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

2 Preliminaries

We denote the entire set of n-qubit stabilizer states as $\mathcal{S}_n := \{|\phi_\alpha\rangle\}$. The size of $\mathcal{S}_n$ scales superexponentially as $|\mathcal{S}_n| = 2^n \prod_{k=0}^{n-1}(2^{n-k}+1) = 2^{\mathcal{O}(n^2)}$ [1, Proposition 1]. Suppose ψ is a normalized n-qubit state. The *stabilizer extent* is introduced in [2, Definition 3] to quantify the state ψ, and is defined as follows:

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

This definition can be simplified as complex L^1-norm minimization problem:

$$\sqrt{\xi(\psi)} = \min_{x \in \mathbb{C}^{|\mathcal{S}_n|}} \left\{ \sum_{\alpha=1}^{|\mathcal{S}_n|} x_\alpha \;\middle|\; A_n x = b \right\}, \tag{2}$$

Here, we define $A_n \in \mathbb{C}^{2^n \times |\mathcal{S}_n|}$ as $(A_n)_{ij} := \langle i|\phi_j\rangle$ and $b \in \mathbb{C}^{|\mathcal{S}_n|}$ as $b_i := \langle i|\psi\rangle$ using the computational basis $\{|i\rangle\}_{i=0}^{2^n-1}$. As in [3], the problem (2) is a second order cone program (SOCP). Thus, its dual problem can be derived [3, Appendix A][4, Section 5.1.6]. By defining $\mathcal{A}_n$ as the set of columns $\{a_j\}$ of A_n,

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

where $\dagger$ denotes the conjugate transpose. Although the actual objective function in (3) should be multiplied by -1, we flipped the sign for simplicity, which does not affect the solution.

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

Algorithm 1: Exact stabilizer extent calculation by Column Generation

Input: vector b corresponding to the state ψ

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

$\mathcal{C} \leftarrow$ Partial set of $\mathcal{A}_n$ `/* Initialize using top overlaps */`

while *true* **do**

  $E, \boldsymbol{y} \leftarrow \texttt{SolveSOCP}(\mathcal{C}, \boldsymbol{b})$

  $\mathcal{C}' \leftarrow \left\{ \boldsymbol{a} \in \mathcal{A}_n \;\middle|\; \left|\boldsymbol{a}^\dagger \boldsymbol{y}\right| > 1 \right\}$ `/* Use of subroutine in Section 3.2 */`

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

    break

  $\mathcal{C} \leftarrow \mathcal{C} \cup \mathcal{C}'$

return E

In order to describe our algorithm in later sections, we denote a function SolveSOCP$(\mathcal{C}, b)$ which takes a set of columns $\mathcal{C}$ and a vector b, and returns the optimal solution E and dual optimal solution $\boldsymbol{y}$ of the SOCP problem (3). In actual numerical computation, this function can be realized by just solving the corresponding primal problem (2).

3 Scaling up The Exact Stabilizer Extent Calculation

Although SOCP are solvable in polynomial time with respect to the matrix size, due to the superexponential growth of the matrix size $|\mathcal{S}_n|$, using the entire A_n is totally impractical for $n > 5$ [?]. In order to tackle this kind of problem, we have proposed to utilize the column generation (CG) method in [5]. The CG method is a classical technique to solve large-scale optimization problems. Here, we show that similar method actually works for the stabilizer extent calculation too.

3.1 CG method

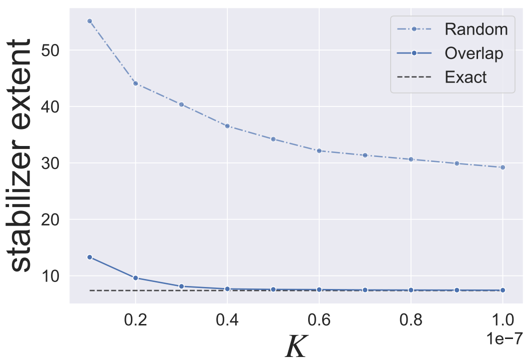

Figure 1: Calculated stabilizer extent for random pure state of 7 qubits under a restricted set of stabilizer states. The ratio $K := |\mathcal{C}|/|\mathcal{A}_n|$ varies from 10^{-8} to 10^{-7}. By selecting the stabilizer states with the top overlaps, we get much better approximation values compared to the randomly selected ones.

3.2 Core Subroutine: Calculating Overlap

As a well-known fact, the stabilizer states have a simple form as shown in the following proposition.

Proposition 1 ([6, Theorem 2], [7, Section 5], [8, Theorem 5.(ii)]). *All stabilizer states can be written in the following form:*

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

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

Theorem 1. *Complexity of computing all stabilizer overlaps Computation of* $A_n^\dagger y$ *can be done in time complexity of* $\mathcal{O}(n|S_n|)$ *and space complexity of* $\mathcal{O}(2^n)$.

