Counting single-qubit Clifford equivalent graph states is # P -complete

Dahlberg, Axel; Helsen, Jonas; Wehner, Stephanie

DOI
10.1063/1.5120591

Publication date
2020

Document Version
Accepted author manuscript

Published in
Journal of Mathematical Physics

Counting single-qubit Clifford equivalent graph states is #ℙ-Complete

Axel Dahlberg,[1] Jonas Helsen,[1] and Stephanie Wehner[1]
QuTech, Delft University of Technology, and Kavli Institute of Nanoscience
Delft, The Netherlands

(Dated: 27 February 2020)

Graph states, which include Bell states, Greenberger-Horne-Zeilinger (GHZ) states, and cluster states, form a well-known class of quantum states with applications ranging from quantum networks to error-correction. Whether two graph states are equivalent up to single-qubit Clifford operations is known to be decidable in polynomial time and have been studied both in the context of producing certain required states in a quantum network but also in relation to stabilizer codes. The reason for the latter is that single-qubit Clifford equivalent graph states exactly correspond to *equivalent* stabilizer codes. We here consider the computational complexity of, given a graph state $|G\rangle$, counting the number of graph states, single-qubit Clifford equivalent to $|G\rangle$. We show that this problem is #ℙ-Complete. To prove our main result we make use of the notion of isotropic systems in graph theory. We review the definition of isotropic systems and point out their strong relation to graph states. We believe that these isotropic systems can be useful beyond the results presented in this paper.

I. INTRODUCTION

Graph states form a well-studied class of quantum states and are applied in many fields such as in quantum networks and quantum computers. In a quantum network, graph states are a resource used by applications such as secret sharing[1], anonymous transfer[2] and others. In a quantum computer, graph states are the logical codewords of many quantum error-correcting codes[3] and form a universal resource for measurement-based quantum computing[4]. The action of single-qubit Clifford operations on graph states is well-understood and can be characterized completely in terms of operations called *local complementations*, acting on the corresponding graph[5]. When faced with a class of objects and an action on them it is natural to consider the orbits induced by this action. The orbit of graph states under single-qubit Clifford operations can be studied by considering the orbits of simple graphs under local complementations, through the mapping mentioned above. The orbits of graph states have been studied in for example[6]. There the motivation came from quantum error correction: graph states can be mapped to stabilizer codes and moreover, the number of orbits for a given number of qubits is equal to the number of *equivalent* stabilizer codes. This gives a method to count the number of inequivalent stabilizer codes on a fixed number of qubits. In[6] the number of inequivalent stabilizer codes is computed for up to 12 qubits by counting the number of orbits of graphs under local complementations.

Furthermore, when studying entanglement measures and the equivalence of quantum states under local operations, the orbits of graphs states under single-qubit Cliffords is naturally an important question. In the excellent survey on graph states [7] it is stated that the computational complexity of generating the orbit of a given graph states is unknown. Here we show that given a graph G, counting the number of graph states equivalent to $|G\rangle$ under single-qubit Clifford operations, i.e. deciding the size of the orbit, is #ℙ-Complete. #ℙ-Complete problems are of great interest in the field of quantum computing. The reason being that the problem of boson-sampling[8], efficiently solvable using a quantum computer, has very strong similarities with the #ℙ-Complete problem of computing the permanent of a matrix[9].

RELATED WORK

The action of single-qubit Clifford operations on graph states was characterized by Van den Nest et al. in[5], where it was shown that these operations acting on a graph state can be completely described by the action of local complementations on the corresponding graph. Furthermore, in[10], Van den Nest et al. used this fact to extend the efficient algorithm by Bouchet for deciding equivalence of graphs under local complementations[11] to an efficient algorithm for deciding equivalence of graph states under single-qubit Clifford operations.

If one also allows for single-qubit Pauli measurements and classical communication, the problem turns out to be equivalent[12] to the known graph theory problem of deciding if a graph is a vertex-minor[13,14] of another. We have previously used this fact to show that deciding if a graph state $|H\rangle$ can be reached from another $|G\rangle$ using only single-qubit Clifford operations, single-qubit Pauli measurements and classical communication (LC + LPM + CC) is NP-Complete, even if $|H\rangle$ is restricted to be (1) a Greenberger-Horne-Zeilinger (GHZ) state on a fixed subset of the qubits of $|G\rangle$[15], (2) a GHZ-state on some subset of the qubits of $|G\rangle$[16] and (3) the tensor product of Bell pairs between fixed qubits[17].

However, even if a problem is ℕℙ-Complete, one can often find efficient algorithms for certain restrictions of the problem. A general concept is that of fixed-parameter tractability, where an algorithm solving a hard problem is shown to have a runtime $\mathcal{O}(f(r)\cdot \mathrm{poly}(n))$, where f is some computable function, r is some parameter of the input and n is the size of the input. For ℕℙ-Complete problems, $f(r)$ is necessarily super-polynomial in n, unless $\mathbb{P} = \mathbb{NP}$. Nonetheless, a fixed-parameter tractable problem can therefore be solved in polynomial time on inputs where the parameter r is bounded. An extremely powerful result in this context is that of Courcelle[18], which states that any graph problem, expressible in a certain rich logic, called monadic second-order logic (MS),

can be solved in time $\mathcal{O}(f(\mathrm{rwd}(G))\cdot|V(G)|^3)$, where $\mathrm{rwd}(G)$ is the rank-width[14] of G and $|V(G)|$ is the number of vertices of G. In[13] Courcelle and Oum showed that the vertex-minor problem is expressible in MS and therefore that it is fixed-parameter tractability in the rank-width of the input graph. It turns out that the rank-width of a graph G equals one plus the *Schmidt-rank width* the graph state $|G\rangle$[19]. Using these results, we applied Courcelle's theorem to the problem of transforming graph states under LC + LPM + CC in[12] and thus showed that this problem is fixed-parameter tractable in the Schmidt-rank width of the input graph state.

In this paper we will focus on the computational complexity of counting the number of graph states equivalent to some graph state using only single-qubit Clifford operations. We point out that, since the property of whether a graph is locally equivalent to another is also expressible in MS[12,13], Courcelle's machinery can also be applied to this problem. In fact, Courcelle's theorem also holds for counting the number of satisfying solutions[18], which is what we are interested in here. The details for how to apply Courcelle's theorem to the problem at hand, we leave for another paper. Here, we instead show that the problem is #ℙ-Complete, and thus has no efficient algorithm in the general case, unless $\mathbb{P}=\mathbb{NP}$.

Overview

In section II we recall how graph states and single-qubit Cliffords relate to graphs and local complementations. In section III we review the graph theoretical notion of an isotropic system and relate this to stabilizer and graph states. In section IV we review the complexity class #ℙ-Complete. In section V we prove our main result that counting the number of graph states equivalent under single-qubit Cliffords is #ℙ-Complete.

