Math. Struct. in Comp. Science (2013), *vol.* 23, *pp.* 360–385. © Cambridge University Press 2013
doi:10.1017/S096012951200014X

Extended phase map decompositions for unitaries

VEDRAN DUNJKO[†] and ELHAM KASHEFI[‡]

[†]*Department of Physics, Heriot-Watt University, Edinburgh, United Kingdom, and*
School of Informatics, University of Edinburgh, Edinburgh, United Kingdom
Email: vd51@hw.ac.uk
[‡]*School of Informatics, University of Edinburgh, Edinburgh, United Kingdom*
Email: ekashefi@gmail.com

Received 3 November 2010; revised 28 June 2011

We give a complete structural characterisation of the map implemented by the positive branch of a one-way pattern. Our approach is based on the phase map decomposition (de Beaudrap *et al.* 2006; de Beaudrap *et al.* 2008) and leads to some preliminary results on the connection between the column structure of a given unitary and the angles of measurements in a pattern that implements it. Our characterisation highlights the role of entanglement in the efficiency of the simulation of a one-way pattern, and it is a step forward towards a full characterisation of those unitaries that have an efficient one-way model implementation.

1. Introduction

The one-way model of quantum computation has attracted considerable attention, mainly because it suggests different physical realisations of quantum computing (Raussendorf and Briegel 2001; Raussendorf *et al.* 2003). In this model, quantum states are transformed using single qubit measurements on an entangled state (called the *open graph state*), which is prepared from an input state by performing controlled-Z operations on pairs of qubits, including the input system and auxiliary qubits prepared in the

$$|+\rangle = \frac{1}{\sqrt{2}}(|0\rangle + |1\rangle)$$

state. Quantum measurements are probabilistic in general, and can drive the computation over 2^n different branches, where n is the number of measurements. However, there exist sufficient conditions based on the structure of the graph state where the computation can be controlled by means of single qubit corrections, which are dependent on the previous measurement outcomes, so that the entire computation becomes deterministic (Raussendorf and Briegel 2001; Raussendorf *et al.* 2003; Danos and Kashefi 2006; Browne *et al.* 2007). In such a deterministic computation, all the branches implement the same unitary map introduced by the *positive branch*, which corresponds to the scenario in which every measurement collapses the qubit states to pre-selected states, typically

$$|+_{\alpha_i}\rangle = \frac{1}{\sqrt{2}}(|0\rangle + e^{i\alpha_i}|1\rangle).$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

We will give a complete structural characterisation of the map implemented by the positive branch of a one-way pattern. The positive branch of a one-way pattern can be expressed in terms of a *phase map decomposition* $R\Phi P$ (de Beaudrap *et al.* 2006; de Beaudrap *et al.* 2008), which we then further analyse to obtain the primary structure of the matrix M that represents $R\Phi P$ in the computational basis. The columns of M can be written as

$$M\mathrm{e}_i = \varepsilon_i B_i \overrightarrow{\varphi}$$

where ε_i are complex scalars of norm one parametrised by the measurement angles of the input qubits, B_i are sign matrices depending on the geometry of the underlying open graph state, and $\overrightarrow{\varphi}$ is a vector parametrised by the measurement angles of measured auxiliary qubits. This primary structure allows the following simple observations concerning the matrix M:

— The first column is determined only by the geometry of the open graph state and the measurement angles of the auxiliary qubits.
— All the entries of each column are sums of complex numbers of a fixed set, possibly differing in signs.
— The measurement angles of input qubits parametrise the global phase factors of columns of matrix M, which otherwise depend only on the geometry of the open graph state and the measurement angles of the auxiliary qubits.
— Moreover, we can use this characterisation to give an easy proof of the independence of the uniform determinism from the measurement angles of the input qubits.

We then proceed meticulously to dissect the B_i matrices to reveal their structure given by the following decomposition:

$$B_i = \gamma_i \Delta_i S B N \Omega_i,$$

where γ_i is a sign, which depends on the adjacency of the input qubits, and Δ_i, S, N and Ω_i are diagonal sign matrices parameterised by the adjacencies of the set of input to the set of output qubits, the adjacency of output qubits, the adjacency of measured auxiliary qubits, and the adjacency of the set of input qubits to the set of measured auxiliary qubits, respectively. Note that matrix B is a full sign matrix, parametrised by the adjacency between the set of output and the set of measured auxiliary qubits. The scalars and the matrices are given in terms of explicit functions on graphs, represented purely graph-theoretically as adjacency matrices and as lists of edges. These functions have group-theoretical properties, which we feel could be further utilised to help elucidate the following open problems:

— The simulation of given unitaries directly in the one-way model, that is, without reference to the circuit-based model.
— The characterisation of graph states that implement the same map in the positive branch.
— A refined characterisation of determinism.
— The characterisation of the pointless measurement (Kashefi *et al.* 2009), which is a key element in defining new error correcting codes.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

— The characterisation of one-way patterns that are efficiently simulatable on a classical computer.

2. Preliminaries

In this section we present a brief summary of the notation used throughout the paper and give a brief review of the one-way model.

2.1. *Qubit states*

A *qubit* is represented by a two-dimensional complex Hilbert space, called the qubit's state space. A *qubit state* is a vector of unit length in the qubit's state space. The *state space of an ensemble of qubits* is represented by the tensor product of the component state spaces, and the *state of an ensemble of qubits* is a vector of unit length in the state space of the ensemble. We use $|0\rangle$ and $|1\rangle$ to denote unit orthonormal vectors in the state space of a qubit, and these constitute the *standard computational basis* of a qubit. We use $|\pm_\alpha\rangle$ to denote vectors parameterised by the real angle α (and the choice of $+$ or $-$) defined with respect to the computational basis vectors as

$$|\pm_\alpha\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle \pm e^{i\alpha}|1\rangle\right).$$

When $\alpha = 0$, we simply write $|\pm\rangle$.

2.2. *Unitary transformations*

We use Z_α to denote a family of *phase shift unitary transformations*, parametrised by the real angle α, and this is represented in the computational basis by the matrix

$$Z_\alpha = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\alpha} \end{bmatrix}.$$

When Z_α rotation is applied to the ith qubit of an ensemble of m qubits, the transformation of the state space of the ensemble is denoted by $\mathbf{Z}_\alpha^{(i)}$, which can be given explicitly by

$$\mathbf{Z}_\alpha^{(i)} = \mathbf{I}^{\otimes(i-1)} \otimes Z_\alpha \otimes \mathbf{I}^{\otimes(m-i)},$$

where $\mathbf{I}$ denotes the identity operator on a single qubit state space. The controlled-Z operator, which is denoted by $\wedge Z$, is a unitary transformation on the state space of two qubits. In the computational basis it is given by the matrix

$$\wedge Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Note that this operator cannot be represented as a tensor product of single qubit transformations, so it can be used to create *entangled states,* which are multi-qubit states that cannot be represented as tensor products of single-qubit states.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

When the $\wedge Z$ transformation is applied to the component state subspace of the ith and jth qubit of an ensemble of m qubits, the transformation of the entire ensemble is denoted by $\wedge\mathbf{Z}_{i,j}$. The eigenvectors of the $\wedge\mathbf{Z}_{i,j}$ transformation are the vectors of the computational basis of the ensemble, with eigenvalue -1 if both the ith and jth qubit are in the state $|1\rangle$, and eigenvalue 1 otherwise.

2.3. *Miscellaneous*

— We use e to denote the basis of the natural logarithm.

— We use e_i to denote the ith vector of the canonical basis, that is, a vector with entries 0 everywhere, except a 1 at the ith entry.

— $\otimes$ represents the tensor product. $X^{\otimes n}$ denotes the nth tensor power of X, explicitly,

$$X^{\otimes n} = \overbrace{X \otimes \cdots \otimes X}^{\text{n times}}.$$

The tensor product of matrices (and also numerical vectors, as they are isomorphic to single row or column matrices) is called the Kroenecker, which is defined explicitly as follows:

If A is an m-by-n matrix and B is a p-by-q matrix, then the Kroenecker product $A \otimes B$ is the block matrix

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}.$$

— The Kroenecker product has the mixed-product property, that is, if A, B, C and D are matrices such that matrix products AC and BD are defined, then

$$(A \otimes B)(C \otimes D) = AC \otimes BD.$$

— We use $SP(n)$, for an integer n, to denote the *sign parity* function defined as

$$SP(n) = (-1)^n.$$

— We use X^τ to denote the transpose of the matrix (or vector) X.

— We use $(\cdot,\cdot)$ to denote the symmetric dot product, so if $X = [x_1, \ldots, x_n]^\tau$ and $Y = [y_1, \ldots, y_n]^\tau$ are vectors, then

$$(X, Y) = \textstyle\sum_{i=1}^{n} x_i y_i.$$

— If Γ is a graph, and $A \subseteq V(\Gamma)$ is a subset of the vertices of Γ, we use Γ_A to denote the vertex-induced subgraph of the graph Γ induced by the set of vertices A.

