Transforming graph states to Bell-pairs is NP-Complete

Axel Dahlberg Jonas Helsen Stephanie Wehner

QuTech - TU Delft, Lorentzweg 1, 2628CJ Delft, The Netherlands

Critical to the construction of large scale quantum networks, i.e. a quantum internet, is the development of fast algorithms for managing entanglement present in the network. One fundamental building block for a quantum internet is the distribution of Bell pairs between distant nodes in the network. Here we focus on the problem of transforming multipartite entangled states into the tensor product of bipartite Bell pairs between specific nodes using only a certain class of local operations and classical communication. In particular we study the problem of deciding whether a given graph state, and in general a stabilizer state, can be transformed into a set of Bell pairs on specific vertices using only single-qubit Clifford operations, single-qubit Pauli measurements and classical communication. We prove that this problem is $\mathbb{NP}$-Complete.

1 Introduction

Entanglement takes center stage in the modern understanding of quantum mechanics. Apart from its usefulness as a theoretical tool, entanglement can also be seen as a resource that can be harnessed for secure communication and many other tasks, see e.g. [1], not achievable by any protocol using only classical communication. One can imagine a network of quantum-enabled nodes, a quantum internet, generating entanglement and harnessing it to perform tasks. However, entanglement over large distances is very difficult to produce and even with rapidly improving technology, the amount of entanglement available in a network will for the foreseeable future be the limiting factor when performing the tasks mentioned above. Moreover entanglement comes in different classes that are not necessarily mutually inter-convertible by operations performed locally on the nodes. These two considerations: (1) the scarcity of entanglement as a resource and (2) the lack of local inter-convertibility of classes of entanglement sets the stage for the present work. In this paper we assume that we already have some existing shared entangled state in a quantum network and we ask the question of whether this state can be transformed into a set of Bell pairs between specific nodes, using only a restricted set of local operations. Examples of such a situation can be found in [2, 3], where an approach is presented of first probabilistically generating a large graph state and then transforming this to the desired target state using local operations. In [2] this target state is precisely a set of Bell pairs between multiple pairs of nodes.

Any decision on this transformation process must be made fast, since entanglement decays over time [4]. Hence there is a need for fast algorithms to decide whether different entangled states can be converted into each other by local operations. In [5], measurement-based quantum network coding was introduced, where one step in the procedure includes transforming general graph states into Bell-pairs using single-qubit Clifford operations, single-qubit Pauli measurement and classical communication: LC + LPM + CC. However, the computational complexity of finding the correct operations or even deciding if it can be done, was never mentioned. In this work we answer this question. Specifically we study the mathematic problem of deciding whether a given 'graph state'

can be decomposed into a series of Bell pairs on specific vertices, using only LC + LPM + CC. In particular we determine the computational complexity of this problem. We consider this set of operations since they are on many hardware platforms the simplest and fastest operations to perform. For example on Nitrogen-Vacancies in diamond, single-qubit gates on one of the qubits (electron) is many orders faster than two-qubits gates and single-qubit gates on the other qubits (nuclear spins) [6]. Furthermore, restricting to only Cliffords, instead of allowing general unitary operations, is natural on systems where, for example, the qubits are logical qubits of some error-correcting code. Since for many quantum error-correction codes, doing non-Clifford operations is a costly process involving, in many cases, the consumption of magic states. While it is true that the high relative cost of logical non-Clifford gates is not a universal phenomena, for instance 3D color codes [7, 8] have the ability to perform transversal T-gates, it is the case for many popular codes. Moreover, over the years many techniques from graph theory have been put to bear on specifically the behavior of graph states under LC + LPM + CC [9, 10, 11, 12, 13, 14]. This gives us the ability to ask very precise questions and also answer them.

Our main result is that we prove that the problem of deciding whether a given graph state can be converted into Bell Pairs using only LC + LPM + CC (BELLVM) is in general NP-Complete. This means that unless $\mathbb{P} = \mathbb{NP}$, there will be no efficient algorithm for solving this problem on arbitrary graph states. In order to prove our results we make heavy use of results in algorithmic graph theory, and in fact we prove new graph theoretical results in the process of proving our main theorem.

Related work

The study of single-qubit operations on graph states started with the work by Raussendorf and Briegel in [15] and lead to the development of measurement-based quantum computing [16]. In [10] Van den Nest et al. showed how the action of single-qubit Clifford operations can be described purely by the graph operation *local complementation* on the corresponding graph. Bouchet had developed back in 1991 ([17]) an efficient algorithm to decide if two graphs are equivalent under local complementations which was used by Van den Nest et al. in [18] to present an efficient algorithm for deciding the equivalence of graph states under single-qubit Clifford operations. Adding single-qubit Pauli measurements and classical communication to the set of operations is equivalent to also allow for vertex-deletion on the corresponding graph, next to the local complementations [11]. Graphs that can be reached by the action of local complementations and vertex-deletions is a known concept in graph theory and are called vertex-minors [19, 20]. Using the fact that the graph states reachable from a graph state $|G\rangle$ using LC + LPM + CC are exactly captured by the vertex-minors of G, we have previously shown that deciding if a GHZ-state can be reached from a given graph state using LC + LPM + CC is NP-Complete, both if (1) the GHZ-state is on a given set of vertices [12] and (2) the GHZ-state is allowed to be on any subset of a fixed size [14].

Due to the different *target state* considered in this paper, the computational complexity of BELLVM does not follow from [12]. Concretely, a tensor-product of Bell pairs is not equivalent to a GHZ-state under the operations studied here. For this reason we will here perform a different reduction to prove that BELLVM is NP-complete compared to the one used to prove the same for the problem in [12]. In [12] we reduced from the problem of finding a Hamiltonian tour on a three-regular graph. Here on the other hand we reduce from the problem of finding disjoint paths between sources and sinks in a graph. While doing so we also show new results, purely related to the disjoint path problem in graph theory.

However, the fact that a problem is NP-Complete does not mean that there is no efficient algorithm, if one allows to put certain restrictions on the input to the problem. Such a restriction is for example that a certain parameter r of the input should be bounded. If a problem can be solved in polynomial time on inputs where this parameter r is bounded, the problem is said to be fixed-parameter tractable (in r). Concretely, there might exist an algorithm solving the problem in time $\mathcal{O}(f(r) \cdot \text{poly}(n))$, where f is a computable function and n is the size of the input. For NP-Complete problems, $f(r)$ is necessarily super-polynomial in n, unless $\mathbb{P} = \mathbb{NP}$. Individual problems have been shown to be fixed-parameter tractable in various parameters. However, Courcelle [21] showed that

a large class of graph problems are fixed-parameter tractable in what is called the rank-width [19] of the graph. In fact, any graph problem, expressible in a certain rich logic (MS)[1], can be solved in time $\mathcal{O}(f(\mathrm{rwd}(G)) \cdot |V(G)|^3)$, where $\mathrm{rwd}(G)$ is the rank-width of G and $|V(G)|$ is the number of vertices of G. In [20] Courcelle and Oum showed that the vertex-minor problem is fixed-parameter tractability in the rank-width of the input graph by showing that the problem is expressible in MS. The rank-width of a graph G equals one plus the *Schmidt-rank width* the graph state $|G\rangle$ [22]. Using these results, we applied Courcelle's theorem to the problem of transforming graph states under $LC + LPM + CC$ in [11] and thus showed that this problem is fixed-parameter tractable in the Schmidt-rank width of the input graph state.

Courcelle's theorem states that there exists an efficient algorithm for graph states with bounded Schmidt-rank width. However, a direct implementation of the algorithm from Courcelle's theorem is not usable in practice, due to a huge constant factor in the runtime [23]. In [12] we presented two efficient algorithms, not suffering from this huge constant factor, for the problem of deciding whether $|G\rangle$ can be transformed to $|G'\rangle$ using $LC + LPM + CC$ if: (1) $|G'\rangle$ is a GHZ-state and $|G\rangle$ has Schmidt-rank width 1 [12] and (2) $|G'\rangle$ is a GHZ-state of bounded size and G is a circle graph [12]. We point out that the second algorithm is in fact not captured by Courcelle's theorem since circle graphs have unbounded rank-width [12].

Here we show a hardness proof for the case when the target graph is the tensor product of some number of Bell pairs on specified qubits. However, out result is not only a negative one, since it can also give a hint for heuristic algorithms that can be used to solve the problem at hand, which do not suffer from a huge constant pre-factor in the runtime as for those based on Courcelle's theorem. In particular, since we show the relation between BELLVM and certain problems related to finding disjoint paths in a graph, we can make use of results for solving the disjoint path problem also for BELLVM. Even though the disjoint path problem is in general $\mathbb{NP}$-complete, there are many heuristic algorithms. For example in [24], a two-step approach using an integer linear program formulation of the problem with an evolutionary algorithm. By using the reduction we show in this paper, one might be able to use the same approach for the problem at hand.