Notation

We use the following notation for sets of consecutive natural numbers.

$$[n]=\{i\in\mathbb{Z}\,:\,0\leq i<n\} \tag{1}$$

For a vertex u in a graph $G=(V,E)$ we will denote the *neighborhood*, i.e. the adjacent vertices as

$$N_G(v)=\{u\in V:(u,v)\in E\}. \tag{2}$$

Furthermore given a subset $X\subseteq V$ we use the following notation for the symmetric difference of the neighborhoods of the vertices in X

$$N_G(X)=\bigtriangleup_{v\in X}N_G(v). \tag{3}$$

Given a graph G, we denote by $\overline{G}$ the *complementary* graph, i.e. the graph with vertex-set V and edge-set

$$\overline{E}=\{(u,v)\in V\times V:u\neq v\wedge(u,v)\notin E\}. \tag{4}$$

The Pauli matrices are denoted as

$$I=\begin{pmatrix}1&0\\0&1\end{pmatrix},\qquad X=\begin{pmatrix}0&1\\1&0\end{pmatrix},$$
$$Y=\begin{pmatrix}0&-\mathrm{i}\\\mathrm{i}&0\end{pmatrix},\qquad Z=\begin{pmatrix}1&0\\0&-1\end{pmatrix}. \tag{5}$$

The single-qubit Pauli group $\mathscr{P}_1$ consists of $\{\mathrm{i}^kI,\mathrm{i}^kX,\mathrm{i}^kY,\mathrm{i}^kZ\}$ for $k\in\mathbb{Z}_4$ together with matrix-multiplication. Single-qubit unitaries that take elements of $\mathscr{P}_1$ to elements of $\mathscr{P}_1$ are called single-qubit Clifford operations and formally form the normalizer of $\mathscr{P}_1$. The n-qubit Pauli group $\mathscr{P}_n$ is the n-fold single-qubit Pauli group whose elements are the tensor-products of elements of $\mathscr{P}_1$.

II. GRAPH STATES

Here we review graph states and their properties under single-qubit Clifford operations. We start with reviewing stabilizer states: a superset of graph states.

A. Stabilizer states

A stabilizer state $|\mathscr{S}\rangle$ on n qubits is defined by its stabilizer group $\mathscr{S}$, which is a subgroup of the Pauli group $\mathscr{P}_n$[3]. The stabilizer state is defined to be a state such that it is an eigenstate of all elements of $\mathscr{S}$ with an eigenvalue of $+1$, i.e. $s|\mathscr{S}\rangle=|\mathscr{S}\rangle$ for $s\in\mathscr{S}$. To avoid $|\mathscr{S}\rangle$ being a trivial zero state there are two requirements of $\mathscr{S}$, (1) $-I\notin\mathscr{S}$ and (2) all elements of $\mathscr{S}$ should commute[20]. Furthermore, for $|\mathscr{S}\rangle$ to be a unique state (up to a global phase), $\mathscr{S}$ needs to be of size 2^n and can therefore be described by n independent generators. As an example consider the stabilizer group $\mathscr{S}_0$ generated by $X\otimes X$ and $Z\otimes Z$. One can check that $\mathscr{S}_0$ describes the state

$$|\mathscr{S}_0\rangle=\frac{1}{\sqrt{2}}(|0\rangle\otimes|0\rangle+|1\rangle\otimes|1\rangle) \tag{6}$$

B. Graph states

A graph state is a multi-partite quantum state $|G\rangle$ which is described by a graph G, where the vertices of G correspond to the qubits of $|G\rangle$[7]. The graph state is formed by initializing each qubit $v\in V(G)$ in the state $|+\rangle_v=\frac{1}{\sqrt{2}}(|0\rangle_v+|1\rangle_v)$ and for each edge $(u,v)\in E(G)$ applying a controlled phase gate between qubits u and v. Importantly, all the controlled phase gates commute and are invariant under changing the control- and target-qubits of the gate. This allows the edges describing these gates to be unordered and undirected. Formally, a graph state $|G\rangle$ is given as

$$|G\rangle=\prod_{(u,v)\in E(G)}C_Z^{(u,v)}\left(\bigotimes_{v\in V(G)}|+\rangle_v\right), \tag{7}$$

where $C_Z^{(u,v)}$ is a controlled phase gate between qubit u and v, i.e.

$$C_Z^{(u,v)} = |0\rangle\langle 0|_u \otimes \mathbb{I}_v + |1\rangle\langle 1|_u \otimes Z_v \tag{8}$$

and Z_v is the Pauli-Z matrix acting on qubit v. As an example, the graph state described by the complete graph on two vertices K_2 is single-qubit Clifford equivalent to each of the four Bell pairs since

$$|K_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle_a \otimes |+\rangle_b + |1\rangle_a \otimes |-\rangle_b) = H_b \left|\Phi^+\right\rangle_{ab} \tag{9}$$

where $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$, H_b is a Hadamard gate on qubit b and where

$$\left|\Phi^+\right\rangle_{ab} = \frac{1}{\sqrt{2}}(|0\rangle_a \otimes |0\rangle_b + |1\rangle_a \otimes |1\rangle_b). \tag{10}$$

A graph state is also a stabilizer state[7] with a stabilizer group generated by

$$g_v = X_v \prod_{u\in N_v} Z_u \quad \text{for} \quad v \in V(G) \tag{11}$$

Furthermore, any stabilizer state is single-qubit Clifford equivalent to some graph state[5].

Importantly here is that the above in fact gives a bijective mapping from graphs to graph states. Formally we have the following theorem.

Lemma II.1. *Two graphs states $|G\rangle$ and $|G'\rangle$ are equal if and only if their corresponding graphs G and G' are equal.* ◇

Proof. Let $|G\rangle$ and $|G'\rangle$ be two graph states. If G and G' have differing vertex-sets then clearly $|G\rangle$ and $|G'\rangle$ are different since they are states on different sets of qubits. Assume now that G and G' are graphs with the same vertex-set V. The inner product between $|G\rangle$ and $|G'\rangle$ will then be given as[21]

$$\left\langle G\middle|G'\right\rangle = \left(\bigotimes_{v\in V}\langle +|_v\right) \prod_{(u,v)\in E(G)} C_Z^{(u,v)} \prod_{(u,v)\in E(G')} C_Z^{(u,v)} \left(\bigotimes_{v\in V}|+\rangle_v\right). \tag{12}$$

Using the fact that $C_Z^{(u,v)}$ commute and square to identity for any (u,v) we find that the above equation evaluates to

$$\left\langle G\middle|G'\right\rangle = \left(\bigotimes_{v\in V}\langle +|_v\right)\left|G+G'\right\rangle \tag{13}$$

where $G+G'$ is the graph with vertex-set V and edge-set $E(G)\Delta E(G')$ with Δ being the symmetric difference. The state $|G+G'\rangle$ is equal to $\bigotimes_{v\in V}\langle +|_v$ if and only if $G+G'$ is the empty graph. One can see this by for example considering the Schmidt-rank for a bipartition which separates some adjacent vertices in $G+G'$, since this would be one for $\bigotimes_{v\in V}|+\rangle_v$ and greater than one for $|G+G'\rangle$. We therefore have that $\langle G|G'\rangle$ is one if and only if G and G' are equal. □

It turns out that single-qubit Clifford operations on graph states can be completely captured by an operation called *local complementation* on the corresponding graphs.