— If Γ is a graph, then $\#E(\Gamma)$ is the number of its edges, that is, $\#E(\Gamma) = |E(\Gamma)|$. If Γ_A is a subgraph of Γ, the graph designation can be dropped and $\#E(A)$ denotes $\#E(\Gamma_A)$. If A and B are disjoint subsets of the vertices of the graph Γ, then $\#E(A \leftrightarrow B)$ denotes the number of edges connecting the vertices in A to vertices in B in the graph Γ.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

— We use $\oplus$ to denote the modulo 2 addition, so if $X = [x_1, \ldots, x_n]^\tau$ and $Y = [y_1, \ldots, y_n]^\tau$ are vectors of integers, then

$$X \oplus Y = [x_1 \oplus y_1, \ldots, x_n \oplus y_n]^\tau .$$

— We use $\odot$ to denote the point-wise vector product, so if $X = [x_1, \ldots, x_n]^\tau$ and $Y = [y_1, \ldots, y_n]^\tau$ are vectors, then

$$X \odot Y = [x_1 y_1, \ldots, x_n y_n]^\tau .$$

2.4. *The one-way model*

We will now briefly review the one-way model (Raussendorf and Briegel 2001; Danos *et al.* 2007), and present the Phase Map Decomposition (de Beaudrap *et al.* 2006; de Beaudrap *et al.* 2008) of one-way patterns. The process of computation in the one-way model can be summarised in the following steps:

(1) We set up n input qubits in an input state $|\psi\rangle$.
(2) We add $m - n$ auxiliary qubits, prepared in the state $|+\rangle$.
(3) We create the pairwise entanglement of some qubits by means of the $\wedge Z$ interaction. This interaction is represented by an open graph state, an ordered triplet (Γ, I, O), where Γ represents the entanglement graph (two qubits are entangled if and only if the corresponding vertices are adjacent), I is the set of input qubits/vertices and O is the set of output qubits/vertices, which is a subset of the auxiliary qubits.
(4) We measure the input qubits and non-output auxiliary qubits (which we call *pure auxiliary qubits*) in the (X, Y) Bloch sphere plane, that is, in the basis pairs

$$\{(|+_{\alpha_i}\rangle, |-_{\alpha_i}\rangle)\},$$

parametrised by a set of measurement angles $\{\alpha_1, \ldots, \alpha_{m-n}\}$. The set O corresponds to the qubits that will not be measured.

Without loss of generality, we assume input and output qubits are not overlapping. This is not a restriction, as additional auxiliary qubits can be added, which will correspond to the overlapping qubits, to which the quantum state of the overlapping qubits can be teleported. It is easy to show that these two scenarios are equivalent. As quantum measurements are generally probabilistic, the pattern implements a general completely positive map (Danos *et al.* 2007). The scenario in which each measurement corresponds to the projection into the $|+_{\alpha_i}\rangle$ state is called the positive branch, and the positive branch realises a linear transformation of the Hilbert space of the input qubits to the Hilbert space of the output qubits. The corresponding model is called projection-based quantum computing (Danos *et al.* 2009).

We focus on the positive branch only, and for this not to be a restriction, it will suffice that the graph Γ, defined by the underlying graph state, fulfils the graph-theoretical condition of having *flow* or *generalised flow* (Danos and Kashefi 2006; Browne *et al.* 2007), since then, using local single qubit corrections conditioned on sequential measurement outcomes, the entire quantum evolution of the system can be driven to be equal to the positive branch.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

We will choose a labelling of qubits such that the first n labels correspond to the input qubits, the following $a = m - 2n$ correspond to the measured auxiliary qubits (which we will call pure auxiliaries), and the last n correspond to the output qubits. We have assumed that in a given one-way pattern all the input qubits are measured first (the first round of the computation). It would be easy to adapt the whole discussion of this paper to the scenario where there are no input qubits or some of the pure auxiliary qubits are also measured in the first round by labelling such qubits among the first n labels.

The measuring of a qubit in the $\{|\pm_\alpha\rangle\}$ basis is equivalent to first locally rotating that same qubit by the local Z_α unitary transformation, followed by a measurement in the $\{|\pm\rangle\}$ basis. Hence, since $\wedge Z$ and Z_α are commuting, the projection-based computation process can be restated as follows:

(1) We set up n input qubits in the input state $|\psi\rangle$.
(2) We add $m - n$ auxiliary qubits, set in the state $|+\rangle$.
(3) We apply local Z_α rotations to the input and $m - 2n$ auxiliary qubits, corresponding to the measurement angles $\{\alpha_1, \ldots, \alpha_{m-n}\}$.
(4) We create a pairwise entanglement of some qubits by means of the $\wedge Z$ interaction. This interaction is represented by an open graph state, an ordered triplet (Γ, I, O), where Γ represents the entanglement graph (two qubits are entangled if and only if the corresponding vertices are adjacent), I is the set of input qubits/vertices and O is an n qubit/vertex subset of the auxiliary qubits representing the output qubits.
(5) We project the input qubits and $m - 2n$ auxiliary qubits to the $|+\rangle$ state.

The first and second steps above involve embedding a 2^n dimensional Hilbert space into a 2^m dimensional Hilbert space, which we will denote by P (for *preparation map*), which is given explicitly by

$$P : |\psi\rangle \to |\psi\rangle \otimes |+\rangle^{\otimes(m-n)}.$$

The application of the $m - n$ local rotations implements a map, which we denote by Φ_1:

$$\Phi_1 = \prod_{i=1}^{m-n} \mathbf{Z}_{-\alpha_i}^{(i)}$$

Note that this is an operator on a 2^m dimensional Hilbert space. We collect the entangling interactions, $\wedge Z$, into the map Φ_2:

$$\Phi_2 = \prod_{(i,j)\in E(\Gamma)} \wedge \mathbf{Z}_{i,j}.$$

We call the cumulative action of the latter two maps the *Phase map* and denote it by Φ:

$$\Phi = \Phi_2 \Phi_1.$$

The final step of the computation consists of projecting the first $m - n$ qubits to the state $|+\rangle$, which we denote by R (for *restriction map*):

$$R = \langle +|^{\otimes(m-n)} \otimes \mathbf{I}_{2^n}.$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

where $\mathbf{I}_{2^n}$ is the identity map on the 2^n-dimensional Hilbert space. The entire process of computation in the projection-based model is now represented by

$$R\Phi P,$$

and we call this representation the *Phase map decomposition* of a given unitary operator implemented in the one-way pattern (de Beaudrap *et al.* 2006; de Beaudrap *et al.* 2008). Note that we can also derive a phase map decomposition directly for any unitary operators without any references to the one-way pattern (de Beaudrap *et al.* 2006).

3. Structural characterisation of the phase map decomposition

We use $\{|i\rangle\}_{i=1}^{2^n}$ to denote the standard computational orthonormal basis of a 2^n dimensional complex Hilbert space. Every computational basis in this representation describes a sequence of 0 and 1, which is the binary representation of the integer value $i-1$. Therefore, $i-1$ represented in binary, encodes the choice of states $|0\rangle$ or $|1\rangle$ in the component single qubit state spaces. For example, the state $|3\rangle$, in a four qubit setting, represents the state $|0\rangle|0\rangle|1\rangle|0\rangle$ as $(3-1)=(0010)_2$.

We will now refine the expression $R\Phi P|i\rangle$ to obtain the structure of the ith column of the matrix, which represents $R\Phi P$ in the computational basis.

Theorem 3.1. Let $R\Phi P$ be a phase map decomposition corresponding to a positive branch of a one-way pattern over m qubits, with n non-overlapping input and output qubits and $a=m-2n$ measured auxiliary qubits, and with the set of measurement angles $\{-\alpha_1,\ldots,-\alpha_{n+a}\}$. Then, the matrix M representing $R\Phi P$ is characterised with respect to columns by the following equality:

$$M\mathrm{e}_i = \varepsilon_i B_i \overrightarrow{\varphi} \tag{1}$$

where

— e_i is the ith vector of the canonical basis
— $\varepsilon_i = \left(\bigotimes_{k=1}^{n} \begin{bmatrix} 1 \\ e^{i\alpha_k} \end{bmatrix}, \mathrm{e}_i\right)$, with $(\cdot,\cdot)$ denoting the symmetric dot product
— $\overrightarrow{\varphi} = \bigotimes_{k=n+1}^{n+a} \begin{bmatrix} 1 \\ e^{i\alpha_k} \end{bmatrix}$
— B_i is a matrix of signs of dimension $2^n \times 2^a$, which depends on the underlying graph state – we call the B_i the sign pattern matrices.

