Stabilizer Extent Calculation by Column Generation

Hiroki Hamaguchi[1], Kou Hamada[1], and Nobuyuki Yoshioka[2,3,4]

[1]Graduate School of Information Science and Technology, University of Tokyo, Tokyo, Japan
[2]Department of Applied Physics, University of Tokyo, Japan
[3]Theoretical Quantum Physics Laboratory, RIKEN Cluster for Pioneering Research (CPR), Wako-shi, Saitama 351-0198, Japan
[4]JST, PRESTO, 4-1-8 Honcho, Kawaguchi, Saitama, 332-0012, Japan

todo

1 Introduction

todo

Table 1: The size of $\mathcal{S}_n$, the data size of A_n in sparse matrix format [1], and the average time for 10 Haar random pure state to numerically compute the stabilizer extent by the naive algorithm and our proposed algorithm

n	5	6	7	8	9	10
$\lvert\mathcal{S}_n\rvert$	2.42e+06	3.15e+08	8.13e+10	4.18e+13	4.29e+16	8.79e+19
size of A_n	1011 MiB	254 GiB	153 TiB	153 PiB	305 EiB	1 YiB
naive	>10 min	×	×	×	×	×
proposed	1 s	10 s	3 min	2 h	(Approx)	(Approx)

2 Preliminaries

Let $\mathcal{S}_n := \{|\phi_j\rangle\}$ be the entire set of n-qubit stabilizer states. We also define the density matrix for $|\phi_j\rangle$ as $\sigma_j := |\phi_j\rangle\langle\phi_j|$. The size of $\mathcal{S}_n$ scales superexponentially as $|\mathcal{S}_n| = 2^n \prod_{k=0}^{n-1}(2^{n-k}+1) = 2^{\mathcal{O}(n^2)}$ [2, Proposition 1]. See also Table 1 for the size of $\mathcal{S}_n$.

The *Robustness of Magic* (RoM) is introduced in [3] to quantify an n-qubit state ρ, represented by density matrix, and defined as follows:

$$\mathcal{R}(\rho) := \min_{c \in \mathbb{R}^{|\mathcal{S}_n|}} \left\{ \|c\|_1 \;\middle|\; \rho = \sum_{j=1}^{|\mathcal{S}_n|} c_j \sigma_j \right\}$$

Hiroki Hamaguchi: hamaguchi-hiroki0510@g.ecc.u-tokyo.ac.jp
Kou Hamada: zkouaaa@g.ecc.u-tokyo.ac.jp
Nobuyuki Yoshioka: nyoshioka@ap.t.u-tokyo.ac.jp

On the other hand, the *stabilizer extent* is introduced in [4, Definition 3] to quantify an normalized n-qubit state ψ, represented by state vector, and defined as follows:

$$\xi(\psi) := \min_{c \in \mathbb{C}^{|\mathcal{S}_n|}} \left\{ \|c\|_1^2 \;\middle|\; |\psi\rangle = \sum_{j=1}^{|\mathcal{S}_n|} c_j |\phi_j\rangle \right\} \tag{1}$$

In this paper, our focus lies on numerical computation of the stabilizer extent. This definition of the stabilizer extent can be simplified as complex L^1-norm minimization problem:

$$\sqrt{\xi(\psi)} = \min_{x \in \mathbb{C}^{|\mathcal{S}_n|}} \{\|x\|_1 \mid A_n x = b\} \tag{2}$$

Here, we define $A_n \in \mathbb{C}^{2^n \times |\mathcal{S}_n|}$ as $(A_n)_{ij} := \langle i|\phi_j\rangle$ and $b \in \mathbb{C}^{2^n}$ as $b_i := \langle i|\psi\rangle$ using the computational basis $\{|i\rangle\}_{i=0}^{2^n-1}$. As in [5], the problem (2) is a second order cone program (SOCP). Thus, by defining $\mathcal{A}_n$ as the columns set $\{a_j\}$ of A_n, its dual problem can be derived as [5, Appendix A][6, Section 5.1.6]

$$\sqrt{\xi(\psi)} = \max_{y \in \mathbb{C}^{2^n}} \left\{ \mathrm{Re}(b^\dagger y) \;\middle|\; \left|a_j^\dagger y\right| \le 1 \text{ for all } a_j \in \mathcal{A}_n \right\} \tag{3}$$

where $\dagger$ denotes the conjugate transpose. While the true objective function of the Lagrange dual problem corresponding to (2) is not $\mathrm{Re}(b^\dagger y)$ but $-\mathrm{Re}(b^\dagger y)$, we flipped the sign for simplicity. This is valid as it does not alter the optimal solution, owing to $\left|a_j^\dagger y\right| = \left|a_j^\dagger(-y)\right|$.

Further, in order to describe our algorithm in later sections, we denote a function $\mathrm{SolveSOCP}(\mathcal{C}, b)$ which takes a columns set $\mathcal{C} \subseteq \mathcal{A}$ and a vector b, and returns the optimal primal solution x and dual optimal solution y of the SOCP problem (2) and (3). In actual implementation, this function can be realized by just solving the corresponding primal problem (2) with SOCP solver, such as MOSEK [7] or CVXPY [8, 9].

3 Scaling up The Exact Stabilizer Extent Calculation

In the preceding sections, we introduced two quantum resource measures: Robustness of Magic and stabilizer extent. Despite both being efficiently quantifiable through convex optimization problems, solving them directly for $n > 5$ qubit systems becomes impractical due to the superexponential growth of the number of stabilizer states $|\mathcal{S}_n|$. To address this challenge, in [11], we proposed employing the classical optimization technique known as column generation (CG) method [10] for Robustness of Magic calculation. However, it remained unclear whether the same approach could be applied to stabilizer extent, since SOCP is generally more difficult class of problem than LP, which is used in RoM calculation. Here, we demonstrate that leveraging the specific structure of stabilizer states enables a similar method to work effectively for calculating the stabilizer extent as well.

