 13. For any two stabilizer states $|\psi\rangle$ and $|\varphi\rangle$ with the same generating matrices, there exists a Pauli operator $g \in \mathcal{G}_n$ such that $|\psi\rangle$ is equal to $g|\varphi\rangle$ up to a constant factor.

The proof of Fact 13 is shown in Appendix A.

2.2 Clifford group

Definition 14. Let $U(n)$ be a set of unitary matrices of size n. The *Clifford group* is defined as

$$\mathcal{C}_n := \left\{U \in U(2^n) \mid U \mathcal{G}_n U^\dagger = \mathcal{G}_n\right\}.$$

It is known that $\mathcal{C}_n$ is generated by the following three quantum gates

$$H := \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, \qquad S := \begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}, \qquad \mathrm{CZ} := \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

The set of *local Clifford operators* is the subset of $\mathcal{C}_n$ defined as

$$\mathcal{C}_1^{\otimes n} := \{C_1 \otimes \cdots \otimes C_n \mid C_i \in \mathcal{C}_1, \forall i \in \{1, \ldots, n\}\}.$$

A Clifford operator $C \in \mathcal{C}_n$ transforms a stabilizer state to another stabilizer state. Applying a Clifford operator $C \in \mathcal{C}_N$ transforms a stabilizer $s \in \mathcal{G}_n$ to $CsC^\dagger$. Indeed, X and Z are transformed as follows.

$$\begin{aligned} HXH^\dagger &= Z, & HZH^\dagger &= X, \\ SXS^\dagger &= Y, & SZS^\dagger &= Z \\ \mathrm{CZ}(X \otimes I_2)\mathrm{CZ}^\dagger &= X \otimes Z, & \mathrm{CZ}(I_2 \otimes X)\mathrm{CZ}^\dagger &= Z \otimes X, \\ \mathrm{CZ}(Z \otimes I_2)\mathrm{CZ}^\dagger &= Z \otimes I_2, & \mathrm{CZ}(I_2 \otimes Z)\mathrm{CZ}^\dagger &= I_2 \otimes Z. \end{aligned} \tag{1}$$

Let $\sigma_{f_C(a)} := C\sigma_a C^\dagger$. The following equality holds on $\mathcal{G}_n/\langle iI_2^{\otimes n}\rangle$

$$\sigma_{f_C(a+b)} = C\sigma_{a+b}C^\dagger = C\sigma_a\sigma_b C^\dagger = C\sigma_a C^\dagger C\sigma_b C^\dagger = \sigma_{f_C(a)}\sigma_{f_C(b)} = \sigma_{f_C(a)+f_C(b)}.$$

Hence, the function $f_C\colon (\mathbb{Z}/2\mathbb{Z})^{2n} \to (\mathbb{Z}/2\mathbb{Z})^{2n}$ is linear so that there exists a unique $2n \times 2n$ $\mathbb{F}_2$-matrix L_C satisfying $aL_C = f_C(a)$. From (1), we obtain the following $\mathbb{F}_2$-matrix representations,

$$L_H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \qquad L_S = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \qquad L_{\mathrm{CZ}} = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Note that all six invertible 2×2 matrices over $\mathbb{F}_2$ are generated by L_H and L_S as follows.

$$\begin{aligned} &L_I = L_H^2 = L_S^2 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad L_H = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad L_S L_H = \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \quad L_S = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}, \\ &L_H L_S L_H = L_S L_H L_S = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad L_H L_S = \begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}. \end{aligned} \tag{2}$$

Obviously, $L_{DC} = L_C L_D$ for any Clifford operations $C, D \in \mathcal{C}_n$.

In this study, Clifford operators with the same $\mathbb{F}_2$ matrix representation are not distinguished since they are equivalent up to a Pauli operator.

Fact 15. Let $C, D \in \mathcal{C}_n$ be Clifford operators. Then, $L_C = L_D$ if and only if there exists a Pauli operator $g \in \mathcal{G}_n$ such that C is equal to gD up to a constant factor.

The proof of Fact 15 is shown in Appendix A.

3 Graph state and combinatorial characterizations of local Clifford operations and measurements

For a finite set V, let $\binom{V}{2}$ denote a family of subsets of V of size exactly two. For a finite set V, and $E \subseteq \binom{V}{2}$, a pair (V, E) is called a *graph*. For a graph $G = (V, E)$, $V(G) := V$ and $E(G) := E$ are called *vertex set* and *edge set*, respectively. The *size of graph* G is defined as $|V(G)|$. For a graph G and $v \in V(G)$, the *neighborhood of* v *in* G is defined as $N_G(v) := \{w \in V \mid \{v, w\} \in E\}$.

Definition 16 (Graph state)**.** For a graph $G = (V = \{1, 2, \ldots, n\}, E)$, a *graph state* $|G\rangle$ is a stabilizer state defined by stabilizer generators

$$X_v \otimes \bigotimes_{w \in N_G(v)} Z_w, \qquad v \in V.$$

Here, for any 2×2 matrix U, $U_v := g_1 \otimes \cdots \otimes g_n$ where $g_v = U$ and $g_w = I_2$ for $w \in V \setminus \{v\}$.

The generating matrix of $|G\rangle$ is $\left[I \mid A\right]$ where $A \in \mathbb{F}_2^{n\times n}$ is the *adjacency matrix* of G, i.e., $A_{v,w} = 1$ if and only if $\{v, w\} \in E(G)$.

In Definition 1, the local Clifford operations are free to use. Hence, the two stabilizer states that are equivalent up to the local Clifford operations can be identified in our analysis.

Definition 17 (Local Clifford equivalence)**.** Stabilizer states $|\psi\rangle$ and $|\varphi\rangle$ are *local-Clifford (LC) equivalent* if there exists a local Clifford operator $C \in \mathcal{C}_1^{\otimes n}$ such that $C|\psi\rangle = |\varphi\rangle$.

The following lemma implies that it is sufficient to prepare graph states for preparing general stabilizer states.

Lemma 18 ([1])**.** *For any stabilizer state $|\psi\rangle$, there exists a graph G such that $|\psi\rangle$ and $|G\rangle$ are LC-equivalent.*

For the combinatorial characterization of LC-equivalence of two graph states, the local complementation is defined as follows.

Definition 19 (Local complementation)**.** The *local complementation* is an operation on a graph. For a graph $G = (V, E)$, and a vertex $v \in V$, the local complementation $\tau_v(G) = (V, E')$ is defined as

$$E' = E \triangle \binom{N_G(v)}{2}$$

where $\triangle$ denotes the symmetric difference. In other words, the local complementation at v leaves all adjacencies outside $N_G(v)$ unchanged and complements the subgraph induced by $N_G(v)$.

Two graphs G and H are said to be *local-complement equivalent* if G and H can be transformed to each other by a sequence of local complementations.

Figure 3 describes an example of the local complementation.

Lemma 20 (Local Clifford equivalence of graph states [1])**.** *For any graphs G and H, $|G\rangle$ and $|H\rangle$ are local-Clifford equivalent if and only if G and H are local-complement equivalent.*

Lemma 20 provides a combinatorial characterization of local Clifford operations on graph states. Next, we consider a combinatorial characterization of measurements. In quantum algorithms in Definition 1, general stabilizer states, which are not necessarily graph states, may be measured in the computational basis. Since any stabilizer state is LC-equivalent to some graph state, it is sufficient to consider Pauli measurements for graph states.

Lemma 21 (Pauli measurements for graph states [3])**.** *For any graph G, let $|G^+\rangle$ and $|G^-\rangle$ be two possible stabilizer states obtained by a Pauli measurement, i.e., in the eigenbasis of X, Y or Z, of a qubit in $|G\rangle$. Then, $|G^+\rangle$ and $|G^-\rangle$ are LC-equivalent to some common graph state $|H\rangle$. Here, H is obtained from G by some local complementations and deletion of the vertex that corresponds to the measured qubit.*

Lemma 21 implies that a Pauli measurement and an adaptive local Clifford operation for a graph state $|G\rangle$ give some graph state $|H\rangle$ deterministically. Furthermore, the graph H is obtained by local complementation and a vertex deletion for G. This observation implies the following lemma.

Definition 22 (Vertex minor [24, 10])**.** A graph H is a *vertex minor* of graph G if H is obtained from G by local complementations and vertex deletions.

Lemma 23 ([10, 3, 24])**.** *For any graphs G and H, $|H\rangle$ is obtained from $|G\rangle$ with probability 1 by a local Clifford operation followed by measurement of qubits in the computational basis and an adaptive local Clifford operation if and only if H is a vertex-minor of G.*

Lemmas 23 provides combinatorial characterizations of local Clifford operations and measurements in the computational basis. Dahlberg, Helsen, and Wehner showed that it is NP-hard to decide whether G is a vertex-minor of H for given graphs G and H [45].

Figure 3: Local complementation at the red top vertex transforms the graph on the left into the graph on the right, and vice versa.

4 Combinatorial characterization of two-qubit Clifford operations: Proof of Theorem 3

In this section, we consider a combinatorial characterization of two-qubit Clifford operations. For the combinatorial interpretation of the CZ operation, we define elementary edge-complementations as follows. The three transformations below were derived for graphical descriptions of CZ operations on stabilizer states in [9].

Definition 24 (Elementary edge-complementations)**.** For a graph $G = (V, E)$, an *elementary edge-complementation* is defined as any one of the following:

1. For $\{v, w\} \in \binom{V}{2}$, complement an edge $\{v, w\}$, i.e., add the edge if it does not exist, remove it if it does.
2. For $\{v, w\} \in \binom{V}{2}$, complement all edges between v and $N_G(w) \setminus \{v\}$.
3. For $\{v, w\} \in \binom{V}{2} \setminus E$, complement all edges in $\left\{\{x, y\} \in \binom{V}{2} \mid x \in N_G(v),\, y \in N_G(w),\, \{x, y\} \notin N_G(v) \cap N_G(w)\right\}$.

The second operation is called *the edge complementation between v and $N_G(w)$*. The third operation is called *the edge complementation between $N_G(v)$ and $N_G(w)$*.

It is easy to see that the above operations correspond to some two-qubit Clifford operations. The first operation corresponds to the CZ operation. Let $\zeta^{(1)}_{v,w}$ be a mapping on graphs corresponding to the edge complementation for $\{v, w\} \in \binom{V}{2}$. The second operation is equivalent to

$$\zeta^{(1)}_{v,w} \circ \tau_w \circ \zeta^{(1)}_{v,w} \circ \tau_w = \tau_w \circ \zeta^{(1)}_{v,w} \circ \tau_w \circ \zeta^{(1)}_{v,w}.$$

The third operation for $\{v, w\} \in \binom{V}{2} \setminus E$ is equivalent to

$$\zeta^{(1)}_{v,w} \circ \tau_v \circ \tau_w \circ \tau_v \circ \zeta^{(1)}_{v,w} = \zeta^{(1)}_{v,w} \circ \tau_w \circ \tau_v \circ \tau_w \circ \zeta^{(1)}_{v,w}$$

and swapping the labels of the vertices v and w. Note that $\tau_v \circ \tau_w \circ \tau_v = \tau_w \circ \tau_v \circ \tau_w$ for $\{v, w\} \in E$ is called *the pivot operation*, which is the edge complementation between $N_G(v)$ and $N_G(w)$ and the swapping of the labels of v and w [29, 33]. Indeed, the three operations in Definition 24 explain all two-qubit Clifford operations.

We use a detailed version of Lemma 18 for the proof of Theorem 3.

Lemma 25 ([9])**.** *For any stabilizer state $|\psi\rangle$, there exist a graph $G = (V, E)$ and $A, B \subseteq V$ such that $A \cap B = \varnothing$, G has no edge between vertices in B, and $|G\rangle$ is equal to $\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w |\psi\rangle$ up to a Pauli operator.*

The following lemma was essentially proved in [9].

Lemma 26 ([9])**.** *For any graph G, the following equalities hold up to a Pauli operator:*

$$H_b \mathrm{CZ}_{a,b} H_b |G\rangle = \left|\zeta^{(2)}_{a,b}(G)\right\rangle \qquad \forall \{a, b\} \in \binom{V}{2}$$

$$(H_a \otimes H_b)\mathrm{CZ}_{a,b}(H_a \otimes H_b) |G\rangle = \left|\zeta^{(3)}_{a,b}(G)\right\rangle \qquad \forall \{a, b\} \in \binom{V}{2} \setminus E(G)$$

where $\zeta^{(2)}_{a,b}$ denotes the edge complementation between a and $N_G(b)$, and $\zeta^{(3)}_{a,b}$ denotes the edge complementation between $N_G(a)$ and $N_G(b)$.

From Lemmas 25 and 26, we obtain the combinatorial characterization of quantum algorithms with CZ-complexity one.