Proof. The proof is based on simple linear algebra manipulations. The main properties used are the diagonal form of both Z_α and $\wedge Z$ in the computational basis. The complex phases arising from the Z_α local rotations are collected in the $\overrightarrow{\varphi}$ vector and in the scalars ε_i, and the diagonal of the Φ_2 entangling operation gets spread across the sign pattern matrices B_i. Let $R\Phi P$ be the phase map decomposition (de Beaudrap *et al.* 2006) of the positive branch of a one-way pattern over m qubits, n of which are input, n output, and

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

$a = m-2n$ are pure auxiliary qubits. Also, let $|i\rangle$ be a vector of the standard computational basis. We can then derive the following directly:

$$\begin{aligned}
R\Phi P|i\rangle &= I_{2^n} R\Phi P|i\rangle \\
&= \sum_{j=1}^{2^n} |j\rangle\langle j|R\Phi P|i\rangle \\
&= \sum_{j=1}^{2^n} |j\rangle\langle j|R\Phi \left(|i\rangle \otimes |+\rangle^{\otimes m-n}\right) \\
&= \sum_{j=1}^{2^n} |j\rangle\langle j|R\Phi_2\Phi_1 \left(|i\rangle \otimes |+\rangle^{\otimes m-n}\right) \\
&= \left(\left(\otimes_{k=1}^{n}\langle +_{\alpha_k}|\right)|i\rangle\right) \sum_{j=1}^{2^n} |j\rangle\langle j|R\Phi_2 \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right)
\end{aligned}$$

(defining the row-constant scalar $C_i = ((\otimes_{k=1}^{n}\langle +_{\alpha_k}|)|i\rangle)$)

$$\begin{aligned}
&= C_i \sum_{j=1}^{2^n} |j\rangle\langle j|R\Phi_2 \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right) \\
&= C_i \sum_{j=1}^{2^n} |j\rangle\langle j|\langle +|^{\otimes (m-n)} \otimes I_{2^n}\Phi_2 \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right) \\
&= C_i \sum_{j=1}^{2^n} |j\rangle \left(\langle +|^{\otimes (m-n)} \otimes \langle j|\right)\Phi_2 \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right)
\end{aligned}$$

(noting that $\Phi_2 = \sum_{l=1}^{2^m} b_l|l\rangle\langle l|$ where b_l is the lth diagonal element of the diagonal matrix Φ_2)

$$\begin{aligned}
&= C_i \sum_{j=1}^{2^n} |j\rangle \left(\left(\langle +|^{\otimes (m-n)} \otimes \langle j|\right) \sum_{l=1}^{2^m} b_l|l\rangle\langle l|\right) \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right) \\
&= C_i \sum_{j=1}^{2^n} |j\rangle \left(\sum_{l=1}^{2^{(m-n)}} b_{[(l-1)2^n+j]}\langle l|\langle j|\right) \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right) \qquad (2) \\
&= C_i \sum_{j=1}^{2^n} |j\rangle \left(\sum_{l=1}^{2^{(m-n)}} b_{[(l-1)2^n+j]}\langle l|\langle j| \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right) \otimes |+\rangle^{\otimes n}\right)\right) \\
&= C_i \sum_{j=1}^{2^n} |j\rangle \left(\sum_{l=1}^{2^{(m-n)}} b_{[(l-1)2^n+j]}\langle l| \left(|i\rangle \otimes \left(\otimes_{k=n+1}^{m-n}|+_{\alpha_k}\rangle\right)\right)\right) \qquad (3)
\end{aligned}$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

$$= C_i \sum_{j=1}^{2^n} |j\rangle \left(\sum_{l=1}^{2^{(m-2n)}} b_{[(i-1)2^{(m-n)}+(l-1)2^n+j]} \langle l| \left(\otimes_{k=n+1}^{m-n} |+_{\alpha_k}\rangle \right) \right)$$
$$= C_i \sum_{j=1}^{2^n} \sum_{l=1}^{2^{(m-2n)}} b_{[(i-1)2^{(m-n)}+(l-1)2^n+j]} |j\rangle\langle l| \left(\otimes_{k=n+1}^{m-n} |+_{\alpha_k}\rangle \right).$$

Summarising, and reintroducing the scalar, we have:

$$R\Phi P|i\rangle = \left(\left(\bigotimes_{k=1}^{n} \langle +_{\alpha_k}| \right) |i\rangle \right) \quad \sum_{j=1}^{2^n} \sum_{l=1}^{2^{(m-2n)}} b_{[(i-1)2^{(m-n)}+(l-1)2^n+j]} |j\rangle\langle l| \left(\bigotimes_{k=n+1}^{m-n} |+_{\alpha_k}\rangle \right).$$

For simplicity, in the above we have omitted:

— a global scaling factor of

$$2^{-(\frac{m-n}{2})},$$

which arises from the scalar products

$$\langle +|^{\otimes(m-n)} \langle j||l\rangle$$

when they are non-zero; and

— the global scaling factor

$$2^{-(\frac{n}{2})},$$

caused by the product

$$\langle j|+\rangle^{\otimes n}.$$

The overall (omitted) scaling factor is

$$2^{-\frac{m}{2}}.$$

The expression $((\bigotimes_{k=1}^{n} \langle +_{\alpha_k}|)|i\rangle)$ is a scalar that depends on the column i, and we denote it by ε_i. We also let

$$B_i = \sum_{j=1}^{2^n} \sum_{l=1}^{2^{(m-2n)}} b_{[(i-1)2^{(m-n)}+(l-1)2^n+j]} |j\rangle\langle l| \tag{4}$$

be a $2^n \times 2^{(m-2n)}$ matrix expressed in the computational basis that depends on the choice of column i with entries in $\{-1, 1\}$. Finally, we use $\overrightarrow{\varphi}$ to denote the numeric representation in the computational basis of the vector

$$\left(\bigotimes_{k=n+1}^{m-n} |+_{\alpha_k}\rangle \right),$$

which is independent of the choice of the column. This corresponds to the quantum state of the auxiliary qubits after the local Z_α rotations, but before the entanglement procedure. The entire expression can then be rewritten in matrix notation as

$$M\mathrm{e}_i = \varepsilon_i B_i \overrightarrow{\varphi},$$

where e_i is the ith vector of the canonical basis. □

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

For simplicity, in the expression for $\overrightarrow{\varphi}$ as a numerical vector, we omit a normalising factor of

$$2^{-(\frac{m-2n}{2})},$$

along with the scaling factor

$$2^{-(m-n)}$$

of the B_i matrices since it has no bearing on the structure we wish to present.

A few direct consequences of Theorem 3.1 are easily checked:

— The first column of M is parametrised by the measurement angles of pure auxiliary qubits only, since $\varepsilon_1 = 1$.

— For every column i, the entries per row, are of the form

$$\varepsilon_i(r, \overrightarrow{\varphi}) \tag{5}$$

for some vector of signs r. So, entries of a column are sums of the same set of elements, varying possibly in signs only.

— From the expression (5) it is clear that every entry of every column is a sum of elements of the set of entries of the vector $\overrightarrow{\varphi}$ varying in signs, multiplied by the column's corresponding global phase factor ε_i.

As mentioned above, we can also prove the following simple lemma about the uniform determinism. Recall that a pattern is said to be uniform deterministic if it is deterministic for all possible angles of measurements.

Lemma 3.2. A pattern is uniformly deterministic if and only if it is deterministic for all possible choices of auxiliary measurement angles.

Proof. From Theorem 3.1, the measurement angles of the input qubits appear only as global phase factors of the columns of M, and these global factors ε_i are of norm one. Hence, the choice of measurement angles of input qubits does not influence the norm of the columns. Also, regardless of the measurement angles of the input qubits (since the product of two complex numbers of norm 1 is always norm 1), the matrix M is orthogonal since its columns are orthonormal. Therefore, uniform determinism can only depend on the measurement angles of the measured auxiliary qubits. □

The statement of Theorem 3.1 suggests a direct method for addressing problems of equalities of patterns, and of simulating a given unitary evolution of a quantum system in the one-way model. For the first problem, we have to evaluate and check the equality of two expressions of the form of the right-hand side of equation (1). However, that entails knowing how to construct the sign pattern matrices from given graph states. This demands further analysis of the sign pattern matrices, which is the topic of the next two sections.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

4. Graph-theoretical characterisation of sign pattern matrices

In the proof of Theorem 3.1, the matrices B_i were defined as representations of the expression

$$\sum_{j=1}^{2^n} \sum_{l=1}^{2^{(m-2n)}} b_{[(i-1)2^{(m-n)}+(l-1)2^n+j]} |j\rangle\langle l| \tag{6}$$

in the computational basis, and in that representation b_l corresponds to the the lth diagonal entry of the matrix representation of Φ_2 in that same basis. We now link the graph-theoretical aspects of the graph state defining the pattern and the above expression. It was shown in de Beaudrap *et al.* (2006) that b_l, the lth diagonal element of Φ_2, is given by the expression

$$b_l = SP\left(\sum_{(i,j)\in E(\Gamma)} x_i x_j\right), \tag{7}$$

where x_k was defined as the kth most significant digit (kth digit from the left, counting the leading zeroes) of $l-1$ represented in binary. It is easy to give a graph-theoretical representation for the expression for b_l. It can be shown that the expression $\sum_{(i,j)\in E(\Gamma)} x_i x_j$, where x_i and x_j are functions of l, counts the number of edges of a vertex-induced subgraph of the graph Γ, where the vertex set that induces the subgraph is determined by l. In order to explain how l determines a subset of vertices, we now define a function that realises this vertex determination process as an integer.

