PROCEEDINGS A

royalsocietypublishing.org/journal/rspa

Research

Cite this article: Bradshaw ZP, LaBorde ML, Wilde MM. 2023 Cycle index polynomials and generalized quantum separability tests. *Proc. R. Soc. A* **479**: 20220733. https://doi.org/10.1098/rspa.2022.0733

Received: 3 November 2022
Accepted: 10 May 2023

Subject Areas:
mathematical physics, quantum physics

Keywords:
cycle index polynomial, quantum entanglement, separability, quantum algorithms, quantum computing, symmetry

Author for correspondence:
Zachary P. Bradshaw
e-mail: zbradshaw@tulane.edu

Electronic supplementary material is available online at https://doi.org/10.6084/m9.figshare.c.6662923.

THE ROYAL SOCIETY PUBLISHING

Cycle index polynomials and generalized quantum separability tests

Zachary P. Bradshaw[1], Margarite L. LaBorde[2] and Mark M. Wilde[2,3]

[1]Department of Mathematics, Tulane University, New Orleans, LA, USA
[2]Hearne Institute for Theoretical Physics and Department of Physics and Astronomy, Louisiana State University, Baton Rouge, LA, USA
[3]School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA

ZPB, 0000-0002-3591-7594; MLL, 0000-0001-9754-1468; MMW, 0000-0002-3916-4462

The mixedness of one share of a pure bipartite state determines whether the overall state is a separable, unentangled one. Here we consider quantum computational tests of mixedness, and we derive an exact expression of the acceptance probability of such tests as the number of copies of the state becomes larger. We prove that the analytical form of this expression is given by the cycle index polynomial of the symmetric group S_k, which is itself related to the Bell polynomials. After doing so, we derive a family of quantum separability tests, each of which is generated by a finite group; for all such algorithms, we show that the acceptance probability is determined by the cycle index polynomial of the group. Finally, we produce and analyse explicit circuit constructions for these tests, showing that the tests corresponding to the symmetric and cyclic groups can be executed with $O(k^2)$ and $O(k\log(k))$ controlled-SWAP gates, respectively, where k is the number of copies of the state being tested.

1. Introduction

Entanglement is an inherently quantum mechanical phenomenon and thus is of great interest in quantum information science. Separability tests acting on pure states equivalently test for entanglement, thus creating an avenue to identify potentially resourceful states

and verify entanglement generation protocols. Much work has been done to understand entanglement, including the positive partial transpose (PPT) criterion [1,2] and the related entanglement measure known as logarithmic negativity [3–5]. Another well-known approach considers k-extendibility of a state [6–8], with related quantifiers proposed in [9–11]. Experimental entanglement tests have also been implemented on trapped ion [12] and optical systems [13,14]. These notions exemplify how entanglement is characterized and quantified in the current literature.

The most common quantum computational test of separability of pure states is the swap test, introduced in [15], applied in [16] for entanglement detection, and used in quantum fingerprinting [17]. To understand it, first recall that a pure bipartite state $|\psi\rangle_{AB}$ is separable if it can be written as a tensor product of two states, as

$$|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\varphi\rangle_B \,. \tag{1.1}$$

Now, if we take two copies of this separable state, it has the following form:

$$|\psi\rangle_{A_1B_1} \otimes |\psi\rangle_{A_2B_2} = |\phi\rangle_{A_1} \otimes |\varphi\rangle_{B_1} \otimes |\phi\rangle_{A_2} \otimes |\varphi\rangle_{B_2} \,. \tag{1.2}$$

This state is invariant under a swap of systems A_1 and A_2, as well as a swap of systems B_1 and B_2. Thus, the swap test accepts with certainty in this case; however, if a pure bipartite state is not separable, the two-copy state does not possess the above swap invariance, and the swap test can detect this lack of invariance by means of the phase kickback trick, well known in quantum computation. Here we make use of the swap test, as well as generalizations of it, to study the symmetry of quantum states with respect to an arbitrary finite group [18,19].

The recent work in [19] proposed a generalization of the swap test as a method for detecting entanglement, based on the observation that multiple copies of the separable state in (1.1) are invariant under arbitrary permutations of both the A systems and B systems. Indeed, by writing such a state down explicitly as

$$\bigotimes_{i=1}^{k} |\psi\rangle_{A_iB_i} = \bigotimes_{i=1}^{k} |\phi\rangle_{A_i} \otimes |\varphi\rangle_{B_i} \,, \tag{1.3}$$

it is clear that such a state is invariant as mentioned above. However, if the state $|\psi\rangle_{AB}$ is not separable, then checking for various kinds of permutation invariance of the state $\bigotimes_{i=1}^{k} |\psi\rangle_{A_iB_i}$ leads to more fine-grained tests of entanglement with alternative mathematical expressions for the acceptance probability of the test. Other works, such as [20,21], express interest in systems of a similar nature. Understanding these expressions in more detail is one of the main goals of the present paper. Our intention is to generalize the family of separability tests in [19] further by establishing a one-to-one correspondence between finite groups and a subset of quantum separability tests and to compare the cost of implementing these new tests with that of the swap and symmetric groups tests.

Before we discuss our results, let us indicate how our approach falls within the framework of G-Bose symmetry tests, as developed in [19]. We use this method to discuss the separability of a pure bipartite state [7], although group symmetry tests have been used to great effect to test other properties as well (e.g. [22]). These symmetry tests involve a group G and determine if a given state is invariant under the action of a unitary representation of that group. Bose symmetry tests specifically describe situations where every element of a group is considered simultaneously rather than consecutively. The quantum algorithm is implemented by physically realizing the projector onto the symmetric subspace, and the symmetry of the state is manifested through the acceptance probability of the algorithm (see [23] for a thorough discussion of the symmetric subspace in regards to quantum information). In the G-Bose symmetry framework, a pure state $|\psi\rangle_{AB}$ is given and an S_k-Bose symmetry test is conducted on the tensor-power state $|\psi\rangle_{AB}^{\otimes k}$, where S_k denotes the symmetric group on k letters. Possessing this tensor-power state is equivalent to having access to k copies of our state being tested. The swap test is recovered as a special case in which $k = 2$.

The G-Bose symmetry tests allow for a generalization of the swap test to more copies of a state of interest and higher-order groups. These algorithms exchange simplicity for certainty, analogous to how fingerprinting is both more accurate and complicated when greater numbers of prints are taken. In choosing to investigate group symmetries, rather than merely the swap test, the separability of a state can be determined more quickly and accurately. In what follows, we will explicitly consider the trade-off between the swap test, which may pass some states with high probability even if entangled, and higher-order permutation tests, which will reject entangled states with higher probability than a swap test in exchange for the use of more resources.

The natural question is, when do these more complex tests merit performing? In this work, we derive the acceptance probability of a generalized separability test. In doing so, we present an inherent reliance on the cycle index polynomial, a particularly important polynomial in Pólya theory [24,25] that encodes the structure of a permutation group by storing the number of elements of a given cycle type as its coefficients. This allows us to compare separability tests generated from various groups, as well as investigate the mathematical relationships inherently present in these tests. We directly show that an arbitrary finite group generates a separability test with its acceptance probability given by the cycle index polynomial of that group. We supplement this by then giving explicit quantum circuit descriptions for groups of interest and counting the number of gates needed to realize each test. Combining our acceptance probability results with resource counting gives us a metric to compare when the relative strictness of the test is outweighed by the benefit of fewer gate resources, and we discuss this factor in more detail in §5.

The rest of our paper is organized as follows. In §2, we review the algorithm for the bipartite pure-state separability test in [19], and we prove that the acceptance probability of this algorithm is given by the cycle index polynomial [24,25] of the symmetric group S_k, which is itself related to the complete Bell polynomials [26]. This enables us to write the acceptance probability as both the permanent and the determinant of particular matrices, as a consequence of Newton's identities [27]. In [19], it was conjectured that this acceptance probability does not increase as $k \to \infty$. In §3, we prove that this conjecture is true. In fact, we show that it strictly decreases and converges to zero whenever $\rho_B := \mathrm{Tr}_A[|\psi\rangle\langle\psi|_{AB}]$ is not a pure state.

In §4, we generalize the bipartite pure-state separability test to an algorithm involving any group G, in which a G-Bose symmetry test is performed on the tensor-power state $|\psi\rangle_{AB}^{\otimes k}$. By identifying G with a subgroup of S_k, which is guaranteed to exist by Cayley's theorem, we show, by the same reasoning as in §2, that the acceptance probability of the algorithm is given by the cycle index polynomial of the group G. We discuss how these generalized tests are in fact separability tests for pure, bipartite states, and they have an interesting connection to combinatorics via the cycle index polynomial. Finally, in §5, we analyse the resources needed to implement these tests on quantum computers; in doing so, we show that simpler groups can give comparable performance for fewer resources. Finally, we conclude in §6 with a summary, followed by open questions for future work.

2. Bipartite pure-state separability test

Let us begin by reviewing the construction of the bipartite pure-state separability test in [19]. As discussed therein, it can also be viewed as a G-Bose symmetry test. We now recall the definition of a G-Bose symmetric state.

Definition 2.1. Let G be a finite group with a unitary representation $U_S : G \to U(\mathcal{H})$, where $U(\mathcal{H})$ denotes the set of all unitaries that act on a Hilbert space $\mathcal{H}$. Then a state ρ_S is called G-Bose symmetric if

$$\Pi_S^G \rho_S \Pi_S^G = \rho_S, \tag{2.1}$$

where $\Pi_S^G := (1/|G|) \sum_{g \in G} U_S(g)$ is the group representation projection and $|G|$ is the order of the group (the number of elements in the underlying set).