3.1 Core Subroutine: Calculating Overlap

Before we start to consider about the stabilizer extent, we define *fidelity* of $b \in \mathbb{C}^{2^n}$ as

$$\sqrt{F(b)} := \max_{a_j \in \mathcal{A}_n} \left|a_j^\dagger b\right| = \max_{\phi \in \mathcal{S}_n} |\langle\phi|\psi\rangle|,$$

which is the maximal overlap between y and the stabilizer states [4, Definition 4][5]. It is known that fidelity is deeply related to stabilizer extend [4, Theorem 4] [5, Theorem 4], and fidelity for mixed state was important when computing Robustness of Magic [11]. Further, we will show fidelity plays a crucial role in our proposed algorithm in later sections. In this section, we show that how to compute the fidelity efficiently up to 8-qubit systems. To this end, we introduce the following proposition. As a well-known fact, the stabilizer states have a form as shown in the following proposition.

Proposition 1 ([12, Theorem 2], [13, Section 5], [14, Theorem 5.(ii)]). *All stabilizer states can be written in the following form:*

$$\begin{cases} |\phi\rangle := |t\rangle & \text{if } k = 0 \\ |\phi\rangle := \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} (-1)^{x^\top Q x} i^{c^\top x} |Rx+t\rangle & \text{if } k > 0 \end{cases} \tag{4}$$

where $Q \in \mathbb{F}_2^{k\times k}$, $c \in \mathbb{F}_2^k$, $R \in \mathbb{F}_2^{n\times k}$, $t \in \mathbb{F}_2^n$ *and* $\mathrm{rank}(R) = k$. *Also, any state that can be written in this form is a stabilizer state.*

By modifying the form slightly, we can obtain the following more convenient form. The proof is given in Appendix A.1.

Theorem 1. *The form* (4) *under the following conditions enumerates all stabilizer states without any duplication or omission.*

- Q *is a upper triangular* $\mathbb{F}_2^{k\times k}$ *matrix.*
- R *is a* $\mathbb{F}_2^{n\times k}$ *rref (reduced row echelon form) matrix satisfies* $\mathrm{rank}(R) = k$.
- t *belongs to the complement of the row space of* R.

Let ϕ be a one of stabilizer state in the standard form (4) with $k > 0$, which means $|\phi\rangle = \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} (-1)^{x^\top Q x} i^{c^\top x} |Rx+t\rangle$. Then, by denoting $a \in \mathcal{A}$ as the corresponding vector of ϕ, the overlap between a and b is

$$a^\dagger b = \langle\phi|\psi\rangle = \frac{1}{2^{k/2}} \sum_{x=0}^{2^n-1} (-1)^{x^\top Q x} i^{c^\top x} \langle Rx+t|\psi\rangle = \sum_{x=0}^{2^n-1} (-1)^{x^\top Q x} i^{c^\top x} \left(\frac{1}{2^{k/2}} b^\dagger_{Rx+t}\right). \tag{5}$$

In the following, we define $P_x := \frac{1}{2^{k/2}} b^\dagger_x$, and for the simplicity, we fix $k = n, R = I_n, t = 0$. This assumption is not restrictive since the other cases can be easily reduced to this case. Recall that what we want is $\max_{a_j \in \mathcal{A}_n} \left|a_j^\dagger b\right|$. Owing to (5), this is basically equivalent to the following problem:

$$\max_{c\in\mathbb{F}_2^n, Q\in\mathbb{F}_2^{n\times n}} \left\{ \left| \sum_{x=0}^{2^n-1} (-1)^{x^\top Q x} i^{c^\top x} P_x \right| \right\}. \tag{6}$$

If we solve (6) naively, the time complexity is $\mathcal{O}\left(2^{n+n(n+1)/2} 2^n n^2\right)$, where $2^{n+n(n+1)/2}$ is the number of the possible (c, Q), 2^n is the number of the terms in the summation, and n^2 is the computational cost per each term. However, we can reduce this time complexity by using the following theorem.

Theorem 2. *The problem* (6) *can be solved in* $\mathcal{O}\left(2^{n+n(n+1)/2}\right)$ *time complexity and* $\mathcal{O}(2^n)$ *space complexity.*

The basic algorithm to solve this problem is the depth-first search (DFS) algorithm, which is describe in Figure 1. The detailed proof is given in Appendix A.2 or here?. Consequently, the next theorem is obtained.

Theorem 3. *The fidelity of a n-qubit state $|\psi\rangle$, defined as*

$$\max_{\phi \in \mathcal{S}_n} |\langle \phi | \psi \rangle| = \max_{a_j \in \mathcal{A}_n} \left| a_j^\dagger b \right|$$

can be computed in time complexity of $\mathcal{O}(|S_n|)$ and space complexity of $\mathcal{O}(2^n)$.

As a remark, the fidelity of a random 8-qubit pure state can be computed in ?? minutes with a *laptop* powered by Intel(R) Core(TM) i7-10510U CPU with 16 GB RAM. It's worth noting that the algorithm's speed is enhanced by branch cutting within the DFS algorithm, as detailed in Appendix A.3. We will use this algorithm as a subroutine in later sections.

