Simulating Quantum Computations with Tutte Polynomials

Ryan L. Mann[1, 2, *]

[1] *School of Mathematics, University of Bristol, Bristol, BS8 1UG, United Kingdom*
[2] *Centre for Quantum Computation and Communication Technology, Centre for Quantum Software and Information, Faculty of Engineering & Information Technology, University of Technology Sydney, NSW 2007, Australia*

We establish a classical heuristic algorithm for exactly computing quantum probability amplitudes. Our algorithm is based on mapping output probability amplitudes of quantum circuits to evaluations of the Tutte polynomial of graphic matroids. The algorithm evaluates the Tutte polynomial recursively using the deletion-contraction property while attempting to exploit structural properties of the matroid. We consider several variations of our algorithm and present experimental results comparing their performance on two classes of random quantum circuits. Further, we obtain an explicit form for Clifford circuit amplitudes in terms of matroid invariants and an alternative efficient classical algorithm for computing the output probability amplitudes of Clifford circuits.

I. INTRODUCTION

There is a natural relationship between quantum computation and evaluations of Tutte polynomials [1, 2]. In particular, quantum probability amplitudes are proportional to evaluations of the Tutte polynomial of graphic matroids. In this paper we use this relationship to establish a classical heuristic algorithm for exactly computing quantum probability amplitudes. While this problem is known to be #P-hard in general [3], our algorithm focuses on exploiting structural properties of an instance to achieve an improved runtime over traditional methods. Previously it was known that this problem can be solved in time exponential in the treewidth of the underlying graph [4].

The basis of our algorithm is a mapping between output probability amplitudes of quantum circuits and evaluations of the Tutte polynomial of graphic matroids [2, 5, 6]. Our algorithm proceeds to evaluate the Tutte polynomial recursively using the deletion-contraction property. At each step in the recursion, our algorithm computes certain structural properties of the matroid in order to attempt to prune the computational tree. This approach to computing Tutte polynomials was first studied by Haggard, Pearce, and Royle [7]. Our algorithm can be seen as an adaption of their work to special points of the Tutte plane where we can exploit additional structural properties.

The performance of algorithms for computing Tutte polynomials based on the deletion-contraction property depends on the heuristic used to decide the ordering of the recursion [7–9]. We consider several heuristics introduced by Pearce, Haggard, and Royle [8] and an additional heuristic, which is specific to our algorithm. We present some experimental results comparing the performance of these heuristics on two classes of random quantum circuits corresponding to dense and sparse instances.

The correspondence between output probability amplitudes of quantum circuits and evaluations of Tutte polynomials also allows us to obtain an explicit form for Clifford circuit amplitudes in terms of matroid invariants by a theorem of Pendavingh [10]. This gives rise to an alternative efficient classical algorithm for computing output probability amplitudes of Clifford circuits.

This paper is structured as follows. We introduce matroid theory in Section II and the Tutte polynomial in Section III. In Sections IV, V, and VI, we establish a mapping between output probability amplitudes of quantum circuits and evaluations of the Tutte polynomial of graphic matroids. This is achieved by introducing the Potts model partition function in Section IV, Instantaneous Quantum Polynomial-time circuits in Section V, and a class of universal quantum circuits in Section VI. In Section VII, we use this mapping to obtain an explicit form for Clifford circuit amplitudes in terms of matroid invariants. We also obtain an efficient classical algorithm for computing the output probability amplitudes of Clifford circuits. We describe our algorithm in Section VIII and present some experimental results in Section IX. Finally, we conclude in Section X.

II. MATROID THEORY

We shall now briefly introduce the theory of matroids. The interested reader is referred to the classic textbooks of Welsh [11] and Oxley [12] for a detailed treatment. Matroids were introduced by Whitney [13] as a structure that generalises the notion of linear dependence. There are many equivalent ways to define a matroid. We shall define a matroid by the independence axioms.

Definition 1 (Matroid)**.** A matroid is a pair $M = (\mathcal{S}, \mathcal{I})$ consisting of a finite set $\mathcal{S}$, known as the *ground set*, and a collection $\mathcal{I}$ of subsets of $\mathcal{S}$, known as the *independent sets*, such that the following axioms are satisfied.

1. The empty set is a member of $\mathcal{I}$.

* mail@ryanmann.org; http://www.ryanmann.org

2. Every subset of a member of $\mathcal{I}$ is a member of $\mathcal{I}$.

3. If A and B are members of $\mathcal{I}$ and $|A| > |B|$, then there exists an $x \in A \backslash B$ such that $B \cup \{x\}$ is a member of $\mathcal{I}$.

The rank of a subset A of $\mathcal{S}$ is given by the *rank function* $r : 2^{\mathcal{S}} \to \mathbb{N}$ of the matroid defined by $r(A) \coloneqq \max(|X| \mid X \subseteq A, X \in \mathcal{I})$. The *rank* of a matroid M, denoted $r(M)$, is the rank of the set S.

The archetypal class of matroids are *vector matroids*. A vector matroid $M = (\mathcal{S}, \mathcal{I})$ is a matroid whose ground set $\mathcal{S}$ is a subset of a vector space over a field $\mathbb{F}$ and whose independent sets $\mathcal{I}$ are the linearly independent subsets of $\mathcal{S}$. The rank of a subset of a vector matroid is the dimension of the subspace spanned by the corresponding vectors. We say that a matroid is $\mathbb{F}$-*representable* if it is isomorphic to a vector matroid over the field $\mathbb{F}$. A matroid is a *binary matroid* if it is $\mathbb{F}_2$-representable and is a *ternary matroid* if it is $\mathbb{F}_3$-representable. A matroid that is representable over every field is called a *regular matroid*.

Every finite graph $G = (V, E)$ induces a matroid $M(G) = (\mathcal{S}, \mathcal{I})$ as follows. Let the ground set $\mathcal{S}$ be the set of edges E and let the independent sets $\mathcal{I}$ be the subsets of E that are a forest, i.e., they do not contain a simple cycle. It is easy to check that $M(G)$ satisfies the independence axioms. The rank of a subset A of a cycle matroid is $|V| - \kappa(A)$, where $\kappa(A)$ denotes the number of connected components of the subgraph with edge set A. The rank of the cycle matroid $M(G)$, denoted $r(M(G))$ or simply $r(G)$, is the rank of the set E. The matroid $M(G)$ is called the *cycle matroid* of G. We say that a matroid is *graphic* if it is isomorphic to the cycle matroid of a graph.

Graphic matroids are regular. To see this consider assigning to the graph G an arbitrary *orientation* $D(G)$, that is, for each edge $e = \{u, v\}$ in G, we choose one of u and v to be the *positive end* and the other one to be the *negative end*. Then construct the *oriented incidence matrix* of G with respect to the orientation $D(G)$.

Definition 2 (Oriented incidence matrix)**.** Let $G = (V, E)$ be a graph and let $D(G)$ be an orientation of G. Then the oriented incidence matrix of G with respect to $D(G)$ is the $|V| \times |E|$ matrix $A_{D(G)} = (a_{ve})_{|V| \times |E|}$ whose entries are

$$a_{ve} = \begin{cases} +1, & \text{if } v \text{ is the positive end of } e; \\ -1, & \text{if } v \text{ is the negative end of } e; \\ 0, & \text{otherwise.} \end{cases}$$

The rows of the oriented incidence matrix $A_{D(G)}$ correspond to the vertices of G and the columns correspond to the edges of G. Each column contains exactly one $+1$ and exactly one -1 representing the positive and negative ends of the corresponding edge. If the column space of $A_{D(G)}$ is the ground set of a vector matroid, then it is easy to see that a subset is independent if and only if it is a forest in G. Hence, the oriented incidence matrix provides a representation of a graphic matroid over every field.