In Dirac's notation, a pure state $|\psi\rangle_S$ is G-Bose symmetric if $\Pi_S^G |\psi\rangle_S = |\psi\rangle_S$. This property is equivalent to $||\Pi_S^G |\psi_S\rangle||_2 = 1$, where $||\cdot||_2$ denotes the standard Euclidean 2-norm. For a general

mixed state, the condition $\mathrm{Tr}[\Pi_S^G \rho_S] = 1$ is equivalent to the definition in (2.1), as argued in [19]. Thus, the algorithm from [19] tests for G-Bose symmetry by checking the latter condition (see also [28, ch. 8]). Indeed, we append an ancillary register C in the state $|0\rangle_C$ to the state $|\psi\rangle_S$ (where the symbol '0' is identified with the identity element of the group G) and act on C with the quantum Fourier transform, leaving a composite system in the state

$$|+\rangle_C \otimes |\psi\rangle_S\,, \tag{2.2}$$

where $|+\rangle_C$ is defined to be the uniform superposition over all labels of the group elements:

$$|+\rangle_C := \frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle_C\,. \tag{2.3}$$

Next we act on the state in (2.2) with the controlled unitary $\sum_{g\in G} |g\rangle\langle g|_C \otimes U_S(g)$, producing the state

$$\frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle_C \otimes (U_S(g)\,|\psi\rangle_S). \tag{2.4}$$

We now project register C onto the state $|+\rangle_C$, which can be accomplished probabilistically by applying the inverse quantum Fourier transform to C, measuring C in the $\{|g\rangle\langle g|_C\}_{g\in G}$ basis, and declaring 'accept' if $|0\rangle\langle 0|_C$ occurs and 'reject' otherwise. Then the acceptance probability p is given by

$$p = \left\| (\langle +|_C \otimes I_S)\left(\frac{1}{\sqrt{|G|}} \sum_{g\in G} |g\rangle_C \otimes (U_S(g)\,|\psi\rangle_S) \right) \right\|_2^2 \tag{2.5}$$

$$= \left\| \frac{1}{|G|} \sum_{g\in G} U_S(g)\,|\psi\rangle_S \right\|_2^2 \tag{2.6}$$

$$= ||\Pi_S^G\,|\psi\rangle_S\,||_2^2 \tag{2.7}$$

$$= \mathrm{Tr}[\Pi_S^G\,|\psi\rangle\langle\psi|_S]. \tag{2.8}$$

By convexity, this result is easily generalized to the case in which the state of system S is described by a density operator ρ_S, so that the acceptance probability is equal to $\mathrm{Tr}[\Pi_S^G \rho_S]$.

Now, to test for the separability of a bipartite pure state ψ_{AB}, we suppose that k copies of the state ψ_{AB} are available, which we write as $\psi_{AB}^{\otimes k}$. We also identify the A systems by $A_1 \cdots A_k$ and the B systems by $B_1 \cdots B_k$. We then perform an S_k-Bose symmetry test on the state $\psi_{AB}^{\otimes k}$ by identifying S with $A_1B_1 \cdots A_kB_k$ and $U_S(\pi)$ with $I_{A_1\cdots A_k} \otimes W_{B_1\cdots B_k}(\pi)$, where $\pi \in S_k$ and $W_{B_1\cdots B_k}$: $S_k \to U(\mathcal{H}_{B_1\cdots B_k})$ is the standard unitary representation of S_k that acts on $\mathcal{H}_{B_1\cdots B_k} \equiv \mathcal{H}_{B_1} \otimes \cdots \otimes \mathcal{H}_{B_k}$ by permuting the Hilbert spaces according to the corresponding permutation. Define $\rho_B := \mathrm{Tr}_A[\psi_{AB}]$. By applying (2.1), the acceptance probability for the bipartite pure-state separability algorithm is given by

$$p^{(k)} := \mathrm{Tr}[\Pi_{B_1\cdots B_k} \rho_B^{\otimes k}] \tag{2.9}$$

where

$$\Pi_{B_1\cdots B_k} := \frac{1}{k!} \sum_{\pi\in S_k} W_{B_1\cdots B_k}(\pi). \tag{2.10}$$

Figure 1 reviews the G-Bose symmetry test. The circuit begins with k copies of an initial bipartite state $|\psi\rangle_{AB}$. The B_i subsystems are collected and subject to a controlled unitary gate whose mathematical description involves each unitary $U(g)$. The control register is initialized to the state $|+\rangle_C$, as defined in (2.3). Under the separability test circuit, $G = S_k$, and the unitary representation is a permutation of the B_i subsystems.

Our first main result is a formula for the acceptance probability $p^{(k)}$ in (2.9) as a sum over the partitions of k of a product of traces of ρ_B and its powers and certain scaling factors. In particular, the formula is identical to that of the cycle index polynomial of the symmetric group S_k, with each variable x_j taking the value $\mathrm{Tr}[\rho^j]$ (see the discussion after the following proposition for

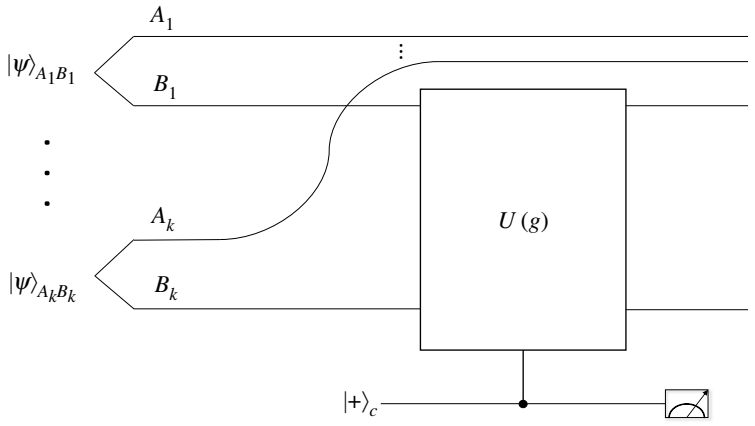

Figure 1. Quantum circuit to implement a G-Bose symmetry test. We take k copies of an initial bipartite state $|\psi\rangle_{AB}$ and consider the reduced state $\rho_B = \mathrm{Tr}_A[|\psi\rangle\langle\psi|_{AB}]$. The collection of these reduced states is then subjected to the separability test determined by the group, where $|+\rangle_c$ is defined in (2.3) and $U(g)$ is an element of the group representation.

more on this). In what follows, we use the standard cycle notation wherein $(i_1\ i_2\ \cdots\ i_k)$ refers to the permutation $\sigma : \{1,\ldots,k\} \to \{1,\ldots,k\}$ that sends i_1 to i_2, i_2 to i_3, and so on with i_k mapping to i_1. For example, $(1\ 2\ 3)$ refers to the permutation σ defined by $\sigma(1) = 2$, $\sigma(2) = 3$ and $\sigma(3) = 1$. We follow the standard convention for functions so that a product of cycles (a composition of functions) is read from right to left. For example, $(1\ 2\ 3)(2\ 3) = (1\ 2)(3) = (1\ 2)$.

Theorem 2.2. *Let ψ_{AB} denote a pure bipartite state and define $\rho_B := Tr_A[\psi_{AB}]$. Then the acceptance probability $p^{(k)}$ for the bipartite pure-state separability test is given by*

$$p^{(k)} = \sum_{a_1+2a_2+\cdots+ka_k=k} \prod_{j=1}^{k} \frac{(Tr[\rho_B^j])^{a_j}}{j^{a_j} a_j!}, \tag{2.11}$$

where the sum is taken over the partitions of k.

Proof. Let $\pi := (1\ 2\ \cdots\ k)$ and consider the representation $W_{B_1\cdots B_k}(\pi)$. It was shown in [29] that $\mathrm{Tr}[W_{B_1\cdots B_k}(\pi)\rho_B^{\otimes k}] = \mathrm{Tr}[\rho_B^k]$, but we include a proof here for completeness. Expanding ρ in the standard basis as $\rho = \sum_{i,j} p_{i,j}|i\rangle\langle j|$, we have

$$\mathrm{Tr}[W_{B_1\cdots B_k}(\pi)\rho_B^{\otimes k}] = \mathrm{Tr}\left[W_{B_1\cdots B_k}(\pi) \sum_{\substack{i_1,\ldots,i_k \\ j_1,\ldots,j_k}} p_{i_1j_1}\cdots p_{i_kj_k}|i_1\rangle\langle j_1| \otimes |i_2\rangle\langle j_2| \otimes \cdots \otimes |i_k\rangle\langle j_k|\right] \tag{2.12}$$

$$= \mathrm{Tr}\left[\sum_{\substack{i_1,\ldots,i_k \\ j_1,\ldots,j_k}} p_{i_1j_1}\cdots p_{i_kj_k}|i_k\rangle\langle j_1| \otimes |i_1\rangle\langle j_2| \otimes \cdots \otimes |i_{k-1}\rangle\langle j_k|\right] \tag{2.13}$$

$$= \sum_{\substack{i_1,\ldots,i_k \\ j_1,\ldots,j_k \\ t_1,\ldots,t_k}} p_{i_1j_1}\cdots p_{i_kj_k}\delta_{t_1i_k}\delta_{j_1t_1}\cdots\delta_{t_ki_{k-1}}\delta_{j_kt_k} \tag{2.14}$$

$$= \sum_{t_1,\ldots,t_k} p_{t_2t_1}p_{t_3t_2}\cdots p_{t_kt_{k-1}}p_{t_1t_k}. \tag{2.15}$$

Meanwhile,

$$\mathrm{Tr}[\rho^k] = \mathrm{Tr}\left[\sum_{i_1,\ldots,i_k,j_1,\ldots,j_k} p_{i_1 j_1}\cdots p_{i_k j_k}|i_1\rangle\langle j_1|i_2\rangle\langle j_2|\cdots|i_k\rangle\langle j_k|\right] \tag{2.16}$$

$$= \sum_{i_1,i_2,\ldots,i_k} p_{i_1 i_2} p_{i_2 i_3}\cdots p_{i_{k-1} i_k} p_{i_k i_1}. \tag{2.17}$$

Thus, by relabelling the indices, we see that

$$\mathrm{Tr}[W_{B_1\cdots B_k}(\pi)\rho_B^{\otimes k}] = \mathrm{Tr}[\rho^k]. \tag{2.18}$$

Similarly, we can show for every m-cycle $\pi_m \in S_k$ that

$$\mathrm{Tr}[W_{B_1\cdots B_k}(\pi_m)\rho_B^{\otimes k}] = \mathrm{Tr}[\rho^m]. \tag{2.19}$$