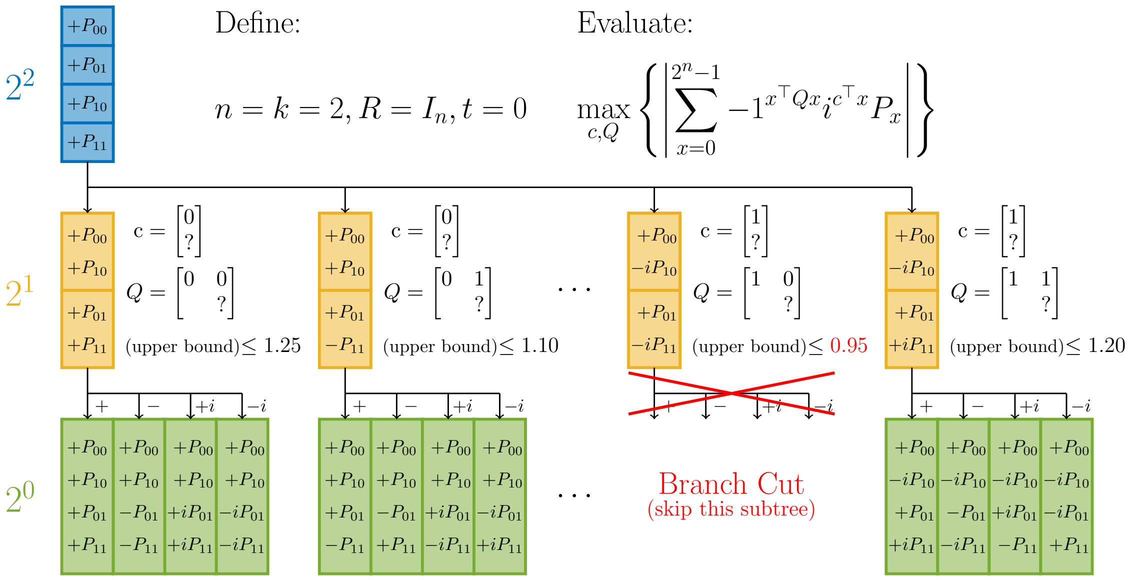

Figure 1: Visualization of the DFS algorithm for Theorem 2. The DFS algorithm is a recursive procedure to calculates the overlap. Each cell stores the evaluated value of the expression, and each leaf node corresponds to the value $\sum_{x=0}^{2^n-1}(-1)^{x^\top Q x_i c^\top x} P_x$. Memory usage is limited to only $\sum_{i=0}^{n} 2^n$. During computation, the maximal solution is either lower bounded by the current best solution or, in certain cases, 1. Thus, branches can be terminated if the upper bound of the current branch is inferior to these values.

3.2 CG method for the stabilizer extent calculation

Next, we introduce the CG method outlined in Algorithm 1. This is exactly the algorithm to compute the stabilizer extent $\xi(\psi)$, and is a iterative algorithm that solves a subproblem restricted to $\mathcal{C} \subseteq \mathcal{A}_n$ per each iterations. It begins with a small subset $\mathcal{C}_0$ and progressively adds columns $\mathcal{C}'$ that violate the constraints of the dual problem (3), and terminate if there are no more violated columns. For further implementation techniques, we direct the reader to [11]. There are two key aspects of this algorithm: the initialization process and the optimality of the solution. We will discuss these in subsequent sections.

3.2.1 Initialization

In the initial step of Algorithm 1, we select a subset $\{a_j\} = \mathcal{C}_0 \subseteq \mathcal{A}_n$ in descending order of $\left| a_j^\dagger b \right|$, which can be computed efficiently as stated in Theorem 3. This choice can be justified with various interpretations. One of them is to consider $\left| a_j^\dagger b \right| = |\langle \phi_j | \psi \rangle|$ as the

Algorithm 1: Exact stabilizer extent calculation by Column Generation

Input: vector b corresponding to the state ψ

Output: Exact stabilizer extent $\xi(\psi)$

$\mathcal{C}_0 \leftarrow$ Partial set of $\mathcal{A}_n$ `/* Initialize using top overlap` $\left|a_j^\dagger b\right|$ `*/`

for $k = 0, 1, 2, \ldots$ **do**

  $x_k, y_k \leftarrow$ `SolveSOCP`$(\mathcal{C}_k, \boldsymbol{b})$

  $\mathcal{C}' \leftarrow \left\{a \in \mathcal{A}_n \,\middle|\, \left|a^\dagger y_k\right| > 1\right\}$ `/* Use of subroutine in Section 3.1 */`

  if $\mathcal{C}' = \emptyset$ **then**

    return $\xi(\psi) = \|x_k\|_1$

  $\mathcal{C}_{k+1} \leftarrow \mathcal{C}_k \cup \mathcal{C}'$

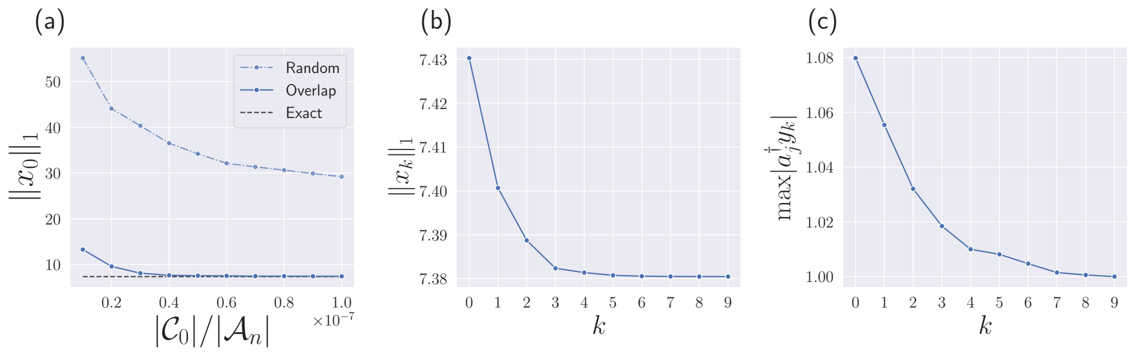

Figure 2: (a) $\|x_0\|_1$ in the Algorithm 1, which can be obtained from `SolveSOCP`$(\mathcal{C}_0, b)$, for a random 8-qubit state. The ratio $|\mathcal{C}_0|/|\mathcal{A}_n|$ varies from 10^{-8} to 10^{-7}. We can get much better results with the top overlap heuristics compared to the random selection of $\mathcal{C}_0$. (b) The convergence of the CG method for the same state. The max violation becomes 1.00 after 9 iterations, which means the optimal solution is found.