Theorem 27. *For any graphs G and H, graph states $|G\rangle$ and $|H\rangle$ can be transformed to each other by an arbitrary number of local Clifford operations and one* CZ *operation if and only if G and H can be transformed to each other by an arbitrary number of local complementations and a single elementary edge-complementation.*

Proof. From Lemmas 20 and 26, the direction $\Leftarrow$ is obvious. We will prove the other direction. Let $|\psi\rangle$ be a stabilizer state just before the CZ operation is performed in the transformation from $|G\rangle$ to $|H\rangle$. From Lemma 25, there exist a graph G' and $A, B \subseteq V$ such that $A \cap B = \varnothing$, G' has no edge between vertices in B and $\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w |\psi\rangle$ is equal to $|G'\rangle$ up to a Pauli operator. Then, some CZ operation is applied to a pair $\{a, b\} \in \binom{V}{2}$ of qubits of $|\psi\rangle$. Assume a and b are both not in B. Then, up to a Pauli operator,

$$\left(\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w\right) \mathrm{CZ}_{a,b} |\psi\rangle = \mathrm{CZ}_{a,b} \left(\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w\right) |\psi\rangle = \mathrm{CZ}_{a,b} |G'\rangle = \left|\zeta^{(1)}_{a,b}(G')\right\rangle$$

since the CZ gate and S gate commute. Assume a is not in B and b is in B. Then, up to a Pauli operator,

$$\begin{aligned}\left(\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w\right) \mathrm{CZ}_{a,b} |\psi\rangle &= \left(\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w\right) \mathrm{CZ}_{a,b} \left(\bigotimes_{v\in A} S_v^\dagger \bigotimes_{w\in B} H_w\right) |G'\rangle \\ &= H_b \mathrm{CZ}_{a,b} H_b |G'\rangle = \left|\zeta^{(2)}_{a,b}(G')\right\rangle.\end{aligned}$$

Finally, assume a and b are both in B. Then, up to a Pauli operator,

$$\begin{aligned}\left(\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w\right) \mathrm{CZ}_{a,b} |\psi\rangle &= \left(\bigotimes_{v\in A} S_v \bigotimes_{w\in B} H_w\right) \mathrm{CZ}_{a,b} \left(\bigotimes_{v\in A} S_v^\dagger \bigotimes_{w\in B} H_w\right) |G'\rangle \\ &= (H_a \otimes H_b) \mathrm{CZ}_{a,b} (H_a \otimes H_b) |G'\rangle = \left|\zeta^{(3)}_{a,b}(G')\right\rangle.\end{aligned}$$

Since $|G\rangle$ and $|G'\rangle$ are LC-equivalent, G and G' are local-complement equivalent. Then, the transformations of graph $G \to G' \to \zeta^{(i)}_{a,b}(G') \to H$ for some $i \in \{1, 2, 3\}$ are realized by an arbitrary number of local complementations and a single elementary edge-complementation. □

From Lemmas 20, 23 and Theorem 27, the CZ-complexity of graph state is now characterized in a combinatorial way.

Definition 28 (The edge-complementation-complexity of graphs)**.** Let $\mathcal{A}$ be an algorithm acting on a graph that consists of the following operations.

1. Deletion of a vertex.
2. Local complementation.
3. Elementary edge-complementation.

The *edge-complementation-complexity of* $\mathcal{A}$ is defined as the number of elementary edge-complementations in $\mathcal{A}$. The *edge-complementation-complexity (EC-complexity)* $\Gamma_{\mathrm{EC}}(G)$ *of a graph* G is defined as the minimum edge-complementation-complexity of $\mathcal{A}$ that generates G with n vertices from the empty graph with $n + s$ vertices for some $s \geq 0$.

Theorem 29 (Equivalent to Theorem 3)**.** *For any graph G, $\Gamma_{\mathrm{CZ}}(|G\rangle) = \Gamma_{\mathrm{EC}}(G)$.*

In the rest of this paper, we mainly consider $\Gamma_{\mathrm{EC}}(G)$ rather than $\Gamma_{\mathrm{CZ}}(|G\rangle)$ since it is easier to analyze.

5 Rank-width

The rank-width is a complexity measure of a graph [17, 18]. The rank-width is defined using a cut-rank.

Definition 30 (Cut-rank)**.** For a graph $G = (V, E)$ and $S \subseteq V$, the *bipartite adjacency matrix* $A_{S,V\setminus S} \in \mathbb{F}_2^{|S|\times|V\setminus S|}$ is a matrix whose rows and columns are indexed by $s \in S$ and $t \in V \setminus S$, respectively. An (s,t)-entry of $A_{S,V\setminus S}$ is 1 if and only if $\{s,t\} \in E$. The *cut-rank* of $S \subseteq V$ for G is defined as $\mathrm{cutrank}_G(S) := \mathrm{rank}_{\mathbb{F}_2}(A_{S,V\setminus S})$. The cut-rank of the empty set is defined as zero.

Obviously, the cut-rank is symmetric, i.e., $\mathrm{cutrank}_G(V \setminus S) = \mathrm{cutrank}_G(S)$ for any $S \subseteq V$. The cut-rank plays an important role in this work since it is invariant under the local complementations.

Lemma 31 ([25, 33])**.** *For any $G = (V, E)$, $S \subseteq V$ and $v \in V$,* $\mathrm{cutrank}_G(S) = \mathrm{cutrank}_{\tau_v(G)}(S)$.

Proof. Let $A_{S,V\setminus S} \in \mathbb{F}_2^{|S|\times|V\setminus S|}$ be the bipartite adjacency matrix of G. Without loss of generality, we can assume $v \in S$. Then, the bipartite adjacency matrix for $\tau_v(G)$ is obtained from $A_{S,V\setminus S}$ by adding the v-th row to all rows that correspond to a neighborhood of v. This transform does not change the rank of the bipartite adjacency matrix. □

The rank-width is defined as follows.

Definition 32 (Rank-decomposition and rank-width [17, 18])**.** Let $G = (V, E)$ be a graph. A *subcubic tree* is a tree whose maximum degree is at most three. The *rank-decomposition* of G is a pair (T, L) of a subcubic tree T with at least two vertices and a bijection L from V to a set of all leaves of T. For a rank-decomposition $(T = (W, F), L)$ and an edge $e \in F$, $T - e$ determines a partition of leaves of T, equivalently a partition of V determined by L. The *width of* $e \in F$ is $\mathrm{cutrank}_G(S)$ where $S, V \setminus S$ is the partition of V determined by e. The *width of the rank-decomposition* T is the maximum of the width among all edges in T. The *rank-width* $\mathrm{rw}(G)$ of G is the minimum width over all rank-decompositions of G. When G has at most one vertices, G has no rank-decomposition. In this case, the rank-width of G is defined as zero.

Obviously, the rank-width of an n-vertex graph is at most $\lceil n/3 \rceil$ since any rank-decomposition containing a node whose removal induces a balanced tripartition of the leaves has width at most $\lceil n/3 \rceil$.

Fact 33. For a graph $G = (V, E)$, let $G[W]$ denote the induced subgraph in G by $W \subseteq V$. In other words, $G[W] := (W, E \cap \binom{W}{2})$. The cut-rank and rank-width are monotone with respect to induced subgraphs, i.e.,

- For any $S \subseteq W \subseteq V$, $\mathrm{cutrank}_{G[W]}(S) \leq \mathrm{cutrank}_G(S)$.
- For any $W \subseteq V$, $\mathrm{rw}(G[W]) \leq \mathrm{rw}(G)$.

The rank-width can be computed in time $\mathrm{poly}(n)2^n$ [35]. There also exist fixed parameter-tractable algorithms computing the rank-width [36, 37, 38, 39].

Here, we provide the proof of Lemma 6 using the information-theoretic counting argument.

Lemma 6. *There exists a positive constant c such that for any positive integer n and $r \leq \lceil n/3 \rceil$, there exists a graph G with n vertices and rank-width at most r such that $\Gamma_{\mathrm{CZ}}(|G\rangle) \geq crn/\log n$.*

Proof. Any bipartite graph with r vertices at one side and $n-r$ vertices at the other side has rank-width at most r since any bipartition has cut-rank at most r. Hence, the number of graphs with n vertices and rank-width at most r is at least $2^{r(n-r)}$. All graph states have different generating matrices. Since quantum algorithms in Definition 1 have to generate $|G\rangle$ with probability 1, the quantum algorithm generates $|G\rangle$ when the all measurement outcomes for the s working qubits are zero. In this case, we can consider a corresponding Clifford circuit with the projectors $|0\rangle\langle 0|$ for the s working qubits. The output of the Clifford circuit with the projectors must be proportional to $|G\rangle$. In the following, we count the number of Clifford circuits with the projectors. There are 324 two-qubit Clifford operators acting on the generating matrices from Figure 7 (b) in Appendix B.

Non-working qubits that do not appear in any two-qubit operation must be $|+\rangle$. Then, the number of Clifford circuits including at most t two-qubit Clifford gates is at most

$$\left(1+324\binom{n+s}{2}\right)^t.$$

We can assume without loss of generality that all working qubits "connect" to some non-working qubit by two-qubit Clifford operations. This observation implies $s \leq t \leq \binom{n}{2}$. If t two-qubit Clifford gates are sufficient for all graph states with n vertices and rank-width r,

$$\left(1+324\binom{n+n^2}{2}\right)^t \geq 2^{r(n-r)} \geq 2^{r\lfloor \frac{2}{3}n \rfloor}$$

since $r \leq \lceil n/3 \rceil$. Then, we obtain

$$t \geq \frac{r}{\log_2\left(1+324\binom{n+n^2}{2}\right)} \left\lfloor \frac{2}{3}n \right\rfloor.$$

□

Here, we also provide the proof of Lemma 8 using the rank-width of random sparse graph analyzed in [19].

Lemma 8. *There exists a constant $c \geq 1$ such that for any r' there exist n' and $r \geq r'$ such that for any $n \geq n'$, there exists a connected graph G with n vertices and rank-width r satisfying $\Gamma_{\mathrm{CZ}}(|G\rangle) \leq n-1+c(r-1)$.*

Proof of Lemma 8. Let $G(n,p)$ denote the Erdős–Rényi random graph. The rank-width of $G(n,c/n)$ with constant $c>1$ was analyzed in [19].

Lemma 34 (Rank-width of random sparse graph [19])**.** *For any constant $c>1$, there exist constants $d>0$ and $e>0$ such that $G(n,c/n)$ contains a connected subgraph H satisfying $|V(H)| \geq dn$ and $\mathrm{rw}(H) \geq en$ asymptotically almost surely.*

Here, $G(n,c/n)$ contains at most $(c/2)n-1$ edges with probability asymptotically $1/2$. Hence, for $c>1$,

$$\Gamma_{\mathrm{EC}}(H) \leq (c/2)n-1 = dn-1+(c/2-d)n \leq |V(H)|-1+[(c/2-d)/e]\,\mathrm{rw}(H). \tag{3}$$

holds with probability asymptotically at least $1/2$.

Then, fix $c>1$ arbitrarily. For any r', let n' be a minimum integer such that there exists a connected graph H satisfying (3), $|V(H)|=n'$ and $\mathrm{rw}(H) \geq r'$. Note that such graph H always exists since asymptotically at least half of H in Lemma 34 satisfy (3) and have linear rank-width. Adding a new vertex of degree one to H gives a new graph H' such that $|V(H')| = |V(H)|+1$, $\Gamma_{\mathrm{EC}}(H') \leq \Gamma_{\mathrm{EC}}(H)+1$ and $\mathrm{rw}(H') = \mathrm{rw}(H)$. Hence, (3) still holds for H'. This argument implies that for any r', there exist n' and $r \geq r'$ such that for any $n \geq n'$ there exists a graph H with n vertices and rank-width r satisfying $\Gamma_{\mathrm{EC}}(H) \leq n-1+[(c/2-d)/e]r$. □

Determining the minimum value of the constant $(c/2-d)/e$ is of interest, particularly whether it can attain the lower bound of 1 given in Theorem 7.

6 Lower bound: Proof of Theorem 7

In this section, we prove Theorem 7.

Theorem 7. *For any connected graph G with $n \geq 2$ vertices and rank-width at least r, $\Gamma_{\mathrm{CZ}}(|G\rangle) \geq n+r-2$.*

The following lemma states that the cut-rank is unchanged by an elementary edge-complementation that does not cross the cut, and is changed by at most one by an elementary edge-complementation that crosses the cut.

Lemma 35. *For a graph $G = (V, E)$ and a pair $\{v, w\} \in \binom{V}{2}$ of vertices, let G' be a graph obtained by an elementary edge-complementation for $\{v, w\}$ from G. Then,*

- *For any $W \subseteq V$, $\mathrm{cutrank}_G(W) = \mathrm{cutrank}_{G'}(W)$ when $(v \in W \wedge w \in W) \vee (v \notin W \wedge w \notin W)$.*
- *For any $W \subseteq V$, $|\mathrm{cutrank}_G(W) - \mathrm{cutrank}_{G'}(W)| \leq 1$ when $(v \in W \wedge w \notin W) \vee (v \notin W \wedge w \in W)$.*

Proof. First, let us consider the case $v \in W \wedge w \in W$. Since the bipartite adjacency matrix $A_{W,V\setminus W}$ does not change by the edge complementation between v and w, its rank does not change. For the edge complementation between v and $N_G(w)$, the new bipartite adjacency matrix is obtained by adding the w-th row to the v-th row for $A_{W,V\setminus W}$. Hence, the rank does not change. For the edge complementation between $N_G(v)$ and $N_G(w)$ for vertices v and w non-adjacent in G, the new bipartite adjacency matrix is obtained by adding the v-th row to rows in $N_G(w) \cap W$ and adding the w-th row to rows in $N_G(v) \cap W$. Hence, the rank does not change. Similarly, for the case $v \notin W \wedge w \notin W$, the rank of the bipartite adjacency matrix does not change.