Overview

In section 1.1 we review the basic theory of graph states and introduce the notion of a Bell-vertex-minor, a graph theoretical concept central to our results. In section 2 we review the notion of circle graphs and related concepts which we make use of in section 4. In section 3 we discuss the Edge-Disjoint path problem, a well-studied computational problem in algorithmic graph theory. In section 4 we show that a certain version of the Edge Disjoint Path problem can be polynomially reduced to BELLVM and in appendix A that this version of the Edge Disjoint Path problem is $\mathbb{NP}$-Complete, implying that BELLVM is $\mathbb{NP}$-hard. Finally, in section 5 we discuss the implications of our result.

Notation

Graphs are assumed to be simple unless otherwise indicated. The vertex-set of a graph $G = (V, E)$ is denoted $V = V(G)$ and the edge-set is denoted $E = E(G)$. Given a vertex v in a graph G we denote the neighborhood of v (the set of vertices adjacent to v in G) by $N_v^{(G)}$. If it is clear which graph the neighborhood concerns we sometimes omit G and simply write N_v. Given a graph G and a subset of its vertices V' we will denote the induced subgraph of G on those vertices by $G[V']$. We denote the fully connected graph on n vertices as K_n.

Throughout this paper we use the following notation for sets of consecutive natural numbers

$$[k, n] \equiv \{i \in \mathbb{N} : k \leq i \leq n\} \tag{1}$$

$$[n] \equiv \{i \in \mathbb{N} : 1 \leq i \leq n\} \tag{2}$$

---

[1]Monadic second-order logic

1.1 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 $|G\rangle$. 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

$$|G\rangle = \prod_{(u,v) \in E(G)} C_Z^{(u,v)} \left(\bigotimes_{v \in V(G)} |+\rangle_v \right), \tag{3}$$

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

$$C_Z^{(u,v)} = |0\rangle\langle 0|_u \otimes \mathbb{I}_v + |1\rangle\langle 1|_u \otimes Z_v \tag{4}$$

and Z_v is the Pauli-Z matrix acting on qubit v.

One advantage of considering graph states is that certain problems can be completely expressed in the language of graph theory, where one can make use of powerful existing tools and techniques. For example, what we make use of here, is the fact that single-qubit Clifford operations (LC), single-qubit Pauli measurements (LPM) and classical communication (CC): LC + LPM + CC, which take graph states to graph states, can be completely characterized by local complementations and vertex-deletions on the corresponding graphs [11]. Local complementation is defined as follows.

Definition 1.1 (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*

$$N_u^{(\tau_v(G))} = \begin{cases} N_u \Delta (N_v \setminus \{u\}) & \text{if } (u,v) \in E(G) \\ N_u & \text{else} \end{cases}, \tag{5}$$

where Δ denotes the symmetric difference between two sets. Given a sequence of vertices $\mathbf{v} = v_1 \ldots v_k$, we denote the induced sequence of local complementations, acting on a graph G, as

$$\tau_{\mathbf{v}}(G) = \tau_{v_k} \circ \cdots \circ \tau_{v_1}(G). \tag{6}$$

$\diamond$

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

$$U_v^{(G)} = \exp\left(-i\frac{\pi}{4}X_v\right) \prod_{u \in N_v} \exp\left(i\frac{\pi}{4}Z_u\right), \tag{7}$$

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 $|G\rangle$

$$U_v^{(G)} |G\rangle = |\tau_v(G)\rangle. \tag{8}$$

Measuring qubit v of a graph state $|G\rangle$ in the Pauli-X (or Pauli-Y, Pauli-Z basis), gives a stabilizer state that is single-qubit Clifford equivalent to a graph state $|G'\rangle$, where G' can be reached from G by a sequence of local complementations and vertex-deletions [9]. The state operations taking the post-measurement state to $|G'\rangle$ depends on the measurement outcome and acts on v and its neighborhood. This means classical communication is required to announce the measurement result at the vertex v to its neighboring vertices. Details can be found in [12], where we introduced to notion of a *qubit-minor* which captures exactly which graph states can be reached from some initial graph state under LC + LPM + CC. Formally we define a qubit-minor as:

Definition 1.2 (Qubit-minor [12])**.** *Assume $|G\rangle$ and $|G'\rangle$ are graph states on the sets of qubits V and U respectively. $|G'\rangle$ is called a qubit-minor of $|G\rangle$ if there exists an adaptive sequence of single-qubit Clifford operations (LC), single-qubit Pauli measurements (LPM) and classical communication (CC) that takes $|G\rangle$ to $|G'\rangle$, i.e.*

$$|G\rangle \xrightarrow[\text{LPM+CC}]{\text{LC}} |G'\rangle \otimes |\text{junk}\rangle_{V \setminus U}. \tag{9}$$

If $|G'\rangle$ is a qubit-minor of $|G\rangle$, we denote this as

$$|G'\rangle < |G\rangle. \tag{10}$$

◇

In [12] we have shown that the notion of qubit-minors for graph states is equivalent to the notion of *vertex-minors* for graphs.

Definition 1.3 (Vertex-minor)**.** *A graph G' is called a vertex-minor of G if and only if there exist a sequence of local complementations and vertex-deletions that takes G to G' [19, 20]. Since vertex-deletions can always be performed last in such a sequence (see [11]), an equivalent definition is the following: A graph G' is called a vertex-minor of G if and only if there exist a sequence of local complementations $\mathbf{v}$ such that $\tau_{\mathbf{v}}(G)[V(G')] = G'$. If G' is a vertex-minor of G we write this as*

$$G' < G. \tag{11}$$

◇

The relation between qubit-minor and vertex-minor is captured by the following theorem.

Theorem 1.4 ([12])**.** *Let $|G\rangle$ and $|G'\rangle$ be two graph states such that no vertex in G' has degree zero. Then $|G'\rangle$ is a qubit-minor of $|G\rangle$ if and only if G' is a vertex-minor of G, i.e.*

$$|G'\rangle < |G\rangle \quad \Leftrightarrow \quad G' < G. \tag{12}$$

◇

Note that one can also include the case where G' has vertices of degree zero. Let's denote the vertices of G' which have degree zero as I. We then have that

$$|G'\rangle < |G\rangle \quad \Leftrightarrow \quad G'[V(G') \setminus I] < G. \tag{13}$$

Theorem 1.4 is very powerful since it allows us to consider graph states under $\text{LC} + \text{LPM} + \text{CC}$, purely in terms of vertex-minors of graphs. We will therefore in the rest of this paper use the formalism of vertex-minors to study the computational complexity of transforming graph states using $\text{LC} + \text{LPM} + \text{CC}$.

1.2 Bell vertex-minors

In this paper we are interested in the question of whether a given graph state $|G\rangle$ can be transformed into some number of Bell pairs using $\text{LC} + \text{LPM} + \text{CC}$ between specific vertices. We will denote the following Bell pair on qubits a and b as

$$\left|\Phi^+\right\rangle_{ab} = \frac{1}{\sqrt{2}}(|0\rangle_a \otimes |0\rangle_b + |1\rangle_a \otimes |1\rangle_b). \tag{14}$$

We formally define the following main problem of this paper.

Problem 1.5 (BELLQM)**.** *Given a graph state $|G\rangle$ and a set of disjoint pairs $B = \{\{p_1, p_1'\}, \ldots, \{p_k, p_k'\}\}$. Let $|G_B\rangle$ be the state following state consisting of Bell pairs between each pair of B*

$$|G_B\rangle = \bigotimes_{\{p,p'\}\in B} |\Phi^+\rangle_{pp'} \tag{15}$$

Decide if $|G_B\rangle$ is a qubit-minor of G. ◇

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

$$|K_2\rangle = \frac{1}{\sqrt{2}}(|0\rangle_a \otimes |+\rangle_b + |1\rangle_a \otimes |-\rangle_b) = H_b |\Phi^+\rangle_{ab} \tag{16}$$

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

Using theorem 1.4 we can turn the question of transforming graph states to Bell pairs using $\text{LC} + \text{LPM} + \text{CC}$ i.e. BELLQM, into the question of whether a disjoint union of K_2's is a vertex-minor of some graph. Formally we have the following problem graph problem.