Now suppose π_m and π_n are disjoint m and n-cycles, respectively. Then they act on different Hilbert spaces and so the trace of the product of their representations acting on $\rho_B^{\otimes k}$ splits into the product of traces. That is,

$$\mathrm{Tr}[W_{B_1\cdots B_k}(\pi_m)W_{B_1\cdots B_k}(\pi_n)\rho^{\otimes k}] = \mathrm{Tr}[\rho^m]\mathrm{Tr}[\rho^n]. \tag{2.20}$$

Now, since every m-cycle yields a factor of $\mathrm{Tr}[\rho^m]$ and products of disjoint cycles split the trace, we have

$$p^{(k)} = \mathrm{Tr}[\Pi_{B_1\cdots B_k}\rho_B^{\otimes k}] \tag{2.21}$$

$$= \mathrm{Tr}\left[\frac{1}{k!}\sum_{\pi\in S_k} W_{B_1\cdots B_k}(\pi)\rho_B^{\otimes k}\right] \tag{2.22}$$

$$= \frac{1}{k!}\sum_{\pi\in S_k}\mathrm{Tr}[W_{B_1\cdots B_k}(\pi)\rho_B^{\otimes k}] \tag{2.23}$$

$$= \frac{1}{k!}\sum_{a_1+2a_2+\cdots+ka_k=k} c(a_1,\ldots,a_k)\prod_{j=1}^{k}\mathrm{Tr}[\rho_B^j]^{a_j} \tag{2.24}$$

where $c(a_1,\ldots,a_k)$ is the number of cycles in S_k with cycle type $(a_1,\ldots,a_k)$, which is known to be $k!/\prod_{j=1}^k j^{a_j}a_j!$ (see [30, eqn (13.3)]). Thus, the equality in (2.11) follows. ■

The cycle index polynomial of a permutation group G is defined by

$$Z(G)(x_1,\ldots,x_n) := \frac{1}{|G|}\sum_{g\in G} x_1^{c_1(g)}\cdots x_n^{c_n(g)}, \tag{2.25}$$

where $c_j(g)$ denotes the number of cycles of length j in the disjoint cycle decomposition of g. Setting $x_j = \mathrm{Tr}[\rho_B^j]$, we see that the acceptance probability of the separability test is given by the cycle index polynomial of the symmetric group S_k (see [30, ch. 37, p. 526]), so that it satisfies the recurrence relation

$$p^{(k)} = \frac{1}{k}\sum_{j=1}^{k}\mathrm{Tr}[\rho^j]p^{(k-j)}. \tag{2.26}$$

Furthermore, the cycle index polynomial of the symmetric group S_k is equivalent to

$$\frac{1}{k!}B_k(x_1, x_2, 2!x_3, 3!x_4, \ldots, (k-1)!x_k), \tag{2.27}$$

where $B_k(x_1,\ldots,x_k)$ is the complete Bell polynomial [31]. From this perspective, the acceptance probability can be interpreted as the kth raw moment of a probability distribution with the first k cumulants given by $1, \mathrm{Tr}[\rho^2], \ldots, \mathrm{Tr}[\rho^k]$.

Using a variant of Newton's identities [27], we can also write the acceptance probability as both a determinant and a permanent of sequences of matrices. To see this, define two sequences of matrices by

$$P_k := \begin{pmatrix} 1 & 1 & 0 & \cdots & \\ \mathrm{Tr}[\rho^2] & 1 & 2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ \mathrm{Tr}[\rho^{k-1}] & \mathrm{Tr}[\rho^{k-2}] & \cdots & 1 & k-1 \\ \mathrm{Tr}[\rho^k] & \mathrm{Tr}[\rho^{k-1}] & \cdots & \mathrm{Tr}[\rho^2] & 1 \end{pmatrix} \tag{2.28}$$

and

$$D_k := \begin{pmatrix} 1 & -1 & 0 & \cdots & \\ \mathrm{Tr}[\rho^2] & 1 & -2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ \mathrm{Tr}[\rho^{k-1}] & \mathrm{Tr}[\rho^{k-2}] & \cdots & 1 & 1-k \\ \mathrm{Tr}[\rho^k] & \mathrm{Tr}[\rho^{k-1}] & \cdots & \mathrm{Tr}[\rho^2] & 1 \end{pmatrix}. \tag{2.29}$$

Let $\{\lambda_i\}_i$ be the eigenvalues of ρ_B. Then $\{\lambda_i^m\}_i$ are the eigenvalues of ρ_B^m. So the trace of ρ_B^m is given by

$$\mathrm{Tr}(\rho_B^m) = \sum_i \lambda_i^m. \tag{2.30}$$

That is, $\mathrm{Tr}(\rho_B^m)$ is the mth power sum of the eigenvalues, so that the acceptance probability $p^{(k)}$ is given by the kth complete homogeneous symmetric polynomial. A variant of Newton's identities [27] relates the kth complete homogeneous symmetric polynomial h_k with the kth power sum p_k by $kh_k = \sum_{j=1}^k h_{k-j}p_j$. Now applying Cramer's rule, we can write

$$h_k = \frac{1}{k!}\det\begin{pmatrix} p_1 & -1 & 0 & \cdots & \\ p_2 & p_1 & -2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ p_{k-1} & p_{k-2} & \cdots & p_1 & 1-k \\ p_k & p_{k-1} & \cdots & p_2 & p_1 \end{pmatrix} \tag{2.31}$$

$$= \frac{1}{k!}\mathrm{perm}\begin{pmatrix} p_1 & 1 & 0 & \cdots & \\ p_2 & p_1 & 2 & 0 & \cdots \\ \vdots & \vdots & \ddots & \ddots & \\ p_{k-1} & p_{k-2} & \cdots & p_1 & k-1 \\ p_k & p_{k-1} & \cdots & p_2 & p_1 \end{pmatrix}. \tag{2.32}$$

Thus, by identifying the acceptance probability with h_k and the trace of ρ^j with p_j, we have

$$p^{(k)} = \frac{1}{k!}\mathrm{perm}(P_k) = \frac{1}{k!}\det(D_k). \tag{2.33}$$

See [32, ch. 1, §2] for more information on equations (2.26), (2.31) and (2.32).

3. Strictly decreasing acceptance probability

In [19], it was conjectured that the acceptance probability of the bipartite pure-state separability test is monotone non-increasing in k. We answer this conjecture in the affirmative as a corollary of the following lemma about complete Bell polynomials. In fact, we show that this inequality is strict and the acceptance probability approaches zero in the limit $k \to \infty$ whenever ρ_B is not a pure state.

Lemma 3.1. *Suppose* $1 = p_1 \geq \cdots \geq p_{n+1} \geq 0$. *Then*

$$\frac{1}{(n+1)!} B_{n+1}(p_1, \ldots, n! p_{n+1}) \leq \frac{1}{n!} B_n(p_1, \ldots, (n-1)! p_n). \tag{3.1}$$

Proof. The complete Bell polynomial satisfies the recurrence relation [33]

$$B_{n+1}(x_1, \ldots, x_{n+1}) = \sum_{j=0}^{n} \binom{n}{j} B_{n-j}(x_1, \ldots, x_{n-j}) x_{j+1}. \tag{3.2}$$

Letting $x_i = (i-1)!\, p_i$ and defining $B_j := B_j(1, \ldots, (j-1)! p_j)$, we have

$$\begin{aligned}
\frac{1}{n!} B_n - \frac{1}{(n+1)!} B_{n+1} &= \frac{1}{n!} B_n - \frac{1}{(n+1)!} \sum_{i=0}^{n} \binom{n}{i} B_{n-i} i! p_{i+1} && (3.3)\\
&= \frac{n}{(n+1)!} B_n - \frac{1}{(n+1)!} \sum_{i=1}^{n} \binom{n}{i} B_{n-i} i! p_{i+1} && (3.4)\\
&= \frac{n}{(n+1)!} B_n - \frac{n}{(n+1)!} \sum_{i=0}^{n-1} \binom{n-1}{i} B_{n-i-1} i! p_{i+2} && (3.5)\\
&\geq \frac{n}{(n+1)!} B_n - \frac{n}{(n+1)!} \sum_{i=0}^{n-1} \binom{n-1}{i} B_{n-i-1} i! p_{i+1} && (3.6)\\
&= \frac{n}{(n+1)!} B_n - \frac{n}{(n+1)!} B_n && (3.7)\\
&= 0, && (3.8)
\end{aligned}$$

where in the second equality, the $i = 0$ term was combined with the first term, and the inequality is justified by $p_{i+2} \leq p_{i+1}$. ∎

Proposition 3.2. *The acceptance probability* $p^{(k)}$ *is strictly decreasing and* $\lim_{k\to\infty} p^{(k)} = 0$ *when* ρ_B *is not a pure state.*

Proof. Writing the acceptance probability in terms of the complete Bell polynomials as in (2.27) and noting that $1 = \mathrm{Tr}[\rho_B] > \mathrm{Tr}[\rho_B^2] > \cdots > \mathrm{Tr}[\rho_B^{k+1}] > 0$ (since the eigenvalues of a mixed state are strictly less than one), we have

$$\begin{aligned}
p^{(k+1)} &= \frac{1}{(k+1)!} B_{k+1}(1, \ldots, k! \mathrm{Tr}[\rho_B^{k+1}]) && (3.9)\\
&< \frac{1}{k!} B_k(1, \ldots, (k-1)! \mathrm{Tr}[\rho_B^k]) && (3.10)\\
&= p^{(k)}, && (3.11)
\end{aligned}$$

where we have applied lemma 3.1, noting that inequality (3.6) is now strict because the inequalities between traces of powers of ρ_B are strict as noted above. Thus, $p^{(k)}$ is strictly decreasing. Since the acceptance probability is given by the cycle index polynomial of the symmetric group, it satisfies the recurrence

$$p^{(k)} = \frac{1}{k} \sum_{j=1}^{k} \mathrm{Tr}[\rho^j] p^{(k-j)}. \tag{3.12}$$

Let r be the total number of eigenvalues of ρ_B. From the fact that $p^{(k-j)} \leq 1$, we have

$$p^{(k)} \leq \frac{1}{k}\sum_{j=1}^{k} \mathrm{Tr}(\rho_B^j) \tag{3.13}$$

$$= \frac{1}{k}\sum_{j=1}^{k}\sum_{i=1}^{r} \lambda_i^j \tag{3.14}$$

$$= \sum_{i=1}^{r}\frac{1}{k}\sum_{j=1}^{k} \lambda_i^j \tag{3.15}$$

$$= \sum_{i=1}^{r}\frac{1}{k}\left(\frac{1-\lambda_i^{k+1}}{1-\lambda_i} - 1\right). \tag{3.16}$$

Since ρ_B is not a pure state, we have $\lambda_i < 1$ for all i, so that $\lambda_i^{k+1} \to 0$ as $k \to \infty$ and the acceptance probability therefore goes to zero in the $k \to \infty$ limit. ■

These results indicate that as $k \to \infty$, fewer repetitions of the test are needed to determine whether a given pure state is entangled. There is a trade-off, however, between increasing k and the computational resources needed to conduct a single test. As k increases, one might suspect that the resources needed will increase in such a way that a large enough k is not feasible. Indeed, as one of our results, we discuss the scaling in this claim in §5.