"closeness" between the states $|\psi\rangle$ and $|\phi_j\rangle$. Hence, choosing the states based on their overlaps is a reasonable choice. The numerical experiments result in Figure 2 also support the effectiveness of this heuristic. In the case of a random pure 8-qubit state, even if we use as small subset as $|\mathcal{C}_0| = 10^{-7}|\mathcal{A}_n|$, the $\|x_0\|_1$ obtained closely approximates the exact value and outperforms randomly selected $\mathcal{C}_0$.

3.2.2 Optimality of the solution

The terminate criterion for Algorithm 1 is the absence of columns that violate the dual constraints $\left|a_j^\dagger y_k\right| \leq 1$, which can be checked efficiently by Theorem 3 as well. This means the optimal solution for the dual problem (3) is found, and the primal solution x_k is also optimal thank to the strong duality of the SOCP problem. Consequently, we can affirm that Algorithm 1 is certain to find the exact stabilizer extent for any n-qubit state $|\psi\rangle$ once it terminates. The convergence of the CG method is also confirmed in numerical experiments. For the same 8-qubit state as in the initialization, $\max_{a_j \in \mathcal{A}} \left|a_j^\dagger y_k\right|$ reaches 1.00 after 9 iterations, indicating the discovery of the optimal solution.

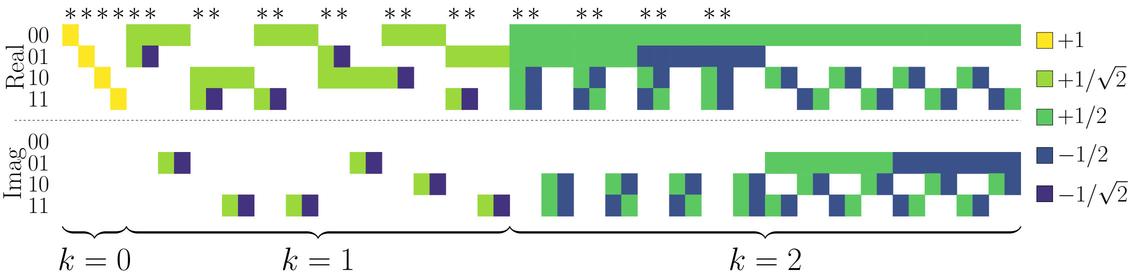

Figure 3: Visualization of the matrix A_n with $n = 2$. The upper half corresponds to the real part, and the lower half corresponds to the imaginary part. The j-th column of this represents the column a_j and its state $|\phi_j\rangle$. The k below the matrix corresponds to the standard form (4). By restricting the matrix A_n to the starred columns which are real vectors, we can obtain the matrix A'_n.

3.3 For the case $|\psi\rangle$ is Real

In some cases, the state $|\psi\rangle$ could be real. For example, todo (Is there any example?) . We show that in such cases the problem can be further simplified. We define the subset of the stabilizer states $\mathcal{S}_n$ that are real as $\mathcal{T}_n = \{|\phi_j\rangle \in \mathcal{S}_n \mid \langle i|\phi_j\rangle \in \mathbb{R} \text{ for all } i\}$, and denote the corresponding subset of the columns in $\mathcal{A}_n$ as $\mathcal{A}'_n$. Then, the next theorem holds.

Theorem 4. *Suppose that the state $|\psi\rangle$ is real. If we substitute the column set $\mathcal{A}_n$ with $\mathcal{A}'_n$ in the problem (2), the optimal solution of the restricted problem is also optimal for the original problem.*

Thanks to Theorem 4, we can reduce the size of the column set size by a factor of 2^n. numerical experiments result?

4 Approximation of the Stabilizer Extent

In the preceding sections, we proposed algorithms to calculate the exact stabilizer extent. However, these algorithms face a significant computational challenge when $n > 9$, since $|\mathcal{S}_n|$ grows superexponentially and we need at least $\mathcal{O}(|\mathcal{S}_n|)$ time complexity to assure the optimality of the solution. On the other hand, techniques, such as those discussed in this section, enable the discovery of columns which can improve the current solution when added to the column set $\mathcal{C}$. These methods can be relatively easily applied even for sufficiently large n. Therefore, it is possible to consider relaxed algorithms based on Algorithm 1 as a method for obtaining approximate value of stabilizer extent. In this section, we describe approximation algorithms for the stabilizer extent calculation.

4.1 Approximation of Fidelity

First, we considered about fidelity in Section 3.1, which can be computed by solving

$$\max_{a_j \in \mathcal{A}_n} \left| a_j^\dagger b \right|. \tag{7}$$

In stead of (7), we consider the following problem to find sufficiently large overlap:

$$\text{Find } a_j \in \mathcal{A}_n \text{ s.t. } \left| a_j^\dagger b \right| > 1. \tag{8}$$

Algorithm 2: Algorithm to find sufficiently large overlap

Input: vector $P \in \mathbb{C}^{2^n}$
Output: Solutions c, Q for (9)

for $k = 0, 1, 2, \dots$ **do**
  $c, Q \leftarrow$ Randomly sampled $c \in \mathbb{F}_2^n, Q \in \mathbb{F}_2^{n \times n}$
  while *True* **do**
    $c', Q' \leftarrow$ best c, Q in the neighborhood of c, Q
    if $\left|\sum_{x=0}^{2^n-1} (-1)^{x^\top Q' x} i^{c'^\top x} P_x\right| > \left|\sum_{x=0}^{2^n-1} (-1)^{x^\top Q x} i^{c^\top x} P_x\right|$ **then**
      $c, Q \leftarrow c', Q'$
    else
      break
  if $\left|\sum_{x=0}^{2^n-1} (-1)^{x^\top Q x} i^{c^\top x} P_x\right| > 1$ **then**
    return c, Q