Definition II.1 (Local complementation)**.** *A local complementation τ_v is a graph operation specified by a vertex v, taking a graph G to $\tau_v(G)$ by replacing the induced subgraph on the neighborhood of v, i.e. $G[N(v)]$, by its complement. The neighborhood of any vertex u in the graph $\tau_v(G)$ is therefore given by*

$$N(u)^{(\tau_v(G))} = \begin{cases} N(u)\Delta(N(v)\setminus\{u\}) & \text{if } (u,v)\in E(G) \\ N(u) & \text{else} \end{cases}, \tag{14}$$

where Δ denotes the symmetric difference between two sets. ◇

The action of a local complementation on a graph induces the following sequence of single-qubit Clifford operations on the corresponding graph state

$$U_v^{(G)} = \exp\left(-\mathrm{i}\frac{\pi}{4}X_v\right)\prod_{u\in N_v}\exp\left(\mathrm{i}\frac{\pi}{4}Z_u\right), \tag{15}$$

where X_v and Z_v are the Pauli-X and Pauli-Z matrices acting on qubit v respectively. Concretely, $U_v^{(G)}$ has the following action on the graph state $|G\rangle$

$$U_v^{(G)}|G\rangle = |\tau_v(G)\rangle. \tag{16}$$

We call two graphs which are related by some sequence of local complementations *locally equivalent*. For example star graphs and complete graphs on the same vertex-set are locally equivalent. As shown by Bouchet in[11], deciding if two graphs are locally equivalent can be done in cubic time in the number of vertices of the graphs. The following theorem, proven by Van den Nest in[5], captures the relation between single-qubit Cliffords on graph states and local complementations on graphs.

Theorem II.1 (Van den Nest[5])**.** *Two graph states $|G\rangle$ and $|G'\rangle$ are single-qubit Clifford equivalent if and only if the two graphs G and G' are locally equivalent.* ◇

As a direct corollary of lemma II.1 and theorem II.1, we therefore have the following result.

Corollary II.1.1. *Let G be a graph with its corresponding graph state $|G\rangle$. The number of graph states which are single-qubit Clifford equivalent to $|G\rangle$ is equal to the number of locally equivalent graphs to G.* ◇

Using corollary II.1.1 we can now restrict ourselves to the problem of counting locally equivalent graphs.

III. ISOTROPIC SYSTEMS

Our main result of this paper makes great use of the concept of an *isotropic system*. In this section we review the definition of an isotropic system and its relation to locally equivalent graphs and graph states. What is interesting to point out, and perhaps never before noted, is that an isotropic system is in fact equivalent to a stabilizer group, see below. For this reason, results obtained for isotropic systems can be of great use when studying stabilizer states and graph states.

Isotropic systems were introduced by Bouchet in[22]. The power of isotropic systems is that they exactly capture the equivalence classes of graphs under local complementation or equivalently equivalence classes of graphs states under single-qubit Clifford operations. Any isotropic system has a set of *fundamental graphs* which are all locally equivalent. As shown in[23], any graph G is a fundamental graph of some isotropic system S. Furthermore, given a isotropic system S with a fundamental graph G, another graph G' is a fundamental graph of S if and only if G and G' are locally equivalent[24].

In section III B we review the formal definition of an isotropic system. In the sections leading up to this, we first set the notation and introduce certain concepts needed.

A. Finite fields and Pauli groups

Let $\{0,1,\omega,\omega^2\}$ be the elements of the finite field of four elements $\mathbb{F}_4$. Under addition we have that $x+x=0$ for any element x of $\mathbb{F}_4$ and furthermore we have that $1+\omega=\omega^2$. Under multiplication we have that $x^i\cdot x^j = x^{i+j \pmod 3}$ for any element $x\neq 0$. An useful inner product on $\mathbb{F}_4$ is the trace inner product, defined as

$$\langle a,b\rangle = a\cdot b^2 + a^2\cdot b \tag{17}$$

What is interesting in relation to quantum information theory is that addition in $\mathbb{F}_4$ corresponds to matrix multiplication in the Pauli group, up to a global phase. Furthermore, the trace inner product captures whether two elements of the Pauli group commute or not. To see this, consider the following mapping α from $\mathbb{F}_4$ to the Pauli group.

$$\alpha(0)=I,\ \alpha(1)=X,\ \alpha(\omega)=Y,\ \alpha(\omega^2)=Z. \tag{18}$$

One can then check that

$$\alpha(a)\alpha(b) = \mathrm{i}^k\alpha(a+b) \quad \text{where} \quad k\in\mathbb{Z}_4. \tag{19}$$

Furthermore, we have that

$$[\alpha(a),\alpha(b)]=0 \quad \Leftrightarrow \quad \langle a,b\rangle = 0. \tag{20}$$

where $[\cdot,\cdot]$ is the commutator. Similarly we can also define a map from the elements of the vector space $\mathbb{F}_4^n$. Let $\mathbf{v}$ be a vector of $\mathbb{F}_4^n$ and $\mathbf{v}^i$ be the i'th element of $\mathbf{v}$. We then define a map from $\mathbb{F}_4^n$ to the Pauli group on n qubits as follows.

$$\alpha(\mathbf{v}) = \bigotimes_{i=0}^{n}\alpha(\mathbf{v}^i). \tag{21}$$

Both eq. (18) and eq. (20) hold for α also acting on vectors of $\mathbb{F}_4^n$.

B. Isotropic systems

Formally an isotropic system is defined as follows[25].

Definition III.1 (isotropic system)**.** *A subspace S of $\mathbb{F}_4^n$ is said to be an isotropic system if: (1) for all $\mathbf{v},\mathbf{w}\in S$ it holds that $\langle\mathbf{v},\mathbf{w}\rangle=0$ and (2) S has dimension n.* ◇

Now note that $\alpha(S)\equiv\{\alpha(\mathbf{v})\ :\ \mathbf{v}\in S\}$ forms a stabilizer group (ignoring global phases). This is because condition (1) of the above definition says that all the elements of $\alpha(S)$ commute, by eq. (20), as required by a stabilizer group.