Definition 4.1. If V is a set of vertices labelled with integers $\{1,\ldots,m\}$ and k is an integer in $\{1,\ldots,2^m\}$, then the selection function Sel is defined as follows: $Sel(V,k)$ is a subset of V such that for the vertex labelled with l (which appears as the subscript) $v_l \in V$, we have $v_l \in Sel(V,k)$ holds if and only if the lth most significant digit of the m digit binary representation (that is, counting leading zeroes) of $k-1$ is 1.

This function easily extends to any finite totally ordered set O through an order-preserving bijection between O and $\{1,\ldots,|O|\}$. Also, we will use an expression of the form *a subset of S, selected by (the integer) k* to mean precisely $Sel(S,k)$. Using this terminology, b_l is the sign parity of the number of edges of the vertex-induced subgraph of Γ induced by a subset of the vertices of Γ, *selected by* l.

We can now direct our attention to the expression (4) and state the following proposition about the graph-theoretical characterisation of the sign pattern matrices B_i.

Proposition 4.2. Every entry $(B_i)_{p,q}$ is the sign parity of the number of edges of the vertex-induced subgraph of Γ defined by the vertices V' that depends on the triplet (i,p,q) as follows:

$$V' = Sel(I,i) \cup Sel(Aux,q) \cup Sel(O,p),$$

where O denotes the subset of output vertices, Aux denotes the subset of pure auxiliary vertices and I denotes the subset of input qubits.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

Proof. First, we recall that all of the entries of the B_i matrices correspond to some diagonal elements of the map Φ_2 induced by the topology of the underlying graph state. From (6), it is clear that the (p,q) element of the matrix B_i is

$$b_{[(i-1)2^{(m-n)}+(q-1)2^n+p]},$$

that is, the $\left((i-1)\,2^{(m-n)}+(q-1)\,2^n+p\right)$th diagonal element of the matrix Φ_2.

Second, the lth diagonal element of the matrix Φ_2 was given by the expression (7), which we repeat here for convenience (but with altered labels to avoid confusion):

$$b_l = SP\left(\sum_{(a,b)\in E(\Gamma)} x_a x_b\right),$$

where x_k was defined as the kth most significant digit (kth digit from the left, counting the leading zeroes) of $(l-1)$ represented in binary.

The sum $\sum_{(a,b)\in E(\Gamma)} x_a x_b$ goes across all the edges of the graph Γ, and the summand corresponding to an edge (a,b) is unity if both the ath and bth most significant digit of $(l-1)$ are unity, and zero otherwise.

This is clearly just the number of edges of a graph Γ' obtained from Γ by removing all the vertices that have the label L such that the Lth most significant digit of $(l-1)$ represented in binary is zero. Using the terminology of the selection function, the sum $\sum_{(a,b)\in E(\Gamma)} x_a x_b$ equals the number of edges of a vertex-induced subgraph Γ' of Γ, where the inducing vertex subset is given by $Sel(V,l)$ (V being the set of vertices, ordered by the labels). Hence, the (p,q) element of the matrix B_i is the sign parity of the number of edges of the graph Γ'.

Since in our case

$$l = (i-1)2^{(m-n)} + (q-1)2^n + p,$$

the binary representation of $(l-1)$ is given as a concatenation of binary strings representing $(i-1)$, $(q-1)$ and $(p-1)$, respectively, while keeping the leading zeroes of each component string. It is then obvious that given the expression

$$Sel\left(V,(i-1)2^{(m-n)}+(q-1)\,2^n+p\right),$$

the value of i selects a subset of the first n vertices (input vertices), q selects the following $m-2n$ vertices (auxiliary vertices) and p selects the final n vertices (output vertices). Hence,

$$Sel(V,l) = Sel(I,i) \cup Sel(Aux,q) \cup Sel(O,p).$$

Since we have shown that the set of vertices $Sel(V,l)$ induces the graph Γ', this concludes the proof. □

The proposition can be elaborated in the following fashion. The (p,q) entry of the sign pattern matrix B_i is the sign parity of the number of edges of a vertex induced subgraph of Γ. The inducing subset of vertices is a union of subsets of the input, output and pure auxiliary vertices. The index of B_i (the corresponding column of the matrix M) i selects a subset of the input vertices; the row p selects a subset of the output vertices; and, finally, the column q selects a subset of the pure auxiliary vertices.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

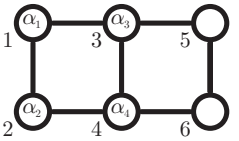


Fig. 1. The 2×3 cluster state. The leftmost pair of qubits (labelled 1 and 2) are the input qubits, and the rightmost (labelled 5 and 6) are the outputs. We denote the local rotation angles (corresponding to the measurement angles) using $\{\alpha_1, \ldots, \alpha_4\}$. The vertex labels are written at the lower left of the vertices.

Example 1. We will now illustrate the application of Theorem 3.1 and Proposition 4.2 on a simple but non-trivial underlying graph state: the 2×3 cluster state. This cluster state is illustrated in Figure 1.

The table given in Figure 2 illustrates the application of Proposition 4.2 to this cluster state by calculating the corresponding matrix B_k.

In summary, the matrix B_4 is given by

$$B_4 = \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 \end{bmatrix}.$$

By Theorem 3.1, the fourth column of the matrix M representing the positive branch of a computation preformed on this cluster state can now be given as:

$$M\mathrm{e}_4 = \varepsilon_4 B_4 \overrightarrow{\varphi}.$$

Note that $\varepsilon_4 = e^{i(\alpha_1+\alpha_2)}$ and $\overrightarrow{\varphi} = [\begin{smallmatrix} 1 \\ e^{i\alpha_3} \end{smallmatrix}] \otimes [\begin{smallmatrix} 1 \\ e^{i\alpha_4} \end{smallmatrix}]$, so the column is characterised by the expression

$$M\mathrm{e}_4 = e^{i(\alpha_1+\alpha_2)} \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} 1 \\ e^{i\alpha_4} \\ e^{i\alpha_3} \\ e^{i\alpha_3+\alpha_4} \end{bmatrix}.$$

Potentially, Theorem 3.1 and Proposition 4.2 could be used to address a number of problems, such as the equality of patterns and the simulation of a given unitary. In doing so, the essential expression we need to calculate is the expression $\varepsilon_i(r, \overrightarrow{\varphi})$. If we are interested in verifying the equality of two patterns, we need to calculate and compare the matrices of their phase map decompositions, given by Theorem 3.1. This entails calculating the dot product of the rows of the matrices B_i and the vectors $\overrightarrow{\varphi}$.

Similarly, if we are trying to simulate a given unitary, then expression $\varepsilon_i(r, \overrightarrow{\varphi})$, which contains variables, will form the left-hand sides of a system of equations that we will have to solve (the right-hand side being the entries of the given unitary).

The dot product of rows of the sign pattern matrices and the vector $\overrightarrow{\varphi}$ is in general hard to evaluate since both vectors have exponential lengths in the number of pure auxiliary

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

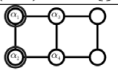
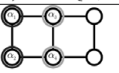
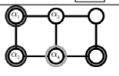
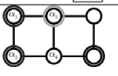
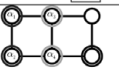
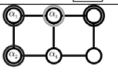
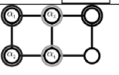
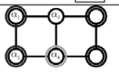
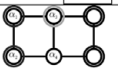
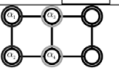
B_i, $\underline{i=4}$ $(\underline{11}) \to \{1,2\}$	q (column)	1	2	3	4
$\overline{p}$ (row)		$(00) \xrightarrow{Sel} \{\}$	$(01) \xrightarrow{Sel} \{4\}$	$(10) \xrightarrow{Sel} \{3\}$	$(11) \xrightarrow{Sel} \{3,4\}$
$\overline{1}$	$(\overline{00}) \xrightarrow{Sel} \{\}$	 $(\underline{11}00\overline{00})_2$ $\#E=1$ $SP=\boxed{-1}$	 $(\underline{11}01\overline{00})_2$ $\#E=2$ $SP=\boxed{1}$	 $(\underline{11}10\overline{00})_2$ $\#E=2$ $SP=\boxed{1}$	 $(\underline{11}11\overline{00})_2$ $\#E=4$ $SP=\boxed{1}$
$\overline{2}$	$(\overline{01}) \xrightarrow{Sel} \{6\}$	 $(\underline{11}00\overline{01})_2$ $\#E=1$ $SP=\boxed{-1}$	 $(\underline{11}01\overline{01})_2$ $\#E=3$ $SP=\boxed{-1}$	 $(\underline{11}10\overline{01})_2$ $\#E=2$ $SP=\boxed{1}$	 $(\underline{11}11\overline{01})_2$ $\#E=5$ $SP=\boxed{-1}$
$\overline{3}$	$(\overline{10}) \xrightarrow{Sel} \{5\}$	 $(\underline{11}00\overline{10})_2$ $\#E=1$ $SP=\boxed{-1}$	 $(\underline{11}01\overline{10})_2$ $\#E=2$ $SP=\boxed{1}$	 $(\underline{11}10\overline{10})_2$ $\#E=3$ $SP=\boxed{-1}$	 $(\underline{11}11\overline{10})_2$ $\#E=5$ $SP=\boxed{-1}$
$\overline{4}$	$(\overline{11}) \xrightarrow{Sel} \{5,6\}$	 $(\underline{11}00\overline{11})_2$ $\#E=2$ $SP=\boxed{1}$	 $(\underline{11}01\overline{11})_2$ $\#E=4$ $SP=\boxed{1}$	 $(\underline{11}10\overline{11})_2$ $\#E=4$ $SP=\boxed{1}$	 $(\underline{11}11\overline{11})_2$ $\#E=7$ $SP=\boxed{-1}$