To solve this, a method called hill climbing is effective. This method repeatedly performs local improvements to find the solutions for the problem (8). It's worth noting that a similar approach for maximizing overlap has been proposed in [15, Section 4.1]. Our approach, however, utilizes the form in (4). Again, setting $k = n, R = I_n, t = 0$, we restrict our consideration to

$$\text{Find } c \in \mathbb{F}_2^n, Q \in \mathbb{F}_2^{n \times n} \text{ s.t. } \left|\sum_{x=0}^{2^n-1} (-1)^{x^\top Q x} i^{c^\top x} P_x\right| > 1. \tag{9}$$

This can easily be generalized to other cases as well. The neighborhood of (c, Q) is defined as the set of (c', Q') that can be obtained by flipping a single bit of c or Q, so there are $n + \frac{n(n+1)}{2}$ neighbors in total. We present the pseudo code in Algorithm 2 to solve (9).

4.2 Numerical Experiments for the case $|\psi\rangle = |\phi_0\rangle^{\otimes n}$

Here, to demonstrate the effectiveness of this approximation method, we present computational results for the case $|\psi\rangle = |\phi_0\rangle^{\otimes n}$. By the multiplicativity of the stabilizer extent [4, Proposition 1], we have $\xi(|\psi\rangle) = \xi(|\phi_0\rangle)^n$ if $|\psi_0\rangle$ is at most 3-qubit state. In this section, $|\phi_0\rangle$ is a single qubit state. numerical experiments result here.

We can actually confirm that our approximate solution is close to the exact solution by comparing the results of the two methods. OR we fail to confirm it.

5 Discussion

In this paper, we have shown that the fidelity and stabilizer extent can be efficiently calculated by leveraging the specific structure of stabilizer states. We have proposed a novel algorithm based on the column generation method to compute the exact stabilizer extent, and demonstrated that it can be applied to sufficiently large systems. We have also proposed an approximation algorithm for more large systems.

There is still room for improvement in some specific cases. As for Robustness of Magic, there is a marvelous algorithm proposed in [16] which focuses on copies of symmetric pure magic states, and we enhanced this result in [11]. Applying such techniques to the stabilizer extent calculation would be promising and is left for future work.

Furthermore, there is some more future direction. For example, todo.

Acknowledgements.—

We would like to thank N. Marumo for valuable comments on the manuscript. N.Y. wishes to thank JST PRESTO No. JPMJPR2119 and the support from IBM Quantum. This work was supported by JST Grant Number JPMJPF2221. This work was supported by JST ERATO Grant Number JPMJER2302 and JST CREST Grant Number JPMJCR23I4, Japan.

References

[1] SciPy. "scipy.sparse.csc_matrix". link.

[2] Scott Aaronson and Daniel Gottesman. "Improved simulation of stabilizer circuits". Physical Review A: Atomic, Molecular, and Optical Physics **70**, 052328 (2004).

[3] Mark Howard and Earl Campbell. "Application of a resource theory for magic states to fault-tolerant quantum computing". Physical Review Letters **118**, 090501 (2017).

[4] Sergey Bravyi, Dan Browne, Padraic Calpin, Earl Campbell, David Gosset, and Mark Howard. "Simulation of quantum circuits by low-rank stabilizer decompositions". Quantum **3**, 181 (2019).

[5] Arne Heimendahl, Felipe Montealegre-Mora, Frank Vallentin, and David Gross. "Stabilizer extent is not multiplicative". Quantum **5**, 400 (2021).

[6] Stephen Boyd and Lieven Vandenberghe. "Convex optimization". Cambridge University Press. Cambridge (2004).

[7] MOSEK ApS. "Mosek modeling cookbook". link (2019).

[8] Steven Diamond and Stephen Boyd. "CVXPY: A python-embedded modeling language for convex optimization". Journal of Machine Learning Research **17**, 2909–2913 (2016).

[9] Akshay Agrawal, Robin Verschueren, Steven Diamond, and Stephen Boyd. "A rewriting system for convex optimization problems". Journal of Control and Decision **5**, 42–60 (2018).

[10] Guy Desaulniers, Jacques Desrosiers, and Marius M. Solomon Solomon, editors. "Column Generation". Springer US. (2005).

[11] Hiroki Hamaguchi, Kou Hamada, and Nobuyuki Yoshioka. "Handbook for Efficiently Quantifying Robustness of Magic" (2023). arxiv:2311.01362.

[12] G.I. Struchalin, Ya. A. Zagorovskii, E.V. Kovlakov, S.S. Straupe, and S.P. Kulik. "Experimental Estimation of Quantum State Properties from Classical Shadows". PRX Quantum **2**, 010307 (2021).

[13] Maarten Van den Nest. "Classical simulation of quantum computation, the gottesman-Knill theorem, and slightly beyond". Quantum Inf. Comput. **10**, 258–271 (2010).

[14] Jeroen Dehaene and Bart De Moor. "Clifford group, stabilizer states, and linear and quadratic operations over GF(2)". Physical Review A **68**, 042318 (2003).

[15] Héctor J. García, Igor L. Markov, and Andrew W. Cross. "On the geometry of stabilizer states". Quantum Info. Comput. **14**, 683–720 (2014).

[16] Markus Heinrich and David Gross. "Robustness of Magic and Symmetries of the Stabiliser Polytope". Quantum **3**, 132 (2019).

[17] Hiroki Hamaguchi, Kou Hamada, and Nobuyuki Yoshioka. "stabilizer_extent". link (2024).