C. Complete and Eulerian vectors

Here we review some further concepts related to isotropic systems which we need for the proof of our main result. Certain isotropic systems can be represented as Eulerian tours on 4-regular multi-graphs (see section III E). These Eulerian tours correspond to what are called Eulerian vectors of the isotropic system. The definition of a Eulerian vector also generalizes to all isotropic systems, even those not representable as Eulerian vectors on 4-regular multi-graphs. In order to give the definition of an Eulerian vector we must first define what are called complete vectors.

Definition III.2 (complete vector)**.** *A vector $\mathbf{v}$ of $\mathbb{F}_4^n$ such that $\mathbf{v}^i\neq 0$ for all $i\in[n]$ is called* complete. ◇

We will also need to notion of supplementary vectors.

Definition III.3 (supplementary vectors)**.** *Two vectors $\mathbf{v},\mathbf{w}$ of $\mathbb{F}_4^n$ are called supplementary if (1) they are complete and (2) $\mathbf{v}^i\neq\mathbf{w}^i$ for all $i\in[n]$.* ◇

Complete vectors come equipped with a notion of *rank*. To define the rank of a complete vector we need some further notation. Let $\mathbf{v}$ be a complete vector of $\mathbb{F}_4^n$. Let X be a subset of $[n]$ and let $\mathbf{v}[X]$ be a vector such its elements are

$$(\mathbf{v}[X])^i = \begin{cases}\mathbf{v}^i & \text{if } i\in X\\ 0 & \text{else}\end{cases} \tag{22}$$

We can now define the following set

$$V_{\mathbf{v}} = \{\mathbf{v}[X]\ :\ X\subseteq[n]\}. \tag{23}$$

Note that $V_{\mathbf{v}}$ forms a subspace of $\mathbb{F}_4^n$. The rank of $\mathbf{v}$ with respect to S is now defined as the dimension of the intersection of $V_{\mathbf{v}}$ and S.

Definition III.4 (rank of a complete vector)**.** *Let $\mathbf{v}$ be a complete vector of $\mathbb{F}_4^n$. The rank of $\mathbf{v}$, $r_S(\mathbf{v})$, with respect to S is the dimension of the intersection of $V_{\mathbf{v}}$ and S, i.e.*

$$r_S(\mathbf{v}) = \dim(V_{\mathbf{v}}\cap S) \tag{24}$$

◇

We are now ready to formally define an Eulerian vector of an isotropic system.

Definition III.5 (Eulerian vector)**.** *A complete vector $\mathbf{v}$ of $\mathbb{F}_4^n$, such that $r_S(\mathbf{v})=0$ is called an Eulerian vector of S.* ◇

D. Fundamental graphs

As mentioned, the power of isotropic systems is that their *fundamental graphs* are exactly the graphs in an equivalence class under local complementations. Here we review the definition of fundamental graphs of an isotropic system, which is defined by a Eulerian vector through a *graphic description*.

Definition III.6 (graphic presentation)**.** *Let G be a graph with vertices*[26] $V(G) = [n]$ *and* $\mathbf{v}, \mathbf{w}$ *be supplementary vectors of* $\mathbb{F}_4^n$*. The following is then an isotropic system*

$$S = \{\mathbf{v}[N_X] + \mathbf{w}[X] \ : \ X \subseteq V(G)\}. \tag{25}$$

The tuple $(G, \mathbf{v}, \mathbf{w})$ *is called a graphic presentation of the isotropic system S. Furthermore, G is called a fundamental graph of S.* ◇

Note that

$$\{\mathbf{v}[N_v] + \mathbf{w}[\{v\}] \ : \ v \in V(G)\} \tag{26}$$

forms a basis for S. In[23] it is shown that if $(G, \mathbf{v}, \mathbf{w})$ is a graphic presentation of S then $\mathbf{v}$ is a Eulerian vector of S. Furthermore, it is shown that given an Eulerian vector $\mathbf{v}$ of S, there exists a unique graphic presentation $(G, \tilde{\mathbf{v}}, \mathbf{w})$ of S such that $\mathbf{v} = \tilde{\mathbf{v}}$. Note that two Eulerian vectors can still represent the same fundamental graph.

The observant reader will notice the close similarity between eq. (26) and eq. (11). Indeed, consider the two supplementary vectors $\mathbf{v}_{\omega^2} = (\omega^2, \ldots, \omega^2)$ and $\mathbf{w}_1 = (1, \ldots, 1)$. Now, let G be an arbitrary graph and S be the isotropic system with $(G, \mathbf{v}_{\omega^2}, \mathbf{w}_1)$ as a graph presentation, as by eq. (25). We will here call S_G the *canonical* isotropic system of G. We then have that $\alpha(S_G)$ is exactly the stabilizer group of the graph state $|G\rangle$. To see this note that

$$g_v = \alpha(\mathbf{v}[N_v] + \mathbf{w}[\{v\}]) \quad \forall v \in V(G) \tag{27}$$

using eq. (21) and eq. (11).

As mentioned before, any two graphs G and G' are locally equivalent if and only if they are fundamental graphs of the same isotropic system[23]. Furthermore, any graph is a fundamental graph of some isotropic system. We therefore see that, for any isotropic system S, there exists a surjective map from the set of Eulerian vectors of S to the graphs in an equivalence class of graphs under local complementations. As described in the section III F, for certain isotropic systems, the number of Eulerian vectors equals the number of Eulerian tours on some 4-regular multi-graph. We will make use of this fact to prove our main result.

E. Graphic systems

Certain isotropic systems, called *graphic* systems, can be represented as a 4-regular multi-graphs. There is then a surjective map from the Eulerian tours on the 4-regular multi-graph to the fundamental graphs of the graphic system[23]. The set of fundamental graphs for graphic systems is exactly the set of circle graphs[23]. We will briefly describe this relation here, however leaving out some details which are out of scope for this paper. For details on graphic systems see[23], for circle graphs see[27,28] and it's relation to graph states see[15].

A 4-regular multi-graph F is a multi-graph (i.e. can contain multi-edges and self-loops) where each vertex has degree 4, i.e. $|N_F(v)| = 4 \ \forall v \in V(F)$. A *walk* P on F is an alternating sequence of vertices and edges

$$P = v_1 e_1 v_2 \ldots e_k v_{k+1} \tag{28}$$

such that e_i is incident on v_i and v_{i+1} for $i \in [n]$. A *trail* is a walk with no repeated edges. A *tour* is a *trail* such that $v_1 = v_{k+1}$. An Eulerian tour is a tour which traverses all edges of F. A 4-regular multi-graph has at least one Eulerian tour, since all vertices have even degree[29]. Any Eulerian tour on a 4-regular multi-graph F traverses each vertex exactly twice, except for the vertex which is both the start and the end of the tour. The order in which these vertices are traversed is captured by the *induced double-occurrence word*.