Problem 1.6 (BELLVM)**.** *Given a graph G and a set of disjoint pairs $B = \{\{p_1, p_1'\}, \ldots, \{p_k, p_k'\}\}$. Let G_B be the graph with vertices $V(G_B) = \bigcup_{i\in[k]}\{p_i, p_i'\}$ and edges $E(G_B) = \bigcup_{i\in[k]}\{(p_i, p_i')\}$. Decide if G_B is a vertex-minor of G.* ◇

To be precise $(|G\rangle, B)$ is a YES-instance of BELLQM, if and only if (G, B) is a YES-instance of BELLVM.
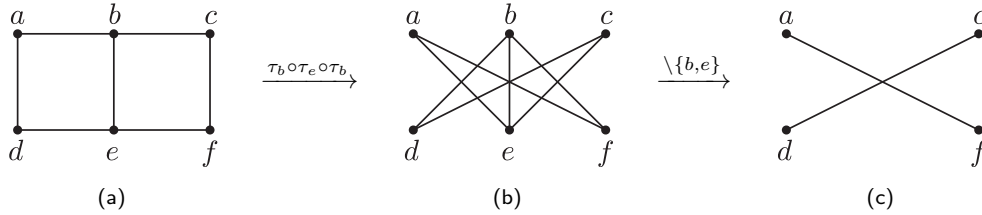

Figure 1: An example of a graph being transformed to a union of two K_2 graphs, using local complementations and vertex-deletions. The original graph in fig. 1a get transformed to the graph in fig. 1b by performing local complementations on the vertices b, e and then b again. Finally, when vertices b and e are deleted, the disjoint union of two K_2 graphs, on vertices $\{a, f\}$ and $\{c, d\}$ respectively, is reached (fig. 1c).

2 Circle graphs

Circle graphs are the graphs which can be described as a Eulerian tours on 4-regular multi-graphs, as described below. What is interesting in this context is that circle graphs are equivalent under local complementations if and only if they can be described as Eulerian tours on the same 4-regular multi-graph [25]. We will make use of this property to prove that BELLVM is $\mathbb{NP}$-Complete, and therefore also BELLQM, in section 4. Here we review circle graphs and certain properties we will later need. For more details on circle graphs, see for example [26, 27, 28], the book by Golumbic [29] or [12] for the use of circle graphs in the context of graph states.

A 4-regular multi-graph is a graph where each vertex has exactly four incident edges and can contain multiple edges between each pair of vertices or edges only incident to a single vertex (self-loops).

Definition 2.1 (Eulerian tour)**.** *Let F be a connected multi-graph. A* walk *W on F is an alternating sequence of vertices and edges*

$$W = v_1 e_1 v_2 \ldots e_k v_{k+1} \tag{17}$$

such that e_i is incident on v_i and v_{i+1} for $i \in [n]$. A tour on F is a closed walk, i.e. $v_1 = v_{k+1}$, without repeated edges. An Eulerian tour U on F is a tour that visits each edge in F exactly once. ◇

Any 4-regular multi-graph is Eulerian, i.e. has a Eulerian tour, since each vertex has even degree [30].

Furthermore, 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 2.2 (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*

$$U = x_1 e_1 x_2 \ldots x_{2k-1} e_{2k-1} x_{2k} e_{2k} x_1. \tag{18}$$

with $x_i \in V(F)$ and $e_i \in E(F)$. From a Eulerian tour U as in eq. (18) we define an induced double-occurrence word as

$$m(U) = x_1 x_2 \ldots x_{2k-1} x_{2k}. \tag{19}$$

◇

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

Definition 2.3 (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 $\mathcal{A}(m(U))$ 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*

$$\ldots u \ldots v \ldots u \ldots v \ldots \quad or \quad \ldots v \ldots u \ldots v \ldots u \ldots, \tag{20}$$

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

Figure 2 shows an example of a 4-regular multi-graph and one of its induced alternance graphs.
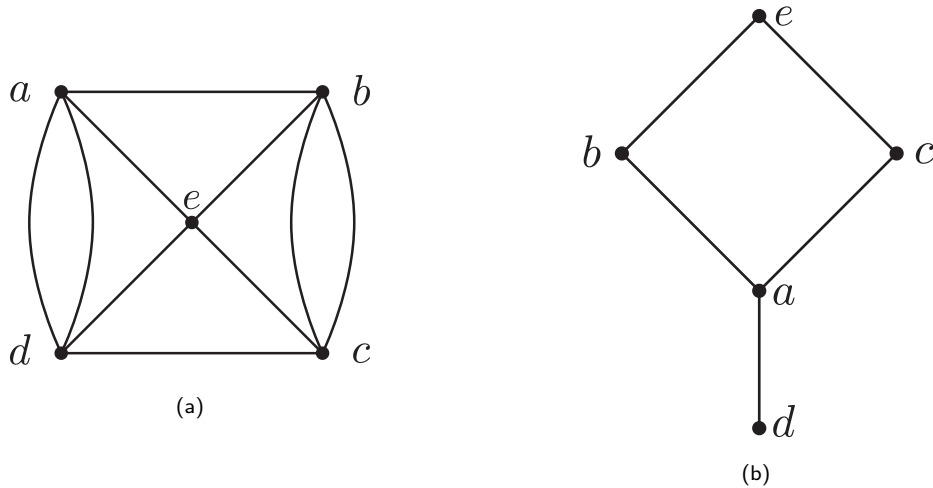

(a)

(b)

Figure 2: Figure 2a shows an example of a 4-regular multi-graph F, which has an Eulerian tour U_0 with an induced double-occurrence word $m(U) = adcbaebced$. The alternating vertex-pairs of $m(U_0)$ are thus (a, b), (a, c), (a, d), (b, e), (c, e) and their mirrors. The alternance graph $\mathcal{A}(m(U_0))$ is therefore the graph in fig. 2b.

Circle graphs are exactly the graphs which are the alternance graph described by some Eulerian tour on some 4-regular multi-graph [28]. Given a Eulerian tour U of some 4-regular multi-graph F and a subset V' of $V(F)$ we will sometimes write $m(U)[V']$ to mean the word $m(U)$ with all letters in $V(F) \setminus V'$ deleted. The notation is intentionally similar to that of taking an induced subgraph since the alternance graph of $m(U)[V']$ is exactly the induced subgraph of the alternance graph of $m(U)$, i.e.

$$\mathcal{A}\Big(m(U)[V']\Big) = \mathcal{A}\Big(m(U)\Big)[V']. \tag{21}$$

Importantly here is that one can answer whether a circle graph has a certain vertex-minor by considering the Eulerian tours of a certain 4-regular multi-graph. Formally we have the following theorem, which is proven in [12].

Theorem 2.4 ([12])**.** *Let F be a connected 4-regular multi-graph and let G be a circle graph such that $G = \mathcal{A}(U)$ for some Eulerian tour U on F. Then G' is a vertex-minor of G if and only if there exist a Eulerian tour U' on F such that*

$$G' = \mathcal{A}\Big(m(U')\big[V(G')\big]\Big). \tag{22}$$

$\diamond$

Using theorem 2.4, we can now ask what property a 4-regular multi-graph should have, such that it's induced alternance graphs are YES-instances to BELLVM, given a set of disjoint pairs P. As we show in lemma 4.4, this question will be directly related to a restricted version of the edge-disjoint path problem, which we define in the next section.

3 The Edge-disjoint path problem

In section 4 we show that BELLVM is $\mathbb{NP}$-Complete by reducing the 4-regular EDPDT (Edge Disjoint Paths with Disjoint Terminals) problem (see below) to BELLVM. We then show that that 4-regular EDPDT is $\mathbb{NP}$-Complete, see corollary 3.5, which therefore implies that the same is true for BELLVM. This is done by reducing the EDP (Edge Disjoint Path) problem to EDPDT and then EDPDT to 4-regular EDPDT. We begin by formally defining all problems just mentioned. We will denote the set of edges in a path $P = v_0 e_1 v_1 \ldots e_l v_l$ as $E(P) = \{e_1, \ldots, e_l\}$. Moreover, given two graphs $G = (V(G), E(G))$ and $D = (V(D), E(D))$ we denote by $G \cup D$ the graph formed by the vertices $V(G \cup D) = V(G) \cup V(D)$ and the edge multi-set $E(G \cup D) = E(G) \cup E(D)$. A path on a graph is a walk without repeated vertices. A closed path is called a circuit.

We first define the EDP (Edge Disjoint Path) problem.

Problem 3.1 (EDP)**.** *Let G and D be graphs such that $V(D) \subseteq V(G)$. Decide whether there exists a set of edge-disjoint circuits $\mathcal{C}$ on the graph $G \cup D$ such that every edge $e \in E(D)$ is part of exactly one circuit in $\mathcal{C}$.* $\diamond$
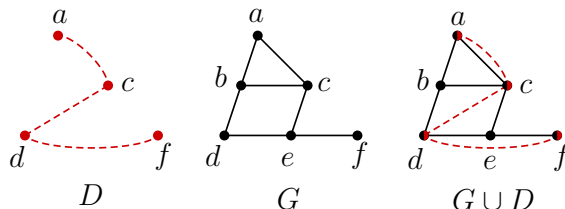

Figure 3: The EDP problem concerns deciding if there exists a set of edge-disjoint circuits in $G \cup D$ such that each edge in D is in part of exactly one of the circuits. In this example a satisfying solution is a set consisting of the three circuits $a(a,c)c(c,a)a$, $d(d,b)b(b,c)c(c,d)d$ and $d(d,e)e(e,f)f(f,d)d$.