4. Generalization of the algorithm

It is possible to consider a generalization of the bipartite pure-state separability algorithm to groups other than the symmetric one. Furthermore, we can show that these algorithms are also separability tests. Let G be a finite group, and let ψ_{AB} be a pure state. Recall that Cayley's theorem [34] guarantees that every finite group is isomorphic to a subgroup of a permutation group. Then there is a representation of G that takes every $g \in G$ to an element $\pi \in S_k$ for some $k \in \mathbb{N}$ and maps π to the operator that permutes the Hilbert spaces in the composite Hilbert space $\mathcal{H}^{\otimes k}$. Then a generalization of the bipartite pure-state separability algorithm is given by performing a G-Bose symmetry test on the state $\psi_{AB}^{\otimes k}$.

By the argument in the proof of theorem 2.2, we see that one simply has to count the number of cycles of any given cycle type in the permutation subgroup isomorphic to G to obtain a formula for the acceptance probability of the algorithm. That is, the argument in theorem 2.2, combined with Cayley's theorem, proves the following theorem:

Theorem 4.1. *Let p_G denote the acceptance probability with respect to the group G for the generalization of the bipartite pure-state separability algorithm. Then*

$$p_G = Z(G)(1, \ldots, Tr[\rho^k]). \tag{4.1}$$

That is, the acceptance probability p_G is given by the cycle index polynomial (2.25) of G evaluated at $x_j = Tr[\rho^j]$ for $j \in \{1, \ldots, k\}$.

As an aside, we note that (4.1) has an interesting combinatorial meaning. Let $\{\lambda_i\}_{i=1}^r$ denote the eigenvalues of ρ. By Pólya's enumeration theorem [35,36], we can interpret (4.1) as a generating function for the number of non-equivalent colourings of a set S with the r colours $\{\lambda_i\}_{i=1}^r$. The role of G here is to define the equivalence between colourings through its action on S.

We now list several examples involving finite groups, including the symmetric groups, which reproduces the result obtained in §2 (the acceptance probability of the bipartite pure-state separability test). The most trivial example is the identity group.

Example 4.2. Let $G = \{e\}$ be the identity group and let $W_{B_1 \ldots B_k} : G \to U(\mathcal{H}_{B_1 \ldots B_k})$ be the unitary representation mentioned above. Then $W_{B_1 \ldots B_k}(e) = I$, and the acceptance probability is given by

the cycle index polynomial of the identity group. That is,

$$p_{\mathrm{id}}^{(k)} = \mathrm{Tr}[\rho_B]^k = 1. \tag{4.2}$$

Example 4.3. Let $G = C_k = \langle g \rangle$ be the cyclic group of rotations of a regular k-gon. We have many options for the representation, as there are in general many k-cycles to which we can map the generator of C_k. However, each choice produces an isomorphic permutation subgroup, and so we may as well choose $W_{B_1 \cdots B_k}(g) = (1\ \ 2\ \ \cdots\ \ k)$. Then the acceptance probability is given by the cycle index polynomial of C_k. That is,

$$p_{\mathrm{cyc}}^{(k)} = \frac{1}{k} \sum_{m|k} \phi(m) (\mathrm{Tr}[\rho_B^m])^{k/m}, \tag{4.3}$$

where the sum is over all m dividing k, and $\phi(m)$ is the Euler ϕ-function [34], which gives the number of $n \in \mathbb{N}$ such that $n \leq m$ and $\gcd(n, m) = 1$.

Example 4.4. Let $G = D_k$ be the dihedral group, which is generated by the rotation r and reflection f of a regular k-gon. As for the case of the cyclic group, we choose a unitary representation so that it sends the rotation r to a k-cycle. The reflection f is then mapped to a product of $\lfloor k/2 \rfloor$-cycles. Then the acceptance probability is given by the cycle index polynomial of D_k. That is, when k is even,

$$p_{\mathrm{dihedral}}^{(k)} = \frac{1}{2k} \sum_{m|k} \phi(m) (\mathrm{Tr}[\rho_B^m])^{k/m} + \frac{1}{4} \left((\mathrm{Tr}[\rho_B^2])^{(k-2)/2} + (\mathrm{Tr}[\rho_B^2])^{k/2} \right) \tag{4.4}$$

$$= \frac{1}{2} p_{\mathrm{cyc}}^{(k)} + \frac{1}{4} \left((\mathrm{Tr}[\rho_B^2])^{(k-2)/2} + (\mathrm{Tr}[\rho_B^2])^{k/2} \right), \tag{4.5}$$

and when k is odd,

$$p_{\mathrm{dihedral}}^{(k)} = \frac{1}{2k} \sum_{m|k} \phi(m) (\mathrm{Tr}[\rho_B^m])^{k/m} + \frac{1}{2} (\mathrm{Tr}[\rho_B^2])^{(k-1)/2} \tag{4.6}$$

$$= \frac{1}{2} p_{\mathrm{cyc}}^{(k)} + \frac{1}{2} (\mathrm{Tr}[\rho_B^2])^{(k-1)/2}. \tag{4.7}$$

Example 4.5. Let $G = A_k$ be the alternating group, which is already a permutation group. Then the acceptance probability is given by the cycle index polynomial of A_k. That is,

$$p_{\mathrm{alt}}^{(k)} = \sum_{a_1 + 2a_2 + \cdots + ka_k = k} \prod_{j=1}^{k} \frac{1 + (-1)^{a_2 + a_4 + \cdots}}{j^{a_j} a_j!} (\mathrm{Tr}[\rho_B^j])^{a_j}, \tag{4.8}$$

where $(-1)^{a_2 + a_4 + \cdots}$ denotes $(-1)^{a_2 + a_4 + \cdots + a_k}$ if k is even and $(-1)^{a_2 + a_4 + \cdots + a_{k-1}}$ if k is odd.

Example 4.6. We also list the example already discussed in theorem 2.2. Let $G = S_k$ be the symmetric group, which is already a permutation group. Then the acceptance probability is given by the cycle index polynomial of S_k. That is,

$$p_{\mathrm{sym}}^{(k)} = \sum_{a_1 + 2a_2 + \cdots + ka_k = k} \prod_{j=1}^{k} \frac{(\mathrm{Tr}[\rho_B^j])^{a_j}}{j^{a_j} a_j!}. \tag{4.9}$$

Example 4.7. Let $G = Q_8$ be the quaternion group. We can represent Q_8 as a subgroup of S_8 by identifying $\{1, -1, i, -i, j, -j, k, -k\}$ with $\{e, (1\ 2)(3\ 4)(5\ 6)(7\ 8), (1\ 3\ 2\ 4)(5\ 7\ 6\ 8), (1\ 4\ 2\ 3)(5\ 8\ 6\ 7), (1\ 5\ 2\ 6)(3\ 8\ 4\ 7), (1\ 6\ 2\ 5)(3\ 7\ 4\ 8), (1\ 7\ 2\ 8)(3\ 5\ 4\ 6), (1\ 8\ 2\ 7)(3\ 6\ 4\ 5)\}$. Then the acceptance probability is given by the cycle index polynomial of Q_8, which can be read off from the permutation representation. That is,

$$p_{\mathrm{quat}} = \frac{1}{8} \left(1 + (\mathrm{Tr}[\rho_B^2])^4 + 6(\mathrm{Tr}[\rho_B^4])^2 \right). \tag{4.10}$$

Example 4.8. In this example, we generalize the cyclic test to products of cyclic groups. Let $G = \mathbb{Z}_m^k$ be the product of k copies of the group $\mathbb{Z}_m$. We represent G as a permutation subgroup by labelling its elements and letting them act on the group to construct a permutation. For example, if $k = 1$, then $G = \{0, 1, \ldots, m-1\}$. Since 0 has no effect on any element of the group, we map it to the identity element e. Meanwhile, 1 acts on each element of the group by sending 0 to 1, 1 to 2, and so on. So we identify 1 with the cycle $(1 \cdots m)$. The remaining permutations are defined similarly.

Returning to the more general setting, we see that the elements of each order n correspond to products of n-cycles. Now, for an element of G to have order n, each component must contain an element of an order that divides n, with at least one component filled by an element of order n. So the number of elements of order n is given by

$$\sum_{i=1}^{k} \binom{k}{i} (\phi(n))^i \Big(\sum_{\substack{l|n \\ l<n}} \phi(l) \Big)^{k-i} = \Big(\phi(n) + \sum_{\substack{l|n \\ l<n}} \phi(l) \Big)^k - \Big(\sum_{\substack{l|n \\ l<n}} \phi(l) \Big)^k \tag{4.11}$$

$$= n^k - (n - \phi(n))^k, \tag{4.12}$$

where ϕ denotes the Euler ϕ-function. The acceptance probability given by the cycle index polynomial of $\mathbb{Z}_m^k$ is then

$$p_{\mathbb{Z}_m^k}^{(k)} = \frac{1}{m^k} \sum_{n|m} (n^k - (n - \phi(n))^k) (\mathrm{Tr}[\rho_B^n])^{m^k/n}. \tag{4.13}$$

Finally, we show that the above non-trivial examples, as well as any other example involving a non-trivial finite group, are tests for separability of a pure bipartite state. Thus, we have produced an entire class of separability tests.