Next, let us consider the case $v \in W \wedge w \notin W$. For the edge complementation between v and w, the new bipartite adjacency matrix is obtained by flipping the (v, w)-entry. Hence, the rank changes by at most one. For the edge complementation between v and $N_G(w)$, the new bipartite adjacency matrix is obtained by flipping (v, u)-entry for $u \in N_G(w) \setminus W$. Hence, the rank changes by at most one. For the edge complementation between $N_G(v)$ and $N_G(w)$ for vertices v and w non-adjacent in G, the new bipartite adjacency matrix is obtained by adding the v-th row to rows in $N_G(w) \cap W$ and adding a row vector $1_{N_G(w)\setminus W}$ to rows in $N_G(v) \cap W$ where $1_{N_G(w)\setminus W}$ is a row vector whose $u \in V \setminus W$-th element is 1 if and only if $u \in N_G(w) \setminus W$. Hence, the rank changes by at most one. Similarly, for the case $v \notin W \wedge w \in W$, the rank of the bipartite adjacency matrix changes by at most one. □

Theorem 36 (Equivalent to Theorem 7)**.** *For any connected graph G with $n \geq 2$ vertices and rank-width at least r, $\Gamma_{\mathrm{EC}}(G) \geq n + r - 2$.*

Proof. Let $G = (V, E)$ be a connected graph with at least two vertices. Then, $\mathrm{rw}(G) \geq 1$ so that we assume $r \geq 1$ in the following. From the connectivity of G, $\Gamma_{\mathrm{EC}}(G) \geq n - 1$. Hence, Theorem holds for $r = 1$.

In the following, we assume $r \geq 2$ and show that if $\Gamma_{\mathrm{EC}}(G) \leq n + r - 3$, then, $\mathrm{rw}(G) \leq r - 1$. Let S be the set of vertices of size s that are deleted in an algorithm in Definition 28. Without loss of generality, we can assume that the vertices are deleted at the end of the algorithm. Let $G' = (V \cup S, E')$ be a graph just before the vertex deletion. Then, $G'[V] = G$. From Fact 33, it is sufficient to show $\mathrm{rw}(G') \leq r - 1$. In the following, we will show $\mathrm{rw}(G') \leq r - 1$. Let $k := \Gamma_{\mathrm{EC}}(G)$. Let $e_1, e_2, \ldots, e_k$ be a sequence of pairs of vertices of G' for which the elementary edge-complementations are applied in this order. Furthermore, let

$$F := \Big\{ e_i = \{v, w\} \mid i \in \{1, \ldots, k\} \quad v \text{ and } w \text{ belong to different connected components in a graph } (V \cup S, \{e_1, \ldots, e_{i-1}\}) \Big\}.$$

Let $T := (V \cup S, F)$. From the definition of F, T is a forest. From the connectivity of G, all vertices in V must belong to a common connected component of T. Since all other vertices that do not belong to the connected component including V does not affect the generation of the graph G, we can assume that there does not exist such vertices without loss of generality. In the following, we assume that $|F| = n + s - 1$, and T is a tree.

We use T to construct a rank-decomposition of G' of width at most $r - 1$ as follows. For each $v \in V \cup S$, let $e_{v_1}, \ldots, e_{v_{d_v}} \in F$ be edges of T incident to v where $v_1 < v_2 < \cdots < v_{d_v}$. A gadget H_v corresponding to the vertex v is defined as

$$V(H_v) := \{l_v\} \cup \{b_{v,i} \mid i = 1, \ldots, d_v\}$$
$$E(H_v) := \{\{l_v, b_{v,d_v}\}\} \cup \{\{b_{v,i}, b_{v,i+1}\} \mid i = 1, \ldots, d_v - 1\}$$

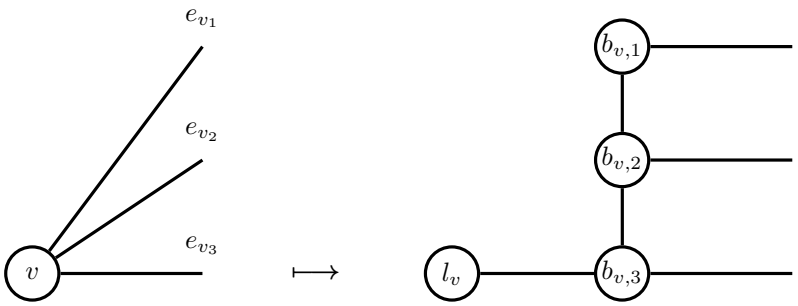


Figure 4: Construction of rank-decomposition on the basis of the tree T. Left: A vertex v in T. Right: A leaf of the rank-decomposition corresponding to v.

Figure 4 shows the correspondence between v and H_v. We now define the rank-decomposition H of G' as

$$V(H) := \bigcup_{v \in V \cup S} V(H_v)$$
$$E(H) := \bigcup_{v \in V \cup S} E(H_v) \cup \bigcup_{\{v,w\} \in F} \big\{\{b_{v,i}, b_{w,j}\} \mid v_i = w_j, i \in \{1, \dots, d_v\}, j \in \{1, \dots, d_w\}\big\}.$$

Then, it is obvious that H is a subcubic tree and the set of leaves of H is $\{l_v \mid v \in V \cup S\}$. Hence, $(H, L \colon v \mapsto l_v)$ is a rank-decomposition of G'.

We will show that the width of the rank-decomposition (H, L) is at most $r-1$. Each edge $e \in E(H)$ defines a partition $P_e := \{V_e, (V \cup S) \setminus V_e\}$ of vertices of G'. In the following, we will show $\mathrm{cutrank}_{G'}(V_e) \leq r-1$ for all $e \in E(H)$.

If $e = \{l_v, b_{v,d_v}\}$ for some $v \in V \cup S$, then $P_e = \{\{v\}, (V \cup S) \setminus \{v\}\}$. This implies $\mathrm{cutrank}_{G'}(V_e) = 1$.

If $e = \{b_{v,i}, b_{w,j}\}$ for some $v \neq w$, edges in $F \setminus \{v, w\}$ do not cross the cut V_e. The number of elementary edge-complementations that cross the cut V_e is at most $k - |F| + 1 = k - n - s + 2 \leq (n + r - 3) - n - s + 2 = r - s - 1 \leq r - 1$. From Lemmas 31 and 35, $\mathrm{cutrank}_{G'}(V_e) \leq r - 1$.

Finally, assume $e = \{b_{v,i}, b_{v,i+1}\}$, for some $v \in V \cup S$. Without loss of generality, we can assume $v \in V_e$. Before the elementary edge-complementation corresponding to $e_{v_{i+1}}$ is applied, v is the unique vertex in V_e that may connect to $(V \cup S) \setminus V_e$. This means that the cut-rank of V_e is at most one at this time. After $e_{v_{i+1}}$, the number of elementary edge-complementations that cross the cut V_e is at most $k - |F| \leq r - 2$. From Lemmas 31 and 35, $\mathrm{cutrank}_{G'}(V_e) \leq r - 1$.

From these observations, $\mathrm{cutrank}_{G'}(V_e) \leq r - 1$ for all $e \in E(H)$, which implies that the width of the rank-decomposition H is at most $r - 1$. □

7 Upper bound: Proof of Theorem 4

In this section, we present algorithms that generate G with EC-complexity $O(rn)$ and prove the following simplified version of Theorem 4.

Theorem 37 (Simplified version of Theorem 4)**.** *Let G be a graph with n vertices and rank-width at most $r \geq 1$. Then,*

$$\Gamma_{\mathrm{CZ}}(|G\rangle) \leq \begin{cases} \frac{5r^2-1}{4r} n + O(1) & \text{if } r \text{ is odd} \\ \frac{5r}{4} n + O(1) & \text{otherwise.} \end{cases}$$

The constant terms $O(1)$ with respect to n are calculated in Appendix C, which proves Theorem 4.

The following is the key lemma for the algorithms in this section.

Lemma 38. *For any graph G and $S \subseteq V(G)$, if $\mathrm{cutrank}_G(S) < |S|$, i.e., the rows of $A_{S,V \setminus S}$ are linearly dependent, there exists $v \in S$ such that $\Gamma_{\mathrm{EC}}(G) \leq \Gamma_{\mathrm{EC}}(G - v) + |S| - 1$.*

Algorithm 1: A framework of algorithms to generate a graph.

Input : A graph G
Output A graph G generated by the operations in Definition 28.
:
1 **Procedure** GenGraph(G)
2 **if** *G has at most two vertices* **then**
3 **return** a graph G generated in a trivial way;
4 Find a small set $S \subseteq V(G)$ with $\mathrm{cutrank}_G(S) < |S|$;
5 Find $v \in S$ such that the v-th row of $A_{S,V\setminus S}$ is spanned by other rows;
6 $G' \leftarrow$ GenGraph($G[V(G) \setminus \{v\}]$);
7 Add v to G' with EC-complexity at most $|S| - 1$ by the algorithm in Lemma 38;
8 **return** G';

Proof. From the linear dependence of rows of $A_{S,V\setminus S}$, there exists $v \in S$ and $T \subseteq S \setminus \{v\}$ such that the v-th row of $A_{S,V\setminus S}$ is a sum of rows of $A_{S,V\setminus S}$ in T. Let A be the adjacency matrix of G. Let A_v be the v-th row of A. Let $A_T := \sum_{w\in T} A_w \bmod 2$. Then, for any $u \in V \setminus S$, an u-th element of A_v and A_T are equal. Let

$$U := \{u \in T \mid u\text{-th elements of } A_v \text{ and } A_T \text{ are different}\}.$$

Then, we can generate G from $G - v$ by the following algorithm. First, for $w \in U$, we apply $\tau_w \circ \zeta^{(1)}_{v,w} \circ \tau_w$, and for $w \in T \setminus U$, we apply $\zeta^{(2)}_{v,w}$. Then, the neighborhood of v is correctly generated for all $(V \setminus S) \cup T$. Finally, we correct the neighborhood of v for $S \setminus (T \cup \{v\})$ by $\zeta^{(1)}_{v,w}$ if needed. The EC-complexity of this algorithm is at most $|S| - 1$. □

Algorithm 1 provides the algorithmic framework for generating graphs that we will consider in the rest of this section. The EC-complexity of Algorithm 1 is determined by the size of S in Line 4.

Definition 39. For a graph G, a subset $S \subseteq V(G)$ of vertices satisfying $\mathrm{cutrank}_G(S) < |S|$ is called a *dependent set*.

Indeed, S in Definition 39 is a dependent set of the independence system in Appendix C. In the following, we consider upper bounds on the size of the minimum dependent sets.

The following lemma provides an upper bound on the size of the minimum dependent set when a graph G has rank-width r.

Lemma 40. *For any positive integer r and a subcubic tree T with $n \geq 3r$ leaves, there exists an edge $e \in E(T)$ such that one of the two trees obtained by removing e from T contains $k \in \{r+1, r+2, \ldots, 2r\}$ leaves of T.*

Proof. We will show that the edge $e \in E(T)$ selected by Algorithm 2 satisfies the conditions of the lemma.

We first assume $n \geq 3r + 1$. Consider the tree containing the vertex b when Algorithm 2 outputs $\{a, b\}$. The tree has at most $2r$ leaves of T from the stopping condition of the algorithm. We will show that the tree has at least $r + 1$ leaves of T. If $a = a^*$, the tree with root a is the entire tree T, and since a has at most three children, the tree with root b contains at least $\lceil n/3 \rceil \geq \lceil (3r+1)/3 \rceil = r + 1$ leaves of T. If $a \neq a^*$, the tree with root a contains at least $2r + 1$ leaves of T from the stopping condition. Since a has at most two children, the tree with root b contains at least $\lceil (2r+1)/2 \rceil = r + 1$ leaves of T.

We assume $n = 3r$. Then, the above proof works except in the case where three subtrees incident to a^* each have exactly r leaves. In this case, by removing the edge $\{a^*, b\}$, the other tree rooted at a^* rather than b has $2r$ leaves. □

Lemma 40 implies an upper bound on the size of the minimum dependent sets.

Algorithm 2: Algorithm to find an edge that satisfies the condition in Lemma 40.

Input : r: A positive integer. T: A subcubic tree with $n \geq 3r$ leaves.
Output An edge e in T that satisfies the condition in Lemma 40.
:

Procedure FindEdge(T, r)
  Procedure aux(T, a)
    $(T', b) \leftarrow$ The rooted tree obtained by removing a from T that has the largest number of leaves where the root b is a child of a in T;
    if *the number of leaves in* $T' \leq 2r$ **then**
      return $\{a, b\}$;
    return aux(T', b);
  $a^* \leftarrow$ Arbitrary non-leaf vertex of T;
  return aux(T, a^*);

Corollary 41. *For any graph G with n vertices and rank-width $r \leq n/3$, there exists $S \subseteq V(G)$ such that $|S| \in \{r+1, \ldots, 2r\}$ and* $\mathrm{cutrank}_G(S) \leq r$.

Hence, if $n \geq 3r$, there exists a vertex $v \in V(G)$ such that G can be generated from $G - v$ with EC-complexity at most $2r - 1$. This argument implies that for any fixed r, the EC-complexity of graphs with n vertices and rank-width r is at most $(2r-1)n + O(1)$. In the following, we will improve the constant factor $2r - 1$.

First, we show the size of a dependent set S can be reduced while maintaining $\mathrm{cutrank}_G(S) < |S|$.