A *minor* of a matroid M is a matroid that is obtained from M by a sequence of *deletion* and *contraction* operations.

Definition 3 (Deletion)**.** Let $M = (\mathcal{S}, \mathcal{I})$ be a matroid and let e be an element of the ground set. Then the deletion of M with respect to e is the matroid $M \backslash \{e\} = (\mathcal{S}', \mathcal{I}')$ whose ground set is $\mathcal{S}' = \mathcal{S} \backslash \{e\}$ and whose independent sets are $\mathcal{I}' = \{I \subseteq \mathcal{S} \backslash \{e\} \mid I \in \mathcal{I}\}$.

The deletion of an element from the cycle matroid of a graph corresponds to removing an edge from the graph.

Definition 4 (Contraction)**.** Let $M = (\mathcal{S}, \mathcal{I})$ be a matroid and let e be an element of the ground set. Then the contraction of M with respect to e is the matroid $M/\{e\} = (\mathcal{S}', \mathcal{I}')$ whose ground set is $\mathcal{S}' = \mathcal{S} \backslash \{e\}$ and whose independent sets are $\mathcal{I}' = \{I \subseteq \mathcal{S} \backslash \{e\} \mid I \cup \{e\} \in \mathcal{I}\}$.

The contraction of an element from the cycle matroid of a graph corresponds to removing an edge from the graph and merging its two endpoints.

An element e of a matroid is said to be a *loop* if $\{e\}$ is not an independent set and said to be a *coloop* if e is contained in every maximally independent set. If an element e of a matroid is either a loop or a coloop then the deletion and contraction of e are equivalent.

III. THE TUTTE POLYNOMIAL

We shall now briefly introduce the Tutte polynomial, which is a well-known invariant in matroid and graph theory.

Definition 5 (Tutte polynomial of a matroid)**.** Let $M = (\mathcal{S}, \mathcal{I})$ be a matroid with rank function $r : 2^{\mathcal{S}} \to \mathbb{N}$. Then the Tutte polynomial of M is the bivariate polynomial defined by

$$\mathrm{T}(M; x, y) \coloneqq \sum_{A \subseteq \mathcal{S}} (x-1)^{r(M)-r(A)} (y-1)^{|A|-r(A)}.$$

The Tutte polynomial may also be defined recursively by the *deletion-contraction property*.

Definition 6 (Deletion-contraction property)**.** Let $M = (\mathcal{S}, \mathcal{I})$ be a matroid. If M is the empty matroid, i.e., $\mathcal{S} = \varnothing$, then

$$\mathrm{T}(M; x, y) = 1.$$

Otherwise, let e be an element of the ground set. If e is a loop, then

$$\mathrm{T}(M; x, y) = y\mathrm{T}(M \backslash \{e\}; x, y).$$

If e is a coloop, then

$$\mathrm{T}(M;x,y) = x\mathrm{T}(M/\{e\};x,y).$$

Finally, if e is neither a loop nor a coloop, then

$$\mathrm{T}(M;x,y) = \mathrm{T}(M\backslash\{e\};x,y) + \mathrm{T}(M/\{e\};x,y).$$

The deletion-contraction property immediately gives an algorithm for recursively computing the Tutte polynomial. This algorithm is in general inefficient, but the performance may be improved by using isomorphism testing to reduce the number of recursive calls [14]. The performance of this algorithm depends on the heuristic used to choose elements of the ground set [7–9]. Björklund et al. [15] showed that the Tutte polynomial can be computed in time exponential in the number of vertices.

The Tutte polynomial of a graph may be recovered by considering the Tutte polynomial of the cycle matroid of a graph and using the fact that the rank of a subset A of a cycle matroid is $|V| - \kappa(A)$, where $\kappa(A)$ denotes the number of connected components of the subgraph with edge set A.

Definition 7 (Tutte Polynomial of a graph)**.** Let $G = (V, E)$ be a graph and let $\kappa(A)$ denote the number of connected components of the subgraph with edge set A. Then the Tutte polynomial of G is a polynomial in x and y, defined by

$$\mathrm{T}(G;x,y) \coloneqq \sum_{A\subseteq E} (x-1)^{\kappa(A)-\kappa(E)}(y-1)^{\kappa(A)+|A|-|V|}.$$

The Tutte polynomial is trivial to evaluate along the hyperbola $(x-1)(y-1) = 1$ for any matroid. In the case of graphic matroids, Jaeger, Vertigan, and Welsh [16] showed that the Tutte polynomial is #P-hard to evaluate, except along this hyperbola and when (x, y) equals one of nine special points.

Theorem 1 (Jaeger, Vertigan, and Welsh [16])**.** *The problem of evaluating the Tutte polynomial of a graphic matroid at an algebraic point in the (x, y)-plane is* #P-hard *except when $(x-1)(y-1) = 1$ or when (x, y) equals one of $(1, 1)$, $(-1, -1)$, $(0, -1)$, $(-1, 0)$, $(i, -i)$, $(-i, i)$, (j, j^2), (j, j^2), or (j^2, j), where $j = \exp(2\pi i/3)$. In each of these exceptional cases the evaluation can be done in polynomial time.*

Vertigan [17] extended this result to vector matroids.

Theorem 2 (Vertigan [17])**.** *The problem of evaluating the Tutte polynomial of a vector matroid over a field $\mathbb{F}$ at an algebraic point in the (x, y)-plane is* #P-hard *except when $(x-1)(y-1) = 1$, (x, y) equals $(1, 1)$, or when*

1. *$|\mathbb{F}| = 2$ and (x, y) equals one of $(-1, -1)$, $(0, -1)$, $(-1, 0)$, $(i, -i)$, or $(-i, i)$;*
2. *$|\mathbb{F}| = 3$ and (x, y) equals one of (j, j^2) or (j^2, j), where $j = \exp(2\pi i/3)$; or*
3. *$|\mathbb{F}| = 4$ and (x, y) equals $(-1, -1)$.*

In each of these exceptional cases, except when (x, y) equals $(1, 1)$, the evaluation can be done in polynomial time.

Snook [18] showed that when (x, y) equals $(1, 1)$ and $\mathbb{F}$ is either a finite field of fixed characteristic or a fixed infinite field, then evaluating the Tutte polynomial is #P-hard. It is an open problem to understand the complexity of evaluating the Tutte polynomial at $(1, 1)$ over any fixed field.

IV. THE POTTS MODEL PARTITION FUNCTION

The Potts model is a statistical physical model described by an integer $q \in \mathbb{Z}^+$ and a graph $G = (V, E)$, with the vertices representing spins and the edges representing interactions between them. A set of edge weights $\{\omega_e\}_{e\in E}$ characterise the interactions and a set of vertex weights $\{\upsilon_v\}_{v\in V}$ characterise the external fields at each spin. A configuration of the model is an assignment σ of each spin to one of q possible states. The *Potts model partition function* is defined as follows.

Definition 8 (Potts model partition function)**.** Let $q \in \mathbb{Z}^+$ be an integer and let $G = (V, E)$ be a graph with the weights $\Omega = \{\omega_e\}_{e\in E}$ assigned to its edges and the weights $\Upsilon = \{\upsilon_v\}_{v\in V}$ assigned to its vertices. Then the q-state Potts model partition function is defined by

$$\mathrm{Z}_{\mathrm{Potts}}(G;q,\Omega,\Upsilon) \coloneqq \sum_{\sigma\in\mathbb{Z}_q^V} w_G(\sigma),$$

where

$$w_G(\sigma) = \exp\left(\sum_{\{u,v\}\in E} \omega_{\{u,v\}}\delta(\sigma_u,\sigma_v) + \sum_{v\in V}\upsilon_v\delta(\sigma_v)\right).$$

The Potts model partition function with an external field is equivalent to the zero-field case on an augmented graph $G' = (V', E')$. To construct G' from G, for each of the connected components $\{C_i\}_{i=1}^{\kappa(E)}$ of G add a new vertex u_i and for every vertex $v \in V(C_i)$ add an edge $e_v = \{u_i, v\}$ with the weight υ_v assigned to it. Then we have the following proposition.