Proposition 4.9. *Let ψ_{AB} denote a pure bipartite state. Then the generalized bipartite pure-state separability algorithm is, in fact, a faithful test for separability of ψ_{AB} for any non-trivial finite group G, meaning that the acceptance probability is equal to one if and only if the pure state is a separable state.*

Proof. Suppose ψ_{AB} is separable. That is, $|\psi\rangle_{AB} = |\phi\rangle_A \otimes |\varphi\rangle_B$ for some states $|\phi\rangle_A \in \mathcal{H}_A$ and $|\varphi\rangle_B \in \mathcal{H}_B$. Then

$$\rho_B := \mathrm{Tr}_A[\psi_{AB}] = \mathrm{Tr}_A[|\phi\rangle\langle\phi|_A \otimes |\varphi\rangle\langle\varphi|_B] = |\varphi\rangle\langle\varphi|_B . \tag{4.14}$$

That is, ρ_B is a pure state. From theorem 4.1, the acceptance probability of the algorithm is given by the cycle index polynomial evaluated at the traces of increasing powers of ρ_B. But since ρ_B is pure, $\mathrm{Tr}[\rho_B^j] = 1$ for all $j \in \{1, \ldots, n\}$. Then the acceptance probability is equal to the cycle index polynomial at $x_j = 1$ for all $j \in \{1, \ldots, n\}$. That is,

$$p_G = Z(G)(1, \ldots, \mathrm{Tr}[\rho^n]) \tag{4.15}$$

$$= Z(G)(1, \ldots, 1) \tag{4.16}$$

$$= \frac{1}{|G|} \sum_{g \in G} 1^{c_1(g)} \cdots 1^{c_n(g)} \tag{4.17}$$

$$= \frac{1}{|G|} \sum_{g \in G} 1 \tag{4.18}$$

$$= 1 \tag{4.19}$$

where $c_i(g)$ denotes the number of cycles of length i in the disjoint cycle decomposition of g. Thus, ψ_{AB} separable implies that the acceptance probability is identically one.

Now suppose ρ_B is a mixed state. Then $\mathrm{Tr}[\rho_B^j] < 1$ for all $j > 1$ and we have

$$p_G = Z(G)(1, \mathrm{Tr}[\rho_B^2], \ldots, \mathrm{Tr}[\rho_B^n]) \tag{4.20}$$

$$= \frac{1}{|G|} \sum_{g \in G} 1^{c_1(g)} (\mathrm{Tr}[\rho_B^2])^{c_2(g)} \cdots (\mathrm{Tr}[\rho_B^n])^{c_n(g)} \tag{4.21}$$

$$< \frac{1}{|G|} \sum_{g \in G} 1^{c_1(g)} \cdots 1^{c_n(g)} \tag{4.22}$$

$$= \frac{1}{|G|} \sum_{g \in G} 1 \tag{4.23}$$

$$= 1, \tag{4.24}$$

where we have used the assumption that G is nontrivial to guarantee that at least one of the $c_j(g)$ is nonzero so that the inequality holds. Thus, the test is faithful. ■

5. Resource comparison of symmetry tests

Given the generalization in §4, we can now compare the performance of these separability tests. There are two practical concerns to consider when implementing such a test: the rate at which the acceptance probability decays and the resources required to construct it. The cycle index polynomial results described above allow for direct analysis of the former topic, but the latter requires additional consideration before it can be adequately addressed. First, we will specify how resources are counted for each algorithm. Then we compare the resource cost for each algorithm given this framework. We accompany this with a discussion of the acceptance probability of the compared methods.

We now clarify what is meant by resources in this context. For the G-Bose symmetry test described in [19] and tests of that nature, the two primary resources are the number of gates used to construct the test and how many qubits are needed in the control register. We begin with a discussion of gate counting.

(a) Resource counting of quantum gates

The unitary representation in this context is always formed from a collection of SWAP gates used to permute the subsystems. SWAP gates can be realized by a sequence of three CNOT gates in alternating direction. Often, the literature commonly counts the number of CNOT gates used as a resource (e.g. [37]); however, particular architectures may have more efficient realizations of the SWAP gate. Furthermore, this algorithm actually calls for controlled-SWAP gates, which may have vastly different compilations between architectures. For the purposes of this discussion, we will be counting the necessary number of controlled-SWAPs alone. Furthermore, we do not restrict to swapping between consecutive Hilbert spaces, although in principle this could be a limitation of particular systems.

Here, we give an explicit construction for two example groups. The first is the cyclic group test, which is a simple Abelian subgroup of the symmetric group and therefore of interest as a point of comparison. Although constructions of cyclic shifts exist in the literature, our construction follows binary encoding procedures [38,39] and uses fewer gates than a naive implementation and thus warrants discussion. The second construction given describes a recursive implementation of the full permutation test. Similarly, although the quantum Schur transform [40–42] gives a recipe for implementing the symmetric group in principle, the gate construction is abstract and thus difficult to use for accurate gate counts compared to other approaches. As such, we use the construction given in [15]. In the following two subsections, we show that a cyclic group test can be implemented with $\mathcal{O}(k \log(k))$ controlled-SWAP gates and a full symmetric group test (also known as a permutation test) with $\mathcal{O}(k^2)$ controlled-SWAP gates.

Note that our tests are only valid for pure states. If it is unknown whether the input state is pure or mixed, one can first check whether it is indeed pure. To accomplish this, perform a swap test on the state in question to verify that it has purity approximately equal to one and is sufficiently pure.

(i) Cyclic group

Analysis of the cyclic group benefits from established literature. Any cyclic permutation can be achieved in constant depth with $k-1$ gates, where k is the order of the cycle [37]. We will now show that any cyclic group test can be generated by implementing solely the elements in that cycle that are powers of two. This means that the resource cost of implementing the cyclic test of order k is $(k-1)\log_2(k)$, and the constant depth condition above from [37] gives a corresponding depth of $\mathcal{O}(\log_2(k))$ in the separability test.

First, recall that the k-order cyclic group is isomorphic to the set $\mathbb{Z}_k$ of integers modulo k under addition. This will allow us to symbolically represent each element by a single number, understood in this context to be modulo k.

Since the case of $k=1$ is trivial, let us first consider the base case of $k=2$. This example illustrates the general construction of cyclic tests and recreates the well-established swap test [15,17]. The controlled-SWAP element corresponds to the element $1=2^0$, and is the sole gate needed, and the identity element is naturally 0. (Note that 1 is the sole power of two in $\mathbb{Z}_2=\{0,1\}$.) The control state for this test is given by a single qubit state of

$$|+\rangle_{C(2)}=\frac{1}{\sqrt{2}}(|0\rangle+|1\rangle), \tag{5.1}$$

where we employ the computational basis. It is clear that each element in the ancillary basis will give rise to its corresponding group element with this test.

How does this construction generalize? For each given k, we follow a similar recipe as above. As C_k is isomorphic to $\mathbb{Z}_k$, start by identifying each cycle in C_k with a number in $\mathbb{Z}_k$. If we always map the first k-cycle to one, then this map follows simply by mapping cycle composition to integer addition by one. Consider, for instance, the case of C_5. Then the first cycle is (1 2 3 4 5). Map this to 1. Then the next element, (1 3 5 2 4) = (1 2 3 4 5)(1 2 3 4 5) maps to $1+1=2$. After we have identified each element of C_k with an element of $\mathbb{Z}_k$, we can always rewrite these numbers in binary. The beauty of binary construction, as is well appreciated in computer science, is that only elements corresponding to powers of two need to be individually defined, and every other number can be generated from combinations of them. Thus, after this second rewrite, we have a mapping between every cycle in C_k and a binary number. Now to construct the circuit, we only need to implement controlled gates that correspond to cycles that have mapped to a power of two. For C_5, this would be gates that have mapped to 001, 010 and 100 (in decimal: 1, 2 and 4 respectively).

To show how this construction grows, it is most convenient to denote the gates by which power of two they implement. In figure 2, we label gates as 2^j where j ranges from 0 to $\lfloor\log_2(k-1)\rfloor$. To see why $\lfloor\log_2(k-1)\rfloor$ is the final gate, recall the convention that $\mathbb{Z}_k$ always contains 0 instead of k. Then the bound falls out from inspection. Revisiting our above example of $k=5$, the gates we identified as necessary can be equivalently represented as $001=2^0$, $010=2^1$ and $100=2^2$.

This construction can also be achieved by considering the labelling of the control state. If the computational basis is read as a number in binary, we can clearly define the relationship between the computational basis and the group element construction as $|g\rangle=|g_{\text{binary}}\rangle=|g_{\text{decimal}}\rangle$, where the abstract construction is equivalent to a computational basis in binary, which equivalently realizes the familiar group element in decimal. For example, following the above convention, the basis state for $k=5$ given by $|(1\ 3\ 5\ 2\ 4)\rangle=|10\rangle=|2\rangle$ indicates that the element (1 3 5 2 4) can be labelled as the 2 element of the group. As 2 is obviously a power of 2, this group element must be encoded in the circuit. This construction is shown generally in figure 2 and for our specific

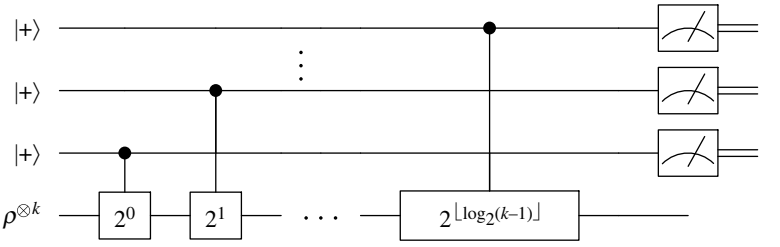

Figure 2. Figure demonstrating how to systematically generate a test for the cyclic group of order k. The notation (2^j) indicates the unitary representation of the element in C_k labelled by the jth power of two. Alternatively, this element is obtained by the full k-cycle $(1, 2, \ldots, k)$ acting on itself 2^j times. Note that the final power is always given by $\lfloor \log_2(k-1) \rfloor$. Also, $|+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$ in the circuit diagram above and the final measurements are performed in the Hadamard basis, accepting if all $+1$ outcomes occur.