Definition III.7 (Induced double-occurrence word)**.** *Let F be a connected 4-regular multi-graph on k vertices $V(F)$. Let U be a Eulerian tour on F of the form*

$$U = x_1 e_1 x_2 \ldots x_{2k-1} e_{2k-1} x_{2k} e_{2k} x_1. \tag{29}$$

with $x_i \in V(F)$ and $e_i \in E(F)$. From a Eulerian tour U as in eq. (29) *we define an induced double-occurrence word as*

$$m(U) = x_1 x_2 \ldots x_{2k-1} x_{2k}. \tag{30}$$

◇

We can now define a mapping from an induced double-occurrence word $m(U)$ to a graph $\mathscr{A}(m(U))$, where the edges of $\mathscr{A}(m(U))$ are exactly the pairs of vertices in $m(U)$ which alternate. Formally we have the following definition.

Definition III.8 (Alternance graph)**.** *Let $m(U)$ be the induced double-occurrence word of some Eulerian tour U on some 4-regular multi-graph F. Let now $\mathscr{A}(m(U))$ be a graph with vertices $V(F)$ and the edges E, such that for all $(u,v) \in V(F) \otimes V(F)$, $(u,v) \in E$ if and only if $m(U)$ is of the form*

$$\ldots u \ldots v \ldots u \ldots v \ldots \quad \text{or} \quad \ldots v \ldots u \ldots v \ldots u \ldots, \tag{31}$$

i.e. u and v are alternating in $m(U)$. We will sometimes also write $\mathscr{A}(U)$ as short for $\mathscr{A}(m(U))$. ◇

It turns out that the set of alternating graphs induced by the Eulerian tours on some 4-regular multi-graph F are exactly the fundamental graphs of some isotropic system S. We then say that S is *associated* to F. An isotropic system that is associated to some 4-regular multi-graph is called *graphic*. There is a formal mapping λ from a 4-regular multi-graph F together with an ordering T of its edges to an isotropic system $S = \lambda_T(F)$. However, this mapping is rather complex and the interested reader can find the details in[23]. What is important here is that, for any T, there is a bijective mapping from the Eulerian tours of F to the Eulerian vectors of $S = \lambda_T(F)$[23]. This statement is implied by the results developed in[23], however in a non-trivial way. For this reason, we here point out why this follows in section III F.

F. Eulerian decompositions

An Eulerian decomposition D of a 4-regular multi-graph F is a set of tours on F such that each edges of F is in exactly one of the tours. As shown in[23], given a 4-regular multi-graph on n vertices, any Eulerian decomposition can be describe by a complete vector of $\mathbb{F}_4^n$. To see this, note that an Eulerian decomposition on a 4-regular multi-graph F can be described by, for each vertex v in F, a pairing of the incident edges on v. For example, let e_v^1, e_v^2, e_v^3 and e_v^4 be the four edges incident on the vertex v and consider now a pairing where e_v^1 is paired with e_v^2 and e_v^3 with e_v^4, written as $((e_v^1,e_v^2),(e_v^3,e_v^4))$. We can then construct an Eulerian decomposition by walking along the vertices and edges of F and when we reach v through the edge e_v^1 we should exit through the edge e_v^2 and vice versa. Note that there are exactly three different ways to pair the four edges of a vertex and we can thus represent this pairing by a non-zero element of $\mathbb{F}_4$ as

$$1 \mapsto ((e_v^1,e_v^2),(e_v^3,e_v^4)) \tag{32}$$

$$\omega \mapsto ((e_v^1,e_v^3),(e_v^2,e_v^4)) \tag{33}$$

$$\omega^2 \mapsto ((e_v^1,e_v^4),(e_v^2,e_v^3)). \tag{34}$$

Furthermore we can represent the pairings of all the vertices of F as a complete vector of $\mathbb{F}_4^n$. Note that the Eulerian decomposition for a given complete vector depends on the ordering of the edges incident on a vertex. However this ordering simply changes which Eulerian decomposition is related to which complete vector, but not the fact that we now have a mapping from complete vectors of $\mathbb{F}_4^n$ to Eulerian decompositions of F. This ordering T is exactly the ordering mentioned in section III E, which can be used to map F to an isotropic system $S = \lambda_T(F)$. Let now $D_T(\mathbf{v})$ be the Eulerian decomposition induced by the complete vector $\mathbf{v}$ as described above.

Importantly here, as stated in[23], is that, for any Eulerian decomposition D of F there is a unique complete vector $\mathbf{v} \in \mathbb{F}_4^n$ such that $D = D_T(\mathbf{v})$, for a fixed T. Furthermore, the Eulerian decomposition $D_T(\mathbf{v})$ consists of an Eulerian tour if and only if $\mathbf{v}$ is an Eulerian vector of $S = \lambda_T(F)$. We therefore have the following corollary.

Corollary III.1.1 (Implied by[23])**.** *Let F be a 4-regular multi-graph with n vertices. Let T be an ordering of its vertices as described above and formally defined in[23]. The number of Eulerian tours on F equals the number of Eulerian vectors of $S = \lambda_T(F)$.* ◇

Proof. From above we already know that a Eulerian decomposition of F is described by exactly one complete vector of $\mathbf{F}_4^n$ through the mapping D_T. Furthermore, the Eulerian decomposition $D_T(\mathbf{v})$ consists of exactly one Eulerian tour if and only if $\mathbf{v}$ is a Eulerian vector of $S = \lambda_T(F)$. Finally the number of Eulerian decompositions of F that consists of exactly one Eulerian tour are clearly equal to the number of Eulerian tours on F. □

G. Number of locally equivalent graphs

In[30] Bouchet showed that $l(G)$, the number of graphs locally equivalent to some graph G, is given by

$$l(G) = \frac{e(S)}{k(S)} \tag{35}$$

where S is an isotropic system with G as a fundamental graph and $e(S)$ is the number of Eulerian vectors of S and $k(S)$ is an index of S. We also have that if S and S' are isotropic systems which both have G as a fundamental graph, then $e(S) = e(S')$ and $k(S) = k(S')$. Using the canonical isotropic system we introduced in section III D we can therefore also define

$$e(G) \equiv e(S_G), \quad k(G) \equiv k(S_G), \tag{36}$$

such that

$$l(G) = \frac{e(G)}{k(G)}. \tag{37}$$

Below, we review the definition of $k(G)$ as presented in[30]. The index $k(G)$ of a graph is given as

$$k(G) = \begin{cases} \left|\nu(G)^\perp\right| + 2 & \text{if } G \text{ is in the class } \mu \\ \left|\nu(G)^\perp\right| & \text{else} \end{cases} \tag{38}$$

where the bineighborhood space $\nu(G)$ and the graph class μ are defined below and $^\perp$ denotes the orthogonal complement. Firstly, we introduce the following notation that will help simplify some later expressions.