Lemma 42. *Let $S \subseteq V(G)$ and $k = \mathrm{cutrank}_G(S)$. Assume $k < |S|$. Then, any $T \subseteq S$ of size at least $\left\lceil \frac{k+|S|+1}{2} \right\rceil$ satisfies* $\mathrm{cutrank}_G(T) < |T|$.

Proof. If $|S| - k \in \{1, 2\}$, $\left\lceil \frac{k+|S|+1}{2} \right\rceil = |S|$. Hence, $T = S$ and the condition is satisfied. Assume $|S| - k > 2$. For any $v \in S$, let $S' = S \setminus \{v\}$. $A_{S', V \setminus S'}$ is obtained from $A_{S, V \setminus S}$ by deleting v-th row and adding v-th column. Hence, $\mathrm{rank}_{\mathbb{F}_2}(A_{S', V \setminus S'}) \leq \mathrm{rank}_{\mathbb{F}_2}(A_{S, V \setminus S}) + 1$. In the process $S \to S'$, the size of the subset decreases by one while the cutrank of the subset increases by at most one. Then,

$$0 < |S| - \mathrm{cutrank}_G(S) - 2 \leq |S'| - \mathrm{cutrank}_G(S') \leq |S| - \mathrm{cutrank}_G(S).$$

We can repeatedly apply this process until $|S| - \mathrm{cutrank}_G(S) > 2$. □

Corollary 41 and Lemma 42 imply the following improved upper bound on the size of the minimum dependent sets.

Corollary 43. *For any graph G with n vertices and rank-width $r \leq n/3$, there exists $S \subseteq V(G)$ such that $|S| \leq \left\lceil \frac{3r+1}{2} \right\rceil$ and* $\mathrm{cutrank}_G(S) < |S|$.

Corollary 43 implies that for any fixed r, the EC-complexity of a graph with n vertices and rank-width r is at most $\left\lceil \frac{3r-1}{2} \right\rceil n + O(1)$.

We can further improve the constant factor. Let

$$C(n, r) := \max_{G\colon n\text{-vertex graph with rank-width at most } r} \Gamma_{\mathrm{EC}}(G).$$

Lemma 44. *For odd r and any $n \geq 3r$, there exists $k \in \{1, 2, \ldots, r\}$ such that*

$$C(n, r) \leq \frac{5r^2 - 1}{4r} k + C(n - k, r).$$

For even r and any $n \geq 3r$, there exists $k \in \{1, 2, \ldots, r\}$ such that

$$C(n, r) \leq \frac{5r}{4} k + C(n - k, r).$$

Proof. Let G be a graph with n vertices and rank-width at most r that satisfies $\Gamma_{\mathrm{EC}}(G) = C(n,r)$. There exists a set $S \subseteq V(G)$ with $|S| \in \{r+1, \ldots, 2r\}$ and $\mathrm{cutrank}_G(S) \leq r$. There also exists $v_1 \in S$ such that G can be generated from $G' = G[V(G) \setminus \{v_1\}]$ with EC-complexity at most $\left\lceil \frac{|S|+r-1}{2} \right\rceil$. If $|S| \geq r+2$, G' has a vertex set $S' = S \setminus \{v_1\}$ with $\mathrm{cutrank}_{G'}(S') \leq \mathrm{cutrank}_G(S) \leq r < |S'|$. Hence, there exists $v_2 \in S'$ such that G' can be generated from $G'[V(G') \setminus \{v_2\}]$ with EC-complexity at most $\left\lceil \frac{|S|+r-2}{2} \right\rceil$. By repeating this argument, there exist vertices $v_1, \ldots, v_{|S|-r}$ such that G can be generated from $G[V(G) \setminus \{v_1, \ldots, v_{|S|-r}\}]$ with EC-complexity per vertex at most

$$\frac{1}{|S|-r} \sum_{k=1}^{|S|-r} \left\lceil \frac{|S|+r-k}{2} \right\rceil = \frac{1}{|S|-r} \sum_{k=r+1}^{|S|} \left\lceil \frac{k+r-1}{2} \right\rceil. \tag{4}$$

Since (4) is an average of $|S|-r$ integers, this is maximized when $|S| = 2r$. In this case, if r is odd, (4) is

$$\begin{aligned} \frac{1}{r} \sum_{k=r+1}^{2r} \left\lceil \frac{k+r-1}{2} \right\rceil &= \frac{1}{r} \left(r + (r+1) + (r+1) + (r+2) + (r+2) + \cdots + \frac{3r-1}{2} + \frac{3r-1}{2} \right) \\ &= \frac{5r^2-1}{4r}, \end{aligned}$$

and if r is even,

$$\begin{aligned} \frac{1}{r} \sum_{k=r+1}^{2r} \left\lceil \frac{k+r-1}{2} \right\rceil &= \frac{1}{r} \left(r + (r+1) + (r+1) + (r+2) + (r+2) + \cdots + \frac{3r-2}{2} + \frac{3r-2}{2} + \frac{3r}{2} \right) \\ &= \frac{5r}{4}. \end{aligned}$$

Hence,

$$C(n,r) = \Gamma_{\mathrm{EC}}(G) \leq \begin{cases} \frac{5r^2-1}{4r}(|S|-r) + C(n-|S|+r, r) & \text{if } r \text{ is odd} \\ \frac{5r}{4}(|S|-r) + C(n-|S|+r, r) & \text{otherwise.} \end{cases}$$

□

Lemma 44 implies that for any fixed r,

$$C(n,r) \leq \begin{cases} \frac{5r^2-1}{4r} n + O(1) & \text{if } r \text{ is odd} \\ \frac{5r}{4} n + O(1) & \text{otherwise} \end{cases}$$

and hence Theorem 37 is proved. The constant terms $O(1)$ with respect to n are calculated in Appendix C, which completes the proof of Theorem 4.

8 Interval graphs and circle graphs

8.1 Definitions of graph classes

Definition 45 (Interval graph and circle graph)**.** Let $V = \{v_1, v_2, \ldots, v_n\}$. A *double occurrence word* over the alphabet V is a word in which every letter in V appears exactly twice. The *interval graph* $G = (V, E)$ for a double occurrence word m is defined as

$$E = \left\{ \{a,b\} \in \binom{V}{2} \,\middle|\, a \text{ and } b \text{ appear in } m \text{ in the order } abba \text{ or } abab \right\}.$$

The *circle graph* $G = (V, E)$ for a double occurrence word m is defined as

$$E = \left\{ \{a,b\} \in \binom{V}{2} \,\middle|\, a \text{ and } b \text{ appear in } m \text{ in the order } abab \right\}.$$

Algorithm 3: Interval graph preparation.

Input : A double occurrence word m over $\{v_1, v_2, \ldots, v_n\}$.
Output : An interval graph defined by m generated by the operations in Definition 28.

Procedure GenIntervalGraph(m)
  Prepare a vertex z;
  Prepare vertices $\{u_i\}_{1,\ldots,n}$;
  for $t = 1, \ldots, 2n-2$ **do**
    if *the t-th element of m is the first appearance in m* **then**
      Local complementation on z;
      Edge complementation between $u_{p_1^{-1}(t)}$ and z (elementary edge-complementation);
      Local complementation on z;
    else
      Edge complementation between $u_{p_2^{-1}(t)}$ and z (elementary edge-complementation);
  Delete z;
  return $\{u_1, \ldots, u_n\}$;

It is known that the classes of interval graphs and circle graphs have unbounded rank-width [46, 17].

A circle graph defined by a double occurrence word m is an intersection graph of a *chord diagram* determined by m. Figure 5 shows an example of a double occurrence word and a corresponding chord diagram. A circle graph for m has an edge $\{v, w\}$ if and only if the chords for v and w intersect. Note that the class of circle graphs is closed under vertex-minors. The class of circle graphs, in the context of vertex-minors and rank-width, is analogous to the class of planar graphs in the context of graph-minors and branch-width.

Theorem 46 (The grid minor theorem for vertex-minors [47])**.** *For any circle graph H, there exists an integer r such that every graph with rank-width at least r contains a vertex-minor isomorphic to H.*

Hence, the class of circle graphs plays a key role within vertex-minor theory [22].

8.2 Interval graphs

For a double occurrence word m and $a \in \{1, \ldots, n\}$, let $p_1(a)$ and $p_2(a)$ denote the positions of the first and second appearances of v_a in m. For example, $V = \{v_1, v_2, v_3\}$, and $m = v_3v_2v_2v_1v_3v_1$,

$$p_1(1) = 4, \qquad p_2(1) = 6, \qquad p_1(2) = 2, \qquad p_2(2) = 3, \qquad p_1(3) = 1, \qquad p_2(3) = 5.$$

Theorem 47. *For any interval graph G with n vertices, Algorithm 3 generates G with EC-complexity $2n-2$.*

Proof. The EC-complexity of Algorithm 3 is obviously $2n-2$. The correctness of Algorithm 3 is obvious if t runs from 1 to $2n$. We can omit the case $t = 2n$ since the $(2n)$-th element of m is the second appearance in m. We can also omit the case $t = 2n-1$ since if the $(2n-1)$-th element of m is the second appearance in m, we can simply ignore this, and if the $(2n-1)$-th element of m is the first appearance, then the $(2n-1)$-th and $(2n)$-th elements are equal and represent an isolated vertex in G. □

Although Algorithm 3 uses the single working vertex z, we can use u_a in place of z where $a = \arg\max p_1(a)$ without any overhead. Hence, the EC-complexity $2n-2$ is achievable without working vertices.

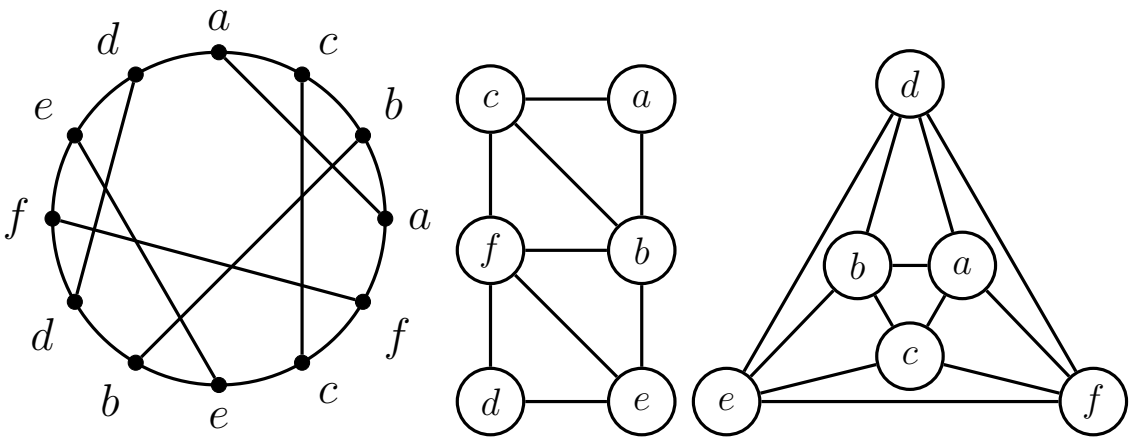


Figure 5: Left: The chord diagram of a double occurrence word acbafcebdfed. Middle: The circle graph. Right: The tour graph.

8.3 Circle graphs

Recently, Davies and Jena proved that for any circle graph G with n vertices and the chromatic number k, $\Gamma_{\mathrm{CZ}}(|G\rangle) \leq 2n\lceil \log_2 k \rceil$ [20, 21]. Their algorithm is a simple divide-and-conquer algorithm. In this section, we show an alternative algorithm based on the algorithmic framework in Algorithm 1.

For a double occurrence word m over V, we define a *tour graph* that is a connected (not necessarily simple) 4-regular graph that has the Eulerian cycle visiting the vertices in the order in m. Figure 5 shows an example of a chord diagram, a circle graph and a tour graph corresponding to a double occurrence word. A local complementation on a vertex v in a circle graph G corresponds to a reversion of the subsequence in m between two occurrences of v. The corresponding transformation of an Eulerian cycle in a tour graph is called the *κ-transformation* [30]. Kotzig proved that the κ-transformation is strong enough to transform an arbitrary Eulerian cycle of a connected 4-regular graph into another arbitrary Eulerian cycle.

Lemma 48 ([30])**.** *Let G be a connected 4-regular graph. Let m and m' be arbitrary Eulerian cycles of G. Then, m can be transformed to m' by a sequence of κ-transformations.*

Lemma 48 implies that the tour graph for G without a specific choice of Eulerian cycle represents the class of circle graphs local-complement equivalent to G.

Theorem 49. *For any circle graph G with n vertices, $\Gamma_{\mathrm{EC}}(G) \leq 2\lfloor \log_3(n+1) \rfloor (n-1) \leq 1.262 \cdot (n-1)\log_2(n+1)$. There exists an algorithm within the algorithmic framework in Algorithm 1 satisfying the upper bound.*

Proof. We will show that for any circle graph G with n vertices, there exists a circle graph G' that is local-complement equivalent to G and includes a vertex v of degree at most $2\lfloor \log_3(n+1) \rfloor$. This implies $\Gamma_{\mathrm{EC}}(G) \leq 2\lfloor \log_3(n+1) \rfloor (n-1)$. The algorithm falls within the algorithmic framework in Algorithm 1 since $S = \{v\} \cup N_{G'}(v)$ has a cutrank at most $|S| - 1$.