Fig. 2. Calculation of matrix B_4 for the 2×3 cluster state. The rows and columns of the table correspond to the rows and columns of the matrix B_4. The entries of the table are coded as follows: the underlined data is associated with the choice of the matrix B_4; the over-lined data is associated with the choice of row of B_4, that is, p; and the grey data is associated with the choice of the column of B_4. Each entry of the table contains a cluster state with the vertex-inducing set emphasised by the encircling of the qubits. The final entries of the matrix B_4 are shown as the boxed values.

qubits. However, the vector $\overrightarrow{\varphi}$ is represented as

$$\overrightarrow{\varphi} = \bigotimes_{k=n+1}^{n+a} \begin{bmatrix} 1 \\ e^{i\alpha_k} \end{bmatrix}.$$

Such a representation contains the same number of 2-dimensional vectors as there are measured auxiliary qubits, and so is efficient. The ability to represent the rows of the sign pattern matrices in such a compact form might assist in deriving techniques for solving and evaluating such expressions efficiently.

Hence, in the next section we will focus our attention on the structure of the rows of the sign pattern matrices and present the decomposition theorem for the sign pattern matrices.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

5. Decomposition of the sign pattern matrices

If we turn our attention to any row p of any matrix B_i, we can see from Proposition 4.2 that by selecting the index i (equivalently, a column of M) and a row p we have fixed a subset of the input qubits and a subset of output qubits, respectively. The pth row of B_i is then generated by the sign parities of the numbers of edges of the vertex-induced subgraphs of Γ, where the inducing set is a union of the selected fixed sets of input and output vertices, and the entry of that row (the column of B_i) then selects the additional subset of the pure auxiliary vertices. Therefore, for fixed p and i, the corresponding row of B_i, which we denote by r, is given entry-wise by the expression

$$(r)_k = SP\left(\#E\left(\Gamma_{Sel(I,i)\cup Sel(O,p)\cup Sel(Aux,k)}\right)\right). \tag{8}$$

Note that in the above expression, only the subsets of pure auxiliary vertices change as we go across the entries of r.

The subgraph inducing vertex subset is expressed as a union of three subsets, of which two are constant and one is variable. We use the denotations $A = Sel(O,p)$, $B = Sel(I,i)$ and $X = Sel(Aux,k)$. As we will be dealing with only one graph at a time, we will drop the graph designation and use the shorthand $\#E(A)$ instead of $\#E(\Gamma_A)$. Also, we use $\#E(Y \leftrightarrow Z)$ to denote the number of edges joining vertices in Y with vertices in Z in the graph we are observing. It is then easy to see that

$$\begin{aligned}\#E(A \cup X \cup B) = \#E(A) + \#E(X) + \#E(B) + \#E(B \leftrightarrow A)\\ + \#E(A \leftrightarrow X) + \#E(B \leftrightarrow X).\end{aligned} \tag{9}$$

Equality (9) and the fact that the sign parity function is a homomorphism from the additive monoid of integers to the multiplicative monoid of integers,

$$SP(i+j) = SP(i)SP(j),$$

will give a basis for the decomposition of the sign parity matrices. Therefore, we can express the vector r with respect to the entries as follows:

$$\begin{aligned}(r)_k = SP\left(\#E(B)\right) SP\left(\#E(B \leftrightarrow A)\right) SP\left(\#E(A)\right)\\ SP\left(\#E(A \leftrightarrow X)\right) SP\left(\#E(X)\right) SP\left(\#E(B \leftrightarrow X)\right).\end{aligned} \tag{10}$$

Note the following dependencies of the factors of the right-hand side of the above expression:

(1) $SP\left(\#E(B)\right)$ depends on i (the choice of the column) only since it corresponds to a choice of the subset of output vertices.

(2) $SP\left(\#E(B \leftrightarrow A)\right)$ depends on both i and p (the row selection of B_i), but is independent of k.

(3) $SP\left(\#E(A)\right)$ depends on p only since it corresponds to a choice of input vertices.

(4) $SP\left(\#E(A \leftrightarrow X)\right)$ depends on p and k.

(5) $SP\left(\#E(X)\right)$ depends on k only.

(6) $SP\left(\#E(B \leftrightarrow X)\right)$ depends on i and k.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

We have represented the fixed row of a sign pattern matrix r by its entries. We will now represent r using vector functions defined on graphs since that will allow for a simple characterisation of B_i matrix entries.

First, note that in the list of dependencies of factors that make up an entry of f, the first three are constants in k. The last three factors depend on k, and we shall represent them as components of values (which are vectors) of two different vector functions on graphs. We define a function on simple graphs whose set of vertices are equipped with a strict order.

Definition 5.1. Let Γ be a simple graph, where the set of vertices is equipped with a strict order. We define $\mathscr{P}(\Gamma)$ to be a vector of signs of length $2^{|V|}$ given by the following components

$$(\mathscr{P}(\Gamma))_k = SP\,(\#E(Sel(V,k)))$$

for all $k = 1,\ldots,2^{|V|}$.

Since we will often be expressing the $\mathscr{P}$ function of some vertex-induced subgraph of a graph, it is convenient to adopt a shorthand notation. If the graph Γ in question is clear and S is a subset of its set of vertices, we will write $\mathscr{P}(S)$ as shorthand for $\mathscr{P}(\Gamma_S)$. The other useful function is defined on bipartite graphs.

Definition 5.2. Let Γ be a bipartite graph with partitions V and W where the set of vertices is equipped with a strict order. We define $\mathscr{B}_\Gamma(V,W)$ to be a vector of signs of length $2^{|W|}$ given by the components

$$(\mathscr{B}_\Gamma(V,W))_k = SP\,(\#E(V \cup Sel(W,k)))$$

for $k = 1,\ldots,2^{|W|}$.

Again, if the graph Γ is clear from context, we will omit the subscript Γ and simply write $\mathscr{B}(V,W)$ instead of $\mathscr{B}_\Gamma(V,W)$. These two functions can be defined explicitly on different representations of graphs, and these representations have potentially useful properties. We will give these properties after we have given the theorem about the decomposition of the sign pattern matrices.

The row r can now be expressed (as its transpose, that is as a column) using $\mathscr{B}$ and $\mathscr{P}$ functions. As the goal is to represent a general column r (that is, any of the rows of any matrix B_i), we will introduce these parameters for row r in the form of its row index p and its sign pattern matrix, which will be denoted by i. Therefore, $r_{p,i}$ is now expressed as

$$r_{p,i} = \gamma_i\, c^1_{p,i}\, c^2_p \cdot (\mathscr{B}(Sel(O,p),Aux) \odot \mathscr{P}(Aux) \odot \mathscr{B}(Sel(I,i),Aux))\,. \tag{11}$$

The order of the components corresponds to the order of factors in the entry-wise representation of r in equation (10).

In (11), γ_i is a scalar, corresponding to the factor $SP\,(\#E(B))$, so we can represent it using the $\mathscr{P}$ function as

$$\gamma_i = (\mathscr{P}(I))_i\,.$$

Also, $c^1_{p,i}$ is a constant scalar for every entry of a fixed row (and thus depends on the row and the choice of B_i) and corresponds to the expression $SP\,(\#E(B \leftrightarrow A))$. It can be

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

represented using the $\mathscr{B}$ function:

$$c^1_{p,i} = (\mathscr{B}(Sel(I,i),O))_p.$$

Finally, c^2_p is a constant scalar for a fixed row, does not depend on the choice of B_i and corresponds to the term $SP\,(\#E(A))$. It can be represented using the function $\mathscr{P}$:

$$c^2_p = (\mathscr{P}(O))_p\,.$$

The three row-wise constant factors have been defined as components of vectors that depend on i or are constant. Then, by collecting the components across rows, and indexes i, we can easily note the following deconstruction of the sign pattern matrices.