Proposition 3.

$$\mathrm{Z}_{\mathrm{Potts}}(G;q,\Omega,\Upsilon) = q^{-\kappa(E)}\mathrm{Z}_{\mathrm{Potts}}(G';q,\Omega\cup\Upsilon,0).$$

A similar proposition appears in Welsh's monograph [19]; we prove Proposition 3 in Appendix A.

It will be convenient to consider the Potts model with weights that are all positive integer multiples of a complex number θ. We shall implement this model on the augmented graph G' with all weights equal to θ by replacing

each edge with the appropriate number of parallel edges. Let us denote the partition function of this model by $\mathrm{Z}_{\mathrm{Potts}}(G';q,\theta)$. Then, we have the following proposition relating the partition function of this model to the Tutte polynomial of the augmented graph G'.

Proposition 4.

$$\mathrm{Z}_{\mathrm{Potts}}(G';q,\theta) = q^{\kappa(E')}(e^{\theta}-1)^{r(G')}\mathrm{T}\left(G';x,y\right),$$

where $x = \frac{e^{\theta}+q-1}{e^{\theta}-1}$ *and* $y = e^{\theta}$.

In particular, the q-state Potts model partition function is related to the Tutte polynomial along the hyperbola $(x-1)(y-1)=q$. For a proof of Proposition 4, we refer the reader to Welsh's monograph [19, Section 4.4].

The 2-state Potts model partition function specialises to the *Ising model partition function*.

Definition 9 (Ising model partition function)**.** Let $G=(V,E)$ be a graph with the weights $\Omega=\{\omega_e\}_{e\in E}$ assigned to its edges and the weights $\Upsilon=\{\upsilon_v\}_{v\in V}$ assigned to its vertices. Then the Ising model partition function is defined by

$$\mathrm{Z}_{\mathrm{Ising}}(G;\Omega,\Upsilon) \coloneqq \sum_{\sigma\in\{-1,+1\}^V} w_G(\sigma),$$

where

$$w_G(\sigma) = \exp\left(\sum_{\{u,v\}\in E}\omega_{\{u,v\}}\sigma_u\sigma_v + \sum_{v\in V}\upsilon_v\sigma_v\right).$$

Proposition 5.

$$\mathrm{Z}_{\mathrm{Potts}}(G;2,\Omega,\Upsilon) = w_G\mathrm{Z}_{\mathrm{Ising}}\left(G;\frac{\Omega}{2},\frac{\Upsilon}{2}\right),$$

where $w_G = \exp\left(\frac{1}{2}\sum_{e\in E}\omega_e + \frac{1}{2}\sum_{v\in V}\upsilon_v\right)$.

Proof. The proof follows from some simple algebra. ∎

V. INSTANTANEOUS QUANTUM POLYNOMIAL TIME

We shall now briefly introduce the class of commuting quantum circuits, known as *Instantaneous Quantum Polynomial-time* (IQP) circuits [5]. These circuits exhibit many interesting mathematical properties. In particular, the output probability amplitudes of IQP circuits are proportional to evaluations of the Tutte polynomial of binary matroids [2]. IQP circuits comprise only gates that are diagonal in the Pauli-X basis and are described by an *X-program*.

Definition 10 (X-program)**.** An X-program is a pair (P,θ), where $P=(p_{ij})_{m\times n}$ is a binary matrix and $\theta\in[-\pi,\pi]$ is a real angle. The matrix P is used to construct a Hamiltonian of m commuting terms acting on n qubits, where each term in the Hamiltonian is a product of Pauli-X operators,

$$\mathrm{H}_{(P,\theta)} \coloneqq -\theta\sum_{i=1}^{m}\bigotimes_{j=1}^{n}X_j^{p_{ij}}.$$

Thus, the columns of P correspond to qubits and the rows of P correspond to interactions in the Hamiltonian.

An X-program induces a probability distribution $\mathcal{P}_{(P,\theta)}$ known as an *IQP distribution*.

Definition 11 ($\mathcal{P}_{(P,\theta)}$)**.** For an X-program (P,θ) with $P=(p_{ij})_{m\times n}$, we define $\mathcal{P}_{(P,\theta)}$ to be the probability distribution over binary strings $x\in\{0,1\}^n$, given by

$$\mathbf{Pr}[x] \coloneqq \left|\psi_{(P,\theta)}(x)\right|^2,$$

where

$$\psi_{(P,\theta)}(x) = \langle x|\exp\left(-i\mathrm{H}_{(P,\theta)}\right)|0^n\rangle.$$

The principal probability amplitude $\psi_{(P,\theta)}(0^n)$ of an IQP distribution is directly related to an evaluation of the Tutte polynomial of the binary matroid whose ground set is the row space of P.

Proposition 6. *Let* (P,θ) *be an X-program with* $P=(p_{ij})_{m\times n}$. *Let* $M=(\mathcal{S},\mathcal{I})$ *be the binary matroid whose ground set* $\mathcal{S}$ *is the row space of* P, *then,*

$$\psi_{(P,\theta)}(0) = e^{i\theta(r(M)-m)}(i\sin(\theta))^{r(M)}\mathrm{T}\left(M;x,y\right),$$

where $x=-i\cot(\theta)$ *and* $y=e^{2i\theta}$.

A similar result may be obtained for the other probability amplitudes. This can easily be seen when $\theta=\frac{\pi}{2k}$ for $k\in\mathbb{Z}^+$, by firstly letting $P\|^k x$ be the matrix obtained from P by appending k rows identical to x, and then observing that $\psi_{(P,\theta)}(x) = -i\psi_{(P\|^k x,\theta)}(0^n)$. For a proof of Proposition 6 and a treatment of the general θ case, we refer the reader to Ref. [2, Section 3].

We shall consider X-programs that are induced by a weighted graph.

Definition 12 (Graph-induced X-program)**.** For a graph $G=(V,E)$ with the weights $\{\omega_e\in[-\pi,\pi]\}_{e\in E}$ assigned to its edges and the weights $\{\upsilon_v\in[-\pi,\pi]\}_{v\in V}$ assigned to its vertices, we define the X-program induced by G to be an X-program $\mathcal{X}_G$ such that

$$\mathrm{H}_{\mathcal{X}_G} = -\sum_{\{u,v\}\in E}\omega_{\{u,v\}}X_uX_v - \sum_{v\in V}\upsilon_vX_v.$$

Any X-program can be efficiently represented by a graph-induced X-program [5]. The principal probability amplitude $\psi_{\mathcal{X}_G}(0^n)$ of the IQP distribution generated by a graph-induced X-program is directly related to the Ising model partition function of the graph with imaginary weights.

Proposition 7. *Let $G = (V, E)$ be a graph with the weights $\Omega = \{\omega_e \in [-\pi, \pi]\}_{e\in E}$ assigned to its edges and the weights $\Upsilon = \{\upsilon_v \in [-\pi, \pi]\}_{v\in V}$ assigned to its vertices, then,*

$$\psi_{\mathcal{X}_G}\left(0^{|V|}\right) = \frac{1}{2^{|V|}} \mathrm{Z}_{\mathrm{Ising}}(G; i\Omega, i\Upsilon).$$

Proposition 7 is well known [20, 21]; we provide a proof in Appendix B. It will be convenient to consider graph-induced X-programs $\mathcal{X}_{G(\theta)}$ with weights that are all positive integer multiples of a real angle θ. As in Section III, this model can be implemented on the augmented graph $G' = (V', E')$ with all weights equal to θ by replacing each edge with the appropriate number of parallel edges. Let us denote the graph-induced X-program of this model by $\mathcal{X}_{G'(\theta)}$. Then, we have the following proposition.