Definition III.9. *Let $S = \{s_1,\ldots,s_k\}$ be a set and $P \subseteq S$ a subset of S. We will associate to P a binary vector $\overrightarrow{P}$ of length k as follows:*

$$\overrightarrow{P}^{(i)} = \begin{cases} 1 & \text{if } s_i \in P \\ 0 & \text{else} \end{cases} \tag{39}$$

where $\overrightarrow{P}^{(i)}$ is the i-th element of $\overrightarrow{P}$. We denote the number of nonzero elements of $\overrightarrow{P}$ as $\left|\overrightarrow{P}\right|$, such that $\left|\overrightarrow{P}\right| = |P|$. ◇

Here, the base-set S will here be the vertices V of a graph G and from the context it will always be clear which graph. We will also use $\cdot$ to denote the element-wise product between two binary vectors, such that

$$\overrightarrow{P_1 \cap P_2} = \overrightarrow{P_1} \cdot \overrightarrow{P_2}. \tag{40}$$

To define the graph class μ we first need to review the notion of a bineighborhood space.

Definition III.10 (bineighborhood space)**.** *Let $G = (V,E)$ be a simple graph and $\overline{G} = (V,\overline{E})$ the complementary graph of G. For any $u,v \in V$ let*

$$\nu_G(e) = \overrightarrow{N_G(u) \cap N_G(v)}. \tag{41}$$

For any subset $E' \subseteq E \cup \overline{E}$, let

$$\nu_G(E') = \sum_{e \in E'} \nu_G(e). \tag{42}$$

We will sometimes write $\nu(e)$ or $\nu(E')$ if it is clear which graph is considered. A subset $C \subseteq E$ such that the number of edges in C incident to any vertex in G is even is called a cycle. We denote the set of cycles of G as $\mathscr{C}(G)$. Let $\mathfrak{V} = \mathbb{Z}_2^{|V|}$ be the binary vector space of dimensions $|V|$ and consider the two subspaces

$$\overline{\mathfrak{E}} = \{\nu(E') : E' \subseteq \overline{E}\}, \quad \mathfrak{C} = \{\nu(C) : C \subseteq \mathscr{C}(G)\} \tag{43}$$

The bineighborhood space $\nu(G)$ is defined as the sum of the two subspaces $\overline{\mathfrak{E}}$ and $\mathfrak{C}$, i.e.

$$\nu(G) = \overline{\mathfrak{E}} + \mathfrak{C}. \tag{44}$$

◇

Finally the graph class μ is defined as follows.

Definition III.11 (graph class μ)**.** *A simple graph $G = (V, E)$ is said to be in the class μ if:*

1. $d_G(v) = 1 \pmod 2$ *for every vertex $v \in V$. I.e. all vertices in G should have an odd degree.*
2. $|\nu(e)| = 0 \pmod 2$ *for all edges $e \in \overline{E}$. I.e. for every edge (u,v), not in G, the symmetric difference of the neighborhoods of u and v should have an even size.*
3. $|\nu(C)| = |C| \pmod 2$ *for all cycles $C \in \mathscr{C}(G)$. I.e., for all cycles C of G, the number of non-zero elements of the $\nu(C)$ and the number of edges of C should both be even or both be odd.*

◇

IV. COMPLEXITY

The problems in $\mathbb{NP}$ are decision problems where YES-instances to the problem have proofs that can be checked in polynomial time. For example the SAT-problem is in $\mathbb{NP}$, where one is asked to decide if a given boolean formula has a satisfying assignment of its variables[31]. On the other hand, problems where the NO-instances have proofs that can be checked in polynomial time are the problems in co-$\mathbb{NP}$. A problem is said the be $\mathbb{NP}$-Complete if (1) it is in $\mathbb{NP}$ and (2) any other problem in $\mathbb{NP}$ can be reduced to this problem in polynomial time. $\mathbb{NP}$-Complete problems are therefore informally the hardest problem in $\mathbb{NP}$.

#$\mathbb{P}$ problems are the *counting* versions of the $\mathbb{NP}$ problems. For example, the *counting* version of SAT (#SAT) is to compute how many satisfying assignments a given boolean formula has. #$\mathbb{P}$-Complete problems are the problems in #$\mathbb{P}$ for which any other problem in #$\mathbb{P}$ can be polynomially reduced to. For example #SAT is #$\mathbb{P}$-Complete[9]. Note that #$\mathbb{P}$-Complete is at least as hard as $\mathbb{NP}$-Complete, since if we know the number of satisfying assignments we know if at least one exists. Other well-known problems #$\mathbb{P}$-Complete are for example computing the permanent of a given boolean matrix or finding how many perfect matchings a given bipartite graph has[9].

Recently, #$\mathbb{P}$-Complete problems have been the interest of the quantum computing community due to the problem of boson sampling[8]. The boson sampling problem can be solved efficiently on a quantum computer. Furthermore, the boson sampling problem can be related to the problem of estimating the permanent of a complex matrix. Since computing the permanent is in general a #$\mathbb{P}$-Complete problem and thus believed the be infeasible to solve efficiently on a classical computer, the boson sampling problem is therefore is a strong candidate for a problem showing 'quantum supremacy'.

V. COUNTING THE NUMBER OF LOCALLY EQUIVALENT GRAPHS IS #ℙ-COMPLETE

Here we show our following main result.

Theorem V.1 (main)**.** *Counting the number, $l(G)$, of locally equivalent graphs to a given graph G is #$\mathbb{P}$-Complete.* ◇

We do this by showing that counting the number of Eulerian tours of a 4-regular multi-graph can be reduced in polynomial time to computing $l(G)$, where G is a circle graph. Since counting the number of Eulerian tours of a 4-regular multi-graph is #$\mathbb{P}$-Complete[32], the result follows. By corollary II.1.1 we have the following corollary.

Corollary V.1.1. *Counting the number of graph states which are single-qubit Clifford equivalent to a given graph state $|G\rangle$ is #$\mathbb{P}$-Complete.* ◇

Proof. Directly implied by theorem V.1 and corollary II.1.1. □

A. Reducing # of Eulerian tours to # of local equivalent graphs

Here we show how the problem of computing the number of Eulerian tours on a 4-regular multi-graph can be reduced in polynomial time to the problem of computing the number of locally equivalent graphs to some circle graph and thus provide the proof for theorem V.1.

Proof of theorem V.1. From corollary III.1.1 we know that for any 4-regular multi-graph F, there exists an isotropic system $S = \lambda_T(F)$ such that the number of Eulerian vectors $e(S)$ equals the number of Eulerian tours on F. Let now G be a fundamental graph of S. We then have that $e(G) = e(S)$, by eq. (36) and[23]. Furthermore, recall the G is necessarily also an alternance graph induced by some Eulerian tour on F, see section III E. We can therefore compute the number of Eulerian tours on F by computing $l(G) \cdot k(G)$, as by eq. (37). As we show below, we can both find G and compute $k(G)$ in polynomial time from which the theorem follows.