Let $G = (V, E)$ be a circle graph and $T = (V, F)$ be a tour graph of G. Since T is a 4-regular graph, T has a cycle. Let C be a smallest cycle in T and $v \in V(C)$ be an arbitrary vertex in C. Here, a self-loop and double edges are regarded as cycles of length one and two, respectively. Let T' be the connected component in $H = (V, F - E(C))$ including v. Let m' be an arbitrary Eulerian cycle on T' starting and ending at v. There exists an Eulerian cycle on $T'' = (V \setminus (V(T') \setminus V(C)), F - E(T'))$ since T'' is connected and all vertices have even degrees. Here, T'' is connected since every connected component of H contains at least one vertex of C. Let m'' be an arbitrary Eulerian cycle on T'' starting and ending at v. Let m be the Eulerian cycle on T obtained by the concatenation of m' and m''. Let G' be the circle graph corresponding to the Eulerian cycle m on T. From Lemma 48, G' is local-complement equivalent to G. In the Eulerian cycle for G', a vertex $w \in V \setminus \{v\}$ alternates with v if and only if w is included in both

m' and m''. This implies that the neighborhood of v on G' is a subset of $V(C) \setminus \{v\}$. Hence, the degree of v is at most $|V(C)| - 1$ on G'.

From the Moore bound [48], any 4-regular graph with n vertices and girth at least g must satisfy

$$n \geq \begin{cases} 2 \cdot 3^{\frac{g-1}{2}} - 1 & \text{if } g \text{ is odd} \\ 3^{\frac{g}{2}} - 1 & \text{if } g \text{ is even.} \end{cases}$$

Hence, any graph with n vertices and girth g satisfies $g \leq 2\lfloor \log_3(n+1) \rfloor + 1$. This implies that the degree of v is at most $2\lfloor \log_3(n+1) \rfloor$ on G'. Hence, the EC-complexity of G is at most $2\lfloor \log_3(n+1) \rfloor (n-1)$. □

Davies and Jena proved that for any circle graph G that is local-complement equivalent to a bipartite graph, the EC-complexity of G is at most $2n$ using a single auxiliary vertex and $3(n-1)$ without using an auxiliary vertex [20, 21]. We will prove that the algorithmic framework in Algorithm 1 gives the EC-complexity $2n$ without using an auxiliary vertex. Every connected 4-regular planar graph has an A-trail, which is an Eulerian cycle without straight-ahead transitions [30]. Furthermore, de Fraysseix and Ossona de Mendez proved that an A-trail always represents a bipartite circle graph, and vice versa [49].

Lemma 50 ([30, 49])**.** *A circle graph G is local-complement equivalent to a bipartite graph if and only if G has a planar tour graph.*

The girth of a 4-regular planar graph is upper bounded by three, which is much better than the Moore bound for general 4-regular graphs $O(\log n)$.

Lemma 51. *For any circle graph G with n vertices that is local-complement equivalent to a bipartite graph, $\Gamma_{\mathrm{EC}}(G) \leq 2(n-1)$. There exists an algorithm within the algorithmic framework in Algorithm 1 satisfying the upper bound.*

Proof. The proof is almost the same as the proof of Theorem 49. Let G be a circle graph with n vertices that is local-complement equivalent to a bipartite graph. From Lemma 50, G has a planar tour graph T. Let C be a smallest cycle of T.

We will show that the size of C is at most three. From Euler's formula, a planar graph with v vertices, e edges and f faces satisfies

$$v - e + f = 2.$$

For 4-regular graph, $4v = 2e$. Let g be the girth of the planar graph. Then, $fg \leq 2e$. It implies

$$g \leq \frac{2e}{f} = \frac{4v}{2 + e - v} = \frac{4v}{2 + v} < 4.$$

Hence, the girth of a 4-regular planar graph is at most three.

This implies that there exists a graph G' that is local-complement equivalent to G and includes a vertex of degree at most two. When we remove v in T, we can choose one of the two transitions of the Eulerian cycle at v that correspond to G' and $\tau_v(G')$. The tour graphs of $G' - v$ and $\tau_v(G') - v$ are described in Figure 6. Since the left resulting graph is planar, at least one of $G' - v$ and $\tau_v(G') - v$ has a planar tour graph. It implies $\Gamma_{\mathrm{EC}}(G) \leq 2(n-1)$. □

The same proof for Lemma 51 does work if T can be embedded in a surface of Euler genus 1.

Lemma 52. *For any circle graph G with n vertices whose tour graph can be embedded in a surface of Euler genus one, $\Gamma_{\mathrm{EC}}(G) \leq 2(n-1)$. There exists an algorithm within the algorithmic framework in Algorithm 1 satisfying the upper bound.*

Proof. The proof is almost the same as Lemma 51. We use generalized Euler's formula $v - e + f \geq 1$ instead. □

For general Euler genus, we obtain a similar upper bound.

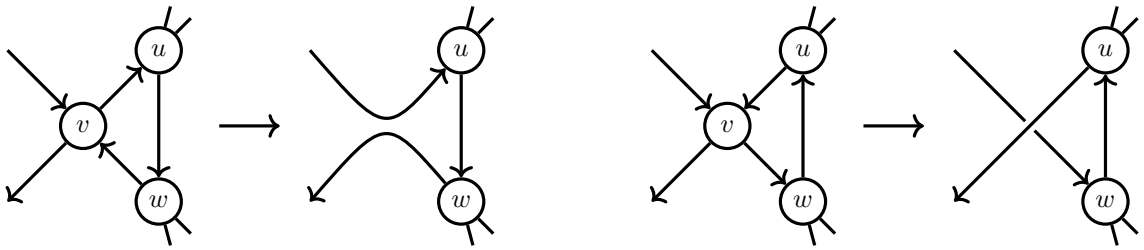


Figure 6: Two possible tour-graph reductions obtained by deleting v of degree two in the circle graph. In the left Eulerian cycle, the vertices are visited in the order v, u, w, v. In the right Eulerian cycle, the vertices are visited in the order v, w, u, v. The two resulting graphs are tour graphs of $G' - v$ and $\tau_v(G') - v$.

Lemma 53. *Let $\mathcal{C}$ be a class of circle graphs whose tour graph can be embedded in a surface of Euler genus ϵ. The EC-complexity of graphs in $\mathcal{C}$ is at most $3n + O(\epsilon \log \epsilon)$. There exists an algorithm within the algorithmic framework in Algorithm 1 satisfying the upper bound.*

Proof. The proof is almost the same as the proof of Lemma 51. By generalized Euler's formula,

$$2 - \epsilon \leq v - e + f.$$

For $v \geq \max\{0,\, \epsilon - 1,\, 5\epsilon - 9\} = \max\{0,\, 5\epsilon - 9\}$, $fg \leq 2e$ implies

$$g \leq \frac{2e}{f} \leq \frac{4v}{2 - \epsilon + e - v} = \frac{4v}{2 - \epsilon + v} = 5 - \frac{10 - 5\epsilon + v}{2 - \epsilon + v} < 5.$$

Hence, the girth of T is at most four if $n \geq \max\{0,\, 5\epsilon - 9\}$. From Theorem 49, EC-complexity of a circle graph with $5\epsilon - 10$ vertices is $O(\epsilon \log \epsilon)$. This implies the upper bound $3n + O(\epsilon \log \epsilon)$. □

Lemma 50 implies the equivalence of the planarity of tour graph and the bipartiteness of circle graph up to the local complementation. It would be interesting to see how the Euler genus of tour graph is reflected to properties of corresponding local-complement equivalence class of circle graphs.

Algorithm 3 for interval graphs is the unique graph generation algorithm in this paper that is not included within the algorithmic framework in Algorithm 1. It is an open question whether Algorithm 1 gives algorithms with EC-complexity $O(n)$ for interval graphs.

9 Conclusion

In this work, we studied the complexity of graph-state preparation in a general model of quantum algorithms that allows measurements in the computational basis, single-qubit Clifford operations, and two-qubit Clifford operations, and introduced the CZ-complexity, defined as the minimum number of two-qubit Clifford operations required to generate a target graph state. We first showed that this complexity measure is well justified, since every optimal graph-state preparation algorithm can be assumed to use only CZ operations as its two-qubit gates. We then established a combinatorial characterization of graph-state transformations and proved that the CZ-complexity of a graph state coincides with the edge-complementation-complexity of the corresponding graph. More precisely, a graph state $|G\rangle$ can be generated from $|H\rangle$ with CZ-complexity at most t if and only if G can be obtained from H by vertex deletions, local complementations, and at most t elementary edge-complementations. This yields a purely graph-theoretic description of graph-state preparation.

Based on this characterization, we related graph-state preparation complexity to rank-width. For an n-vertex graph of rank-width at most r, we obtained an upper bound of $O(rn)$ on the CZ-complexity, while for connected graphs of rank-width at least r we proved the lower bound $n + r - 2$. We also showed that these bounds are close to optimal in two complementary senses. There exist graphs of rank-width at most r whose CZ-complexity is $\Omega(rn/\log n)$, and there exist connected graphs for which the additive lower bound beyond $n - 1$ is tight up to a constant factor.

In particular, for connected graphs, rank-width one exactly characterizes the graph states with CZ-complexity $n-1$.

We also considered special graph classes with unbounded rank-width. For interval graphs and circle graphs, we presented preparation algorithms with CZ-complexities $O(n)$ and $O(n \log n)$, respectively. These examples show that although rank-width provides a useful general framework for bounding graph-state preparation complexity, finer combinatorial descriptions can yield substantially better bounds for specific graph families.

Measurement-based quantum computation (MBQC) is one of the main applications of graph states. It is known that MBQC on graph states of logarithmically bounded rank-width can be simulated efficiently on a classical computer [5]. Efficient classical simulation is also known for MBQC on circle graph states [6, 7, 8]. From the perspective of MBQC, a natural goal is therefore to identify classes of graph states that admit efficient preparation but for which efficient classical simulation of MBQC is not known. In view of the results of this paper, natural candidate classes include interval graphs and graph classes of moderate rank-width, for example rank-width $r = \Theta(\sqrt{n})$.

Several other problems remain open. It would be worthwhile to sharpen the constants in the rank-width-based bounds and to identify other local-complementation-invariant graph parameters that yield meaningful lower bounds on CZ-complexity. On the algorithmic side, it remains open whether the general framework developed in this paper can also achieve $O(n)$ EC-complexity for interval graphs. For circle graphs, it would be natural to understand how the Euler genus of a tour graph is reflected in the structure of the corresponding local-complementation equivalence class. All preparation algorithms developed in this paper avoid the use of working qubits. It remains unclear whether allowing working qubits can reduce the number of CZ operations. Clarifying this question, and more generally understanding the trade-off between the number of working qubits and the number of CZ operations, would be important for practical implementations. Another natural direction is to enlarge the class of single-qubit operations that do not contribute to the two-qubit count and allow arbitrary single-qubit unitaries. In this direction, Claudet and Perdrix recently gave a graphical characterization of local unitary equivalence in terms of generalized local complementation [12]. Developing an analogous graphical characterization for transformations generated by an arbitrary number of local unitary operations together with at most t CZ operations would substantially broaden the scope of the present framework. We hope that the combinatorial viewpoint developed in this paper will be useful for further studies on graph-state preparation, stabilizer-state transformations, and resource states for MBQC.

Acknowledgments

The work of RM was supported by JST FOREST Program Grant Number JPMJFR216V and JSPS KAKENHI Grant Numbers JP20H04138, JP20H05966 and JP22H00522. We thank Simon Martiel and Tristan Cam for identifying errors in the algorithms for generating permutation graphs and circle graphs in the earlier version of the paper. We also thank James Davies and Andrew Jena for identifying the same errors independently.

Author contributions. The notion of CZ-complexity and the problem of minimizing it were proposed by Yusei Yoshimura. Yusei Yoshimura and Ryuhei Mori proved preliminary versions of some of the results in this paper. Later, Soh Kumabe and Ryuhei Mori strengthened these results using the cut-rank and rank-width. Large language models were used only to assist with the writing of the manuscript, including language editing, phrasing, and clarity improvements. They were not used to produce mathematical results, proofs, calculations, code, or figures. All authors approved the final manuscript and are fully responsible for its content.

References

[1] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. "Graphical description of the action of local Clifford transformations on graph states". Phys. Rev. A **69**, 022316 (2004).