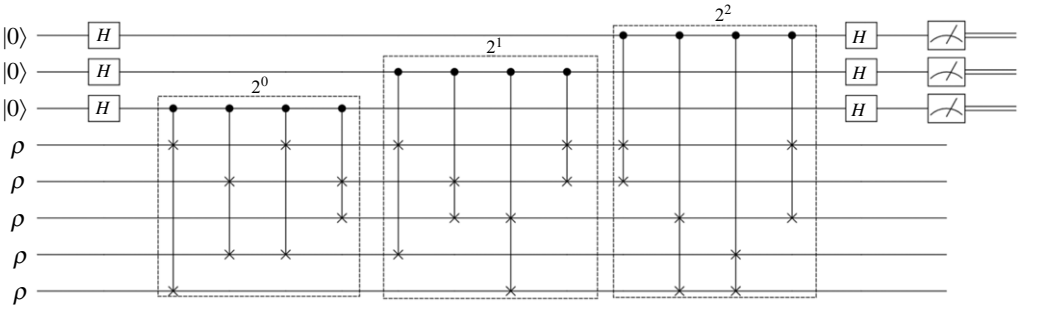

Figure 3. An example of the cyclic group test for $k = 5$. The notation (2^j) indicates the unitary representation of the element in C_k labelled by the jth power of two. For this case, only the elements corresponding to 2^0, 2^1 and 2^2 contribute. Notice that, if the gates are not controlled on the same qubit, each individual cycle collapses to a depth of two with $k - 1$ gates.

example of $k = 5$ in figure 3. Note that all elements of C_k will take at most $k - 1$ SWAP gates to implement.

Furthermore, note that cyclic permutations can be implemented in a constant depth of two [37]. To maintain this depth even for the controlled gates, a GHZ state, $\frac{1}{\sqrt{2}}(|0\rangle^{\otimes m} + |1\rangle^{\otimes m})$, where $m = \lfloor k/2 \rfloor$, can be used instead of a single plus state, similar to the approach employed in [43]. Then the controls can act on different qubits of the state, and the final measurement is taken by projecting back to the GHZ state. This state preparation and the corresponding measurement may add complexity to the ancilla register; however, since the circuit to prepare a k-qubit GHZ state has depth $\mathcal{O}(\log_2(k))$ (with the circuit to project onto it being its inverse), this gives the cyclic group test a depth that grows as $\mathcal{O}(\log_2(k))$.

To see that this circuit is capable of generating every element of the cyclic group, we again refer to the isomorphism between C_k and $\mathbb{Z}_k$. Writing every element of $\mathbb{Z}_k$ in binary, it becomes obvious that every element can be written as an addition of powers of 2 that form the basis of binary numbers. As such, only elements corresponding to new 'digits' need to be considered.

(ii) Symmetric group

We now review a recursive algorithm for the construction of the symmetric group. Necessary to this construction is the proof that the entire group S_k can be generated in a convenient way, using solely transpositions. This construction is equivalent to that given in [15], but we explain it here for completeness.

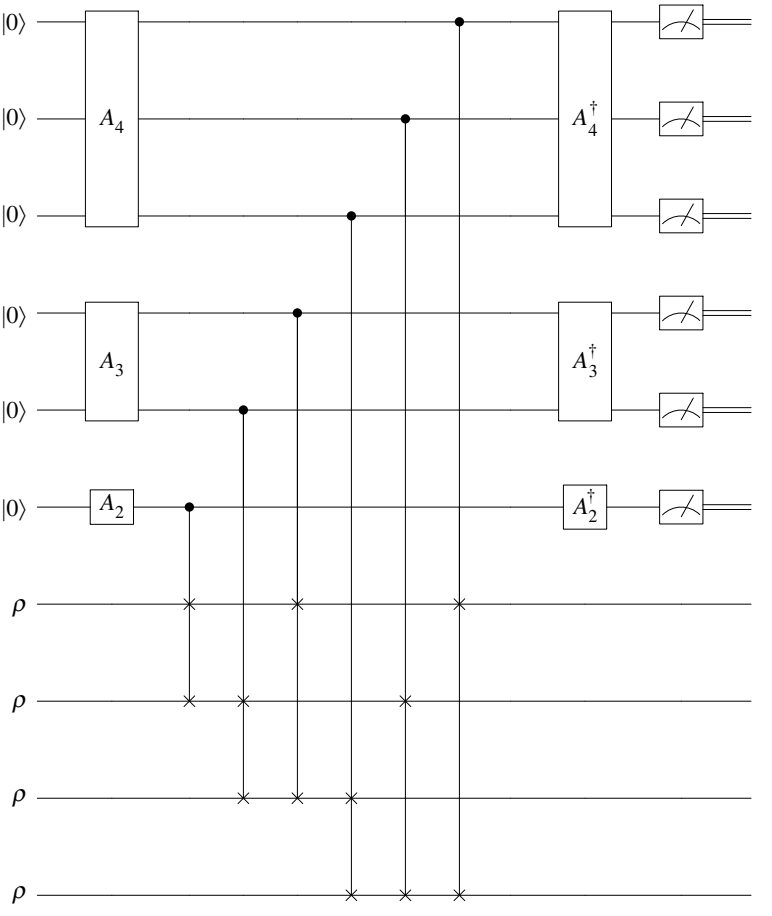

Figure 4. Figure demonstrating how to systematically generate a test for the symmetric group of order four.

Observe that S_2 can be generated by the element $(1\ 2)$. To generate S_3, we need only act on this element from the left by $(1\ 3)$ and $(2\ 3)$. Indeed, the remaining elements of S_3 are given by $(1\ 2\ 3) = (1\ 3)(1\ 2)$ and $(1\ 3\ 2) = (2\ 3)(1\ 2)$. This serves as our base case, and we now proceed by induction. Suppose we can generate every element of S_{k-1} in this way. We must show that the remaining elements of S_k are given by acting on S_{k-1} from the left by the transpositions of the form $(i\ k)$ for $i \in \{1, 2, \ldots, k-1\}$. To see this, let $(i_1\ i_2\ \cdots\ i_m)$ be an arbitrary m-cycle in S_{k-1}. Then acting from the left by $(i_j\ k)$ for some $j \in \{1, \ldots, m\}$ yields $(i_j\ k)(i_1\ i_2\ \cdots\ i_m) = (i_1\ i_2\ \cdots\ i_{j-1}\ k\ i_j\ \cdots\ i_m)$. In this way, we can generate every cycle in S_k. Since every element of S_k can be decomposed into a product of disjoint cycles, we can now generate every element of S_k recursively by appending only transpositions of the form $(i\ k)$. We can visualize this construction by the circuit given in figure 4 for an example when $k = 4$.

Given a way to generate S_k, we now need an appropriate control state to implement these elements. By supposition, the identity can always be implemented via the state $|0\rangle$ tensored with itself to some power. What then for the remaining states? Consider only one 'layer' of the recursive construction of S_k. It suffices to only ever use one transposition at a time. Thus, the control state for every ith layer of transpositions should take the form

$$|+\rangle_{S_i} = \frac{1}{\sqrt{i+1}}(|0\rangle^{\otimes i} + |10\cdots0\rangle + |01\cdots0\rangle + \cdots + |00\cdots1\rangle), \tag{5.2}$$

as given in [15]. These individual control states should be concatenated together to form the control register for the entire algorithm. For a quick sanity check, when considering the tensor product of such states as i ranges from 1 to k, the normalization constant out in front becomes $\sqrt{k!} = \sqrt{|S_k|}$.

However, a question remains; can the control register for such a circuit also be generated recursively? Observe, in figure 4, that we denote a series of gates A_j that act on the control register to create superpositions. Furthermore, notice that we have arranged the transpositions in a consistent manner such that each gate is appended in ascending order of transposition. Then we define the gate A_j to act as such:

$$A_j |0\rangle^{\otimes j-1} = \frac{1}{\sqrt{j}}(|0\rangle^{\otimes j-1} + |W_{j-1}\rangle), \tag{5.3}$$

where $|W_{j-1}\rangle = (1/\sqrt{j-1}) \sum_{i=1}^{j-1} |2^i\rangle$ is the W-state on $j-1$ qubits. Here $|2^i\rangle$ is the state with a one in the ith component and a zero elsewhere. We can observe by inspection that this action, when taken recursively from $j=2$ to $j=k$, will generate a superposition over $k!$ basis elements. An example of this construction can be seen in figure 4 for $k=4$.

There are several choices available to construct these A_j gates. We review two here. One recursive approach is to begin by designing the circuit for A_i; then the next gate A_{i+1} is given by adding $i+1$ control qubits, initializing the first qubit to a superposition of $((1/\sqrt{i})|0\rangle + (\sqrt{i-1}/\sqrt{i})|1\rangle)$, then controlling off of this state, implement A_i on the remaining new qubits. However, this naive approach will use numerous gates and quickly grow in size. In [15], they assume the first i qubits are initialized and then add $i+1$ qubits for the recursive step. The $(i+1)$-th qubit can be acted on by a one-qubit gate U_i given by

$$U_i := \frac{1}{\sqrt{i+1}} \begin{pmatrix} 1 & -\sqrt{i} \\ \sqrt{i} & 1 \end{pmatrix}. \tag{5.4}$$

Following this, act simultaneously on the $i+1$ qubit and the remaining qubits with a series of two-qubit gates given by

$$T_{j,j+1} := \frac{1}{\sqrt{i-j+1}} \begin{pmatrix} \sqrt{i-j+1} & 0 & 0 & 0 \\ 0 & 1 & \sqrt{i-j} & 0 \\ 0 & -\sqrt{i-j} & 1 & 0 \\ 0 & 0 & 0 & \sqrt{i-j+1} \end{pmatrix}, \tag{5.5}$$

where j ranges from 1 to $i-1$. This will give the desired control state. In all likelihood, there are even more ways to generate the desired control register. Whichever approach is chosen, the control state should remain the same. Note that the ancilla cost of the control state should be at least $\mathcal{O}(k \log_2 k)$ ancilla qubits regardless simply from the magnitude of the symmetric group, $|S_k| = k!$.