We can find G in polynomial time as follows.

1. Find an Eulerian tour U on F, can be done in polynomial time by Fleury's algorithm[33].

2. Construct the alternance graph $G = \mathscr{A}(U)$ induced by U, can be done in polynomial time, see[15].

In the rest of this section we show that $k(G)$ can be computed in polynomial time from which the main result follows.We will start by showing that determining if a graph G is in the class μ, see definition III.11, can be done in time $\mathscr{O}(|V|^5)$. Note that there might be even faster ways to compute this, but we are here only interested to show that this can be done in polynomial time. We assume that the graph $G = (V,E)$ is represented by its adjacency matrix. To check if G is in the class μ one needs to check the three properties in definition III.11:

1. Checking if all vertices have odd degree can be done in $\mathscr{O}(|V|^2)$ time.

2. Checking if $|\nu(e)|$ is even for all edges can be done in $\mathscr{O}(|V|^3)$ time, since there are $\mathscr{O}(|V|^2)$ edges and computing $|\nu(e)|$ can be done in linear time[34].

3. For the last property is not directly clear whether this can be done in polynomial time since we need to a priori check the property $|\nu(C)| = |C| \pmod 2$ for all cycles in G, which might be exponentially many. As we will now show, we only need to check the property for the cycles in a cycle basis of G. A cycle basis $\mathscr{C}\mathscr{B} = \{C_1,\ldots,C_k\}$, where $k = \mathscr{O}(|V|^2)$, is a set of cycles such that any cycle of G can be written as the symmetric difference of the elements of a subset of $\mathscr{C}\mathscr{B}$. As shown in[35] a cycle basis of an undirected graph can be found in $\mathscr{O}(|V|^2)$ time. Thus any cycle of G can be written as

$$\bigtriangleup_{C'\in\mathscr{C}} C' \tag{45}$$

where $\mathscr{C}$ is a subset of the cycle basis $\mathscr{C}\mathscr{B}$. We then have that

$$\nu\left(\bigtriangleup_{C'\in\mathscr{C}} C'\right) = \sum_{C'\in\mathscr{C}} \nu(C'). \tag{46}$$

Thus we need to show that for any $\mathscr{C}\mathscr{B}$

$$\left|\sum_{C'\in\mathscr{C}} \nu(C')\right| = \left|\bigtriangleup_{C'\in\mathscr{C}} C'\right| \pmod 2 \quad \forall \mathscr{C} \subseteq \mathscr{C}\mathscr{B} \tag{47}$$

if and only if

$$|\nu(C)| = |C| \pmod 2 \quad \forall C \in \mathscr{C}\mathscr{B}. \tag{48}$$

Lets first show that eq. (47) implies eq. (48). Equation (47) states that the equation holds for every subset $\mathscr{C}$ of the elements of the cycle basis $\mathscr{C}\mathscr{B}$. In particular it should hold for the singletons $\mathscr{C} = \{C\}$, where $C \in \mathscr{C}\mathscr{B}$. Note that this directly implies eq. (48). For the rest of this section we now prove that eq. (48) implies eq. (47). We will do this by induction on the size of $\mathscr{C}$. This is obviously true if $|\mathscr{C}| = 1$. Lets therefore assume that the statement is true for $|\mathscr{C}| \leq k$ which we will show implies that it is also true for $|\mathscr{C}| = k+1$. Lets assume that $\mathscr{C}$ is a subset of $\mathscr{C}\mathscr{B}$ of size $k+1$ and that $\tilde{C}$ is an element of $\mathscr{C}$. Lets then consider the left-hand side of eq. (47)

$$\left|\sum_{C'\in\mathscr{C}} \nu(C')\right| = \left|\sum_{C'\in\mathscr{C}\setminus\{\tilde{C}\}} \nu(C') + \nu(\tilde{C})\right|. \tag{49}$$

We will now make use of the fact that the size of the symmetric difference of two sets S_1 and S_2 is $|S_1 \Delta S_2| = |S_1| + |S_2| - 2|S_1 \cap S_2|$. Expressed in terms of binary vectors this relation reads $\left|\overrightarrow{S_1} + \overrightarrow{S_2}\right| = \left|\overrightarrow{S_1}\right| + \left|\overrightarrow{S_2}\right| - 2\left|\overrightarrow{S_1} \cdot \overrightarrow{S_2}\right|$. We therefore have that eq. (49) evaluates to

$$\left|\sum_{C'\in\mathscr{C}\setminus\{\tilde{C}\}} \nu(C') + \nu(\tilde{C})\right| = \left|\sum_{C'\in\mathscr{C}\setminus\{\tilde{C}\}} \nu(C')\right| + \left|\nu(\tilde{C})\right| - 2\left|\left(\sum_{C'\in\mathscr{C}\setminus\{\tilde{C}\}} \nu(C')\right) \cdot \nu(\tilde{C})\right| \tag{50}$$

Wr can then see that eq. (48) implies eq. (47) since when taking $\pmod 2$, the last term in the above expression vanishes and the two first evaluate to

$$\left|\bigtriangleup_{C'\in\mathscr{C}\setminus\{\tilde{C}\}} C'\right| + \left|\tilde{C}\right| \tag{51}$$

where we used the induction hypothesis. By a similar argument one can see that the expression in eq. (51) equals $\pmod 2$

$$\left|\bigtriangleup_{C'\in\mathscr{C}} C'\right|. \tag{52}$$

Thus the total time to check property 3 in definition III.11 is $\mathscr{O}(|V|^5)$. To see this, note that we need to check $|\nu(C)| = |C| \pmod 2$ for all $C \in \mathscr{C}\mathscr{B}$, which contains $\mathscr{O}(V^2)$ elements. To compute $|\nu(C)|$, we com-

pute $\nu(e)$, in linear time, for each of the $\mathcal{O}(V^2)$ elements of C, and add these together, also in linear time.