A Fast Algorithm for Overlap

In this section, we will explain the detail of DFS algorithm in Section 3.1 and introduce some heuristics to improve the efficiency.

A.1 Efficient Enumeration of Stabilizer States

In this section, we prove the Theorem 1.

Theorem 1. *The form* (4) *under the following conditions enumerates all stabilizer states without any duplication or omission.*

- *Q is a upper triangular $\mathbb{F}_2^{k\times k}$ matrix.*
- *R is a $\mathbb{F}_2^{n\times k}$ rref (reduced row echelon form) matrix satisfies* $\mathrm{rank}(R) = k$*.*
- *t belongs to the complement of the row space of R.*

Proof. Main Ideas come from [12]. The assertion is trivial for $k = 0$. We will only consider the case $k > 0$. Define $f : (Q, c, R, t) \mapsto |\phi\rangle$ as the mapping from (Q, c, R, t) to the corresponding state $|\phi\rangle$. Firstly, we show that f is injective. We can say that

$$\begin{aligned}&\left\{R_1 x + t_1 \,\middle|\, x \in \mathbb{F}_2^{n-k}\right\} = \left\{R_2 x + t_2 \,\middle|\, x \in \mathbb{F}_2^{n-k}\right\}\\ \iff\ &\mathrm{Im}(R_1) = \mathrm{Im}(R_2) \land (t_2 - t_1) \in \mathrm{Im}(R_1)\\ \iff\ &R_1 = R_2 \land t_1 = t_2.\end{aligned}$$

The last equivalence is due to the property of the rref matrix and the complement condition. Given that Q is an upper triangular matrix, both Q and c can be uniquely reconstructed from the coefficients of the state $|\phi\rangle$. Consequently, for any $|\phi\rangle$, the values of (Q, c, R, t) can be uniquely determined, establishing injectivity in the mapping f.

Next, we show that f is surjective. Since f is injective, we only have to show that the cardinality of the domain is equal to that of the codomain, i.e., $-2^n + |\mathcal{S}_n|$. It is known that the number of $\mathbb{F}_2^{n\times k}$ rref matrices with $\mathrm{rank}(R) = k$ is $\begin{bmatrix} n \\ k \end{bmatrix}_2$, which is a q-binomial coefficient with $q = 2$. Therefore, the number of Q, c, R, t is $2^{k(k+1)/2}, 2^k, \begin{bmatrix} n \\ k \end{bmatrix}_2, 2^{n-k}$, respectively, and the total number of states is

$$\sum_{k=1}^{n} 2^{k(k+1)/2} 2^k \begin{bmatrix} n \\ k \end{bmatrix}_2 2^{n-k} = -2^n + 2^n \sum_{k=0}^{n} \begin{bmatrix} n \\ k \end{bmatrix}_2 2^{k(k+1)/2} = -2^n + 2^n \prod_{k=1}^{n} (2^k + 1) = -2^n + |\mathcal{S}_n|.$$

In the second last equation, we used the q-binomial theorem. Therefore, the mapping is surjective, which concludes the proof. □

In Theorem 1, we used $\mathbb{F}_2$. However, from the perspective of the DFS algorithm, it is more practical to use $\{0, 1\} \subset \mathbb{Z}$ and permit the term $c^\top x$ to be any integer value. Otherwise, $-1 = i^{1+1} \neq i^0 = 1$ although $1 + 1 = 0$ in $\mathbb{F}_2$, which makes the algorithm more complicated. Hence, the subsequent corollary is valuable.

Corollary 1. *In Theorem 1, We can substitute* $\mathbb{F}_2$ *with* $\{0,1\} \subset \mathbb{Z}$.

Proof. By changing $\mathbb{F}_2$ to $\{0,1\} \subset \mathbb{Z}$, the term $(-1)^{x^\top Qx}$ is invariant, and the term $i^{c^\top x}$ is multiplied by -1 iff $p \equiv 2,3 \pmod 4$, where p is the number of i such that $c_i = 1$ and $x_i = 1$. Now, we consider the following form:

$$\begin{cases} |\phi\rangle := |t\rangle & \text{if } k = 0, \\ |\phi\rangle := \dfrac{1}{2^{k/2}} \displaystyle\sum_{x=0}^{2^k-1} (-1)^{x^\top (Q+Q')x} i^{c^\top x} |Rx + t\rangle & \text{if } k > 0, \end{cases} \tag{10}$$

where $Q \in \{0,1\}^{k\times k}, c \in \{0,1\}^k, R \in \{0,1\}^{n\times k}, t \in \{0,1\}^n, \operatorname{rank}(R) = k$ and $Q'_{ij} = 1$ iff $(i < j) \wedge (c_i = c_j = 1)$. Now, if the pair (Q, c, R, t) in (10) is the same as that of the original form (4), then the two states are representing the exactly same state since

$$(-1)^{x^\top Q' x} = (-1)^{\binom{p}{2}} = \begin{cases} 1 & \text{if } p \equiv 0,1 \pmod 4, \\ -1 & \text{if } p \equiv 2,3 \pmod 4. \end{cases}$$

Therefore, by identifying the $Q+Q'$ in $\mathbb{Z}$ with new Q'' in $\mathbb{F}_2$, we can conclude the proof. $\square$

A.2 Calculating the Overlap

In this section, we prove the Theorem 2. Be aware that the problem (6) is equivalent to the following problem thank to the corollary 1:

$$\max_{c\in\{0,1\}^n, Q\in\{0,1\}^{n\times n}} \left\{ \left| \sum_{x=0}^{2^n-1} (-1)^{x^\top Qx} i^{c^\top x} P_x \right| \right\}.$$

Theorem 2. *The problem (6) can be solved in* $\mathcal{O}\left(2^{n+n(n+1)/2}\right)$ *time complexity and* $\mathcal{O}(2^n)$ *space complexity.*