Theorem 5.3. Let $V = I \cup Aux \cup O$ be the set of vertices of the graph Γ, partitioned into input, auxiliary and output vertices. Let

$$\begin{aligned}
\gamma_i &= (\mathscr{P}(\Gamma_I))_i \\
\Delta_i &= diag(\mathscr{B}(Sel(I,i),O)) \\
S &= diag\,(\mathscr{P}(O)) \\
B &= [\mathscr{B}(Sel(O,1),Aux),\ldots,\mathscr{B}(Sel(O,2^{|O|}),Aux)]^\tau \\
N &= diag(\mathscr{P}(Aux)) \\
\Omega_i &= diag(\mathscr{B}(Sel(I,i),Aux)).
\end{aligned}$$

Then

$$B_i = \gamma_i \Delta_i S B N \Omega_i.$$

Proof. The origin of γ_i is straightforward and the Δ_i and S matrices are a direct consequence of the $\mathscr{P}$ and $\mathscr{B}$ function representations of the scalars $c^1_{p,i}$ and c^2_p given above. The B matrix contains the first factor in the brackets in equation (11) in each row, which is constant in i, but variable in row p.

Matrix N is the second factor in brackets in equation (11) spread across the diagonal of a diagonal matrix. That factor was constant in p and i, and by presenting it as a diagonal matrix that multiplies B from the right, we achieve the point-wise multiplication of each row of B with that factor. Analogous reasoning can be used for the matrix Ω_i, except that it is variable in i. □

Combining the results of Theorems 3.1 and 5.3, we get the following theorem.

Theorem 5.4. Using the notation of Theorems 3.1 and 5.3, the matrix M can be represented with respect to columns as

$$M\mathrm{e}_i = \varepsilon_i \gamma_i \Delta_i S B N \Omega_i \overrightarrow{\varphi}.$$

While Theorem 5.3 completely decomposes the sign pattern matrices, for an actual calculation of this decomposition, it is useful to have the explicit forms of the functions $\mathscr{B}$ and $\mathscr{P}$. In the next section we will present different representations of these functions,

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

and present some of their properties that could be helpful in the application of the decomposition given by Theorem 5.4.

6. Explicit representations of $\mathscr{P}$ and $\mathscr{B}$

The function $\mathscr{B}$ has an elegant representation in terms of the adjacency matrix of the bipartite graph. If $\mathscr{A}$ is the adjacency matrix of a bipartite graph, with partitions V and W, and all the labels of V precede the labels of W, then it has the block form

$$\mathscr{A} = \begin{bmatrix} 0 & C \\ C^{\tau} & 0 \end{bmatrix}.$$

We now define the function $\hat{\mathscr{B}} : \{0,1\}^n \to \{-1,1\}^{2^n}$ such that

$$\hat{\mathscr{B}}(b_1, \ldots, b_n) = \bigotimes_{i=1}^{n} \begin{bmatrix} 1 \\ SP(b_i) \end{bmatrix}. \tag{12}$$

It can be shown that if v is the modulo 2 sum of the columns of C, then

$$\mathscr{B}(V, W) = \hat{\mathscr{B}}(v).$$

That is, the $\mathscr{B}$ function can be calculated directly from the adjacency matrix of the bipartite graph in question using the $\hat{\mathscr{B}}$ function. Moreover, the $\hat{\mathscr{B}}$ function is a monomorphism from the group $(\{0,1\}^n, \oplus)$ to the group $(\{-1,1\}^{2^n}, \odot)$, where $\oplus$ and $\odot$ represent modulo 2 addition and point-wise multiplication, respectively.

The $\hat{\mathscr{B}}$ representation of $\mathscr{B}$ and the monomorphism property are important, as described below. Given the adjacency matrix of the underlying graph, we can efficiently compute a polynomial number of entries of the matrix-vector multiplication $B\overrightarrow{\varphi}$, even though the mere length of a row of B is exponential in the number of auxiliary qubits. It will be enough for us to use the representation $\hat{\mathscr{B}}$ given on the right-hand side of equation (12), and $\overrightarrow{\varphi}$ represented in the Kroenecker product form, together with the following property of the scalar product on tensor spaces:

$$\left(\bigotimes_{i=1}^{n} X_i, \bigotimes_{i=1}^{n} Y_i \right) = \prod_{i=1}^{n} (X_i, Y_i), \tag{13}$$

where X_i and Y_i are of equal dimensions.

The monomorphism property also helps in the scenario where we want to calculate $B\Omega_i\overrightarrow{\varphi}$. Since both the rows of B and the diagonal of Ω_i are represented by the $\mathscr{B}$ functions, and hence by the $\hat{\mathscr{B}}$ functions, because of the monomorphism property, the point-wise product of a row in B and the diagonal of Ω_i is again representable in the $\hat{\mathscr{B}}$ form, which is easily read out of the adjacency matrix, so this becomes efficiently solvable as well.

The efficient calculation of a single entry of the matrix M is just the efficient evaluation of one entry of $\varepsilon_i\gamma_i\Delta_i SBN\Omega_i\overrightarrow{\varphi}$, where i designates the column of the entry we wish to evaluate. Note that γ_i is an easily calculated scalar, and Δ_i and S are diagonal matrices whose single entry is easily calculated as well. Hence, the ability to calculate $B\Omega_i\overrightarrow{\varphi}$ efficiently brings us close to the immediate goal. However, there remains the problem of

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

the matrix N, which is represented by the $\mathscr{P}$ function, since the hard part of the calculation we wish to perform is calculating an entry of

$$BN\Omega_i\overrightarrow{\varphi}. \tag{14}$$

The $\mathscr{P}$ function gives the sign parities of the number of edges of all subgraphs of a given graph as a binary vector. One way to represent it explicitly is by taking the positive part of the directed adjacency matrix of the given graph Γ. That is, we direct the graph in an arbitrary fashion, and in its directed adjacency matrix (with elements having values 1 and -1 depending on the direction of the directed edges of the now directed graph) replace all -1's with zeroes. If $\mathscr{A}$ is that matrix, it is easy to see that

$$(\mathscr{P}(\Gamma))_i = SP\left((\mathscr{A}\,[i]_2\,,[i]_2)\right),$$

where $[i]_2$ is the binary representation of $i-1$ given as a vector.

If n is the number of vertices, this representation takes n^2 binary digits on input since they make up the $\mathscr{A}$ matrix.

An alternative representation uses $\binom{n}{2}$ binary digits in the form of an *edge binary list*, which we will now define. Let E be an ordered set of pairs of vertices of Γ such that the label of the first vertex in a pair is strictly smaller than the label of the second – all in all, there are $\binom{n}{2}$ of them. Also, let E be ordered lexicographically according to edges:

$$E = \{(v_i, v_j) | v_k \in V \;\&\; i < j\}.$$

Then, for a given graph Γ, $V = V(\Gamma)$ we use $\mathscr{E}$ to denote the binary vector of length $\binom{n}{2}$ such that the ith entry of $\mathscr{E}$ is 1 if the ith pair of vertices of E is adjacent in Γ and 0 otherwise. We call this vector the *edge binary list*. It is easy to see that the edge binary list uniquely characterises a simple graph. If Γ is a graph and $\mathscr{E} = (b_1, \ldots, b_{\binom{n}{2}})$ is its *edge binary list*, then $\mathscr{P}(\Gamma)$ can be given explicitly as

$$(\mathscr{P}(\Gamma))_i = \left(\mathscr{P}\left(b_1, \ldots, b_{\binom{n}{2}}\right)\right)_i = \prod_{k=1}^{\binom{n}{2}} (-1)^{b_k X_1(i,k) X_2(i,k)},$$

where

$$X_1(i,k) = \left\lfloor \frac{i}{2^{f(k)}} \right\rfloor \mod 2,$$

and

$$X_2(i,k) = \left\lfloor \frac{i}{2^{\left(k - \binom{f(k)}{2} - 1\right)}} \right\rfloor \mod 2,$$

with

$$f(k) = \left\lfloor \frac{\sqrt{8k-7}+1}{2} \right\rfloor.$$

The unappealing functions X_1 and X_2 can be explained more simply. Let (v_p, v_q) be the kth entry of the set E. Then $X_1(i,k)$ is the qth binary digit of the binary representing $i-1$, counting from the least significant digit. With the same notation, $X_2(i,k)$ is the pth binary digit of the binary representing $i-1$, counting from the least significant digit. This representation, even though it seems to be the least elegant, has one significant property.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

For $\mathscr{P}$ defined on edge binary lists,

$$\mathscr{P} : \{0,1\}^{\binom{n}{2}} \to \{-1,1\}^{2^n}$$

is a monomorphism from the group $(\{0,1\}^{\binom{n}{2}}, \oplus)$ to the group $(\{-1,1\}^{2^n}, \odot)$ where $\oplus$ denotes point-wise modulo 2 addition, and $\odot$ denotes point-wise multiplication.