[2] Wolfgang Dür, Hans Aschauer, and Hans J. Briegel. "Multiparticle entanglement purification for graph states". Phys. Rev. Lett. **91**, 107903 (2003).
[3] Marc Hein, Jens Eisert, and Hans J. Briegel. "Multiparty entanglement in graph states". Phys. Rev. A **69**, 062311 (2004).
[4] Robert Raussendorf, Daniel E. Browne, and Hans J. Briegel. "Measurement-based quantum computation on cluster states". Phys. Rev. A **68**, 022312 (2003).
[5] Maarten Van den Nest, Wolfgang Dür, Guifré Vidal, and Hans J. Briegel. "Classical simulation versus universality in measurement-based quantum computation". Phys. Rev. A **75**, 012337 (2007).
[6] Sergey Bravyi and Robert Raussendorf. "Measurement-based quantum computation with the toric code states". Phys. Rev. A **76**, 022304 (2007).
[7] Brent Harrison, Vishnu Iyer, Ojas Parekh, Kevin Thompson, and Andrew Zhao. "Fermionic insights into measurement-based quantum computation: Circle graph states are not universal resources" (2025). arXiv:2510.05557.
[8] Frederik Hahn, Rose McCarty, Hendrik Poulsen Nautrup, and Nathan Claudet. "The structure of circle graph states" (2026). arXiv:2603.08847.
[9] Matthew B. Elliott, Bryan Eastin, and Carlton M. Caves. "Graphical description of the action of Clifford operators on stabilizer states". Phys. Rev. A **77**, 042307 (2008).
[10] Axel Dahlberg and Stephanie Wehner. "Transforming graph states using single-qubit operations". Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences **376**, 20170325 (2018).
[11] Jeremy C. Adcock, Sam Morley-Short, Axel Dahlberg, and Joshua W. Silverstone. "Mapping graph state orbits under local complementation". Quantum **4**, 305 (2020).
[12] Nathan Claudet and Simon Perdrix. "Local equivalence of stabilizer states: A graphical characterisation". In 42nd International Symposium on Theoretical Aspects of Computer Science (STACS 2025). Volume 327 of Leibniz International Proceedings in Informatics (LIPIcs), pages 27:1–27:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik (2025).
[13] Adán Cabello, Lars Eirik Danielsen, Antonio J. López-Tarrida, and José R. Portillo. "Optimal preparation of graph states". Phys. Rev. A **83**, 042314 (2011).
[14] Hemant Sharma, Kenneth Goodenough, Johannes Borregaard, Filip Rozpędek, and Jonas Helsen. "Minimising the number of edges in LC-equivalent graph states". Quantum **10**, 2001 (2026).
[15] Zhengfeng Ji, Jianxin Chen, Zhaohui Wei, and Mingsheng Ying. "The LU-LC conjecture is false". Quantum Information and Computation **10**, 97–108 (2010).
[16] Ketan N. Patel, Igor L. Markov, and John P. Hayes. "Optimal synthesis of linear reversible circuits". Quantum Information and Computation **8**, 0282–0294 (2008).
[17] Sang-il Oum and Paul Seymour. "Approximating clique-width and branch-width". Journal of Combinatorial Theory, Series B **96**, 514–528 (2006).
[18] Sang-il Oum. "Rank-width: Algorithmic and structural results". Discrete Applied Mathematics **231**, 15–24 (2017).
[19] Choongbum Lee, Joonkyung Lee, and Sang-il Oum. "Rank-width of random graphs". Journal of Graph Theory **70**, 339–347 (2012).
[20] James Davies and Andrew Jena. "Preparing graph states forbidding a vertex-minor" (2025). arXiv:2504.00291.
[21] Andrew Jena. "Graph-theoretic techniques for optimizing NISQ algorithms". PhD thesis. University of Waterloo. (2024). url: https://hdl.handle.net/10012/20343.
[22] Donggyu Kim and Sang-il Oum. "Vertex-minors of graphs: A survey". Discrete Applied Mathematics **351**, 54–73 (2024).
[23] André Bouchet. "Isotropic systems". European Journal of Combinatorics **8**, 231–244 (1987).
[24] André Bouchet. "Graphic presentations of isotropic systems". Journal of Combinatorial Theory, Series B **45**, 58–76 (1988).
[25] André Bouchet. "Connectivity of isotropic systems". Annals of the New York Academy of Sciences **555**, 81–93 (1989).
[26] André Bouchet. "κ-transformations, local complementations and switching". In Geňa Hahn, Gert Sabidussi, and Robert E. Woodrow, editors, Cycles and Rays. Pages 41–50. Springer Netherlands, Dordrecht (1990).

[27] André Bouchet. "An efficient algorithm to recognize locally equivalent graphs". Combinatorica **11**, 315–329 (1991).
[28] André Bouchet. "Recognizing locally equivalent graphs". Discrete Mathematics **114**, 75–86 (1993).
[29] André Bouchet. "Circle graph obstructions". Journal of Combinatorial Theory, Series B **60**, 107–144 (1994).
[30] Anton Kotzig. "Eulerian lines in finite 4-valent graphs and their transformations". In Theory of Graphs (Proc. Colloq., Tihany, 1966). Pages 219–230. Academic Press, New York-London (1968).
[31] Dmitriĭ G. Fon-Der-Flaass. "Local complementations of simple and directed graphs". In Alekseĭ D. Korshunov, editor, Discrete Analysis and Operations Research. Pages 15–34. Springer Netherlands, Dordrecht (1996).
[32] André Bouchet. "Digraph decompositions and Eulerian systems". SIAM Journal on Algebraic Discrete Methods **8**, 323–337 (1987).
[33] Sang-il Oum. "Rank-width and vertex-minors". Journal of Combinatorial Theory, Series B **95**, 79–100 (2005).
[34] Sang-il Oum and Paul Seymour. "Testing branch-width". Journal of Combinatorial Theory, Series B **97**, 385–393 (2007).
[35] Sang-il Oum. "Computing rank-width exactly". Information Processing Letters **109**, 745–748 (2009).
[36] Bruno Courcelle and Sang-il Oum. "Vertex-minors, monadic second-order logic, and a conjecture by Seese". Journal of Combinatorial Theory, Series B **97**, 91–126 (2007).
[37] Fedor V. Fomin and Tuukka Korhonen. "Fast FPT-approximation of branchwidth". SIAM Journal on Computing **53**, 1085–1131 (2024).
[38] Tuukka Korhonen and Marek Sokołowski. "Almost-linear time parameterized algorithm for rankwidth via dynamic rankwidth". In Proceedings of the 56th Annual ACM Symposium on Theory of Computing. Pages 1538–1549. STOC 2024. Association for Computing Machinery (2024).
[39] Tuukka Korhonen and Sang-il Oum. "Branch-width of connectivity functions is fixed-parameter tractable" (2026). arXiv:2601.04756.
[40] Peter Høyer, Mehdi Mhalla, and Simon Perdrix. "Resources required for preparing graph states". In Algorithms and Computation: 17th International Symposium, ISAAC 2006, Kolkata, India, December 18-20, 2006. Proceedings 17. Pages 638–649. Springer (2006).
[41] Jérôme Javelle, Mehdi Mhalla, and Simon Perdrix. "On the minimum degree up to local complementation: Bounds and complexity". In Graph-Theoretic Concepts in Computer Science. Pages 138–147. Springer Berlin Heidelberg (2012).
[42] David Cattanéo and Simon Perdrix. "Minimum degree up to local complementation: Bounds, parameterized complexity, and exact algorithms". In Algorithms and Computation. Pages 259–270. Springer Berlin Heidelberg (2015).
[43] Nathan Claudet and Simon Perdrix. "Covering a graph with minimal local sets". In Graph-Theoretic Concepts in Computer Science. Pages 136–150. Springer Nature Switzerland (2025).
[44] Daniel Gottesman. "Stabilizer codes and quantum error correction". PhD thesis. California Institute of Technology. (1997). arXiv:quant-ph/9705052.
[45] Axel Dahlberg, Jonas Helsen, and Stephanie Wehner. "How to transform graph states using single-qubit operations: Computational complexity and algorithms". Quantum Science and Technology **5**, 045016 (2020).
[46] Martin C. Golumbic and Udi Rotics. "On the clique-width of some perfect graph classes". International Journal of Foundations of Computer Science **11**, 423–443 (2000).
[47] Jim Geelen, O-joung Kwon, Rose McCarty, and Paul Wollan. "The grid theorem for vertex-minors". Journal of Combinatorial Theory, Series B **158**, 93–116 (2023).
[48] Norman Biggs. "Algebraic graph theory". Cambridge Mathematical Library. Cambridge University Press. (1993). 2nd edition.
[49] Hubert de Fraysseix and Patrice Ossona de Mendez. "On a characterization of Gauss codes". Discrete & Computational Geometry **22**, 287–295 (1999).
[50] Sergey Bravyi, Joseph A. Latone, and Dmitri Maslov. "6-qubit optimal Clifford circuits". npj Quantum Information **8**, 79 (2022).

[51] Lars Eirik Danielsen and Matthew G. Parker. "On the classification of all self-dual additive codes over GF(4) of length up to 12". Journal of Combinatorial Theory, Series A **113**, 1351–1367 (2006).
[52] Lars Eirik Danielsen. "Graph-based classification of self-dual additive codes over finite fields". Advances in Mathematics of Communications **3**, 329–348 (2009).
[53] Eric M. Rains and Neil J.A. Sloane. "Self-dual codes" (2002). arXiv:math/0208001.
[54] Gabriele Nebe, Eric M. Rains, and Neil J.A. Sloane. "Self-dual codes and invariant theory". Springer. (2006).
[55] Mithilesh Kumar, Srimathi Varadharajan, and Håvard Raddum. "Graphs and self-dual additive codes over GF(4)". In Proc. The Eleventh International Workshop on Coding and Cryptography (WCC 2019). (2019). url: https://www.lebesgue.fr/sites/default/files/proceedings_WCC/WCC_2019_paper_24.pdf.

Appendix A Proofs of Facts 13 and 15

A.1 Proof of Fact 13

Fact 13. For any two stabilizer states $|\psi\rangle$ and $|\varphi\rangle$ with the same generating matrices, there exists a Pauli operator $g \in \mathcal{G}_n$ such that $|\psi\rangle$ is equal to $g|\varphi\rangle$ up to a constant factor.

Proof. Let G be the common generating matrix for $|\psi\rangle$ and $|\varphi\rangle$. Let $v \in \mathbb{F}_2^n$ be a vector where $v_i = 1$ if and only if the sign of the i-th stabilizer for $|\psi\rangle$ and $|\varphi\rangle$ differs. There exists a row vector $a \in \mathbb{F}_2^{2n}$ satisfying

$$G \begin{bmatrix} O_n & I_n \\ I_n & O_n \end{bmatrix} a^T = v$$

since $G \begin{bmatrix} O_n & I_n \\ I_n & O_n \end{bmatrix} \in \mathbb{F}_2^{n \times 2n}$ has linearly independent rows. Then, $S_i \sigma_a |\varphi\rangle = (-1)^{v_i} \sigma_a |\varphi\rangle$ where S_i is the i-th stabilizer of $|\varphi\rangle$. Hence, $\sigma_a |\varphi\rangle$ has the same stabilizer as $|\psi\rangle$. □

A.2 Proof of Fact 15

Fact 15. Let $C, D \in \mathcal{C}_n$ be Clifford operators. Then, $L_C = L_D$ if and only if there exists a Pauli operator $g \in \mathcal{G}_n$ such that C is equal to gD up to a constant factor.

Proof. First, we prove that $L_C = I_{\mathbb{F}_2^{2n}}$ implies C is in $\mathcal{G}_n$ up to a constant factor. Assume $L_C = I_{\mathbb{F}_2^{2n}}$. Then, for any $h \in \mathcal{G}_n$, there exists $\alpha_h \in \{\pm 1\}$ such that $ChC^\dagger = \alpha_h h$. For any $h_1, h_2 \in \mathcal{G}_n$, $\alpha_{h_1 h_2} h_1 h_2 = C h_1 h_2 C^\dagger = C h_1 C^\dagger C h_2 C^\dagger = \alpha_{h_1} \alpha_{h_2} h_1 h_2$. Hence, $\alpha_{h_1 h_2} = \alpha_{h_1} \alpha_{h_2}$ for any $h_1, h_2 \in \mathcal{G}_n$. This implies that the map $h \mapsto \alpha_h$ is a group homomorphism from $\mathcal{G}_n$ to $\{\pm 1\}$.

Let $a \in \mathbb{F}_2^{2n}$ be a vector satisfying

$$\alpha_{X_i} = (-1)^{a_{n+i}}, \qquad \alpha_{Z_i} = (-1)^{a_i} \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Then, $g = \sigma_a$ satisfies

$$gX_i = \alpha_{X_i} X_i g, \qquad gZ_i = \alpha_{Z_i} Z_i g \qquad \forall i \in \{1, 2, \ldots, n\}.$$

Since every element of $\mathcal{G}_n$ is a product of the generators $X_1, \ldots, X_n, Z_1, \ldots, Z_n$ up to a constant factor, $gh = \alpha_h hg$ for any $h \in \mathcal{G}_n$ from the homomorphism. Then, $(Cg)h(Cg)^\dagger = \alpha_h ChC^\dagger = \alpha_h^2 h = h$. Hence, Cg commutes with any matrix in $\mathcal{G}_n$, and hence commutes with any matrix in $U(2^n)$. This means that Cg is equal to the identity matrix up to a constant factor. Hence, C is in $\mathcal{G}_n$ up to a constant factor.

Next, assume $L_C = L_D$. Then, $L_{CD^{-1}} = L_{D^{-1}} L_C = L_D^{-1} L_C = I_{\mathbb{F}_2^{2n}}$. This means that there exists $g \in \mathcal{G}_n$ such that CD^{-1} is equal to g up to a constant factor. □

Appendix B Sufficiency of the CZ operators: Proof of Proposition 2

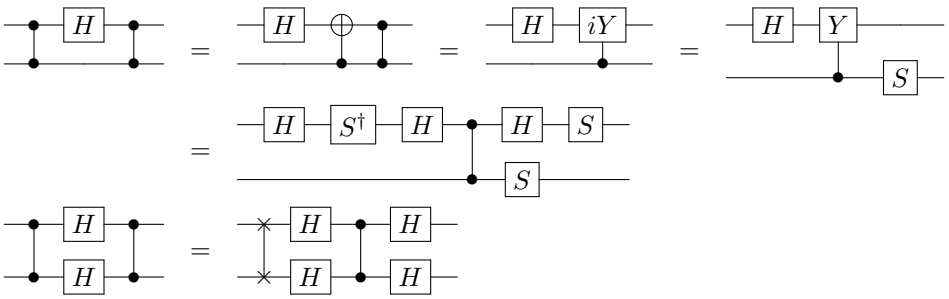
In this section, we prove Proposition 2.