Proof. Define $x := \begin{bmatrix} x_0 \\ \overline{x} \end{bmatrix}$ $(x_0 \in \{0,1\}, \overline{x} \in \{0,1\}^{n-1})$, $c := \begin{bmatrix} c_0 \\ \overline{c} \end{bmatrix}$ $(c_0 \in \{0,1\}, \overline{c} \in \{0,1\}^{n-1})$, and $Q := \begin{bmatrix} Q_{00} & Q_0^\top \\ 0 & \overline{Q} \end{bmatrix}$ $(Q_{00} \in \{0,1\}, Q_0 \in \{0,1\}^{n-1}, \overline{Q} \in \{0,1\}^{(n-1)\times(n-1)})$. Since $x^\top Qx = x_0(Q_{00} + Q_0^\top \overline{x}) + \overline{x}^\top \overline{Q}\,\overline{x}$ and $c^\top x = c_0 x_0 + \overline{c}^\top \overline{x}$, we can derive that

$$\begin{aligned} \sum_{x=0}^{2^n-1} (-1)^{x^\top Qx} i^{c^\top x} P_x &= \sum_{\overline{x}=0}^{2^{n-1}-1} (-1)^{\overline{x}^\top \overline{Q}\,\overline{x}} i^{\overline{c}^\top \overline{x}} \left(P_{2\overline{x}} + (-1)^{Q_{00} + Q_0^\top \overline{x}} i^{c_0} P_{2\overline{x}+1} \right) \\ &= \sum_{\overline{x}=0}^{2^{n-1}-1} (-1)^{\overline{x}^\top \overline{Q}\,\overline{x}} i^{\overline{c}^\top \overline{x}} P'_x \end{aligned} \tag{11}$$

where we identify a vector $\begin{bmatrix} x_0 & x_1 & \cdots & x_{n-1} \end{bmatrix}^\top$ as a integer $\sum_{i=0}^{n-1} x_i 2^i$, and define $P'_x := P_{2\overline{x}} + (-1)^{Q_{00} + Q_0^\top \overline{x}} i^{c_0} P_{2\overline{x}+1}$. Since (11) is the same form as the original problem, this problem can be solved recursively by fixing the value c_0, Q_{00} and Q_0.

We now analyze the time complexity of this recursive algorithm. Considering each possible combination of c_0, Q_{00}, and Q_0, there are 2^{n+1} options. For each such combination, P'_x can be computed in $\mathcal{O}(n2^{n-1})$ time. Hence, we establish the following recurrence relation for the time complexity $T(n)$:

$$T(n) = 2^{n+1}(T(n-1) + n2^{n-1}), \quad T(1) = 4.$$

Solving this recurrence relation yields

$$T(n) = 2^{n+\frac{n(n+1)}{2}} + \sum_{d=2}^{n} 2^{n+\frac{n(n+1)}{2}-\frac{d(d-1)}{2}} d$$

$$\frac{T(n)}{2^{n+\frac{n(n+1)}{2}}} = 1 + \sum_{d=2}^{n} 2^{-\frac{d(d-1)}{2}} d \le 1 + \sum_{d=2}^{n} 2^{-d+1} d \le 4 - (n+2)2^{-n+1} \to 4.$$

Hence, the time complexity is $\mathcal{O}\Big(2^{n+n(n+1)/2}\Big)$. □

While the algorithm and proof presented above may seem somewhat rough, our actual implementation is significantly more precise and efficient. You can access it at GitHub [17]. Moreover, we can enhance efficiency further by employing branch cut heuristics, as we will explain in the next section.

A.3 Branch Cut For The DFS

In the previous section, we explained the efficient algorithm for the overlap calculation. However, this algorithm can be much more faster by using the branch cut heuristics we will introduce in this section.

Firstly, please recall that we are maximizing the following:

$$\max_{c,Q} \left\{ \left| \sum_{x=0}^{2^n-1} (-1)^{x^\top Qx} i^{c^\top x} P_x \right| \right\}$$

This can be easily bounded by

$$\max_{c,Q} \left\{ \left| \sum_{x=0}^{2^n-1} (-1)^{x^\top Qx} i^{c^\top x} P_x \right| \right\} \le \max_{c,Q} \left\{ \sum_{x=0}^{2^n-1} \left| (-1)^{x^\top Qx} i^{c^\top x} P_x \right| \right\} = \sum_{x=0}^{2^n-1} |P_x|$$

Such a bound is important for the branch cut heuristics, because it allows us to terminate the branch if the current value is inferior to the bound. However, this bound can be more refined. Since each coefficient takes only $1, -1, i$ or $-i$, we can obtain

$$\max_{c,Q} \left\{ \left| \sum_{x=0}^{2^n-1} (-1)^{x^\top Qx} i^{c^\top x} P_x \right| \right\} \le \max_{c,Q} \left\{ \left| \sum_{x=0}^{2^n-1} i^{c_x} P_x \right| \right\} \tag{12}$$

where $c_x \in \{0,1,2,3\}$ is the independent variable for each x. Let $P^* = \sum_{x=0}^{2^n-1} i^{c_x^*} P_x$ be one of the optimal solutions of (12), i.e., $|P^*| = \max_{c,Q} \left\{ \left| \sum_{x=0}^{2^n-1} i^{c_x} P_x \right| \right\}$. Then, without loss of generality, we can assume that $\frac{\pi}{2} \le \arg P^* < \frac{3\pi}{2}$, and by sorting and multiplying $i, -1$ or $-i$ to P_x appropriately, we can also assume that

$$0 \le \arg(P_0) \le \arg(P_1) \le \cdots \le \arg(P_{2^n-1}) < \pi/2. \tag{13}$$

It is evident that all c_x satisfies $\arg(P^*) - \pi/4 \le \arg\left(i^{c_x} P_x\right) < \arg(P^*) + \pi/4$. Otherwise, adjusting c_x would yield a larger value. Moreover, if $\arg(P^*)$ is fixed within the range $[\pi/2, 3\pi/2)$, such c_x values yield the maximum value. Thus, Algorithm 3 is validated, as it covers all possible cases regardless of the specific value of $\arg(P^*) \in [\pi/2, 3\pi/2)$. Refer to Figure 4 for a visual representation of this algorithm. The time complexity of this approach is $\mathcal{O}(n2^n)$ owing to the sorting of 2^n elements.