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

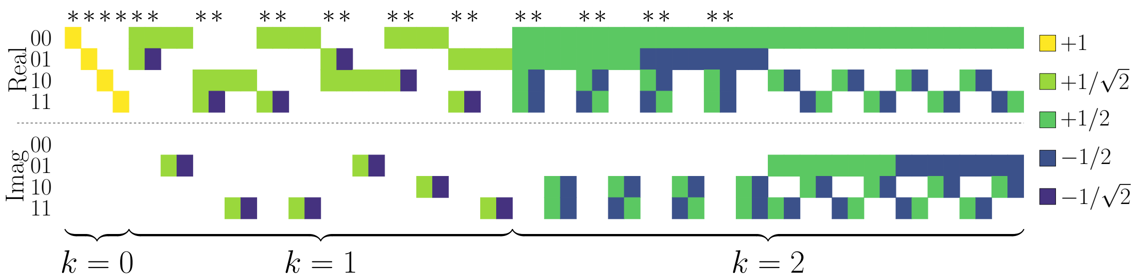

Figure 2: Visualization of the matrix A_n with $n = 2$. The upper half corresponds to the real part, and the lower half corresponds to the imaginary part. The j-th column of this represents the column a_j and its state $|\phi_j\rangle$. The k below the matrix corresponds to the standard form (4). By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

In some cases, the state $|\psi\rangle$ could be real. For example, ... We show that in such cases the problem can be further simplified. We define the subset of the states in $\mathcal{S}_n$ with real coefficients as $\mathcal{T}_n$, and the corresponding subset of the columns in $\mathcal{A}_n$ as $\mathcal{A}'_n$. Then, the next theorem holds.

Theorem 2. *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 this theorem, we can reduce the size of the column set size by a factor of 2^n.

4 Approx Solutions

5 Discussion

In this paper, we have shown that todo.

There is still room for improvement in some specific cases. As for Robustness of Magic, there is a marvelous algorithm proposed in [**?**] which focuses on copies of symmetric pure magic states, and we enhanced this result in [5]. 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] Scott Aaronson and Daniel Gottesman. "Improved simulation of stabilizer circuits". Physical Review A: Atomic, Molecular, and Optical Physics **70**, 052328 (2004).

[2] 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).

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

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

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

[6] 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).

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

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

A Fast Algorithm for Overlap

As we stated in Theorem [**?**], we can calculate the all overlap of the stabilizer states efficiently. In this section, we will explain the algorithm in detail and introduce some heuristics to improve the efficiency.

A.1 Efficient Enumeration of Stabilizer States

As we stated in the proposition 1, there are the form of the stabilizer states. Here, we will show a more efficient way to enumerate all the stabilizer states by modifying the form slightly.

Theorem 3. *The form (4) with the following conditions enumerates all the stabilizer states without any duplication or omission:*

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

Proof. Main Ideas come from [6]. Firstly, we show that the mapping $\{(Q,c,R,t)\}\to\mathcal{S}_n$ is injective. We can say that

$$\begin{aligned}&\left\{R_1x+t_1 \;\middle|\; x\in\mathbb{F}_2^{n-k}\right\}=\left\{R_2x+t_2 \;\middle|\; x\in\mathbb{F}_2^{n-k}\right\}\\ \iff&\operatorname{Im}(R_1)=\operatorname{Im}(R_2)\wedge(t_2-t_1)\in\operatorname{Im}(R_1)\\ \iff&R_1=R_2\wedge t_1=t_2.\end{aligned}$$

The last equivalence is due to the property of the rref matrix and the complement condition. Since Q is a upper triangular matrix, we can uniquely determine Q and c for given state $|\phi\rangle$. Thus, if two states are the same, then the corresponding (Q,c,R,t) are also the same, which means that the mapping is injective.

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

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

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

In theorem 3, we used $\mathbb{F}_2$. However, in the viewpoint of the dfs algorithm, it is more practical to use $\{0,1\}\subset\mathbb{Z}$ and allow the term $c^\top x$ to be any integer. Therefore, the following corollary is useful.