Proposition 2. *For any graph G, there exists a quantum algorithm $\mathcal{A}$ generating $|G\rangle$ of the type in Definition 1 with CZ-complexity $\Gamma_{\mathrm{CZ}}(|G\rangle)$ where all two-qubit Clifford operations in $\mathcal{A}$ are the* CZ *operations.*

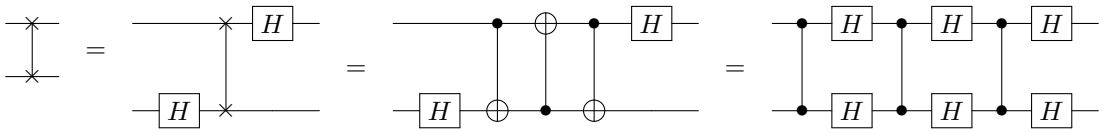
We first prove that any two-qubit Clifford gate can be replaced by single-qubit gates, at most one CZ gate and at most one SWAP gate.

Lemma 54. *Any two-qubit Clifford operator is represented by a quantum circuit consisting of arbitrary number of single-qubit Clifford gates, at most one* CZ *gate and at most one SWAP gate.*

Proof. From Fact 15, for a given two-qubit Clifford operator $\mathcal{C}$, it is sufficient to show the existence of the quantum circuit $\mathcal{D}$ satisfying $L_{\mathcal{C}} = L_{\mathcal{D}}$ and the conditions of the lemma. Since the S, H and CZ gates generate the Clifford group, any two-qubit Clifford operator is represented by a Clifford circuit consisting of these gates. When the Clifford circuit includes more than one CZ gates, we can reduce the number of CZ gates by the following procedure. There are six possible single-qubit Clifford gates (up to succeeding Pauli gates) in (2). Since S commutes with CZ, we can assume without loss of generality that single-qubit gates between two specific CZ gates are all Hadamard gates. In the two possible cases, the number of CZ gates can be reduced by using the SWAP gate as follows.



The second relation is obtained from the following Clifford circuit for the SWAP gate.



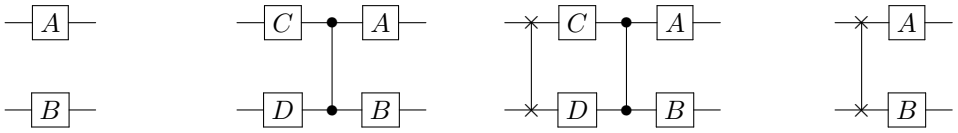
By repeating this procedure, the number of the CZ gates is reduced to at most one. □

Then, by replacing all two-qubit Clifford gates in quantum circuits with Clifford circuits with at most one CZ gate and SWAP gate, we obtain the following theorem.

Lemma 55. *Let $\mathcal{C}$ be a Clifford circuit that consists of arbitrary number of single-qubit Clifford gates and t two-qubit Clifford gates. Then, there exists a Clifford circuit $\mathcal{D}$ that consists of arbitrary number of single-qubit Clifford gates and at most t* CZ *gates and a permutation of qubits $\mathcal{P}$ such that $\mathcal{D}\mathcal{P} = \mathcal{C}$.*

Proof. From Lemma 54, all two-qubit Clifford gates in $\mathcal{C}$ can be replaced by Clifford circuits consisting of arbitrary number of single-qubit Clifford gates, at most one CZ gates and at most one SWAP gates. The SWAP gate commutes with an arbitrary quantum gate U by changing the qubits for which U is applied. Hence, the SWAP gate can be placed at the beginning of the quantum circuit, and obtain Lemma 55. □

Now, we are ready to prove Proposition 2.



(a) Zero CZ gate. (b) One CZ gate. (c) Two CZ gates. (d) Three CZ gates.

Figure 7: Classification of two-qubit Clifford operators (up to succeeding Pauli gates): The gates A and B are chosen from the six single-qubit Clifford operators in (2). The gates C and D are chosen from I, H and HS.

Proof of Proposition 2. From Lemma 21, regardless of the measurement outcome, an adaptive local Clifford operation gives some graph state deterministically. Hence, it is sufficient to show that there is a quantum algorithm that includes at most $\Gamma_{\mathrm{CZ}}(|G\rangle)$ CZ operations and no other two-qubit Clifford operations, and produces $|G\rangle$ when all measurement outcomes are zero. There exist a Clifford circuit $\mathcal{C}$ including arbitrary number of single-qubit Clifford gates and $\Gamma_{\mathrm{CZ}}(|G\rangle)$ two-qubit Clifford gates such that

$$\langle 0|^{\otimes s} \mathcal{C} |0\rangle^{\otimes(n+s)} \propto |G\rangle$$

From Lemma 55, there exists a Clifford circuit $\mathcal{D}$ including some single-qubit Clifford gates and at most $\Gamma_{\mathrm{CZ}}(|G\rangle)$ CZ gates such that

$$\langle 0|^{\otimes s} \mathcal{D} |0\rangle^{\otimes(n+s)} \propto |G\rangle$$

This proves Proposition 2. □

Remark 56. In terms of the number of CZ gates when the SWAP gate is not allowed, two-qubit Clifford operators are classified as in Figure 7. The number of two-qubit Clifford operators of types (a), (b), (c) and (d) are $36 = 6^2$, $324 = 3^2 \times 6^2$, 324 and 36, respectively. Note that the numbers of required CNOT gates (equivalently the CZ gates) for Clifford operators up to six qubits are calculated by computer search in [50].

Appendix C Independence systems, additive codes and the EC-complexity

C.1 Isotropic independence systems

In this appendix, we consider upper bounds on the minimum dependent set that depend only on n, and prove Theorem 4. For any graph G, let

$$\mathcal{I}_G := \{S \subseteq V(G) \mid \mathrm{cutrank}_G(S) = |S|\}.$$

We first regard the structure $(V(G), \mathcal{I}_G)$ as an independence system in matroid theory. We then switch the understanding of the structure from matroid theory to coding theory.

Definition 57 (Independence system)**.** Let V be a finite set and $\mathcal{I} \subseteq 2^V$. Then, $(V, \mathcal{I})$ is an *independence system* if it satisfies the following conditions:

1. $\varnothing \in \mathcal{I}$.
2. For any $X \in \mathcal{I}$ and $Y \subseteq X$, $Y \in \mathcal{I}$.

Then, $(V(G), \mathcal{I}_G)$ is obviously an independence system. Furthermore, $(V(G), \mathcal{I}_G)$ is an isotropic independence system.

Definition 58 (Linear 2-extendible system and isotropic independence system)**.** An independence system $(V, \mathcal{I})$ is a *linear 2-extendible system* on a field $\mathbb{F}$ if there exists $m \in \mathbb{N}$ and two m-dimensional vectors $a^v \in \mathbb{F}^m$ and $b^v \in \mathbb{F}^m$ associated to each $v \in V$, and

$$\mathcal{I} = \{S \subseteq V \mid \text{A set of } 2|S| \text{ vectors } a^v \text{ and } b^v \text{ for } v \in S \text{ are linearly independent}\}.$$

Especially, a linear 2-extendible system on $\mathbb{F}_2$ is a *binary 2-extendible system*.

A binary 2-extendible system is an *isotropic independence system* if $m = |V|$ and for any $s, t \in \{1, 2, \ldots, |V|\}$

$$\sum_{v \in V} (a_s^v b_t^v + b_s^v a_t^v) = 0 \tag{5}$$

where $a_s^v \in \mathbb{F}_2$ denotes an s-th element of $a^v \in \mathbb{F}_2^{|V|}$.

Lemma 59. *For any graph G, $(V(G), \mathcal{I}_G)$ is an isotropic independence system.*

Proof. For each $v \in V(G)$, let a^v, $b^v \in \mathbb{F}_2^n$ be vectors defined as

$$\begin{aligned} &\text{A } w\text{-th element of } a^v \text{ is } 1 \iff w = v \\ &\text{A } w\text{-th element of } b^v \text{ is } 1 \iff \{v, w\} \in E. \end{aligned} \tag{6}$$

We first confirm that $((a^v, b^v))_{v \in V(G)}$ satisfies the isotropic condition (5). Let A be the adjacency matrix of G. Then,

$$\begin{aligned} \sum_{v \in V(G)} (a_s^v b_t^v + b_s^v a_t^v) &= \sum_{v \in V(G)} (\delta_{v,s} A_{v,t} + A_{v,s} \delta_{v,t}) \\ &= A_{s,t} + A_{t,s} = 0 \end{aligned}$$

Hence, $((a^v, b^v))_{v \in V(G)}$ satisfies the isotropic condition (5).

We now confirm that the binary 2-extendible system defined by $((a^v, b^v))_{v \in V(G)}$ represents $(V(G), \mathcal{I}_G)$. For any $S \subseteq V(G)$,

$$\begin{aligned} \dim \operatorname{span}_{\mathbb{F}_2} \left(\bigcup_{v \in S} \{a_v, b_v\} \right) &= \operatorname{rank}_{\mathbb{F}_2} \left(\begin{bmatrix} I_S & A_{S,S} \\ 0 & A_{V(G) \setminus S, S} \end{bmatrix} \right) \\ &= |S| + \operatorname{rank}_{\mathbb{F}_2}(A_{V(G) \setminus S, S}) \\ &= |S| + \operatorname{cutrank}_G(S). \end{aligned}$$

Hence, all the $2|S|$ vectors are linearly independent if and only if

$$|S| + \operatorname{cutrank}_G(S) = 2|S| \iff \operatorname{cutrank}_G(S) = |S|.$$

This implies that the binary 2-extendible system represents $(V(G), \mathcal{I}_G)$. □

The isotropic system was introduced by Bouchet [23]. However, the isotropic system has not been regarded as an independence system. Note that for a bipartite graph $G = (A \cup B, E)$, the binary 2-extendible system $(A \cup B, \mathcal{I}_G)$ is an intersection of the binary matroid $\mathcal{M} = \mathrm{Bin}(G, A, B)$ and its dual $\mathcal{M}^* = \mathrm{Bin}(G, B, A)$, defined in [33], i.e., $S \in \mathcal{I}_G$ if and only if S is independent both for $\mathcal{M}$ and $\mathcal{M}^*$. Generally, for any stabilizer state with generating matrix $G \in \mathbb{F}_2^{n \times 2n}$, there exists corresponding isotropic independence system $(V, \mathcal{I})$. Here, V corresponds to the set of qubits, and $\mathcal{I}$ is an independent set defined by $((a^v, b^v))_{v \in V}$ where a^v and b^v are v-th column and $(n+v)$-th column of G, respectively.

C.2 Self-dual additive code over $\mathbb{F}_4$ with respect to the Hermitian trace inner product

Since from Lemma 38, we are interested in small dependent sets of the isotropic independence system $(V(G), \mathcal{I}_G)$ rather than large independent sets, we switch our perspective from matroid theory to coding theory. Indeed, self-dual additive codes over $\mathbb{F}_4$ with respect to the Hermitian trace inner product are essentially equivalent to isotropic systems [51]. Let $\mathbb{F}_4 = \{0, 1, \omega, \omega^2\}$ where $\omega^2 + \omega + 1 = 0$. The *conjunction* of $x \in \mathbb{F}_4$ is defined as $\overline{x} = x^2$. The *trace map* $\mathrm{Tr}\colon \mathbb{F}_4 \to \mathbb{F}_2$ is defined as $\mathrm{Tr}(x) = x + \overline{x}$.

$$\begin{bmatrix} X & X & I & I \\ I & X & X & I \\ I & I & X & X \\ Z & Y & Y & Z \end{bmatrix} \longleftrightarrow \left(\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right), \left(\begin{bmatrix} 1 \\ 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right), \left(\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right), \left(\begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right) \longleftrightarrow \begin{bmatrix} \omega & \omega & 0 & 0 \\ 0 & \omega & \omega & 0 \\ 0 & 0 & \omega & \omega \\ 1 & \omega^2 & \omega^2 & 1 \end{bmatrix}$$

Figure 8: Correspondence between the stabilizer generators, isotropic independence system and generator matrix of $4^{\mathrm{H}+}$-code.