Let (G, D) be a YES-instance of EDP and let $\mathcal{C}$ be the edge-disjoint circuits of problem 3.1. For each edge e in $E(D)$ denote the circuit in $\mathcal{C}$ which e is part of as C_e. These edge disjoint circuits C_e on $G \cup H$ correspond to edge disjoint paths on the graph G with *terminals*, i.e. beginning and ending points of the path, precisely at the vertices u, v s.t. $e = (u, v)$ for all $e \in D$. The EDP problem is known to be $\mathbb{NP}$-Complete even in the case where $G \cup D$ is a Eulerian graph [31]. The EDPDT problem is now the EDP problem but with the demand graph D restricted to be a disjoint union of connected graphs on two vertices, i.e. of the form $K_2^{\times k}$ for some k, such that the terminals are distinct.

Problem 3.2 (EDPDT). *Let G and $D = K_2^{\times k}$ be graphs such that $V(D) \subseteq V(G)$. Decide EDP with the instance (G, D).* ◇

The 4-regular EDPDT problem is then a further restriction of this problem to the case where $G \cup D$ is 4-regular (and $D = K_2^{\times k}$). Formally we have

Problem 3.3 (4-regular EDPDT). *Let G and $D = K_2^{\times k}$ be graphs such that $V(D) \subseteq V(G)$ and $G \cup D$ is 4-regular. Decide EDP with the instance (G, D).* ◇

Note that the 4-regularity of $G \cup D$ means that all vertices in G which are not in D must have degree 4, while all vertices also in D must have degree 3. We can equivalently formulate the 4-regular EDPDT problem as a problem involving only G and a set of terminal pairs on G, which will be a more useful definition for some of the proofs.

Problem 3.4 (4-regular EDPDT (equivalent formulation)). *Let G be a multi-graph where each vertex has degree either 3 or 4. Let $T = \{\{t_1, t_1'\}, \ldots, \{t_k, t_k'\}\}$ be the set of disjoint terminal pairs, such that $t_i, t_i' \in V(G)$ and $d_G(t_i) = d_G(t_i') = 3$ for all $i \in [k]$. Furthermore, assume that there are no other vertices of degree 3 in G, i.e. $\bigcup_{i \in [k]} \{t_i, t_i'\} = \{v \in V(G) : d_G(v) = 3\}$. Decide if there exists k edge-disjoint paths P_i for $i \in [k]$ such that the ends of P_i are t_i and t_i'.* ◇

In the appendix (theorem A.3) we show that EDP can be reduced to 4-regular EDPDT in polynomial time. A corollary to this theorem is therefore.

Corollary 3.5. *4-regular EDPDT is $\mathbb{NP}$-Complete.* ◇

4 BellVM is NP-Complete

Now we move on to proving the main result of this paper. The 4-regular EDPDT problem can be reduced to the BELLVM problem as shown below in theorem 4.3. Since we also show that BELLVM is in $\mathbb{NP}$ we conclude that BELLVM is $\mathbb{NP}$-Complete.

Corollary 4.1. *BELLVM is $\mathbb{NP}$-Complete.* ◇

Proof. Theorem 4.3 states that there exists a Karp reduction from 4-regular EDPDT to BELLVM. This implies that BELLVM is $\mathbb{NP}$-hard, by corollary 3.5. Since any instance of BELLVM is also an instance of the general vertex-minor problem, which is in $\mathbb{NP}$ [12], also BELLVM is in $\mathbb{NP}$ and hence $\mathbb{NP}$-Complete. □

As a directly corollary we then also have that.

Corollary 4.2. *BELLQM is $\mathbb{NP}$-Complete.* ◇

Proof. Follows directly from corollary 4.1 and theorem 1.4. □

The remaining part of this section will be used to prove theorem 4.3. The main part of the proof consists of proving lemma 4.4 which provides an explicit reduction from 4-regular EDPDT to BELLVM. This is done by constructing a 4-regular multi-graph $H_{(G,T)}$ from an instance (G, T) of 4-regular EDPDT, together with the graph $G_B = K_2^{\times |T|}$ on a certain subset B of the vertices of $H_{(G,T)}$. The construction is such that G_B is a vertex-minor of any alternance graph $\mathcal{A}(U)$ induced by an Eulerian tour U on $H_{(G,T)}$ if and only if (G, T) is a YES-instance of 4-regular EDPDT. In other words, $(\mathcal{A}(U), B)$ is a YES-instance of BELLVM if and only if (G, T) is a YES-instance of 4-regular EDPDT. What is left to show is that reduction can be done in polynomial time, which is done in the proof of theorem 4.3 below.

Theorem 4.3. *The 4-regular EDPDT problem is polynomially reducible to BELLVM.* ◇

Proof. From lemma 4.4 below we see that any *yes(no)*-instance of 4-regular EDPDT can be mapped to a *yes(no)*-instance of BELLVM. What remains to be shown is that this mapping can be performed in polynomial time. The reduction consist of the following to three steps:

1. Construct the multi-graph $H_{(G,T)}$ as defined in lemma 4.4.

2. Find an Eulerian tour U on $H_{(G,T)}$.

3. Construct the alternance graph $\mathcal{A}(U)$ induced by U.

Computing the graph $H_{(G,T)}$ can be done in polynomial time by simply adding the vertices and edges described in lemma 4.4. Note that the number of vertices and vertices in $H_{(G,T)}$ is $|V(G)| + 2 \cdot |T|$ and $|E(G)| + 4 \cdot |T|$. Finding an Eulerian tour U on $H_{(G,T)}$ can be done in polynomial time [32] in the size of $H_{(G,T)}$. Furthermore, constructing the alternance graph $\mathcal{A}(U)$ can be done in polynomial time as shown in [28]. □

Lemma 4.4. *Let $(G, T = \{\{t_1, t_1'\}, \ldots, \{t_k, t_k'\}\})$ be an instance of 4-regular EDPDT (see problem 3.4). As illustrated in fig. 4a, let $H_{(G,T)}$ be the multi-graph constructed from G by adding the distinct vertices $V_B = \bigcup_{i \in [k]} \{p_i, p_i'\}$ and the edges (t_i, p_i), (t_i', p_i'), (p_i, p_i') and[2] (p_i, p_i') for $i \in [k]$ and the edges (p_i', p_{i+1}) for $i \in [k-1]$ and the edge (p_k', p_1). Let G_B be the graph with vertices V_B and edges $\bigcup_{i \in [k]} \{(p_i, p_i')\}$. Let $\mathcal{A}(U)$ be a circle graph described by the Eulerian tour U on $H_{(G,T)}$. Then G_B is a vertex-minor of $\mathcal{A}(U)$ if and only if (G, T) is a yes-instance of 4-regular EDPDT.* ◇

─────────────────────

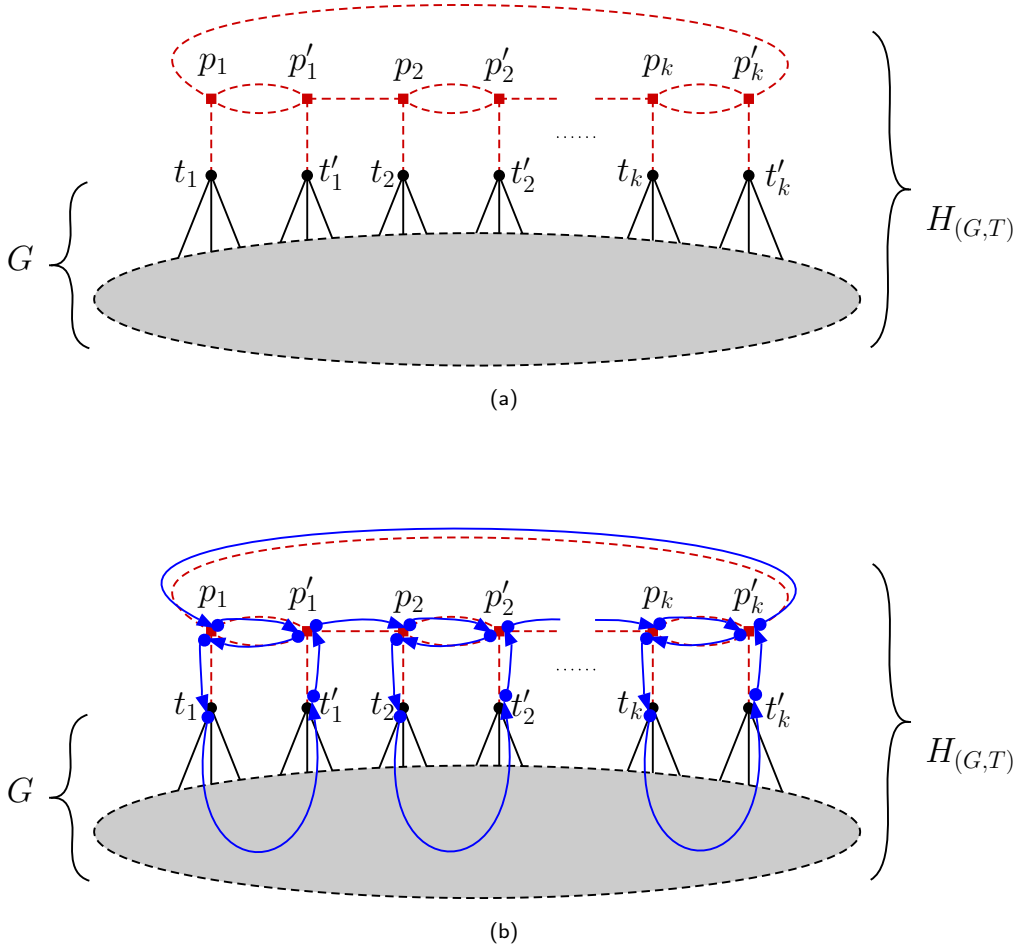[2]Note that (p_i, p_i') is added twice.