Corollary 1. *In theorem 3, 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:

$$\begin{cases}|\phi\rangle := |t\rangle & \text{if } k=0,\\ |\phi\rangle := \dfrac{1}{2^{k/2}}\displaystyle\sum_{x=0}^{2^k-1}(-1)^{x^\top(Q+Q')x}i^{c^\top x}\,|Rx+t\rangle & \text{if } k>0,\end{cases}\tag{5}$$

where $Q\in\{0,1\}^{k\times k},c\in\{0,1\}^k,R\in\{0,1\}^{k\times(n-k)},t\in\{0,1\}^{n-k},\operatorname{rank}(R)=k$ and $Q'_{ij}=1$ iff $(i<j)\wedge(c_i=c_j=1)$. Now, if the pair (Q,c,R,t) in (5) 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. $\square$

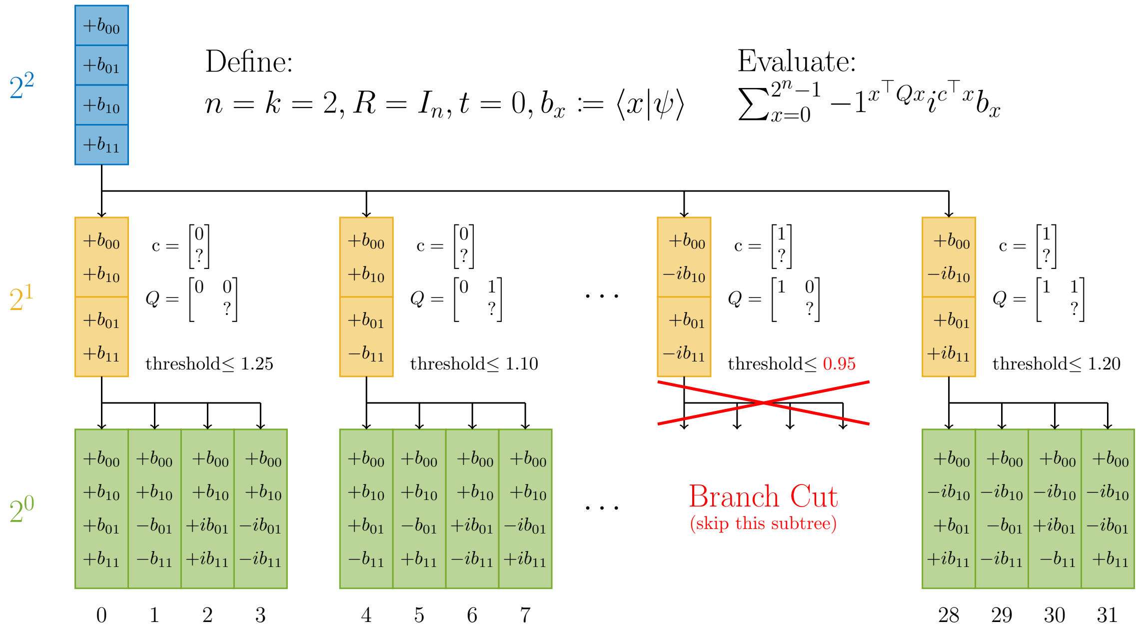

Figure 3: The visualization of the DFS algorithm. The DFS algorithm is a recursive algorithm that calculates the overlap of the stabilizer states by recursively calculating the overlap of the substates. The algorithm is efficient since the overlap of the substates can be calculated with very small computational cost.

A.2 Calculating the Overlap

Thanks to the corollary 1, we can prove the following theorem.

Theorem 4. *Fix k, R, t in the standard form (4). Then, we can compute the overlap $\langle\phi|\psi\rangle$ efficiently.*

Proof. We only consider the case $k > 0, R = 0, t = 0$ for the simplicity. Other cases are trivial or can be reduced to this case. Define $x := \begin{bmatrix} x_0 \\ \overline{x} \end{bmatrix}$, $c := \begin{bmatrix} c_0 \\ \overline{c} \end{bmatrix}$, and $Q := \begin{bmatrix} Q_{00} & Q_0^\top \\ 0 & \overline{Q} \end{bmatrix}$ (x_0, c_0 and Q_{00} are all in $\{0, 1\}$). 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 rewrite the state as

$$
\begin{aligned}
|\phi\rangle &= \sum_{x=0}^{2^k-1} (-1)^{x^\top Qx} i^{c^\top x} |x\rangle \\
&= \sum_{\overline{x}=0}^{2^{k-1}-1} (-1)^{\overline{x}^\top \overline{Q}\overline{x}} i^{\overline{c}^\top \overline{x}} \Big(|2\overline{x}\rangle + (-1)^{Q_{00} + Q_0^\top \overline{x}} i^{c_0} |2\overline{x} + 1\rangle \Big) \\
&= \sum_{\overline{x}=0}^{2^{k-1}-1} (-1)^{\overline{x}^\top \overline{Q}\overline{x}} i^{\overline{c}^\top \overline{x}} |\overline{x}'\rangle
\end{aligned}
$$

by defining $|\overline{x}'\rangle := |2\overline{x}\rangle + (-1)^{Q_{00} + Q_0^\top \overline{x}} i^{c_0} |2\overline{x} + 1\rangle$. (Question: Is it natural to equate integer $2\overline{x} + 1$ to the vector $\begin{bmatrix} 1 \\ \overline{x} \end{bmatrix}$?)