Proposition 8.

$$\psi_{\mathcal{X}_{G(\theta)}}\left(0^{|V|}\right) = \psi_{\mathcal{X}_{G'(\theta)}}\left(0^{|V'|}\right).$$

We prove Proposition 8 in Appendix C. We also have the following proposition relating the principal probability amplitude to the Tutte polynomial of the augmented graph.

Proposition 9.

$$\psi_{\mathcal{X}_{G'(\theta)}}\left(0^{|V'|}\right) = e^{i\theta\left(r(G')-|E'|\right)}(i\sin(\theta))^{r(G')}\mathrm{T}\left(G'; x, y\right),$$

where $x = -i\cot(\theta)$ and $y = e^{2i\theta}$.

We prove Proposition 9 in Appendix D. Notice that if we let $M = (\mathcal{S}, \mathcal{I})$ be the binary matroid whose ground set $\mathcal{S}$ is the column space of the orientated incidence matrix $A_{D(G')}$ of G' with an arbitrary orientation $D(G')$ assigned to it, then we can use Proposition 6 to obtain Proposition 9.

VI. QUANTUM COMPUTATION AND THE TUTTE POLYNOMIAL

In this section we show that quantum probability amplitudes may be expressed in terms of the evaluation of a Tutte polynomial. We achieve this by showing that output probability amplitudes of a class of universal quantum circuits are proportional to the principal probability amplitude of some IQP circuit.

It will be convenient to define the following gate set.

Definition 13 ($\mathcal{G}_\theta$)**.** For a real angle $\theta \in [-\pi, \pi]$, we define $\mathcal{G}(\theta)$ to be the gate set

$$\mathcal{G}_\theta \coloneqq \{H, e^{i\theta X}, e^{i\theta XX}\},$$

where H denotes the Hadamard gate.

It is easy to see that the gate set $\mathcal{G}_{\frac{\pi}{4}}$ generates the Clifford group and the gate set $\mathcal{G}_{\frac{\pi}{8}}$ is universal for quantum computation.

In the IQP model it is easy to implement the gates $e^{i\theta X}$ and $e^{i\theta XX}$. So in order to implement the entire gate set $\mathcal{G}_\theta$, it remains to show that we can implement the Hadamard gate. This can be achieved by the use of postselection when $\theta = \frac{\pi}{4k}$ for $k \in \mathbb{Z}^+$ [6]. To apply a Hadamard gate to the target state $|\alpha\rangle_t$ consider the following Hadamard gadget. Firstly introduce an ancilla qubit in the state $|0\rangle_a$ and apply the gate $e^{\frac{i\pi}{4}(\mathbb{I}-X)_t(\mathbb{I}-X)_a}$ to $|\alpha\rangle_t |0\rangle_a$. Then measure qubit t in the computational basis and postselect on an outcome of 0. The output state of this gadget is then $H|\alpha\rangle_a$.

We shall consider quantum circuits that comprise gates from the set $\mathcal{G}_{\frac{\pi}{4k}}$ for an integer $k \in \mathbb{Z}^+$. Let $C_{k,n,m}$ denote such a circuit that acts on n qubits and comprises m Hadamard gates. Further let $\mathcal{X}_G(C_{k,n,m})$ denote the graph-induced X-program that implements the circuit $C_{k,n,m}$ by replacing each of the m Hadamard gates with the Hadamard gadget. Then we have the following proposition.

Proposition 10.

$$\langle 0^n| C_{k,n,m} |0^n\rangle = \sqrt{2}^m \psi_{\mathcal{X}_G(C_{k,n,m})}\left(0^{n+m}\right).$$

Proof. The proof follows immediately from the application of the Hadamard gadgets. ∎

Any quantum amplitude may therefore be expressed as the evaluation of a Tutte polynomial by Proposition 8, Proposition 9, and Proposition 10.

VII. EFFICIENT CLASSICAL SIMULATION OF CLIFFORD CIRCUITS

In this section we show how the correspondence between quantum computation and evaluations of the Tutte polynomial provides an explicit form for Clifford circuit amplitudes in terms of matroid invariants; namely, the *bicycle dimension* and *Brown's invariant*. This gives rise to an efficient classical algorithm for computing the output probability amplitudes of Clifford circuits. We note that it was first observed by Shepherd [2] that to compute the probability amplitude of a Clifford circuit, it is sufficient to evaluate the Tutte polynomial of a binary matroid at the point (x, y) equals $(-i, i)$, which can be efficiently computed by Vertigan's algorithm [17]. We proceed with some definitions.

Let V be a linear subspace of $\mathbb{F}_2^n$. The bicycle dimension and Brown's invariant are defined as follows.

Definition 14 (Bicycle dimension)**.** The bicycle dimension of V is defined by

$$d(V) \coloneqq \dim(V \cap V^\perp).$$

Definition 15 (Brown's invariant)**.** If $|\mathrm{supp}(x)| \equiv 0 \pmod 4$ for all $x \in V \cap V^\perp$, then Brown's invariant $\sigma(V)$ is defined to be the smallest integer such that

$$\sum_{x \in V} i^{|\mathrm{supp}(x)|} = \sqrt{2}^{d(V)+\dim(V)} e^{\frac{i\pi}{4}\sigma(V)}.$$

The following theorem of Pendavingh [10] provides an explicit form for the Tutte polynomial of a binary matroid at $(-i, i)$ in terms of the bicycle dimension and Brown's invariant.

Theorem 11 (Pendavingh [10])**.** *Let V be a linear subspace of $\mathbb{F}_2^{\mathcal{S}}$ and let $M(V)$ be the corresponding binary matroid with ground set $\mathcal{S}$. If $|\mathrm{supp}(x)| \equiv 0 \pmod 4$ for all $x \in V \cap V^\perp$, then,*

$$\mathrm{T}(M(V); -i, i) = \sqrt{2}^{d(V)} e^{\frac{i\pi}{4}(2|S| - 3r(M) - \sigma(V))}.$$

Otherwise, $\mathrm{T}(M(V); -i, i) = 0$. Further, $\mathrm{T}(M(V); -i, i)$ can be evaluated in polynomial time.

As an immediate consequence of Theorem 11, we obtain an explicit form for Clifford circuit amplitudes in terms the bicycle dimension and Brown's invariant of the corresponding matroid. Furthermore, we obtain an efficient classical algorithm for computing the output probability amplitudes of Clifford circuits. For similar results of this flavour see Refs. [22, 23].

VIII. ALGORITHM OVERVIEW

We shall now use the correspondence between quantum computation and evaluations of the Tutte polynomial to establish a heuristic algorithm for computing quantum probability amplitudes. To compute a probability amplitude, it is sufficient to compute the Tutte polynomial of a graphic matroid at $x = -i\cot\left(\frac{\pi}{4k}\right)$ and $y = e^{\frac{i\pi}{2k}}$ for an integer $k \geq 2$ [2, 5, 6]. Our algorithm will use the deletion-contraction property to recursively compute the Tutte polynomial. At each step in the recursion, the algorithm will compute certain structural properties of the graph in order to attempt to prune the computational tree. Our algorithm can be seen an adaption of the work of Haggard, Pearce, and Royle [7] to special points of the Tutte plane.

We note that our approach differs from tensor network-based methods, which involve the contraction of a graph with tensors assigned to its vertices. These methods have been used to simulate quantum computations while exploiting structural properties of the graph [4, 24–27]. However, our approach allows us to exploit an alternative class of structural properties. We proceed by describing the key aspects of our algorithm.

A. Multigraph Deletion-Contraction Formula

To improve the performance of our algorithm, we shall use the following deletion-contraction formula for multigraphs.