(a)

(b)

Figure 4: Figure 4a shows the graph construction used to reduce 4-regular EDPDT to BELLVM. The graph G is a graph with only vertices of degree 4 except the vertices $\{t_1, t'_1, t_2, t'_2, \ldots, t_k, t'_k\}$ which have degree 3. The vertices of G not in $\{t_1, t'_1, t_2, t'_2, \ldots, t_k, t'_k\}$ are visualized as a grey solid area in order to show that we put no further restrictions on G. $H_{(G,T)}$ is constructed from G by adding the red square vertices and red dashed edges. In fig. 4b the tour U_0 described by the word in eq. (23) is illustrated using solid blue arrows.

Proof. Let's first assume that (G, T) is a YES-instance of 4-regular EDPDT. This implies that there exist k edge-disjoint paths P_i for $i \in [k]$ such that the ends of P_i are t_i and t'_i. These paths also exist in $H_{(G,T)}$, since G is a subgraph of $H_{(G,T)}$. Consider then the tour U', see fig. 4b, described by the word

$$m(U') = p_1 p'_1 p_1 m(P_1) p'_1 p_2 \ldots p'_{k-1} p_k p'_k p_k m(P_k) p'_k. \tag{23}$$

Note U' is not necessarily an Eulerian tour but can be extended to one using the efficient Hierholzer's algorithm [33]. Denote the Eulerian tour obtained from U' by U. Consider now the induced subgraph $\mathcal{A}(U)[V_B]$. Using eq. (21) we know that this induced subgraph is the same as the alternance graph given by the induced word

$$m(U)[V_B] = p_1 p'_1 p_1 p'_1 p_2 \ldots p'_{k-1} p_k p'_k p_k p'_k. \tag{24}$$

We then have that $\mathcal{A}(m(U)[V_B])$, and therefore $\mathcal{A}(U)[V_B]$, is the graph G_B. To see this, note that the only alternances of $m(U)[V_B]$ are $p_i p'_i p_i p'_i$ for $i \in [k]$ and the edges of $\mathcal{A}(m(U)[V_B])$ are therefore (p_i, p'_i) for $i \in [k]$. Since $\mathcal{A}(U)[V_B]$ is an induced subgraph of $\mathcal{A}(U)$ it is by definition

also a vertex-minor of $\mathcal{A}(U)$. As we saw above, G_B is equal to $\mathcal{A}(U)[V_B]$ and is therefore also a vertex-minor of $\mathcal{A}(U)$.

Let's now instead assume that (G, T) is a NO-instance of 4-regular EDPDT. We will show that G_B is not a vertex-minor of $\mathcal{A}(U)$. Let U be an Eulerian tour on $H_{(G,T)}$, which exist since $H_{(G,T)}$ is 4-regular. Consider the sub-trails of U for which both ends are in V_B and that use no edges between vertices in V_B. Note that there are k such sub-trails since each vertex in V_B only has one edge which does not go to another vertex in V_B. Let's denote these sub-trails as $\tilde{P}_i$ for $i \in [k]$ and their ends as $\{p_i, \tilde{p}_i'\}$. From the assumption that (G, T) is a *no*-instance of 4-regular EDPDT, we know that $\{p_i, \tilde{p}_i'\} \neq \{p_i, p_i'\}$ for at least one $i \in [k]$. Consider now the graph $\mathcal{A}(U)[V_B]$. We will now show that $\mathcal{A}(U)[V_B]$ cannot be G_P. Since U was an arbitrary Eulerian tour on $H_{(G,T)}$, we then know that G_B is not a vertex-minor of U, by theorem 2.4.

To show that $\mathcal{A}(U)[V_B]$ is not G_B, consider the induced subgraph $H_{(G,T)}[V_B]$. Note that all vertices in $H_{(G,T)}[V_B]$ have degree 3, see fig. 4a. Let $H_{(U,B)}$ be the graph obtained from $H_{(G,T)}[V_B]$ by adding the edges $(p_i, \tilde{p}_i')$. Note now that $m(U)[V_B]$ also describes an Eulerian tour on $H_{(U,B)}$. Thus, $\mathcal{A}(U)[V_B]$ is the alternance graph $\mathcal{A}(U')$ for some Eulerian tour U' on $H_{(U,B)}$. Therefore, by using lemma 4.5 we know that $\mathcal{A}(U)[V_B]$ is not G_B. $\square$
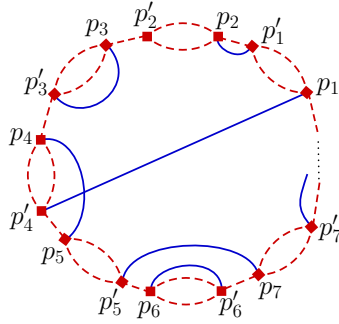

Figure 5: The graph $H_{(B,\tilde{E}_B)}$ used in lemma 4.5, where $\tilde{E}_B$ are the solid blue edges. Lemma 4.5 states that there exist an Eulerian tour U on $H_{(B,\tilde{E}_B)}$ such that $\mathcal{A}(U) = G_B$ if and only if all the solid blue edges $\tilde{E}_B$ are between vertices p_i and p_i' for $i \in [k]$.

Lemma 4.5. *Let B be the set $\{p_1, p_1', \ldots, p_k, p_k'\}$. Let C_B be the cycle graph on vertices $\bigcup_{i \in [k]} \{p_i, p_i'\}$ and with edges $\{(p_1, p_1'), (p_1', p_2), \ldots (p_k, p_k'), (p_k', p_1)\}$. Let $\tilde{C}_B$ be the multi-graph obtained from C_B by duplicating the edges $\{(p_i, p_i')\}_{i \in [k]}$. Let $\tilde{E}_B = \{(p_i, \tilde{p}_i')\}_{i \in [k]}$ be edges such that if they are added to $\tilde{C}_B$, the graph obtained is 4-regular. Denote the graph obtained from $\tilde{C}_B$ by adding the edges $\tilde{E}_B$ by $H_{(B,\tilde{E}_B)}$. Let G_B be the graph $(B, \bigcup_{i \in [k]} \{(p_i, p_i')\})$. Then for any Eulerian tour U on $H_{(B,\tilde{E}_B)}$, the graph $\mathcal{A}(U)$ is equal to G_B if and only if $\{p_i, \tilde{p}_i'\} = \{p_i, p_i'\}$ for all $i \in [k]$.* $\diamond$

Proof. First note that if $\mathcal{A}(U) = G_B$ for some Eulerian tour U on $H_{(B,\tilde{E}_B)}$ then $\mathcal{A}(U') = G_B$ for any Eulerian tour U' on $H_{(B,\tilde{E}_B)}$. This is because the graph G_B is invariant under local complementations.

Assume now that $\{p_i, \tilde{p}_i'\} = \{p_i, p_i'\}$ for all $i \in [k]$. Then the tour U described by the double occurrence word

$$m(U) = p_1 p_1' p_1 p_1' p_2 \ldots p_{k-1}' p_k p_k' p_k p_k'. \tag{25}$$

is an Eulerian tour on $H_{(B,\tilde{E}_B)}$. Furthermore, $\mathcal{A}(U) = G_B$.