Given this construction, it is easy to see the number of controlled-SWAP gates needed to perform the symmetric group test. Indeed, from figure 4, we see that the number of controlled-SWAPs needed when $k=4$ is $1+2+3=6$, where the 1 corresponds to the permutation (1 2) needed to generate S_2, the 2 corresponds to the permutations (2 3) and (1 3) needed to generate S_3 from S_2, and the 3 corresponds to the permutations (3 4), (2 4) and (1 4) needed to generate S_4 from S_3. By induction, the number of controlled-SWAP's needed to perform the kth symmetric group test is $k(k-1)/2$, thus leading to the claimed $\mathcal{O}(k^2)$ gate complexity.

(iii) Dihedral group

The dihedral group, D_k is isomorphic to the semi-direct product of $\mathbb{Z}_k$ with $\mathbb{Z}_2$, with $\mathbb{Z}_2$ acting on $\mathbb{Z}_k$ by inversion. As such, it can be formed in a faithful way using a cyclic group generator and a non-commuting action that squares to identity. Using just the generators of the group, it is clear that the unitary flip action adds a factor of two to the number of cyclic gates needed, plus

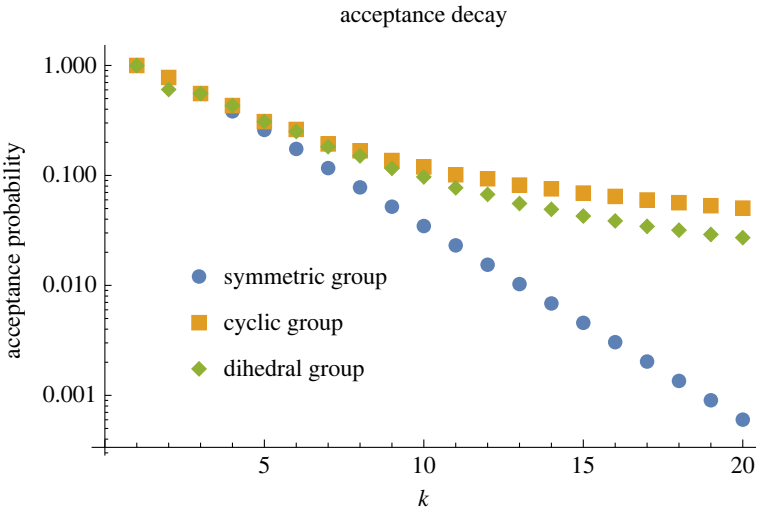

Figure 5. Plot of the acceptance probabilities of each separability test as k increases, for the symmetric group S_k, the cyclic group C_k, and the dihedral group D_k. For this example, we use a reduced W-state as an example to illustrate the algorithmic scaling for an unextendible state. For a separable state, all acceptance probabilities are equal to one.

the additional instance of the flip element acting alone. In this manner, the full dihedral group requires at most $2k \log_2(k)$ gates to implement.

(b) Comparison between subgroups of the symmetric group

Now that we have given a method to count the number of quantum gates necessary for these separability tests, we consider if there is any advantage to using a simpler group as k increases. Essentially, when is the trade-off between additional gates and acceptance probability favourable towards the various tests?

The inherent motivation behind increasing k is to obtain a smaller acceptance probability, prompting the need for proposition 3.2. Clearly, the symmetric test provides the most stringent bound (figure 5), yet it grows quickly in terms of gate resources needed (figure 6). The cyclic group, however, benefits from the simplest construction but does not decay as quickly as the full symmetric group. It should be noted that, while it certainly appears to be true, we have not shown rigorously that the tests corresponding to sequences of groups other than the symmetric groups are decreasing. We suspect that a generating function argument might do the trick, but were not able to produce this result.

To visualize this trade-off, we consider the quantity $R_{\text{test}}/(1 - P_{\text{acc}})$, where R_{test} is the number of resources needed to perform the test via the counting methods described above and P_{acc} is the acceptance probability of the test. We employ the quantity $1 - P_{\text{acc}}$ in the denominator, as we would like the test to have a lower acceptance probability for non-separable states, and thus the denominator will converge to one for better algorithms. This quantifier is very closely aligned with the expected runtime of the algorithm until getting a failure (it would be exactly equal to the expected runtime if we instead used circuit depth over $1 - P_{\text{acc}}$ as the figure of merit). Thus, in comparing this quantity for the various tests, smaller values correspond to more ideal behaviour from the algorithms. In figure 7, we show this quantity for algorithms generated by the cyclic group and the symmetric group.

Examining figure 7, we see a clear difference in the performance between the tests generated by the cyclic group and the symmetric group. The plotted ratio can be thought of as resources-to-rejection, in the sense that $1 - P_{\text{acc}}$ is the probability that a non-separable state is correctly identified—or rather, the failure rate of the algorithm for such a state. Although figure 5 makes it

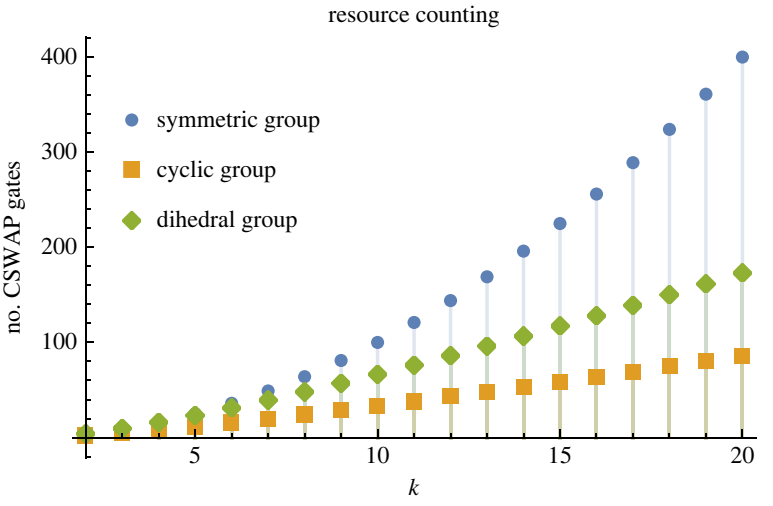

Figure 6. Plot of the resource scaling in terms of the number of controlled-SWAP gates used for each group test as k increases. We consider the symmetric group S_k, the cyclic group C_k, and the dihedral group D_k, and we use the gate counting methods described in the text.

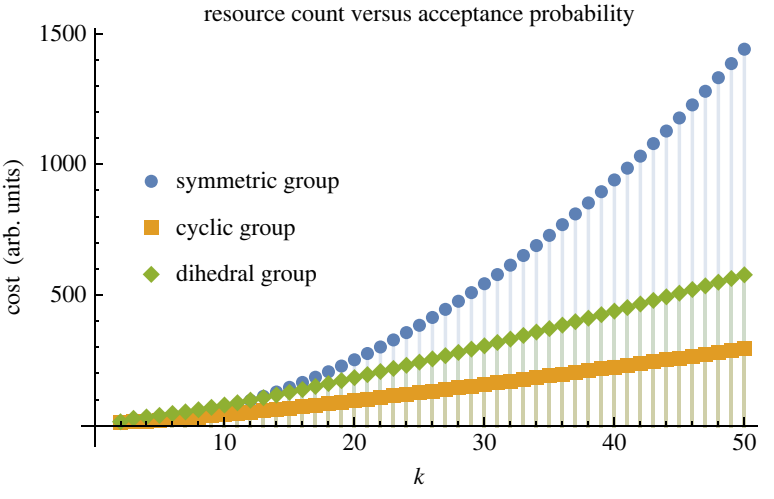

Figure 7. We show the ratio of the resources required to rejection probability as k increases. We consider here the cyclic group of k elements as an example of a simple Abelian group and show it gives an advantage in terms of the resources-to-rejection metric over the test generated by the full symmetric group.

appear that the standard test generated from S_k would always be preferable, we determine from this comparison that the C_k algorithm gives more benefit per gate resource.

From this analysis, we can assert that the simpler test for separability is more cost-efficient than the full permutation test. We show in figure 5 that both tests show a decrease in acceptance probability as k increases, a desirable trait. However, figure 6 shows how quickly circuit sizes grow as k increases, particularly for S_k, which can be considered the standard test. Figure 7 bridges these notions to show that the comparative growth in gate resources of S_k outweighs the relative decrease in acceptance probability given over the C_k test. We thus determine that the cyclic group C_k suffices as a separability test of this nature.

6. Conclusion

In this work, we have presented several separability tests for bipartite pure states, and we have established analytical expressions for their acceptance probabilities. These expressions invariably rely on the cycle index polynomial of the group. Indeed, from a mathematical point of view, this relationship seems natural, due to the inherent combinatorics present in the algorithms. Nonetheless, these expressions give us direct insight into the performance of any separability tests generated from a finite group—which we have shown can be feasibly constructed. Using this perspective, we demonstrate that when using more copies of the state under test, these tests become more stringent. Additionally, we observe that the full symmetric test using a representation of the symmetric group gives a quickly decreasing acceptance probability for an entangled state; however, for the given implementations of these algorithms, other tests can use fewer resources and still show great efficiency.

Here, we have limited ourselves to pure bipartite states; however, we believe multipartite tests may yield results in a similar vein. For instance, a trivial implementation would be to separate all parties into individual tests and then multiply the results. There is a question, however, if more elegant algorithms exist for multipartite cases, and if interesting mathematics arise in the study of such systems. Additionally, generalizing these algorithms to apply to mixed states is a natural next step. In particular, providing tests for separability of multipartite mixed states would be an interesting subject for further study. Furthermore, we think it would be interesting to study the performance of these algorithms on near-term quantum devices; all of our analysis in this paper is for the ideal case.