However, knowing how to use this, or any other representation of the $\mathscr{P}$ function, to help evaluate or express $N\overrightarrow{\varphi}$ efficiently, or $BN\Omega_i\overrightarrow{\varphi}$ in conjunction with the $\hat{\mathscr{B}}$ representation of $\mathscr{B}$, remains an open question.

Example 2. In the previous example we calculated the fourth column of the matrix implemented by a (2×3) cluster computation, using the results of Theorem 3.1. Here we will recalculate the matrix B_4 from the first example by first calculating the component scalars and matrices of the decomposition of the sign pattern matrices given by Theorem 5.3.

By this Theorem, B_4 is given by the expression

$$B_4 = \gamma_4 \Delta_4 S B N \Omega_4.$$

To start off, γ_4 is given by the fourth entry of the vector $\mathscr{P}(\Gamma_I)$. In our running example, the graph Γ_I, that is, the subgraph of the input vertices of the (2×3) cluster state is the connected two vertex graph with the labels 1 and 2. From the definition of the $\mathscr{P}$ function, this is the sign parity of the number of edges of the subgraph of Γ_I induced by (in the case $i = 4$) both vertices, that is, the total graph. Since this number of edges is unity, we have $\gamma_4 = -1$.

Next, we compute the matrix Δ_4, which by Theorem 5.3 is given by

$$\Delta_4 = diag(\mathscr{B}(Sel(I,4), O)).$$

Note that, in our example, the input and output vertices are not connected, and since the components of the vector $\mathscr{B}(Sel(I,4), O)$ track the sign parity of the number of edges connecting the set of input vertices and different subsets of the output vertices, these sign parities are all unity. Hence, it is easy to see that Δ_4 is the identity matrix of dimension 4. Hence,

$$\Delta_4 = diag(1,1,1,1).$$

To compute the matrix S, we first recall its definition:

$$S = diag(\mathscr{P}(O)).$$

Now, the ith entry of $\mathscr{P}(O)$ is the sign party of the number of edges in the subgraph $\Gamma_{Sel(O,i)}$, so it is easy to check that the complete vector is $[1,1,1,-1]^{\tau}$, so

$$S = diag(1,1,1,-1).$$

The next matrix we need to compute is B, which is perhaps the most interesting as it is the only non-diagonal matrix appearing in the decomposition. We compute it row-wise. Recall, by Theorem 5.3, the ith row of the matrix B is given by

$$\mathscr{B}(Sel(O,i), Aux).$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

We will compute the first two rows using the definition of the function $\mathscr{B}$ directly, and, for illustrative purposes, the last two using the characterisation of the function $\mathscr{B}$ given at the beginning of Section 6.

The first row of B is given by $\mathscr{B}(Sel(O,1), Aux)$, and since $Sel(O,1)$ is the empty set, using the same reasoning as for the calculation of Δ_4, the first row of B is given by the vector $[1,1,1,1]$.

The second row is generated by calculating $\mathscr{B}(Sel(O,2), Aux) = \mathscr{B}(\{6\}, Aux)$ (in our cluster graph example). Recall that, by definition, the kth element of the vector $\mathscr{B}(\{6\}, Aux)$ is given by the sign parity of the number of edges connecting the set of vertices $\{6\}$ with the set $Sel(Aux, k)$ (in the graph Γ of the cluster state). By simple counting (going through all the subsets of the set Aux as we go through the entries of the vector), it is easy to see that the second row of B is given by the vector $[1,-1,1,-1]$.

The third and fourth rows of B are given by

$$\mathscr{B}(Sel(O,3), Aux) = \mathscr{B}(\{5\}, \{3,4\})$$

and

$$\mathscr{B}(Sel(O,4), Aux) = \mathscr{B}(\{5,6\}, \{3,4\}),$$

respectively.

To calculate these vectors using the characterisation given in Section 6, we need to generate the corresponding adjacency submatrices, as described in that section (that is, the matrices C). It is easy to verify that the adjacency submatrix C_3 corresponding to the third row of B is a column matrix given by

$$C_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

whereas the adjacency submatrix C_4 corresponding to the the fourth row of B is a square matrix given by

$$C_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

The modulo 2 sum of columns of C_3 is C_3 itself, and by the characterisation we are illustrating, the third row of B is given by

$$[1,-1] \otimes [1,1] = [1\,1,-1,-1].$$

For the final row of B, note that the modulo 2 sum of columns of C_4 is $[1,1]^{\tau}$, hence the fourth row of B is given by

$$[1,-1] \otimes [1,-1] = [1\ -1,-1,1].$$

Putting this all together, the B matrix is then given by

$$B = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

The matrix N contains the sign parities of the numbers of edges of subgraphs of the graph Γ_{Aux}, hence it is, in our example, equal to the matrix S, that is,

$$N = diag(1, 1, 1, -1).$$

Finally, we compute the matrix Ω_4, which is given by

$$\Omega_4 = diag(\mathscr{B}(Sel(I, 4)), O).$$

Note that this vector tracks the sign parity of the numbers of connecting edges between vertices in $Sel(I, 4) = I$ and subsets of the output vertices. In our case, since inputs and outputs are not connected,

$$\Omega_4 = diag(1, 1, 1, 1).$$

Finally, we can write out the decomposition of the matrix B_4 explicitly as

$$\begin{aligned} B_4 &= \gamma_4 \Delta_4 S B N \Omega_4 \\ &= -\mathbf{I} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \mathbf{I} \\ &= \begin{bmatrix} -1 & 1 & 1 & 1 \\ -1 & -1 & 1 & -1 \\ -1 & 1 & -1 & -1 \\ 1 & 1 & 1 & -1 \end{bmatrix}. \end{aligned}$$

7. The role of entanglement

In the previous section we briefly addressed the efficiency of calculating a single entry of the linear map M representing the positive branch of the computation realised by a given one-way computer. Here we explain how the underlying structure of the entanglement affects the efficiency of such computations, which is closely linked to the problems of efficient classical simulations of quantum computations and quantum systems in general. To show this, we analyse the computation of a single entry of M thoroughly. Let $m_{i,j}$ be the (i, j)th entry of the matrix M. Then, from Theorem 5.4, we have

$$m_{i,j} = \mathrm{e}_i^{\tau} M \mathrm{e}_j = \mathrm{e}_i^{\tau} \varepsilon_j \gamma_j \Delta_j S B N \Omega_j \overrightarrow{\varphi}.$$

Note that ε_j and γ_j are scalars, so the expression above can be rewritten as

$$m_{i,j} = \varepsilon_j \gamma_j \mathrm{e}_i^{\tau} \Delta_j S B N \Omega_j \overrightarrow{\varphi},$$

which can be further simplified since Δ_j and S are diagonal matrices. Using δ_{j_i} and s_i to denote the ith diagonal elements of Δ_j and S, respectively, we then have

$$m_{i,j} = \varepsilon_j \gamma_j \delta_{j_i} s_i \mathrm{e}_i^{\tau} B N \Omega_j \overrightarrow{\varphi}.$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

The vector-matrix product $e_i^\tau B$ constitutes the ith row of B, and using r_i to denote this row, we get

$$m_{i,j} = \varepsilon_j \gamma_j \delta_{j_i} s_i \left(r_i, N\Omega_j \overrightarrow{\varphi} \right),$$

where $(\cdot,\cdot)$ is the standard inner product. For a fixed entry (i,j) of M, all the scalars $\varepsilon_j, \gamma_j, \delta_{j_i}, s_i$ are efficiently computable, so for the efficient evaluation of the entry $m_{i,j}$, the computational bottleneck is the evaluation of the inner product

$$(r_i, N\Omega_j \overrightarrow{\varphi}).$$

The vectors r_i and $N\Omega_j\overrightarrow{\varphi}$ are of length 2^a, where a is the number of pure auxiliary qubits, so direct calculation would be inefficient. However, recall that both r_i and $\overrightarrow{\varphi}$ have compact forms

$$r_i = \bigotimes_{k=1}^{a} \begin{bmatrix} 1 \\ SP(b_{k_i}) \end{bmatrix}$$

$$\overrightarrow{\varphi} = \bigotimes_{k=n+1}^{a+n} \begin{bmatrix} 1 \\ e^{i\alpha_k} \end{bmatrix}$$

where the angles α_k are the measurement angles of the pure auxiliary qubits, and parameters b_{k_i} can be read from the adjacency matrix of the graph, as explained in the previous section and briefly summarised below.