Assume now on the other hand that $\{p_i, \tilde{p}_i'\} \neq \{p_i, p_i'\}$ for some $i \in [k]$. Let $(p_i, \tilde{p}_i')$ be a pair for which $\{p_i, \tilde{p}_i'\} \neq \{p_i, p_i'\}$. Consider now the tour

$$U' = p_i (p_i, \tilde{p}_i') P_C \tag{26}$$

on $H_{(B,\tilde{E}_B)}$ where P_C is a path in C_B which ends at p_i, i.e. $E(P_C) \cap \tilde{E}_B = \emptyset$. By Hierholzer's algorithm let U be an Eulerian tour on $H_{(B,\tilde{E}_B)}$ obtained by extending U'. Note now that $(p_i, \tilde{p}'_i)$ is an alternance in $m(U)$ and therefore an edge in $\mathcal{A}(U)$. But since $(p_i, \tilde{p}'_i)$ is not an edge in G_B, we know that $\mathcal{A}(U)$ is not equal to G_B. $\qquad\square$

5 Conclusion

The problem of transforming graph states to Bell-pairs using local operations is a problem with direct applications to the development of quantum networks or distributed quantum processors. Solutions to special cases of this problem have been considered in for example [2] and [5]. However, at least to our knowledge, the computational complexity of this problem was previously unknown. Here we show that deciding whether a given graph state $|G\rangle$ can be transformed into a set of Bell pairs on a given set of vertices, using only single-qubit Clifford operations, single-qubit Pauli measurements and classical communication is $\mathbb{NP}$-Complete. In fact, we show that the problem remains $\mathbb{NP}$-Complete if G is a circle graph.

Acknowledgements

References

[1] Stephanie Wehner, David Elkouss, and Ronald Hanson. Quantum internet: A vision for the road ahead. *Science*, 362(6412):eaam9288, Oct 2018.

[2] Mihir Pant, Hari Krovi, Don Towsley, Leandros Tassiulas, Liang Jiang, Prithwish Basu, Dirk Englund, and Saikat Guha. Routing entanglement in the quantum internet. *npj Quantum Information*, 5(1):25, 2019.

[3] Mihir Pant, Don Towsley, Dirk Englund, and Saikat Guha. Percolation thresholds for photonic quantum computing. *Nature Communications*, 10(1):1070, 2019.

[4] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, Cambridge, 10th anniversary edition edition, 2010.

[5] Takaaki Matsuo, Takahiko Satoh, Shota Nagayama, and Rodney Van Meter. Analysis of measurement-based quantum network coding over repeater networks under noisy conditions. *Physical Review A*, 97:13, 10 2017.

[6] N. Kalb, A. A. Reiserer, P. C. Humphreys, J. J. W. Bakermans, S. J. Kamerling, N. H. Nickerson, S. C. Benjamin, D. J. Twitchen, M. Markham, and R. Hanson. Entanglement distillation between solid-state quantum network nodes. *Science*, 356(6341):928–932, jun 2017.

[7] H. Bombin and M. A. Martin-Delgado. Exact topological quantum order in $d = 3$ and beyond: Branyons and brane-net condensates. *Phys. Rev. B*, 75:075103, Feb 2007.

[8] Héctor Bombín. Gauge color codes: optimal transversal gates and gauge fixing in topological stabilizer codes. *New Journal of Physics*, 17(8):083002, aug 2015.

[9] M. Hein, W. Dür, J. Eisert, R. Raussendorf, M Van den Nest, H. J. Briegel, M. Van den Nest, and H. J. Briegel. Entanglement in Graph States and its Applications. *Quantum Computers, Algorithms and Chaos*, pages 1–99, 2006.

[10] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Graphical description of the action of local clifford transformations on graph states. *Physical Review A*, 69(2):022316, 2004.

[11] Axel Dahlberg and Stephanie 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.

[12] Axel Dahlberg, Jonas Helsen, and Stephanie Wehner. How to transform graph states using single-qubit operations: computational complexity and algorithms. *arXiv preprint arXiv:1805.05306*, 2018.

[13] F Hahn, A Pappa, and J Eisert. Quantum network routing and local complementation. *arXiv preprint arXiv:1805.04559*, 2018.

[14] Axel Dahlberg, Jonas Helsen, and Stephanie Wehner. The complexity of the vertex-minor problem. *arXiv preprint arXiv:1906.05689*, 2019.

[15] Hans J. Briegel and Robert Raussendorf. Persistent entanglement in arrays of interacting particles. *Phys. Rev. Lett.*, 86:910–913, Jan 2001.

[16] Robert Raussendorf and Hans J. Briegel. A one-way quantum computer. *Phys. Rev. Lett.*, 86:5188–5191, May 2001.

[17] André Bouchet. An efficient algorithm to recognize locally equivalent graphs. *Combinatorica*, 11(4):315–329, 1991.

[18] Maarten Van den Nest, Jeroen Dehaene, and Bart De Moor. Efficient algorithm to recognize the local Clifford equivalence of graph states. *Physical Review A*, 70(3):034302, 2004.

[19] Sang Il Oum. Rank-width and vertex-minors. *Journal of Combinatorial Theory. Series B*, 95(1):79–100, 2005.

[20] Bruno Courcelle and Sang il Oum. Vertex-minors, monadic second-order logic, and a conjecture by Seese. *Journal of Combinatorial Theory. Series B*, 97(1):91–126, 2007.

[21] Bruno Courcelle and Joost Engelfriet. *Graph Structure and Monadic Second-Order Logic: A Language Theoretic Approach*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[22] 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(1):012337, 2007.

[23] Alexander Langer, Felix Reidl, Peter Rossmanith, and Somnath Sikdar. Practical algorithms for mso model-checking on tree-decomposable graphs. *Comput. Sci. Rev.*, 13(C):39–74, November 2014.

[24] Bernardo Martín, Ángel Sánchez, Cesar Beltran-Royo, and Abraham Duarte. Solving the edge-disjoint paths problem using a two-stage method. *International Transactions in Operational Research*, 27(1):435–457, 2020.

[25] André Bouchet. Graphic presentations of isotropic systems. *Journal of Combinatorial Theory, Series B*, 45(1):58–76, 1988.

[26] Anton Kotzig. Quelques remarques sur les transformations κ. In *seminaire Paris*, 1977.

[27] André Bouchet. Caracterisation des symboles croises de genre nul. *Comptes Rendus de l'Académie des Sciences*, 274:724–727, 1972.

[28] André Bouchet. Circle Graph Obstructions. *Journal of Combinatorial Theory, Series B*, 60(1):107–144, 1994.

[29] Martin Charles Golumbic. *Algorithmic graph theory and perfect graphs*. North-Holland Publishing Co., 2nd edition, 2004.

[30] Norman Biggs, E Keith Lloyd, and Robin J Wilson. *Graph Theory, 1736-1936*. Oxford University Press, 1976.

[31] Jens Vygen. Np-completeness of some edge-disjoint paths problems. *Discrete Applied Mathematics*, 61(1):83–90, 1995.

[32] M. Fleury. Deux problemes de Geometrie de sitation. *Journal de mathematiques elementaires*, 2nd(2):257–261, 1883.

[33] Shimon Even. *Graph Algorithms*. Cambridge University Press, 2 edition, 2011.

A   The 4-regular EDPDT problem is NP-Complete

From [31] we know that the EDP problem is $\mathbb{NP}$-Complete even when the graph $G \cup D$ is Eulerian. Here we prove that this problem remains $\mathbb{NP}$-Complete if we restrict the demand graph to be of the form $D = K_2^{\times k}$ and restrict $G \cup D$ to be 4-regular. We call this problem 4-regular EDPDT, see problem 3.4. To do so we will first introduce the notion of a grid graph gadget, an essential tool for reducing the Eulerian EDP problem to the 4-regular EDP problem. We make use of these results to show BELLVM (and BELLQM) is $\mathbb{NP}$-Complete in section 4.

Definition A.1. *Let G be an Eulerian multi-graph and let v be a vertex of G of degree $2n$ with incident edges labeled $1, \ldots n, 1', \ldots, n'$. The grid gadget gadget GG_v associated to v, illustrated in fig. 6, is a graph on $n^2 + 2n$ vertices labeled*

$$V(GG_v) = \Big\{ v_i : i \in [n] \Big\} \cup \Big\{ v_{i'} : i \in [n] \Big\} \cup \Big\{ v_{i,j'} : i,j \in [n] \Big\} \setminus \{v_{0,0'}\} \tag{27}$$

with edge set

$$\begin{aligned}
E(GG_v) = \Big\{ &(v_i, v_{i+1}) : i \in [n-1] \Big\} \cup \{(v_n, v_1)\} \\
&\cup \Big\{ (v_{i'}, v_{(i+1)'}) : i \in [n-1] \Big\} \cup \{(v_{n'}, v_{1'})\} \\
&\cup \left[\bigcup_{i \in [n]} \Big\{ (v_i, v_{i,1'}) \Big\} \cup \Big\{ (v_{i,j'}, v_{i,(j+1)'}) : j \in [n-1] \Big\} \right] \\
&\cup \left[\bigcup_{i \in [n]} \Big\{ (v_{i'}, v_{1,i'}) \Big\} \cup \Big\{ (v_{j,i'}, v_{j+1,i'}) : j \in [n-1] \Big\} \right] \\
&\cup \Big\{ (v_{i,n'}, v_{n,i'}) : i \in [n] \Big\} \cup \{(v_1, v_{1'})\} .
\end{aligned} \tag{28}$$