Proposition 12. *Let $G = (V, E)$ be a multigraph and let e be a multiedge of G with multiplicity $|e|$. If e is a loop, then*

$$\mathrm{T}(G; x, y) = y^{|e|}\mathrm{T}(G\backslash\{e\}; x, y).$$

If e is a coloop, then

$$\mathrm{T}(G; x, y) = \left(x + \sum_{i=1}^{|e|-1} y^i\right)\mathrm{T}(G/\{e\}; x, y).$$

Finally, if e is neither a loop nor a coloop, then

$$\mathrm{T}(G; x, y) = \mathrm{T}(G\backslash\{e\}; x, y) + \left(\sum_{i=0}^{|e|-1} y^i\right)\mathrm{T}(G/\{e\}; x, y).$$

Proposition 12 can easily be proven from the deletion-contraction formula by induction; we omit the proof.

If U is the underlying graph of G, then the number of recursive calls may be bounded by $O\left(2^{|E(U)|}\right)$. Alternatively, we may bound the number of recursive calls in terms of the number of vertices plus the number of edges $s = |V(U)| + |E(U)|$ in the underlying graph. The number of recursive calls R_s is then bounded by $R_s \leq R_{s-1} + R_{s-2}$, which is precisely the Fibonacci recurrence. Hence the number of recursive calls is bounded by $O\left(\phi^{|V(U)|+|E(U)|}\right)$, where $\phi = \frac{1+\sqrt{5}}{2}$ is the golden ratio [28]. A careful analysis shows that the number of recursive steps is bounded by $O\left(\tau(U) \cdot |E(U)|\right)$, where $\tau(U)$ denotes the number of spanning trees in U [14].

At each step in the recursion, we use the multigraph deletion-contraction formula to remove all multiedges that correspond to either a loop or a coloop in the underlying graph. This process contributes a multiplicative factor to the proceeding evaluation. Notice that when G is a graph whose underlying graph is a looped forest, then every edge in the underlying graph is either a loop or a coloop. Hence, we obtain the following formula for the Tutte polynomial of G.

Corollary 13. *Let $G = (V, E)$ be a multigraph whose underlying graph U is a looped forest. Further, for each edge e in U, let $|e|$ denote its multiplicity in G. Then,*

$$\mathrm{T}(G; x, y) = \prod_{\substack{e \in E(U) \\ \text{loop}}} y^{|e|} \prod_{\substack{e \in E(U) \\ \text{coloop}}} \left(x + \sum_{i=1}^{|e|-1} y^i\right).$$

Proof. The proof follows immediately from Proposition 12. ∎

B. Graph Simplification

There are a number of techniques that we can use to simplify the graph at each step in the recursion. Firstly, we may remove any isolated vertices, since they do not contribute to the evaluation.

Secondly, when $x = -i\cot\left(\frac{\pi}{4k}\right)$ and $y = e^{\frac{i\pi}{2k}}$ for an integer $k \in \mathbb{Z}^+$, we may replace each multiedge with a multiedge of equal multiplicity modulo $4k$. To account for this, we multiply the proceeding evaluation by a efficiently computable factor. Specifically, we invoke the following proposition.

Proposition 14. *Fix $k \in \mathbb{Z}^+$. Let $G = (V, E)$ be a multigraph and let $G' = (V', E')$ be the graph formed from G by taking the multiplicity of each multiedge in G modulo $4k$. Then,*

$$\mathrm{T}(G; x, y) = \left(ie^{\frac{i\pi}{4k}} \sin\left(\frac{\pi}{4k}\right)\right)^{\kappa(E)-\kappa(E')} \mathrm{T}(G'; x, y),$$

where $x = -i\cot\left(\frac{\pi}{4k}\right)$ and $y = e^{\frac{i\pi}{2k}}$.

We prove Proposition 14 in Appendix E.

C. Vertigan Graphs

The Tutte polynomial of a multigraph whose edge multiplicities are all integer multiples of an integer $k \in \mathbb{Z}^+$ may be evaluated at the point $x = -i\cot\left(\frac{\pi}{4k}\right)$ and $y = e^{\frac{i\pi}{2k}}$ in polynomial time. This can be seen by the following proposition.

Proposition 15. *Fix $k \in \mathbb{Z}^+$. Let $G = (V, E)$ be a multigraph whose edge multiplicities are all integer multiples of k. Further let $G' = (V', E')$ be the graph formed from G by taking the multiplicity of each multiedge in G divided by k. Then,*

$$\mathrm{T}(G; x, y) = \left(\sqrt{2}e^{\frac{i\pi(1-k)}{4k}} \sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)} \mathrm{T}(G'; -i, i),$$

where $x = -i\cot\left(\frac{\pi}{4k}\right)$ and $y = e^{\frac{i\pi}{2k}}$.

We prove Proposition 15 in Appendix F and note that this is a special consequence of the *k-thickening* approach of Jaeger, Vertigan, and Welsh [16]. The Tutte polynomial may then be efficiently computed by Vertigan's algorithm [17]; we call such a multigraph a *Vertigan graph*. We may therefore prune the computational tree whenever the graph is a Vertigan graph with respect to k. Note that this corresponds to quantum circuits comprising gates from the Clifford group.

D. Connected Components

The Tutte polynomial factorises over components.

Proposition 16. *Let $G = (V, E)$ be a graph with connected components $C = \{C_i\}_{i=1}^k$, then,*

$$\mathrm{T}(G; x, y) = \prod_{i=1}^{k} \mathrm{T}(C_i; x, y).$$

Proposition 16 can easily be proven from the deletion-contraction formula; we omit the proof. At each step in the deletion-contraction recursion, if the graph is disconnected, then we may use this property to prune the computational tree and hence improve performance.

E. Biconnected Components

An identical result holds for biconnected components.

Proposition 17 (Tutte [29])**.** *Let $G = (V, E)$ be a graph with biconnected components $B = \{B_i\}_{i=1}^k$, then,*

$$\mathrm{T}(G; x, y) = \prod_{i=1}^{k} \mathrm{T}(B_i; x, y).$$

Proposition 17 can easily be proven from the deletion-contraction formula. For a proof, we refer the reader to Ref. [29, Section 3]. Similarly to the connected component case, we may use this property to prune the computational tree and improve performance. Note that the biconnected components of a graph may be listed in time linear in the number of edges via depth-first search [30].

F. Multi-Cycles

The Tutte polynomial of a multigraph whose underlying graph is a cycle may be computed in polynomial time by invoking the following proposition.

Proposition 18 (Haggard, Pearce, and Royle [7])**.** *Let $G = (V, E)$ be a multigraph whose underlying graph U is an n-cycle with edges indexed by the positive integers. Further, for each edge e in U, let $|e|$ denote its multiplicity in G. Then,*

$$\mathrm{T}(G; x, y) = \sum_{k=1}^{n-2} \left(\prod_{j=k+1}^{n} y_x\left(|e_j|\right) \prod_{j=1}^{k-1} y_1\left(|e_j|\right) \right) + y_x\left(|e_n| + |e_{n-1}|\right) \prod_{j=1}^{n-2} y_1\left(|e_j|\right),$$

where $y_x(j) \coloneqq x + \sum_{i=1}^{j-1} y^i$.

Proposition 18 can easily be proven from the deletion-contraction formula. For a proof, we refer the reader to Ref. [7, Theorem 4]. We may use this proposition to prune the computational tree whenever the underlying graph is a cycle.

G. Planar Graphs

The Tutte polynomial of a planar graph along the hyperbola $(x-1)(y-1)=2$ may be evaluated in polynomial time via the Fisher-Kasteleyn-Temperley (FKT) algorithm [31–33]. We may therefore use this algorithm to prune the computational tree whenever the underlying graph is planar. Note that we may test whether a graph is planar in time linear in the number of vertices [34].