Definition 60 (Self-dual additive code over $\mathbb{F}_4$ with respect to the Hermitian trace inner product)**.** A non-empty subset $\mathcal{C} \subseteq \mathbb{F}_4^n$ is an *additive code* over $\mathbb{F}_4$ if $x + y \in \mathcal{C}$ for all $x, y \in \mathcal{C}$. Each element $x \in \mathcal{C}$ is called a *codeword* of $\mathcal{C}$. A *support* of a codeword $x \in \mathcal{C}$ is defined as

$$\mathrm{support}(x) := \{i \in \{1, 2, \ldots, n\} \mid x_i \neq 0\}.$$

For $x, y \in \mathbb{F}_4^n$, the Hermitian trace inner product is defined as

$$\langle x, y \rangle = \sum_{i=1}^{n} \mathrm{Tr}(x_i \overline{y_i}).$$

Two vectors $x, y \in \mathbb{F}_4^n$ with $\langle x, y \rangle = 0$ are said to be *orthogonal*. The dual code $\mathcal{C}^\perp \subseteq \mathbb{F}_4^n$ of additive code $\mathcal{C}$ is defined as

$$\mathcal{C}^\perp = \{y \in \mathbb{F}_4^n \mid \langle x, y \rangle = 0 \quad \forall x \in \mathcal{C}\}.$$

An additive code $\mathcal{C}$ over $\mathbb{F}_4$ is said to be *self-dual with respect to the Hermitian trace inner product* if $\mathcal{C}^\perp = \mathcal{C}$. Self-dual additive code over $\mathbb{F}_4$ with respect to the Hermitian trace inner product is denoted by $4^{\mathrm{H}+}$-code.

An additive code over $\mathbb{F}_4$ is represented by a *generator matrix* $H \in \mathbb{F}_4^{k \times n}$ satisfying

$$\mathcal{C} = \mathrm{span}_{\mathbb{F}_2}(\text{Rows of } H).$$

When $\mathcal{C}$ is self-dual, $k = n$ and there is another representation using the generator matrix H

$$\mathcal{C} = \{y \in \mathbb{F}_4^n \mid \langle x, y \rangle = 0 \quad \text{for any row } x \text{ of } H\}.$$

We introduce an isomorphism between $\mathcal{G}_1/\langle iI_2 \rangle$ and $\mathbb{F}_4$ defined as

$$I_2 \longleftrightarrow 0, \qquad X \longleftrightarrow \omega, \qquad Y \longleftrightarrow \omega^2, \qquad Z \longleftrightarrow 1.$$

We can also introduce an isomorphism between $\mathcal{G}_n/\langle iI_2^{\otimes n} \rangle$ and $\mathbb{F}_4^n$ by applying the isomorphism for each entry. It is easy to confirm that two Pauli matrices $x, y \in \mathcal{G}_n/\langle iI_2^{\otimes n} \rangle$ commute if and only if the Hermitian trace inner product of their $\mathbb{F}_4^n$ representations is zero. Hence, there is one-to-one correspondence between stabilizer states and $4^{\mathrm{H}+}$-codes. An example of the correspondence is shown in Figure 8.

For a graph state $|G\rangle$ with adjacency matrix $A \in \mathbb{F}_2^{n \times n}$, the corresponding $4^{\mathrm{H}+}$-code with generator matrix $\omega I + A$ is denoted by $\mathcal{C}_G$.

Lemma 61. *For any graph G and non-empty $S \subseteq V(G)$, $S \notin \mathcal{I}_G$ if and only if there exists a non-zero codeword of $\mathcal{C}_G$ whose support is a subset of S.*

Proof. For any $a, b, s, t \in \mathbb{F}_2$,

$$\begin{aligned} \mathrm{Tr}\left((b + a\omega)\overline{(s + t\omega)}\right) &= \mathrm{Tr}\left((b + a\omega)(s + t\omega^2)\right) \\ &= \mathrm{Tr}\left(bs + at + bt\omega^2 + as\omega\right) \\ &= sa + tb \end{aligned}$$

Let $a^v, b^v \in \mathbb{F}_2^n$ be vectors in (6) for each $v \in V(G)$. Then, $\omega a^v + b^v$ is the v-th column of the generator matrix H of $\mathcal{C}_G$. For any graph G and non-empty $S \subseteq V(G)$,

$$\begin{aligned}
S \notin \mathcal{I}_G &\iff \exists s,t \in \mathbb{F}_2^S, \quad \neg(s=t=0^S) \ \wedge \ \sum_{v\in S}(s_v a^v + t_v b^v) = 0^{V(G)} \\
&\iff \exists s,t \in \mathbb{F}_2^S, \quad \neg(s=t=0^S) \ \wedge \ \sum_{v\in S}(s_v a^v_w + t_v b^v_w) = 0 \qquad \forall w \in V(G) \\
&\iff \exists s,t \in \mathbb{F}_2^S, \quad \neg(s=t=0^S) \ \wedge \ \sum_{v\in S}\mathrm{Tr}\left((b^v_w + a^v_w\omega)\overline{(s_v + t_v\omega)}\right) = 0 \qquad \forall w \in V(G) \\
&\iff \exists y \in \mathbb{F}_4^S \setminus \{0^S\}, \quad \sum_{v\in S}\mathrm{Tr}\left(H_{wv}\,\overline{y_v}\right) = 0 \qquad \forall w \in V(G) \\
&\iff \exists y \in \mathbb{F}_4^S \setminus \{0^S\}, \quad \sum_{v\in S}\mathrm{Tr}(x_v\,\overline{y}_v) = 0 \qquad \text{for any row } x \text{ of } H \\
&\iff \exists y \in \mathbb{F}_4^{V(G)} \setminus \{0^{V(G)}\}, \quad \mathrm{support}(y) \subseteq S \ \wedge \ \langle x, y\rangle = 0 \qquad \text{for any row } x \text{ of } H \\
&\iff \exists y \in \mathbb{F}_4^{V(G)} \setminus \{0^{V(G)}\}, \quad \mathrm{support}(y) \subseteq S \ \wedge \ y \in \mathcal{C}_G.
\end{aligned}$$

□

Since we are interested in small dependent sets of the isotropic independence system $(V(G), \mathcal{I}_G)$, we are interested in low-weight codewords of $4^{\mathrm{H}+}$-code $\mathcal{C}_G$. Note that the minimum distance of $\mathcal{C}_G$ is one plus the minimum degree of vertices in graphs that are LC-equivalent to G [51, 40, 52]. Now, we can apply results in coding theory.

Lemma 62 ([53, 54])**.** *Let $\mathcal{C}$ be a $4^{\mathrm{H}+}$-code. Then, the minimum distance of $\mathcal{C}$ is at most $2\lfloor \frac{n}{6} \rfloor + 2$ for $n \not\equiv 5 \bmod 6$, and $2\lfloor \frac{n}{6} \rfloor + 3$ for $n \equiv 5 \bmod 6$.*

It is conjectured that the minimum distance is at most $2\lfloor \frac{n}{6} \rfloor + 1$ for $n \equiv 1 \bmod 6$, which was confirmed for $n \leq 25$ [54].

We now prove the upper bound of the EC-complexity. Let

$$C(n) := \max_{G\colon n\text{-vertex graph}} \Gamma_{\mathrm{EC}}(G).$$

Lemma 63. $C(n) \leq \frac{(n-1)(n+4)}{6}$.

Proof. We prove the Lemma using the induction on n. By a computer search, we obtain $C(1) = 0$, $C(2) = 1$, $C(3) = 2$, $C(4) = 3$, $C(5) = 5$, $C(6) = 7$. Hence, the inequality is satisfied for $n \leq 6$. From Lemma 62, for $n \geq 7$

$$\begin{aligned}
C(n) &\leq C(n-1) + 2\left\lfloor \frac{n}{6} \right\rfloor + 1 + \mathbb{I}\{n \equiv 5 \bmod 6\} \\
&\leq C(n-6) + 2\frac{n + (n-1) + (n-2) + (n-3) + (n-4) + (n-5) - 1 - 2 - 3 - 4 - 5}{6} + 7 \\
&= C(n-6) + 2n - 3 \\
&\leq \frac{(n-7)(n-2)}{6} + 2n - 3 \\
&= \frac{(n-1)(n+4)}{6}.
\end{aligned}$$

□

Note that if we assume that the minimum distance of self-dual additive $\mathbb{F}_4$ codes is at most $2\lfloor \frac{n}{6} \rfloor + 1$ for $n \equiv 1 \bmod 6$, we obtain an upper bound $C(n) \leq \frac{n(n+2)-2}{6}$. Although the optimal asymptotic CZ-complexity is $C(n) = \Theta(n^2/\log n)$ [16], Lemma 63 gives the upper bound on the CZ-complexity for finite n.

Corollary 43 and Lemma 61 imply an upper bound on the minimum distance of $4^{\mathrm{H}+}$-codes.

Corollary 64. *For any graph G with n vertices and the rank-width $r \leq n/3$, the $4^{\mathrm{H}+}$-code $\mathcal{C}_G$ has the minimum distance at most $\lceil (3r+1)/2 \rceil$.*

Corollary 64 for $r = 1$ was shown in [55].

The rank-width of an n-vertex graph is at most $\lceil n/3 \rceil$. In other words, any graph with rank-width r has at least $3r - 2$ vertices. As another consequence of Lemma 62, we show that there is no graph with $3r - 2$ vertices and rank-width r when r is even.

Lemma 65. *For any even $r \geq 2$, a graph G with $3r - 2$ vertices has rank-width at most $r - 1$.*

Proof. Let G be a graph with $3r - 2$ vertices. From Lemma 62, there is a set $S \subseteq V(G)$ of size r with $\mathrm{cutrank}_G(S) \leq r - 1$. Let $A, B \subseteq V(G) \setminus S$ be a pair of subsets satisfying $A \cup B = V(G) \setminus S$, $A \cap B = \varnothing$ and $|A| = |B| = r - 1$. Then, we consider three rooted binary trees T_S, T_A and T_B where their leaf nodes correspond to S, A and B, respectively. Then, we obtain a rank-decomposition of G with a vertex that connects to the roots of T_S, T_A and T_B. The width of the rank-decomposition is at most $r - 1$. □

In order to generalize Lemma 65 for odd r by the same proof, we need the conjecture $d \leq 2\lfloor \frac{n}{6} \rfloor + 1$ for $n \equiv 1 \bmod 6$, which was confirmed for $n \leq 25$.

Lemma 66. *For $r = 1, 3, 5, 7, 9$, a graph G with $3r - 2$ vertices has rank-width at most $r - 1$.*

C.3 The proof of Theorem 4

By combining Lemmas 44 and 63, we prove Theorem 4.

Theorem 67 ((Equivalent to Theorem 4))**.** *For any odd r and $n \geq \frac{13r^2-6r-3}{4r}$,*

$$C(n,r) \leq \frac{5r^2-1}{4r}n - \frac{221r^4 - 180r^3 + 10r^2 + 36r + 9}{96r^2}.$$

For any even r and $n \geq \frac{13r-6}{4}$,

$$C(n,r) \leq \frac{5r}{4}n - \frac{221r^2 - 180r + 100}{96}.$$

Proof. Assume r is odd. We prove the theorem by induction on n. We first prove that if $n \in \left[\frac{13r^2-6r-3}{4r}, \frac{17r^2-6r-3}{4r}\right]$,

$$C(n,r) \leq \frac{5r^2-1}{4r}n - \frac{221r^4 - 180r^3 + 10r^2 + 36r + 9}{96r^2}.$$

This can be proved by

$$\begin{aligned} C(n,r) &\leq \frac{(n-1)(n+4)}{6} \\ &= \frac{5r^2-1}{4r}n + \frac{(n-1)(n+4)}{6} - \frac{5r^2-1}{4r}n \\ &= \frac{5r^2-1}{4r}n + \frac{1}{6}\left(n^2 - \frac{15r^2-6r-3}{2r}n - 4\right) \\ &\leq \frac{5r^2-1}{4r}n - \frac{221r^4 - 180r^3 + 10r^2 + 36r + 9}{96r^2} \end{aligned}$$

where in the last inequality, we substitute $n = \frac{13r^2-6r-3}{4r}$ into the second term since it maximizes this term.

We now assume $n \geq \lfloor \frac{17r^2-6r-3}{4r} + 1 \rfloor \geq 3r$. From Lemma 44,

$$\begin{aligned} C(n,r) &\leq \frac{5r^2-1}{4r}k + C(n-k,r) \\ &\leq \frac{5r^2-1}{4r}k + \frac{5r^2-1}{4r}(n-k) - \frac{221r^4 - 180r^3 + 10r^2 + 36r + 9}{96r^2} \\ &= \frac{5r^2-1}{4r}n - \frac{221r^4 - 180r^3 + 10r^2 + 36r + 9}{96r^2}. \end{aligned}$$

Assume r is even. We prove the theorem by induction on n. We first prove that if $n \in \left[\frac{13r-6}{4}, \frac{17r-6}{4}\right]$,

$$C(n,r) \leq \frac{5r}{4}n - \frac{221r^2 - 180r + 100}{96}.$$

This can be proved by

$$\begin{aligned} C(n,r) &\leq \frac{(n-1)(n+4)}{6} \\ &= \frac{5r}{4}n + \frac{(n-1)(n+4)}{6} - \frac{5r}{4}n \\ &= \frac{5r}{4}n + \frac{1}{6}\left(n^2 - \frac{15r-6}{2}n - 4\right) \\ &\leq \frac{5r}{4}n - \frac{221r^2 - 180r + 100}{96} \end{aligned}$$

where in the last inequality, we substitute $n = \frac{13r-6}{4}$ into the second term since it maximizes this term.

We now assume $n \geq \lfloor \frac{17r-6}{4} + 1 \rfloor \geq 3r$. From Lemma 44,

$$\begin{aligned} C(n,r) &\leq \frac{5r}{4}k + C(n-k,r) \\ &\leq \frac{5r}{4}k + \frac{5r}{4}(n-k) - \frac{221r^2 - 180r + 100}{96} \\ &= \frac{5r}{4}n - \frac{221r^2 - 180r + 100}{96}. \end{aligned}$$

□