$\diamond$

For later convenience we also define the following subsets of edges for $i \in [n]$

$$E_i^{\mathrm{hor}}[j', k'] = \Big\{ (v_{i,l'}, v_{i,(l+1)'}) : l \in [j-1, k-1] \Big\} \quad \text{for} \quad i,j,k \in [n] \tag{29}$$

$$E_{i'}^{\mathrm{ver}}[j, k] = \Big\{ (v_{l,i'}, v_{l+1,i'}) : l \in [j-1, k-1] \Big\} \quad \text{for} \quad i,j,k \in [n]. \tag{30}$$

where $v_{i,0'} = v_i$ and $v_{0,i'} = v_{i'}$. Informally, with respect to fig. 6 the edges in $E_i^{\mathrm{hor}}[j', k']$ are the horizontal edges at level i which are right of j' and left of k'. Similarly, the edges in $E_{i'}^{\mathrm{ver}}[j, k]$ are the vertical edges at level i' which are above j below k.
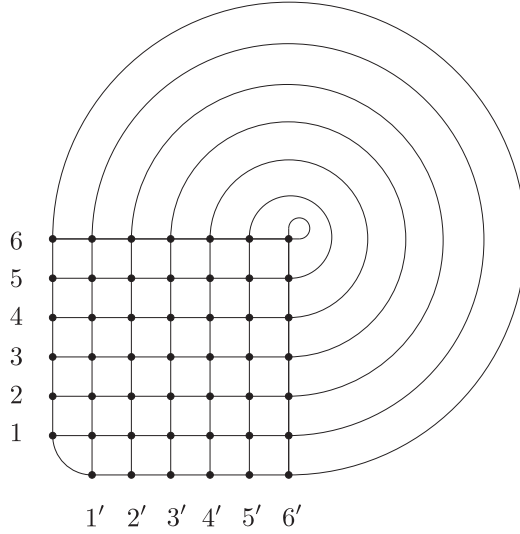
Figure 6: An example of the grid gadget defined in definition A.1.

Note that all the vertices in the grid graph gadget GG_v have degree 4 or 3, with only the vertices on the 'outside' (associated to the incident edges of the vertex v) having degree 3.

We will now prove that the grid gadget, in a specified sense, provides all-to-all connectivity.

Lemma A.2. *Let v be a vertex in a graph with $2n$ incident edges labeled $1, \ldots, n, 1', \ldots n'$ and let GG_v be the associated grid graph gadget as defined in definition A.1. Consider an arbitrary pairing of the vertices $v_1, \ldots, v_n, v_{1'}, \ldots, v_{n'}$ consisting of tuples of the form (v_i, v_j) (unprimed pairs), $(v_k, v_{l'})$ (mixed pairs) and $(v_{m'}, v_{n'})$ (primed pairs). Then there exist edge-disjoint paths $P_1, \ldots P_n$ that connect all pairs and moreover contain all edges in GG_v.* ◇

Proof. Assume without of loss of generality that $i < j$ for the unprimed pairs and $m' < n'$ for the primed pairs (i.e. the first index is always smaller). We will prove the lemma by explicit construction of the paths $P_1, \ldots, P_n$ Let M be the list of mixed pairs, L be the list of unprimed pairs and L' be the list of primed pairs. Note that L and L' are necessarily of equal length. Consider first the list of mixed pairs M. Construct the paths $P_{k,l'}$ by the following algorithm. An example of these paths is shown in fig. 7.

Algorithm 1 Algorithm to construct the paths $P_{k,l'}$.

---
    for $x \in [K]$ where K is the length of M **do**
        Create the path P_{k_x, l'_x} associated to the x'th pair in M by:
            (1) Walking horizontally rightwards from v_{k_x} to v_{k_x, l'_x}
            (2) Walking vertically downwards from v_{k_x, l'_x} to $v_{l'_x}$
    end for

Note that for every pair we have that

$$E(P_{k,l'}) = E_k^{\mathrm{hor}}[0, l'] \cup E_{l'}^{\mathrm{ver}}[0, k] \tag{31}$$

where $E_k^{\mathrm{hor}}[0, l']$ and $E_{l'}^{\mathrm{ver}}[0, k]$ are defined in eqs. (29) and (30). Note also that since these sets are all disjoint the paths $P_{k,l'}$ are all mutually edge-disjoint.

Now consider the lists of of primed and unprimed pairs L and L'. We will construct paths $P_{i,j}$ (unprimed) and $P_{m',n'}$ (primed) using the following algorithm.

Algorithm 2 Algorithm to construct the paths $P_{i,j}$ and $P_{i',j'}$.

$\quad$ **for** $x \in [K]$ where K is the length of L **do** $\qquad \triangleright$ L and L' are necessarily of the same length.
$\qquad$ Create the path P_{i_x,j_x} associated to the k'th pair in L by:
$\qquad\qquad$ (1) Walking horizontally rightwards from v_{i_x} to v_{i_x,m'_x}
$\qquad\qquad$ (2) Walking vertically upwards from v_{i_x,m'_x} to v_{j_x,m'_x}
$\qquad\qquad$ (3) Walking horizontally leftwards from v_{j_x,m'_x} to v_{j_x}
$\qquad$ Create the path $P_{m'_x,n'_x}$ associated to the x'th pair in L' by:
$\qquad\qquad$ (1) Walking vertically upwards from $v_{m'_x}$ to v_{i_x,m'_x}
$\qquad\qquad$ (2) Walking horizontally rightwards from v_{i_x,m'_x} to v_{i_x,n'_x}
$\qquad\qquad$ (3) Walking vertically downwards from v_{i_x,n'_x} to $v_{n'_x}$
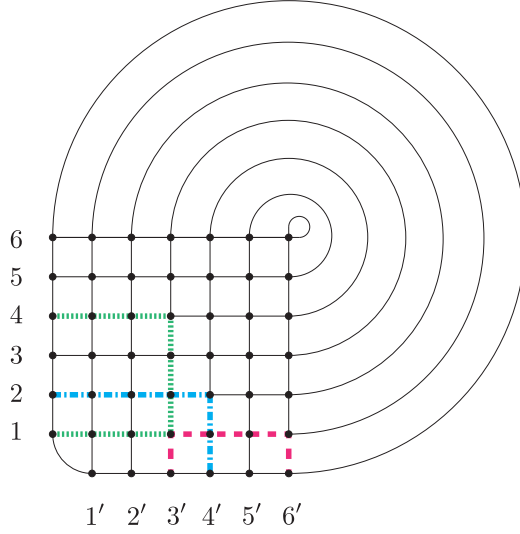$\quad$ **end for**

Figure 7: An example of paths produced by algorithms 1 and 2 on the grid gadget defined in definition A.1. In this example one of the mixed pairs in M is $(2, 4')$ giving rise to the blue dashed-dotted path. Furthermore, the first unprimed pair in L is $(1, 4)$ and the first primed pair in L' is $(3', 6')$, giving rise to the green dotted path and the red dashed path respectively, meeting at $v_{2,3'}$. Note that the x'th primed and unprimed paths always meet at the vertex v_{i_x,m'_x} where v_{i_x} and $v_{m'_x}$ is the first vertex in the x'th primed and unprimed pair respectively.

Now note that the for all $x \in [K]$ (where K is the length of L) we have that

$$E(P_{i_x,j_x}) = E^{\text{hor}}_{i_x}[0, m'_x] \cup E^{\text{ver}}_{m'_x}[i_x, j_x] \cup E^{\text{hor}}_{j_x}[0, m'_x] \tag{32}$$

$$E(P_{m'_x,n'_x}) = E^{\text{ver}}_{m'_x}[0, i_x] \cup E^{\text{hor}}_{i_x}[m'_x, n'_x] \cup E^{\text{ver}}_{n'_x}[0, i_x] \tag{33}$$

This immediately implies that all P_{i_x,j_x} and $P_{m'_x,n'_x}$ are mutually edge-disjoin and moreover that no P_{i_x,j_x} nor any $P_{m'_x,n'_x}$ share edges with any of the mixed-pair paths $P_{k,l'}$. This last point can be seen by noting that an unprimed vertex v_i can either be part of a mixed pair or an unprimed pair but not both at the same time (with a similar argument for the primed vertices).

Hence we have constructed a set of edge-disjoint paths that connect all vertex pairs. However they do not yet contain all edges in $E(GG_v)$. It is however straightforward to extend the paths to include all remaining edges. To see this consider the grid graph gadget GG_v and remove all edges that are contained in one of the paths constructed above. What remains is a not necessarily connected graph G_{rem} of which the connected components, by construction, share a vertex with at least one of the constructed paths. Moreover, all these graphs will be Eulerian. Now choose for each graph G_{rem} a Eulerian tour U and insert it into exactly one of the paths that shares a

vertex with G_{rem}. The resulting set of paths will still have mutually edge disjoint elements (since a Eulerian tour is edge-disjoint by definition) and furthermore the union of all the paths in the set contains all edges in the grid graph gadget GG_v. This completes the proof. $\square$

We will now make use of lemma A.2 to map instances of EDP to instances of 4-regular EDPDT. We will first construct a mapping from arbitrary demand graphs D to demand graphs of the form $K_2^{\times k}$. Then, to make the graph $G \cup D$ 4-regular, we replace any higher-degree vertices in G with the grid gadget above. Using lemma A.2, we can prove that this puts no restriction on the possible paths.