Thus, we can compute the overlap recursively with very small computational cost per each step. This leads to the efficient calculation of the overlaps, which concludes the proof. $\square$

A.3 Branch Cut For The DFS

In this section, we will explain the branch cut methods used in the dfs search. 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 value can easily evaluate as the following:

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

However, since each coefficient takes only $1, -1, i$ or $-i$, we can obtain more tight bound by

$$\max_{c,Q}\left\{\left|\sum_{x=0}^{2^n-1} -1^{x^\top Qx} i^{c^\top x} P_x\right|\right\} \le \max_{c,Q}\left\{\left|\sum_{x=0}^{2^n-1} i^{c_x} P_x\right|\right\} \tag{6}$$

where $c_x \in \{0,1,2,3\}$ is the independent variable for each x. Let $\mathcal{P} := \sum_{x=0}^{2^n-1} i^{c_x^*} P_x$ be the one of optimal solutions for the problem (6). Then, without loss of generality, we can assume that $\frac{\pi}{2} \le \arg \mathcal{P} < \frac{3\pi}{2}$, and by sorting and multiplying $i, -1$ or $-i$ to P_x appropriately, we can also assume that

$$0 \le \arg(P_0) \le \arg(P_1) \le \cdots \le \arg(P_{2^n-1}) < \pi/2. \tag{7}$$

If all c_x satisfies $\arg(\mathcal{P}) - \pi/4 \le \arg(i^{c_x} P_x) < \arg(\mathcal{P}) + \pi/4$, then c_x is optimal for $\mathcal{P}$. Therefore, we can justify the following Algorithm 2 by moving $\arg(\mathcal{P})$ in the range of $[\pi/2, 3\pi/2)$. Also refer to the figure 4 for the visualization of this algorithm. The time complexity of this algorithm is $O(n2^n)$ due to the sorting of 2^n elements.

Algorithm 2: Branch Cut Algorithm

```
Input: Coefficients P_x for x = 0, 1, ..., 2^n − 1
Output: The answer for the problem (6)
Sort and modify the coefficients P_x so that the condition (7) is satisfied
ans ← 0,   c_x ← 0   for all x
for x ← 0 to 2^n − 1 do
    ans ← max(ans, |Σ_{x=0}^{2^n−1} i^{c_x} P_x|)
    c_x ← c_x + 1
return ans
```

B The proof of Theorem 2

In this section, we prove Theorem 2. 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 $a \in \mathcal{A}_n$.*

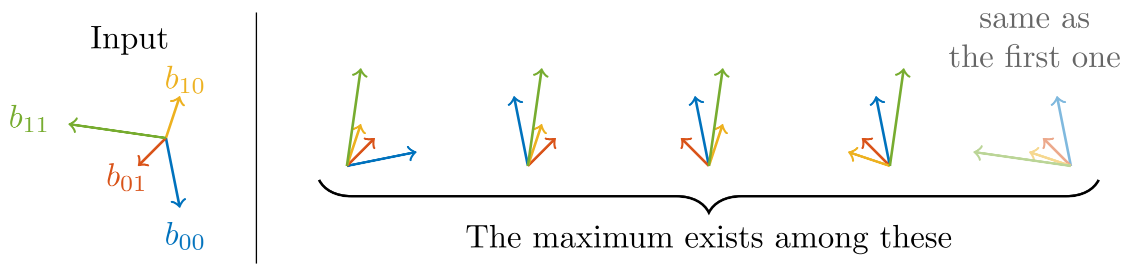

Figure 4: The visualization of Algorithm 2. Suppose that $n=2$ and P_x are represented as the vectors in the complex plane (for example, $P_{00}=1-5i$) by the left figure. Then, by sorting and iterating the loop in Algorithm 2, we can obtain 2^n patterns of the coefficients c_x as the right figure. The maximum of for the problem (6) exists among these 2^n patterns.

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 $\frac{1}{2^{k/2}}\sum_{x=0}^{2^k-1}(-1)^{x^\top Qx} i^{c^\top x}\,|Rx+t\rangle$ with $k>0$ and $a^\dagger y = \alpha + i\beta \quad (\alpha,\beta \in \mathbb{R})$. The following two states

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

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 Theorem 2.

Theorem 2. *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 OPT $\le \|x^*\|_1$. 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 OPT $\ge \mathrm{Re}\left\{b^\dagger y^*\right\}$. Again, by applying the strong duality theorem to the restricted problems, we have $\|x^*\|_1 = \mathrm{Re}\left\{b^\dagger y^*\right\}$, which means that OPT $= \|x^*\|_1 = \mathrm{Re}\left\{b^\dagger y^*\right\}$. Therefore, x^* and y^* are also optimal solutions for the original problems. □