H. Edge-Selection Heuristics

The performance of our algorithm depends on the heuristic used to select edges. We shall consider six edge-selection heuristics: *vertex order*, *minimum degree*, *maximum degree*, *minimum degree sum*, *maximum degree sum*, and *non-Vertigan*. These edge-selection heuristics were first studied by Pearce, Haggard, and Royle [8], with the exception of non-Vertigan, which is specific to our algorithm.

Vertex order: The vertices of the graph are assigned an ordering. A multiedge is selected from those incident to the lowest vertex in the ordering and whose other endpoint is also the lowest vertex of any incident in the ordering. For contractions, the vertex inherits the lowest of the positions in the ordering.

Minimum degree: A multiedge is selected from those incident to a vertex with minimal degree in the underlying graph.

Maximum degree: A multiedge is selected from those incident to a vertex with maximal degree in the underlying graph.

Minimum degree sum: A multiedge is selected from those whose sum of degrees of its endpoints is minimal in the underlying graph.

Maximum degree sum: A multiedge is selected from those whose sum of degrees of its endpoints is maximal in the underlying graph.

Non-Vertigan: A multiedge is selected from those whose multiplicity is not an integer multiple of k; we call such a multiedge *non-Vertigan*. Using this edge-selection heuristic, the number of recursive calls may be bounded by $O\left(2^{\nu(G)}\right)$, where $\nu(G)$ denotes the number of non-Vertigan multiedges in G. This is due to the fact that both the deletion and contraction operation reduce the number of non-Vertigan multiedges by at least one. We note that this is similar to the *Sum-over-Cliffords* approach studied in Refs. [35–37].

I. Other Methods

There are many other methods that may improve the performance of our algorithm, which we do not study. We shall proceed by discussing some of these.

Isomorphism testing: During the computation the graphs encountered and the evaluation of their Tutte polynomial is stored. At each recursive step, we test whether the graph is isomorphic to one already encountered, and if so, we use the evaluation of the isomorphic graph instead. Haggard, Pearce, and Royle [7] showed that isomorphism testing can lead to an improvement in the performance of computing Tutte polynomials. Note that this may not be as effective when the input is a multigraph.

Almost planar: At each step in the recursion, we may test whether the graph is close to being planar, and if so, select edges in such a way that the deletion and contraction operations give rise to a planar graph. For example, if the graph is *apex*, that is, it can be made planar by the removal of a single vertex, then we may select a multiedge incident to such a vertex. Similarly, if the underlying graph is *edge apex* or *contraction apex* [38], then we may select a multiedge such that the deletion or the contraction operation gives rise to a planar graph.

k-connected components: Similarly to the connected and biconnected component case, we may compute the Tutte polynomial in terms of its k-connected components [39, 40].

IX. EXPERIMENTAL RESULTS

In this section we present some experimental results comparing the performance of the edge-selection heuristics described in Section VIII H on two classes of random quantum circuits. Our experiments were performed using SageMath 9.0 [41]. The source code and experimental data are available at Ref. [42].

The first class we consider corresponds to random instances of IQP circuits induced by dense graphs. Specifically, an instance is an IQP circuit induced by a complete graph with edge weights chosen uniformly at random from the set $\{\frac{m\pi}{8} \mid m \in \mathbb{Z}/8\mathbb{Z}\}$. This class of IQP circuits is precisely that studied in Ref. [43], where it is conjectured that approximating the corresponding amplitudes up to a multiplicative error is #P-hard on average.

The second class we consider corresponds to random instances of IQP circuits induced by sparse graphs. Specifically, an instance is an IQP circuit induced by a random graph where each of the possible edges is included independently with probability $1/2$ and with edge weights chosen uniformly at random from the set $\{\frac{m\pi}{8} \mid m \in \mathbb{Z}/8\mathbb{Z}\}$.

We run our algorithm using each of the edge-selection heuristics to compute the principal probability amplitude of 64 random instances of both the dense and sparse class on 12 vertices. The performance of each edge-selection heuristic is measured by counting the number of leaves in the computational tree. Our experimental data is presented in Appendix G. We find that the non-Vertigan edge-selection heuristic performs particularly well for the dense class and the maximum degree sum edge-selection heuristic performs particularly well for the sparse class.

X. CONCLUSION & OUTLOOK

We established a classical heuristic algorithm for exactly computing quantum probability amplitudes. Our algorithm is based on mapping output probability amplitudes of quantum circuits to evaluations of the Tutte polynomial of graphic matroids. The algorithm evaluates the Tutte polynomial recursively using the deletion-contraction property while attempting to exploit structural properties of the matroid. We considered several edge-selection heuristics and presented experimental results comparing their performance on two classes of random quantum circuits. Further, we obtained an explicit form for Clifford circuit amplitudes in terms of matroid invariants and an alternative efficient classical algorithm for computing the output probability amplitudes of Clifford circuits.

ACKNOWLEDGEMENTS

We thank Michael Bremner, Adrian Chapman, Iain Moffatt, Ashley Montanaro, Rudi Pendavingh, and Dan Shepherd for helpful discussions. This research was supported by the QuantERA ERA-NET Cofund in Quantum Technologies implemented within the European Union's Horizon 2020 Programme (QuantAlgo project), EPSRC grants EP/L021005/1, EP/R043957/1, and EP/T001062/1, and the ARC Centre of Excellence for Quantum Computation and Communication Technology (CQC2T), project number CE170100012. Data are available at the University of Bristol data repository, data.bris, at https://doi.org/10.5523/bris.kbhgclva863q21tjkqpyr5uvq.

Appendix A: Proof of Proposition 3

Proposition 3 (restatement)**.**

$$\mathrm{Z}_{\mathrm{Potts}}(G;q,\Omega,\Upsilon) = q^{-\kappa(E)}\mathrm{Z}_{\mathrm{Potts}}(G';q,\Omega\cup\Upsilon,0).$$

Proof. By definition,

$$\begin{aligned} q^{\kappa(E)}\mathrm{Z}_{\mathrm{Potts}}(G;q,\Omega,\Upsilon) &= q^{\kappa(E)}\sum_{\sigma\in\mathbb{Z}_q^V}\exp\left(\sum_{\{u,v\}\in E}\omega_{\{u,v\}}\delta(\sigma_u,\sigma_v)+\sum_{v\in V}\upsilon_v\delta(\sigma_v)\right) \\ &= q^{\kappa(E)}\prod_{i=1}^{\kappa(E)}\sum_{\sigma\in\mathbb{Z}_q^{V(C_i)}}\exp\left(\sum_{\{u,v\}\in E(C_i)}\omega_{\{u,v\}}\delta(\sigma_u,\sigma_v)+\sum_{v\in V(C_i)}\upsilon_v\delta(\sigma_v)\right). \end{aligned}$$

Now, by combining terms that are invariant under a $\mathbb{Z}_q$ symmetry, we have

$$\begin{aligned} q^{\kappa(E)}\mathrm{Z}_{\mathrm{Potts}}(G;q,\Omega,\Upsilon) &= \prod_{i=1}^{\kappa(E)}\sum_{\sigma\in\mathbb{Z}_q^{V(C_i)}}\exp\left(\sum_{\{u,v\}\in E(C_i)}\omega_{\{u,v\}}\delta(\sigma_u,\sigma_v)\right)\sum_{\sigma'\in\mathbb{Z}_q}\exp\left(\sum_{v\in V(C_i)}\upsilon_v\delta(\sigma_v,\sigma')\right) \\ &= \sum_{\sigma\in\mathbb{Z}_q^{V'}}\exp\left(\sum_{\{u,v\}\in E}\omega_{\{u,v\}}\delta(\sigma_u,\sigma_v)+\sum_{\{u,v\}\in E'\setminus E}\upsilon_v\delta(\sigma_u,\sigma_v)\right) \\ &= \mathrm{Z}_{\mathrm{Potts}}(G';q,\Omega\cup\Upsilon,0). \end{aligned}$$