For future work, we propose considering further the nature of the relationship between cycle index polynomials and quantum algorithms. Our results have shown a clear benefit in the use of cycle index polynomials in describing separability tests; is the reverse true? An interesting question is whether algorithms of this form can be used to determine the coefficients of cycle index polynomials in general, and we leave this for future work.

Data accessibility. The datasets generated during and/or analysed during the current study are available in the GitHub repository: https://github.com/mlabo15/GeneralizedSeparability. The data are provided in electronic supplementary material [44].

Authors' contributions. Z.P.B.: conceptualization, formal analysis, investigation, methodology, writing—original draft, writing—review and editing; M.L.L.: formal analysis, investigation, methodology, writing—original draft, writing—review and editing; M.M.W.: investigation, methodology, supervision, validation, writing—review and editing.

All authors gave final approval for publication and agreed to be held accountable for the work performed therein.

Conflict of interest declaration. The authors have no non-financial competing interests to declare.

Funding. M.L.L. supported by the DOD Smart Scholarship program, and M.M.W. from the National Science Foundation (grant no. 1907615).

Acknowledgements. M.L.L. acknowledges support from the DOD Smart Scholarship program, and MMW from the National Science Foundation under grant no. 1907615. Z.P.B. acknowledges Christophe Vignat for his helpful comments on a draft of this work. M.L.L. would like to acknowledge Kate Caillet for helpful discussions.

References

1. Peres A. 1996 Separability criterion for density matrices. *Phys. Rev. Lett.* **77**, 1413–1415. (doi:10.1103/PhysRevLett.77.1413)
2. Horodecki M, Horodecki P, Horodecki R. 1996 Separability of mixed states: necessary and sufficient conditions. *Phys. Lett. A* **223**, 1–8. (doi:10.1016/S0375-9601(96)00706-2)
3. Życzkowski K, Horodecki P, Sanpera A, Lewenstein M. 1998 Volume of the set of separable states. *Phys. Rev. A* **58**, 883–892. (doi:10.1103/PhysRevA.58.883)
4. Eisert J, Plenio MB. 1999 A comparison of entanglement measures. *J. Mod. Opt.* **46**, 145–154. (doi:10.1080/09500349908231260)

5. Vidal G, Werner RF. 2002 Computable measure of entanglement. *Phys. Rev. A* **65**, 032314. (doi:10.1103/PhysRevA.65.032314)
6. Werner RF. 1989 An application of Bell's inequalities to a quantum state extension problem. *Lett. Math. Phys.* **17**, 359–363. (doi:10.1007/BF00399761)
7. Doherty AC, Parrilo PA, Spedalieri FM. 2002 Distinguishing separable and entangled states. *Phys. Rev. Lett.* **88**, 187904. (doi:10.1103/PhysRevLett.88.187904)
8. Navascués M, Owari M, Plenio MB. 2009 Power of symmetric extensions for entanglement detection. *Phys. Rev. A* **80**, 052306. (doi:10.1103/PhysRevA.80.052306)
9. Kaur E, Das S, Wilde MM, Winter A. 2019 Extendibility limits the performance of quantum processors. *Phys. Rev. Lett.* **123**, 070502. (doi:10.1103/PhysRevLett.123.070502)
10. Wang K, Wang X, Wilde MM. Quantifying the unextendibility of entanglement, November 2019. (https://arxiv.org/abs/1911.07433)
11. Kaur E, Das S, Wilde MM, Winter A. 2021 Resource theory of unextendibility and nonasymptotic quantum capacity. *Phys. Rev. A* **104**, 022401. (doi:10.1103/PhysRevA.104.022401)
12. Brydges T, Elben A, Jurcevic P, Vermersch B, Maier C, Lanyon BP, Zoller P, Blatt R, Roos CF. 2019 Probing Rényi entanglement entropy via randomized measurements. *Science* **364**, 260–263. (doi:10.1126/science.aau4963)
13. Shalm LK, Adamson RB, Steinberg AM. 2005 Experimentally measuring the purity of a state without state tomography. In *Conference on lasers and electro-optics/quantum electronics and laser science and photonic applications systems technologies*, , Baltimore, MD, 22-27 May 2005, p. QThD4. New York, NY: IEEE.
14. Cassemiro KN, Laiho K, Silberhorn C. 2010 Accessing the purity of a single photon by the width of the Hong–Ou–Mandel interference. *New J. Phys.* **12**, 113052. (doi:10.1088/1367-2630/12/11/113052)
15. Barenco A, Berthiaume A, Deutsch D, Ekert A, Jozsa R, Macchiavello C. 1997 Stabilization of quantum computations by symmetrization. *SIAM J. Comput.* **26**, 1541–1557. (doi:10.1137/S0097539796302452)
16. Brennen GK. 2003 An observable measure of entanglement for pure states of multi-qubit systems. *Quantum Inform. Comput.* **3**, 619–626. (doi:10.26421/QIC3.6-5)
17. Buhrman H, Cleve R, Watrous J, De Wolf R. 2001 Quantum fingerprinting. *Phys. Rev. Lett.* **87**, 167902. (doi:10.1103/PhysRevLett.87.167902)
18. Burchardt A, Czartowski J, Życzkowski K. 2021 Entanglement in highly symmetric multipartite quantum states. *Phys. Rev. A* **104**, 022426. (doi:10.1103/PhysRevA.104.022426)
19. LaBorde ML, Rethinasamy S, Wilde MM. 2021 Testing symmetry on quantum computers. (http://arXiv.org/abs/2105.12758)
20. Islam R, PM Preiss RM, Tai EM, Lukin A, Rispoli M, Greiner M. 2015 Measuring entanglement entropy in a quantum many-body system. *Nature* **528**, 77–83. (doi:10.1038/nature15750)
21. Nguyen C-H, Tseng K-W, Maslennikov G, Gan HCJ, Matsukevich D. 2021 Experimental swap test of infinite dimensional quantum states. Preprint. (https://arxiv.org/abs/2103.10219)
22. Kada M, Nishimura H, Yamakami T. 2008 The efficiency of quantum identity testing of multiple states. *J. Phys. A: Math. Theor.* **41**, 395309. (doi:10.1088/1751-8113/41/39/395309)
23. Harrow AW. 2013 The church of the symmetric subspace. (https://arxiv.org/abs/1308.6595)
24. Pólya G. 1937 Kombinatorische anzahlbestimmungen für gruppen, graphen und chemische verbindungen. *Acta Math.* **68**, 145–254. (doi:10.1007/BF02546665)
25. Roberts F. 2009 *Applied combinatorics*. Boca Raton, FL: CRC Press.
26. Roman SM, Rota G-C. 1978 The umbral calculus. *Adv. Math.* **27**, 95–188. (doi:10.1016/0001-8708(78)90087-7)
27. Mead DG. 1992 Newton's identities. *Am. Math. Mon.* **99**, 749–751. (doi:10.1080/00029890.1992.11995923)
28. Harrow AW. Applications of coherent classical communication and the Schur transform to quantum information theory. PhD thesis, Massachusetts Institute of Technology, December 2005. (https://arxiv.org/abs/quant-ph/0512255)
29. Ekert AK, Alves CM, Oi DKL, Horodecki M, Horodecki P, Kwek LC. 2002 Direct estimations of linear and nonlinear functionals of a quantum state. *Phys. Rev. Lett.* **88**, 217901. (doi:10.1103/PhysRevLett.88.217901)
30. van Lint JH, Wilson RM. 2001 *A course in combinatorics*. Cambridge, UK: Cambridge University Press.

31. Comtet L. 1974 *Advanced combinatorics: the art of finite and infinite expansions*. Springer Netherlands.
32. Macdonald IG. 1995 *Symmetric functions and hall polynomials*. Oxford Mathematical Monographs. Oxford, UK: Clarendon Press.
33. Bell ET. 1934 Exponential polynomials. *Ann. Math.* **35**, 258–277. (doi:10.2307/1968431)
34. Dummit DS, Foote RM. 2004 *Abstract algebra*, 3rd edn. New York, NY: Wiley.
35. Brualdi RA. 2010 *Introductory combinatorics*. New York, NY: Wiley.
36. Tucker A. 1995 *Applied combinatorics*. New York, NY: Wiley.
37. Grassl M, Beth T. 2000 Cyclic quantum error–correcting codes and quantum shift registers. *Proc. R. Soc. Lond. A* **456**, 2689–2706. (doi:10.1098/rspa.2000.0633)
38. Babbush R, Gidney C, Berry DW, Wiebe N, McClean J, Paler A, Fowler A, Neven H. 2018 Encoding electronic spectra in quantum circuits with linear T complexity. *Phys. Rev. X* **8**, 041015. (doi:10.1103/PhysRevX.8.041015)
39. Low GH, Chuang IL. 2019 Hamiltonian simulation by qubitization. *Quantum* **3**, 163. (doi:10.22331/q-2019-07-12-163)
40. Bacon D, Chuang IL, Harrow AW. 2006 Efficient quantum circuits for Schur and Clebsch-Gordan transforms. *Phys. Rev. Lett.* **97**, 170502. (doi:10.1103/PhysRevLett.97.170502)
41. Bacon D, Chuang IL, Harrow AW. 2007 The quantum Schur and Clebsch-Gordan transforms: I. Efficient qudit circuits. In *Proc. of the Eighteenth Annual ACM-SIAM Symp. on Discrete Algorithms, SODA '07, New Orleans, LA, 7–9 January 2007*, pp. 1235–1244. USA: Society for Industrial and Applied Mathematics.
42. Krovi H. 2019 An efficient high dimensional quantum Schur transform. *Quantum* **3**, 122. (doi:10.22331/q-2019-02-14-122)
43. Quek Y, Wilde MM, Kaur E. 2022 Multivariate trace estimation in constant quantum depth. (https://arxiv.org/abs/2206.15405)
44. Bradshaw ZP, LaBorde ML, Wilde MM. 2023 Cycle index polynomials and generalized quantum separability tests. Figshare. (doi:10.6084/m9.figshare.c.6662923)