Theorem A.3. *Let G and D be graphs such that $V(D) \subseteq V(G)$ and $G \cup D$ is Eulerian. There exist graphs G'' and D' such that $D' = K_2^{\times k}$ where $k = |E(D)|$, $G'' \cup D'$ is 4-regular and (G, D) is a YES-instance of the EDP problem if and only if (G'', D') is.* $\diamond$
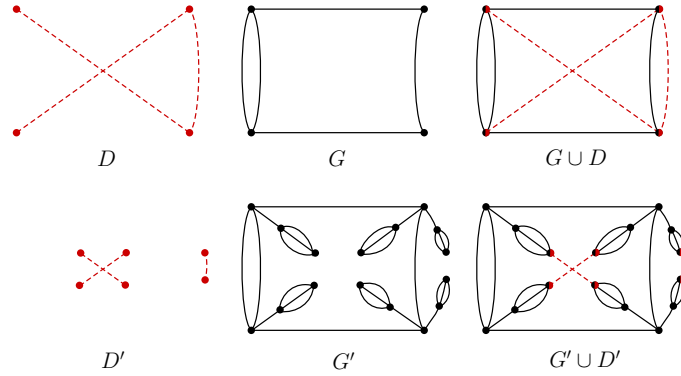

Figure 8: An example showing the mapping from an instance (G, D) of EDP to an instance (G', D') of EDPDT. Black solid edges are edges from G or G' and red dashed edges are edges from D'.

Proof. To prove the theorem we will construct an explicit mapping from the graphs (G, D) to (G'', D'). We do this in two steps: (1) map (G, D) to (G', D') where $D' = K_2^{\times k}$ but $G' \cup D'$ is not necessarily 4-regular and (2) map (G', D') to (G'', D') such that $G'' \cup D'$ is 4-regular. The first mapping is visualized in fig. 8 and formalized in eqs. (35) to (38). Informally, we replace each edge $e = (u, v) \in E(D)$ in the graph $G \cup D$ with the following gadget

$$\tag{34}$$
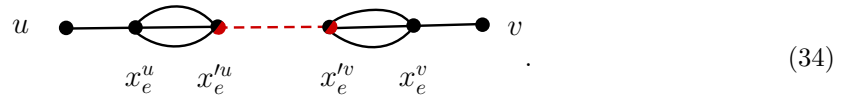
where x_e^u, $x_e^{\prime u}$, $x_e^{\prime v}$ and x_e^v are all new vertices. Note that we label the vertex closer to the corresponding vertex u in the edge e without the prime and the further one with a prime. Formally, we define G' to be

$$V(G') = V(G) \cup \left(\bigcup_{e=(u,v)\in E(D)} \{x_e^u, x_e^{\prime u}, x_e^{\prime v}, x_e^v\} \right) \tag{35}$$

$$E(G') = E(G) \cup \left(\bigcup_{e=(u,v)\in E(D)} \Big\{ (u, x_e^u), (x_e^u, x_e^{\prime u}), (x_e^u, x_e^{\prime u}), (x_e^u, x_e^{\prime u}), \right.$$

$$\left. (x_e^{\prime v}, x_e^v), (x_e^{\prime v}, x_e^v), (x_e^{\prime v}, x_e^v), (x_e^v, v) \Big\} \right). \tag{36}$$

Note that we have added the edges $(x_e^u, x_e'^u)$ and $(x_e'^v, x_e^v)$ three times. We now define the new demand graph D' as

$$V(D') = \bigcup_{e=(u,v)\in E(D)} \{x_e'^u, x_e'^v\} \tag{37}$$

$$E(D') = \bigcup_{e=(u,v)\in E(D)} \{(x_e'^u, x_e'^v)\} \tag{38}$$

Note that $D' = K_2^{\times k}$. Note also that $G' \cup D'$ is still Eulerian. However it is in general not 4-regular.

Now let N be the set of vertices of $G' \cup D'$ of degree different than 4. Note that, by construction $N \cap V(D') = \emptyset$. That is, all vertices of degree other that 4 are exclusively vertices in G'. We will from G' construct a graph G'' such that $G'' \cup D'$ is 4-regular. For every vertex in N which has degree 2, we simply add a self-loop to the vertex, making the vertex have degree 4 without changing any connectivity. Furthermore, for every vertex $v \in N$ with degree larger than 4, we replace it by the grid graph gadget GG_v as defined in definition A.1, attaching the edges incident on v to the vertices $v_1, \ldots v_n, v_{1'}, \ldots, v_{n'}$ (where $n = \deg(v)/2$) in the grid graph gadget. Note the graph $G'' \cup D'$ obtained after this procedure is 4-regular.

To prove the theorem we now need to show that (G, D) is a YES-instance to EDP if and only if (G'', D') is a YES-instance. Again, we will do this in two steps: (1) first show that (G, D) is a YES-instance if and only if (G', D') is and (2) show that (G', D') is a YES-instance if and only if (G'', D') is.

1. We now prove that (G', D') is a YES-instance of EDP if and only (G, D) is. First assume (G, D) is a YES-instance of EDP. This means there exists for each $e = (u, v) \in E(D)$ a path P_e on G that begins and ends at the vertices u, v such that all paths P_e are mutually edge-disjoint. Now, for each $e \in E(D)$ define the path P'_e on G' as

   $$P'_e = x_e'^u x_e^u x_e'^u x_e^u u P_e v x_e^v x_e'^v x_e^v x_e'^v \tag{39}$$

   where we have omitted writing out the edges that the path traverses for clarity. For a visual aid refer to eq. (34) where we instead of starting the path at u as in P_e we start at $x_e'^u$, traverse the edge $(x_x^u, x_e'^u)$ back and fourth three times and then move to u using the edge $(x_e^u, x_e'^u)$. From u the path P'_e is the same as P_e and when arriving at v we instead end at $x_e'^v$ similarly to how we started at $x_e'^u$. Thus P'_e is a path connecting the vertices $x_e'^u, x_e'^u$. Note that by definition $(x_e'^u, x_e'^u)$ is an edge in the demand graph D' (precisely corresponding to the edge $(u, v) \in D$). The paths P'_e are also mutually edge-disjoint, since the paths P_e are. Hence (G', D') is a YES-instance.

   For the other direction, assume that (G', D') is a YES-instance of EDP and thus that there exists edge-disjoint paths P'_e connecting the vertices $x_e'^u, x_e'^v$ for all $e = (x_e'^u, x_e'^v) \in E(D')$. One can then see that P'_e is either of the form as in eq. (39) or as

   $$P'_e = x_e'^u x_e^u u P_e v x_e^v x_e'^v. \tag{40}$$

   This means that the associated path P_e forms a path between u and v in G for all $e =\in E(D')$. Furthermore, since all P'_e are pairwise edge-disjoint, so are the P_e. Hence (G, D) is also a YES-instance of EDP.

2. Next we argue that (G', D') is a YES-instance of EDP if and only if (G'', D') is. If (G'', D') is a YES-instance of EDP then so is (G', D'), since any edge-disjoint paths passing through a grid graph gadget GG_v can also be made into edge-disjoint paths passing through the vertex v. Hence assume that (G', D') is a YES-instance of EDP. Note that the only difference between (G', D') and (G'', D') is the replacement of vertices $v \in N$ with the grid graph gadget GG_v. Moreover recall that $N \cap V(D') = \emptyset$. Finally, by lemma A.2, we know that any $\deg(v)/2$ paths passing through a vertex v can be mapped to $\deg(v)/2$ paths passing through GG_v, and that these paths are mutually edge-disjoint and use all edges in GG_v. This implies that if (G', D') is a YES-instance of EDP, then so is (G'', D'). This proves the theorem.

$\square$

We can now prove corollary 3.5.

Proof of corollary 3.5. Theorem A.3 states that there exists a many-one reduction from Eulerian EDP to 4-regular EDPDT. Furthermore, this reduction consists of constructing the graphs (G'', D') from the graphs (G, D) by the explicit rules in eqs. (35) to (38) and by replacing vertices of degree more than 4 with the grid graph gadget, which is clearly polynomial. This shows that 4-regular EDPDT is $\mathbb{NP}$-Hard. Since any instance of 4-regular EDPDT is also an instance of the general EDP which is in $\mathbb{NP}$, also 4-regular EDPDT is in $\mathbb{NP}$ and thus $\mathbb{NP}$-Complete. $\square$