This completes the proof. ∎

Appendix B: Proof of Proposition 7

Proposition 7 (restatement)**.** *Let $G=(V,E)$ be a graph with the weights $\Omega=\{\omega_e\in[-\pi,\pi]\}_{e\in E}$ assigned to its edges and the weights $\Upsilon=\{\upsilon_v\in[-\pi,\pi]\}_{v\in V}$ assigned to its vertices, then,*

$$\psi_{\mathcal{X}_G}\left(0^{|V|}\right) = \frac{1}{2^{|V|}}\mathrm{Z}_{\mathrm{Ising}}(G;i\Omega,i\Upsilon).$$

Proof. By definition,

$$\begin{aligned}
\psi_{\mathcal{X}_G}(0^{|V|}) &= \left\langle 0^{|V|} \right| \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} X_u X_v + i \sum_{v\in V} \upsilon_v X_v \right) \left| 0^{|V|} \right\rangle \\
&= \left\langle +^{|V|} \right| \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} Z_u Z_v + i \sum_{v\in V} \upsilon_v Z_v \right) \left| +^{|V|} \right\rangle \\
&= \frac{1}{2^{|V|}} \sum_{x,y\in\{0,1\}^V} \langle y| \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} Z_u Z_v + i \sum_{v\in V} \upsilon_v Z_v \right) |x\rangle \\
&= \frac{1}{2^{|V|}} \sum_{x\in\{0,1\}^V} \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} (-1)^{x_u \oplus x_v} + i \sum_{v\in V} \upsilon_v (-1)^{x_v} \right) \\
&= \frac{1}{2^{|V|}} \sum_{z\in\{-1,+1\}^V} \exp\left(i \sum_{\{u,v\}\in E} \omega_{\{u,v\}} z_u z_v + i \sum_{v\in V} \upsilon_v z_v \right) \\
&= \frac{1}{2^{|V|}} \mathrm{Z}_{\mathrm{Ising}}(G; i\Omega, i\Upsilon).
\end{aligned}$$

This completes the proof. ■

Appendix C: Proof of Proposition 8

Proposition 8 (restatement)**.**

$$\psi_{\mathcal{X}_{G(\theta)}}\left(0^{|V|}\right) = \psi_{\mathcal{X}_{G'(\theta)}}\left(0^{|V'|}\right).$$

Proof.

$$\begin{aligned}
\psi_{\mathcal{X}_{G(\theta)}}\left(0^{|V|}\right) &= \frac{1}{2^{|V|}} \mathrm{Z}_{\mathrm{Ising}}(G; i\theta, i\theta) \quad \text{(by Proposition 7)} \\
&= \frac{1}{2^{|V|}} e^{-i\theta(|E|+|V|)} \mathrm{Z}_{\mathrm{Potts}}(G; 2, 2i\theta, 2i\theta) \quad \text{(by Proposition 5)} \\
&= \frac{1}{2^{|V|+\kappa(E)}} e^{-i\theta(|E|+|V|)} \mathrm{Z}_{\mathrm{Potts}}(G'; 2, 2i\theta, 0) \quad \text{(by Proposition 3)} \\
&= \frac{1}{2^{|V|+\kappa(E)}} \mathrm{Z}_{\mathrm{Ising}}(G'; i\theta, 0) \quad \text{(by Proposition 5)} \\
&= \psi_{\mathcal{X}_{G'(\theta)}}\left(0^{|V'|}\right) \quad \text{(by Proposition 7).}
\end{aligned}$$

This completes the proof. ■

Appendix D: Proof of Proposition 9

Proposition 9 (restatement)**.**

$$\psi_{\mathcal{X}_{G'(\theta)}}\left(0^{|V'|}\right) = e^{i\theta\left(r(G')-|E'|\right)} (i\sin(\theta))^{r(G')} \mathrm{T}\left(G'; x, y\right),$$

where $x = -i\cot(\theta)$ *and* $y = e^{2i\theta}$.

Proof.

$$\begin{aligned}
\psi_{\mathcal{X}_{G'}(\theta)}\left(0^{|V'|}\right) &= \frac{1}{2^{|V'|}}\mathrm{Z}_{\mathrm{Ising}}(G';i\theta,0) \quad \text{(by Proposition 7)} \\
&= \frac{1}{2^{|V'|}}e^{-i\theta|E'|}\mathrm{Z}_{\mathrm{Potts}}(G';2,2i\theta,0) \quad \text{(by Proposition 5)} \\
&= \frac{1}{2^{r(G')}}e^{-i\theta|E'|}(e^{2i\theta}-1)^{r(G')}\mathrm{T}\left(G';-i\cot(\theta),e^{2i\theta}\right) \quad \text{(by Proposition 4)} \\
&= e^{i\theta\left(r(G')-|E'|\right)}(i\sin(\theta))^{r(G')}\mathrm{T}\left(G';-i\cot(\theta),e^{2i\theta}\right).
\end{aligned}$$

This completes the proof. ■

Appendix E: Proof of Proposition 14

Proposition 14 (restatement)**.** *Fix $k \in \mathbb{Z}^+$. Let $G=(V,E)$ be a multigraph and let $G'=(V',E')$ be the graph formed from G by taking the multiplicity of each multiedge in G modulo $4k$. Then,*

$$\mathrm{T}(G;x,y) = \left(ie^{\frac{i\pi}{4k}}\sin\left(\frac{\pi}{4k}\right)\right)^{\kappa(E)-\kappa(E')}\mathrm{T}(G';x,y),$$

where $x=-i\cot\left(\frac{\pi}{4k}\right)$ and $y=e^{\frac{i\pi}{2k}}$.

Proof.

$$\begin{aligned}
\mathrm{T}(G;x,y) &= e^{\frac{i\pi}{4k}(|E|-r(G))}\left(i\sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)}\psi_{\mathcal{X}_G\left(\frac{\pi}{4k}\right)}\left(0^{|V|}\right) \quad \text{(by Proposition 9)} \\
&= e^{\frac{i\pi}{4k}(|E|-r(G))}\left(i\sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)}\psi_{\mathcal{X}_{G'}\left(\frac{\pi}{4k}\right)}\left(0^{|V'|}\right) \\
&= e^{\frac{i\pi}{4k}\left(|E|-|E'|\right)}\left(ie^{\frac{i\pi}{4k}}\sin\left(\frac{\pi}{4k}\right)\right)^{r(G')-r(G)}\mathrm{T}(G';x,y) \quad \text{(by Proposition 9)} \\
&= \left(ie^{\frac{i\pi}{4k}}\sin\left(\frac{\pi}{4k}\right)\right)^{\kappa(E)-\kappa(E')}\mathrm{T}(G';x,y).
\end{aligned}$$

This completes the proof. ■

Appendix F: Proof of Proposition 15

Proposition 15 (restatement)**.** *Fix $k \in \mathbb{Z}^+$. Let $G=(V,E)$ be a multigraph whose edge multiplicities are all integer multiples of k. Further let $G'=(V',E')$ be the graph formed from G by taking the multiplicity of each multiedge in G divided by k. Then,*

$$\mathrm{T}(G;x,y) = \left(\sqrt{2}e^{\frac{i\pi(1-k)}{4k}}\sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)}\mathrm{T}(G';-i,i),$$

where $x=-i\cot\left(\frac{\pi}{4k}\right)$ and $y=e^{\frac{i\pi}{2k}}$.