To obtain the vector of parameters $\left[b_{1_i}, \ldots, b_{a_i}\right]^\tau$ for the ith row of the matrix B, which we denoted r_i, we let $\mathscr{A}$ be the adjacency matrix of the underlying graph and $\mathscr{A}'$ be a submatrix of $\mathscr{A}$ obtained by dropping all the rows of $\mathscr{A}$ except for the rows corresponding to the pure auxiliary vertices, and dropping all the columns except the columns that correspond to the vertices that are in the set $Sel(O,i)$, which is the subset of output vertices selected by i. The vector $\left[b_{1_i}, \ldots, b_{a_i}\right]^\tau$ is the modulo 2 sum of the columns in the submatrix $\mathscr{A}'$.

Furthermore, the matrix Ω_j is a diagonal matrix, and by applying the reasoning of the previous section, it is easy to see that its diagonal can also be written in a compact tensorial form

$$\Omega_j = diag\left(\bigotimes_{k=1}^{a} \begin{bmatrix} 1 \\ SP(o_{k_j})) \end{bmatrix} \right)$$

where the parameters o_{k_j} are also read from the adjacency matrix of the graph. Concretely, the vector of parameters $\left[o_{1_i}, \ldots, o_{a_i}\right]^\tau$ is the modulo 2 sum of the columns of the matrix $\mathscr{A}''$, which is the submatrix of $\mathscr{A}$ obtained by dropping all the rows of $\mathscr{A}$ except for the rows corresponding to the pure auxiliary vertices and dropping all the columns except the columns corresponding to the vertices that are in the set $Sel(I,j)$, that is, the subset of input vertices selected by j. Continuing the analysis, the matrix Ω_j can be rewritten as

$$\Omega_j = \bigotimes_{k=1}^{a} \begin{bmatrix} 1 & 0 \\ 0 & SP(o_{k_j}) \end{bmatrix}.$$

Therefore, the vector $\overrightarrow{\varphi}'_j = \Omega_j \overrightarrow{\varphi}$ also has a compact form

$$\overrightarrow{\varphi}'_j = \bigotimes_{k=1}^{a} \begin{bmatrix} 1 \\ SP(o_{k_j}) e^{i\alpha_{(k+n)}} \end{bmatrix}$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

by the mixed-product property of the Kroenecker product. Hence, the inner product $\left(r_i, N\Omega_j \overrightarrow{\varphi}\right)$ can be rewritten as

$$\left(r_i, N\overrightarrow{\varphi}'_j\right)$$

where both r_i and $\overrightarrow{\varphi}'_j$ have a compact representation.

The above analysis highlights the role of matrix N, which depends on the graph of the underlying entangled state, in the efficiency of computing m_{ij}. Now we can illustrate a couple of classes of entanglement graphs for which the calculation of a single entry of the matrix M is a feasible task.

First is the case of graphs for which the induced subgraph of the pure auxiliary qubits is empty. If the pure auxiliary qubits are unconnected (unentangled), the matrix N in the decomposition given by Theorem 5.3 is the identity matrix. This is easy to see as the diagonal entries of N, in this unconnected scenario, are the sign parities of the number of edges of unconnected subgraphs, hence are all zero. For these graphs, the inner product, which forms the bottleneck of the computation of a single entry of M, is simply

$$\left(r_i, \overrightarrow{\varphi}'_j\right).$$

Since both of the vectors in this inner product have a compact representation, the above inner product can be written as

$$\left(\bigotimes_{k=1}^{a} \begin{bmatrix} 1 \\ SP(b_{k_i})) \end{bmatrix}, \bigotimes_{k=1}^{a} \begin{bmatrix} 1 \\ SP(o_{k_j}) e^{i\alpha_{(k+n)}} \end{bmatrix}\right),$$

which is equal to

$$\prod_{k=1}^{a} \left(1 + SP(b_{k_i}) SP(o_{k_j}) e^{i\alpha_{(k+n)}}\right)$$

by the property given in equation (13).

The techniques used in the previous example can be generalised to more complicated families of graphs as follows. Let the pure auxiliary vertices be denoted by $v_{n+1}, \ldots, v_{n+a.}$ For simplicity, we will assume that a is even. Let the graph underlying the one-way computation be such that the pairs of vertices

$$\left\{(v_{n+1+2k}, v_{n+2+2k})\right\}_{k=0}^{a/2}$$

are connected (that is, the pure auxiliary qubits are pairwise entangled, respecting the order of the labels). Then it can be shown that for this graph, the corresponding matrix N is of the form

$$N = \bigotimes_{k=1}^{(a/2)} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}.$$

Then, again using the property given by equation (13) and the mixed-product property of the Kroenecker product, the calculation demanding inner product $\left(r_i, N\overrightarrow{\varphi}'_j\right)$ can be rewritten as the inner product of vectors

$$\bigotimes_{k=1}^{(a/2)} \left(\begin{bmatrix} 1 \\ SP(b_{2k_i-1}) \end{bmatrix} \otimes \begin{bmatrix} 1 \\ SP(b_{2k_i}) \end{bmatrix}\right)$$

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

and

$$\bigotimes_{k=1}^{(a/2)} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix} \left(\begin{bmatrix} 1 \\ SP(o_{2k_j-1})e^{i\alpha_{(2k+n-1)}} \end{bmatrix} \otimes \begin{bmatrix} 1 \\ SP(o_{2k_j})e^{i\alpha_{(2k+n)}} \end{bmatrix} \right),$$

which equals

$$\prod_{k=1}^{(a/2)} \{1 + SP(b_{2k_i})SP(o_{2k_j})e^{i\alpha_{(2k+n)}} + SP(b_{2k_i-1})SP(o_{2k_j-1})e^{i\alpha_{(2k+n-1)}} - SP(b_{2k_i})SP(o_{2k_j})SP(b_{2k_i-1})SP(o_{2k_j-1})e^{i(\alpha_{(2k+n-1)}+\alpha_{(2k+n)})}\},$$

which can again be calculated efficiently.

Note that if the pure auxiliary qubits are arbitrarily connected, no general direct method of efficient calculation of entries of M is known, so the structure of entanglement of the pure auxiliary qubits (captured above with matrix N) proves to be the bottleneck of the computation of a single entry of M. We plan to investigate further the structure of N to fully characterise the class of one-way patterns that have efficient classical simulations.

8. Discussion

We have presented a complete structural characterisation of the positive branch of a one-way pattern in terms of its matrix representation in the computational basis and shown that this structure is intricate and complex, but admitting a high degree of regularity.

While it remains unclear how we can use this regularity directly to tackle problems such as the direct simulation of unitaries in a one-way model or the full characterisation of pointless measurements, the proposed structure clearly emphasises the importance of the entanglement, as discussed in the previous section. We believe that the methods employed for the efficient calculation of single elements of the matrix M in the previous section can be generalised to a wider class of graphs, and potentially to all graphs whose pure auxiliary vertex induced subgraph has its largest connectivity component at most logarithmic in size in terms of the number of vertices of the graph. Such generalisations and the interplay between simulatablity of quantum computation and efficient calculation of single matrix elements of M remain the targets of ongoing research.

It may also be possible to adapt similar techniques to those used for the efficient evaluation of a single element of M to address the dual problem: given a matrix M and a fixed graph, find the angles with which the matrix M is realised on a one-way computer, which corresponds to the simulation of a given unitary.

References

de Beaudrap, N., Danos, V. and Kashefi, E. (2006) Phase map decomposition for unitaries. Available at quant-ph/0603266.

de Beaudrap, N., Danos, V., Kashefi, E. and Roetteler, M. (2008) Quadratic form expansions for unitaries. In: Kawano, Y. and Mosca, M. (eds.) Theory of Quantum Computation,

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press

Communication, and Cryptography Third Workshop, TQC 2008. *Springer-Verlag Lecture Notes in Computer Science* **5106** 29–46.

Browne, D., Kashefi, E., Mhalla, M. and Perdrix, S. (2007) Generalized flow and determinism in measurement-based quantum computation. *New Journal of Physics* **9**.

Danos, V. and Kashefi, E. (2006) Determinism in the one-way model. *Physical Review A* **74** (5) 052310.

Danos, V., Kashefi, E. and Panangaden, P. (2007) The measurement calculus. *Journal of ACM* **54** (2) 8.

Danos, V., Kashefi, E., Panangaden, P. and Perdrix, S. (2009) Extended measurement calculus. In: Gay, S. and Mackie, I. (eds.) *Semantic Techniques in Quantum Computation*, Cambridge University Press.

Kashefi, E., Markham, D., Mhalla, M. and Perdrix, S. (2009) Information flow in secret sharing protocols. *Electronic Proceedings in Theoretical Computer Science* **9** 87–97.

Raussendorf, R. and Briegel, H. J. (2001) A one-way quantum computer. *Physical Review Letters* **86** (5188).

Raussendorf, R., Browne, D. E. and Briegel, H. J. (2003) Measurement-based quantum computation with cluster states. *Physical Review A* **68** 022312.

https://doi.org/10.1017/S096012951200014X Published online by Cambridge University Press