Algorithm 3: Branch Cut Algorithm

Input: Coefficients P_x for $x = 0, 1, \dots, 2^n - 1$

Output: The answer for the problem (12)

Sort and modify the coefficients P_x so that the condition (13) is satisfied

ans $\leftarrow 0$, $\quad c_x \leftarrow 0 \quad$ for all x

for $x \leftarrow 0$ **to** $2^n - 1$ **do**

  ans $\leftarrow \max\left(\text{ans}, \left|\sum_{x=0}^{2^n-1} i^{c_x} P_x\right|\right)$

  $c_x \leftarrow c_x + 1$

return ans

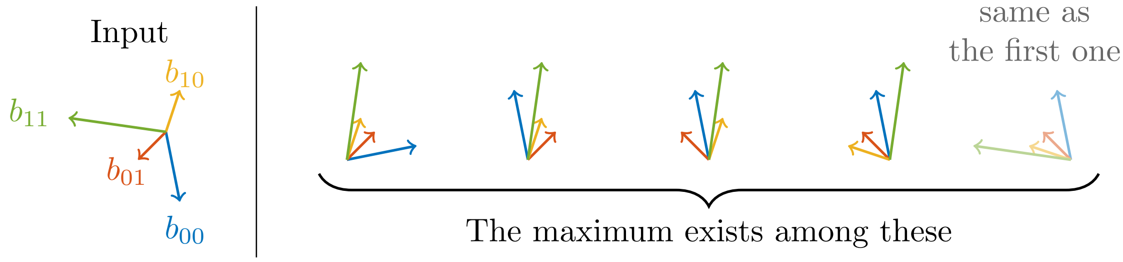

Figure 4: Visualization of Algorithm 3. Suppose that $n = 2$ and P_x are represented as the vectors in the complex plane (e.g., $P_{00} = 1 - 5i$) in the left figure. Sorting and Iterating the loop in Algorithm 3 yields 2^n patterns of the coefficients c_x, as depicted in the right figure. The maximum for the problem (12) exists among these 2^n patterns.

B The proof of Theorem 4

In this section, we prove the Theorem 4. The proof is based on the following lemma.

Lemma 1. *Suppose y is a real vector and satisfies $\left|a^\dagger y\right| \le 1$ for all $a \in \mathcal{A}'_n$ such that a is a real vector. Then, y satisfies $\left|a^\dagger y\right| \le 1$ for all $a \in \mathcal{A}_n$.*

Proof. Fix $a \in \mathcal{A}$ and suppose that a represents the state $|\phi\rangle$. Now, consider to write $|\phi\rangle$ as in the form (4). The case $k = 0$ is trivial since then $a \in \mathcal{A}'_n$. Suppose that $|\phi_i\rangle$ can be written as $\frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} (-1)^{x^\top Q x} i^{c^\top x} |Rx + t\rangle$ with $k > 0$ and $a^\dagger y = \alpha + i\beta \quad (\alpha, \beta \in \mathbb{R})$. The following two states

$$|\phi_+\rangle := \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} (-1)^{x^\top Q x} |Rx + t\rangle, \quad |\phi_-\rangle := \frac{1}{2^{k/2}} \sum_{x=0}^{2^k-1} (-1)^{x^\top Q x + c^\top x} |Rx + t\rangle$$

belongs to $\mathcal{A}'_n$, and denote the column vectors of $|\phi_+\rangle$ and $|\phi_-\rangle$ as a_+ and a_-, respectively. Then, we have $a_+^\dagger y = \alpha + \beta, a_-^\dagger y = \alpha - \beta$ from the assumption, and

$$\left|a^\dagger y\right| = \sqrt{\alpha^2 + \beta^2} \le |\alpha| + |\beta| = \max\{|\alpha + \beta|, |\alpha - \beta|\} \le 1,$$

which completes the proof. □

Now, we are ready to prove the Theorem 4.

Theorem 4. *Suppose that the state $|\psi\rangle$ is real. If we substitute the column set $\mathcal{A}_n$ with $\mathcal{A}'_n$ in the problem (2), the optimal solution of the restricted problem is also optimal for the original problem.*

Proof. Let x^* and y^* be the optimal solutions of the restricted primal and dual problems, namely, the problem (2) and the problem (3) with the column set $\mathcal{A}'_n$ instead of $\mathcal{A}_n$. We can assure such solutions always exists. Now, we show that the x^*, y^* are optimal not only for the restricted problems but also for the original problems.

Let OPT be the optimal value for the original problem. Since x^* can be a feasible solution for the original primal problem, it is clear that $\mathrm{OPT} \leq \|x^*\|_1$. By the strong duality theorem, OPT is also the optimal value for the original dual problem. From the Lemma 1, we can see that y^* is a feasible solution for the original dual problem and $\mathrm{OPT} \geq \mathrm{Re}(b^\dagger y^*)$. Again, by applying the strong duality theorem to the restricted problems, we have $\|x^*\|_1 = \mathrm{Re}(b^\dagger y^*)$, which means that $\mathrm{OPT} = \|x^*\|_1 = \mathrm{Re}(b^\dagger y^*)$. Therefore, x^* and y^* are also optimal solutions for the original problems. $\square$