Proof.

$$\begin{aligned}
\mathrm{T}(G;x,y) &= e^{\frac{i\pi}{4k}(|E|-r(G))}\left(i\sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)}\psi_{\mathcal{X}_G\left(\frac{\pi}{4k}\right)}\left(0^{|V|}\right) \quad \text{(by Proposition 9)} \\
&= e^{\frac{i\pi}{4k}(|E|-r(G))}\left(i\sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)}\psi_{\mathcal{X}_{G'}\left(\frac{\pi}{4}\right)}\left(0^{|V'|}\right) \\
&= \left(\sqrt{2}e^{\frac{i\pi(1-k)}{4k}}\sin\left(\frac{\pi}{4k}\right)\right)^{-r(G)}\mathrm{T}(G';-i,i) \quad \text{(by Proposition 9)}.
\end{aligned}$$

This completes the proof. ■

Appendix G: Tables of Experimental Data

We present our experimental data in Table I and Table II. The rows of the tables represent edge-selection heuristics and the columns represent quantities relating to the number of leaves in the computational trees.

	Sum	Mean	Mean Deviation	#Empty	#Vertigan	#Multicycle	#Planar
Non-Vertigan	8888920	138889	36225	912	2124367	50097	6713544
Vertex Order	37173344	580834	150982	186	167486	553618	36452054
Minimum Degree	57014650	890854	162094	0	353780	1238429	55422441
Maximum Degree	28604576	446947	119407	950	86125	412215	28105286
Minimum Degree Sum	50716183	792440	170360	0	243290	1010284	49462609
Maximum Degree Sum	10993306	171770	39409	6971	8998	78030	10899307

TABLE I. Performance of the edge-selection heuristics on 64 random instances of the dense class on 12 vertices.

	Sum	Mean	Mean Deviation	#Empty	#Vertigan	#Multicycle	#Planar
Non-Vertigan	63958	999	973	30	7994	467	55467
Vertex Order	93642	1463	1518	88	74	813	92667
Minimum Degree	412557	6446	6316	0	2158	7285	403114
Maximum Degree	91218	1425	1476	16	87	933	90182
Minimum Degree Sum	291763	4559	4630	0	1138	4169	286456
Maximum Degree Sum	50375	787	808	14	25	415	49921

TABLE II. Performance of the edge-selection heuristics on 64 random instances of the sparse class on 12 vertices.

[1] D. Aharonov, I. Arad, E. Eban, and Z. Landau, arXiv eprints (2007), arXiv:quant-ph/0702008.
[2] D. Shepherd, arXiv eprints (2010), arXiv:1005.1744.
[3] S. Fenner, F. Green, S. Homer, and R. Pruim, Proceedings of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences **455**, 3953 (1999), arXiv:quant-ph/9812056.
[4] I. L. Markov and Y. Shi, SIAM Journal on Computing **38**, 963 (2008), arXiv:quant-ph/0511069.
[5] D. Shepherd and M. J. Bremner, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences **465**, 1413 (2009), arXiv:0809.0847.
[6] M. J. Bremner, R. Jozsa, and D. J. Shepherd, Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences , rspa20100301 (2010), arXiv:1005.1407.
[7] G. Haggard, D. J. Pearce, and G. Royle, ACM Transactions on Mathematical Software (TOMS) **37**, 24 (2010).
[8] D. J. Pearce, G. Haggard, and G. Royle, in *Proceedings of the Fifteenth Australasian Symposium on Computing: The Australasian Theory-Volume 94* (Australian Computer Society, Inc., 2009) pp. 153–162.
[9] M. Monagan, arXiv eprints (2012), arXiv:1209.5160.
[10] R. Pendavingh, Journal of Algebraic Combinatorics **39**, 141 (2014), arXiv:1203.0910.
[11] D. J. Welsh, *Matroid theory* (Academic Press, 1976).
[12] J. G. Oxley, *Matroid theory*, Vol. 3 (Oxford University Press, USA, 2006).
[13] H. Whitney, American Journal of Mathematics **57**, 509 (1935).
[14] K. Sekine, H. Imai, and S. Tani, in *International Symposium on Algorithms and Computation* (Springer, 1995) pp. 224–233.
[15] A. Björklund, T. Husfeldt, P. Kaski, and M. Koivisto, in *49th Annual IEEE Symposium on Foundations of Computer Science* (IEEE, 2008) pp. 677–686, arXiv:0711.2585.
[16] F. Jaeger, D. L. Vertigan, and D. J. Welsh, in *Mathematical Proceedings of the Cambridge Philosophical Society*, Vol. 108 (Cambridge Univ Press, 1990) pp. 35–53.
[17] D. Vertigan, Journal of Combinatorial Theory, Series B **74**, 378 (1998).
[18] M. Snook, The Electronic Journal of Combinatorics **19**, 41 (2012).
[19] D. J. Welsh, London Mathematical Society Lecture Note Series **186**, 372 (1993).
[20] S. Iblisdir, M. Cirio, O. Boada, and G. Brennen, Annals of Physics **340**, 205 (2014), arXiv:1208.3918.
[21] K. Fujii and T. Morimae, New Journal of Physics **19**, 033003 (2017), arXiv:1311.2128.
[22] C. Guan and K. W. Regan, arXiv eprints (2019), arXiv:1904.00101.
[23] D. Gosset, D. Grier, A. Kerzner, and L. Schaeffer, arXiv eprints (2020), arXiv:2009.03218.
[24] B. O'Gorman, in *14th Conference on the Theory of Quantum Computation, Communication and Cryptography* (Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2019) arXiv:1906.00013.

[25] J. Gray and S. Kourtis, Quantum **5**, 410 (2021), arXiv:2002.01935.

[26] C. Huang, F. Zhang, M. Newman, J. Cai, X. Gao, Z. Tian, J. Wu, H. Xu, H. Yu, B. Yuan, *et al.*, arXiv eprints (2020), arXiv:2005.06787.

[27] F. Pan and P. Zhang, arXiv eprints (2021), arXiv:2103.03074.

[28] H. S. Wilf, *Algorithms and Complexity* (AK Peters/CRC Press, 2002).

[29] W. T. Tutte, Canadian Journal of Mathematics **6**, 80 (1954).

[30] J. Hopcroft and R. Tarjan, Communications of the ACM **16**, 372 (1973).

[31] H. N. Temperley and M. E. Fisher, Philosophical Magazine **6**, 1061 (1961).

[32] P. W. Kasteleyn, Journal of Mathematical Physics **4**, 287 (1963).

[33] P. Kasteleyn, Graph Theory and Theoretical Physics , 43 (1967).

[34] J. Hopcroft and R. Tarjan, Journal of the ACM (JACM) **21**, 549 (1974).

[35] S. Bravyi, G. Smith, and J. A. Smolin, Physical Review X **6**, 021043 (2016), arXiv:1506.01396.

[36] S. Bravyi and D. Gosset, Physical Review Letters **116**, 250501 (2016), arXiv:1601.07601.

[37] S. Bravyi, D. Browne, P. Calpin, E. Campbell, D. Gosset, and M. Howard, Quantum **3**, 181 (2019), arXiv:1808.00128.

[38] M. Lipton, E. Mackall, T. W. Mattman, M. Pierce, S. Robinson, J. Thomas, and I. Weinschelbaum, Involve, a Journal of Mathematics **11**, 413 (2017), arXiv:1608.01973.

[39] A. Andrzejak, Journal of Combinatorial Theory, Series B **70**, 346 (1997).

[40] J. Bonin and A. De Mier, Advances in Applied Mathematics **32**, 31 (2004).

[41] The Sage Developers, *SageMath, the Sage Mathematics Software System (Version 9.0)* (2020).

[42] R. L. Mann, *Data from "Simulating Quantum Computations with Tutte Polynomials"* (2021).

[43] M. J. Bremner, A. Montanaro, and D. J. Shepherd, Physical Review Letters **117**, 080501 (2016), arXiv:1504.07999.