In addition to deciding if the graph G is in the class μ, we also need to compute $\left|\nu(G)^{\perp}\right|$ to determine $k(G)$. This can be done by first finding bases for the subspaces $\overline{\mathfrak{E}}$ and $\mathfrak{C}$. For $\overline{\mathfrak{E}}$ a basis can be found as $\left\{\overrightarrow{\{e\}} : e \in \overline{E}\right\}$. As stated above we can also find a basis for $\mathfrak{C}$, i.e. the cycle basis, in $\mathcal{O}(|V|^2)$ time. From the bases for $\overline{\mathfrak{E}}$ and $\mathfrak{C}$ we can find a basis for $\nu(G)$ in $\mathcal{O}(|V|^3)$ time, by Gaussian elimination. The number of basis vectors we found for $\nu(G)$ is then the dimension of $\nu(G)$. From the dimension of $\nu(G)$ we can find the dimension of $\nu(G)^{\perp}$ as

$$\dim(\nu(G)^{\top}) = |V| - \dim(\nu(G)) \tag{53}$$

and finally the size of $\nu(G)^{\perp}$ as

$$\left|\nu(G)^{\perp}\right| = 2^{\dim(\nu(G)^{\perp})}. \tag{54}$$

Thus there exist an algorithm to compute $k(G)$ with running time $\mathcal{O}(|V|^5)$. This then implies that computing the number of Eulerian tours in a 4-regular multi-graph can be reduced in polynomial time to computing the number of locally equivalent graphs to some circle graph, by using eq. (37), and therefore theorem V.1. □

VI. CONCLUSION

We have shown that counting the number of graph states equivalent under single-qubit Clifford operations is #ℙ-Complete. To do this we have made heavy use of certain concepts in graph theory, mainly developed by Bouchet. As it turns out these concepts, for example isotropic systems, are highly relevant for the study of stabilizer and graph states. We hope that this paper can serve as not only a proof of our main theorem but also as a reference for those in quantum information theory interested in finding use for these graph theory concepts in their research.

ACKNOWLEDGMENTS

AD, JH and SW were supported by an ERC Starting grant, an NWO VIDI grant, and the Zwaartekracht QSC.

[1] D. Markham and B. C. Sanders, "Graph states for quantum secret sharing," Phys. Rev. A **78**, 042309 (2008).
[2] M. Christandl and S. Wehner, "Quantum anonymous transmissions," in *Advances in Cryptology - ASIACRYPT 2005*, edited by B. Roy (Springer Berlin Heidelberg, Berlin, Heidelberg, 2005) pp. 217–235.
[3] D. Gottesman, *Stabilizer Codes and Quantum Error Correction*, Ph.D. thesis, California Institute of Technology (2004).
[4] R. Raussendorf and H. J. Briegel, "A one-way quantum computer," Phys. Rev. Lett. **86**, 5188–5191 (2001).
[5] M. Van den Nest, J. Dehaene, and B. De Moor, "Graphical description of the action of local Clifford transformations on graph states," Physical Review A **69**, 022316 (2004).
[6] L. E. Danielsen and M. 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).
[7] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M. V. den Nest, H. J. Briegel, M. V. den Nest, and H. J. Briegel, "Entanglement in Graph States and its Applications," Quantum Computers, Algorithms and Chaos **162**, 1–99 (2006), arXiv:0602096 [quant-ph].
[8] S. Aaronson and A. Arkhipov, "The computational complexity of linear optics," Theory of Computing **9**, 143–252 (2013).
[9] L. Valiant, "The complexity of computing the permanent," Theoretical Computer Science **8**, 189 – 201 (1979).
[10] M. Van den Nest, J. Dehaene, and B. De Moor, "Efficient algorithm to recognize the local Clifford equivalence of graph states," Physical Review A **70**, 034302 (2004), arXiv:0405023 [quant-ph].
[11] A. Bouchet, "An efficient algorithm to recognize locally equivalent graphs," Combinatorica **11**, 315–329 (1991), arXiv:0702057v2 [cs].
[12] A. Dahlberg and S. Wehner, "Transforming graph states using single-qubit operations," Phil. Trans. R. Soc. A 376, One contribution of 15 to a discussion meeting issue 'Foundations of quantum mechanics and their impact on contemporary society' `http://dx.doi.org/10.1098/rsta.2017.0325`,`arxiv.org/abs/1805.xxxxx` (2018).
[13] B. Courcelle and S. il Oum, "Vertex-minors, monadic second-order logic, and a conjecture by Seese," Journal of Combinatorial Theory. Series B **97**, 91–126 (2007).
[14] S. I. Oum, "Rank-width and vertex-minors," Journal of Combinatorial Theory. Series B **95**, 79–100 (2005).
[15] A. Dahlberg, J. Helsen, and S. Wehner, "How to transform graph states using single-qubit operations: computational complexity and algorithms," arXiv preprint arXiv:1805.05306 (2018).
[16] A. Dahlberg, J. Helsen, and S. Wehner, "The complexity of the vertex-minor problem," arXiv preprint arXiv:1906.05689 (2019).
[17] A. Dahlberg, J. Helsen, and S. Wehner, "Transforming graph states to bell-pairs is np-complete," arXiv preprint arXiv:1907.08019 (2019).
[18] B. Courcelle and J. Engelfriet, *Graph Structure and Monadic Second-Order Logic: A Language Theoretic Approach*, 1st ed. (Cambridge University Press, New York, NY, USA, 2011).
[19] M. Van den Nest, W. Dür, G. Vidal, and H. J. Briegel, "Classical simulation versus universality in measurement-based quantum computation," Physical Review A **75**, 012337 (2007), arXiv:0608060 [quant-ph].
[20] Elements of the Pauli group either commute or anti-commute.
[21] We assume here that when iterating over a set, the order of the elements is always the same.
[22] A. Bouchet, "Isotropic systems," European Journal of Combinatorics **8**, 231 – 244 (1987).
[23] A. Bouchet, "Graphic presentations of isotropic systems," Journal of Combinatorial Theory, Series B **45**, 58–76 (1988).
[24] That is, if and only if $|G\rangle$ and $|G'\rangle$ are equivalent under single-qubit Clifford operations.
[25] The definition here is equivalent to the original one in[22], however we use a slightly different notation than Bouchet used 30 years ago.
[26] Note that we can always choose such a labeling of the vertices of G.
[27] A. Bouchet, "Circle Graph Obstructions," Journal of Combinatorial Theory, Series B **60**, 107–144 (1994).
[28] M. C. Golumbic, *Algorithmic graph theory and perfect graphs*, 2nd ed. (North-Holland Publishing Co., 2004).
[29] L. Euler, "Solutio problematis ad geometriam situs pertinentis," Comment. Acad. Sci. U. Petrop. 8 , 128–140 (1741).
[30] A. Bouchet, "Recognizing locally equivalent graphs," Discrete Mathematics **114**, 75–86 (1993).
[31] S. A. Cook, "The complexity of theorem-proving procedures," in *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71 (ACM, New York, NY, USA, 1971) pp. 151–158.
[32] Q. Ge and D. Stefankovic, "The Complexity of Counting Eulerian Tours in 4-Regular Graphs," in *LATIN 2010: Theoretical Informatics* (Springer Berlin Heidelberg, Berlin, Heidelberg, 2010) pp. 638–649.
[33] M. Fleury, "Deux problemes de Geometrie de sitation," Journal de mathematiques elementaires **2nd**, 257–261 (1883).
[34] By taking the inner product of the corresponding rows in the adjacency matrix.

[35] K. Paton, "An algorithm for finding a fundamental set of cycles of a graph," Communications of the ACM **12**, 514–518 (1969).