Fast computation of magic monotones

Hiroki Hamaguchi [1] Kou Hamada [1] Nobuyuki Yoshioka [2]

[1]Department of Mathematical Informatics, University of Tokyo
[2]Department of Applied Physics, University of Tokyo

Robustness of Magic Stabilizer Extent

1. Summary

Motivation: Computation of Magic Monotones

- **Magic monotones** quantify cost of "hard" operations in FTQC
- Application: classical simulation and synthesis of Clifford+T circuit
- Problem: exact evaluation tends to be **very hard**.
- cf. size of n-qubit stabilizer states $|\mathcal{S}_n| = 2^{\mathcal{O}(n^2)}$; naively only $n \leq 5$.

Contribution: Scale-up of Computation

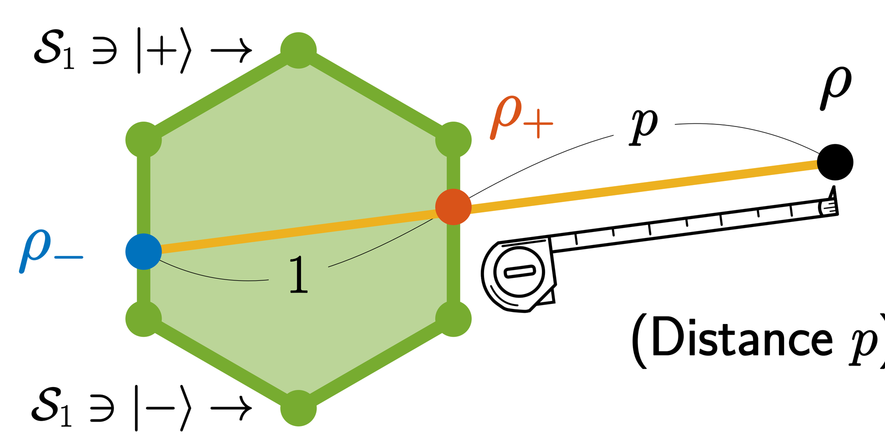
- **Speed up** magic monotone computation $\times 50 \sim \times 300$ faster for $n = 5$
- **Scale up** to $n = 8, 9$ qubits (naive method requires $> 10^{17}$ bytes)
- Our method: **Column Generation** + **Fast computation of overlaps**

2. Magic Monotones

- Interpreted as distance p from set of free states $\mathcal{S}_n$.
- Requirements: faithful, submultiplicative, monotonic, convex.

"Free" states: stabilizer states $\mathcal{S}_n$

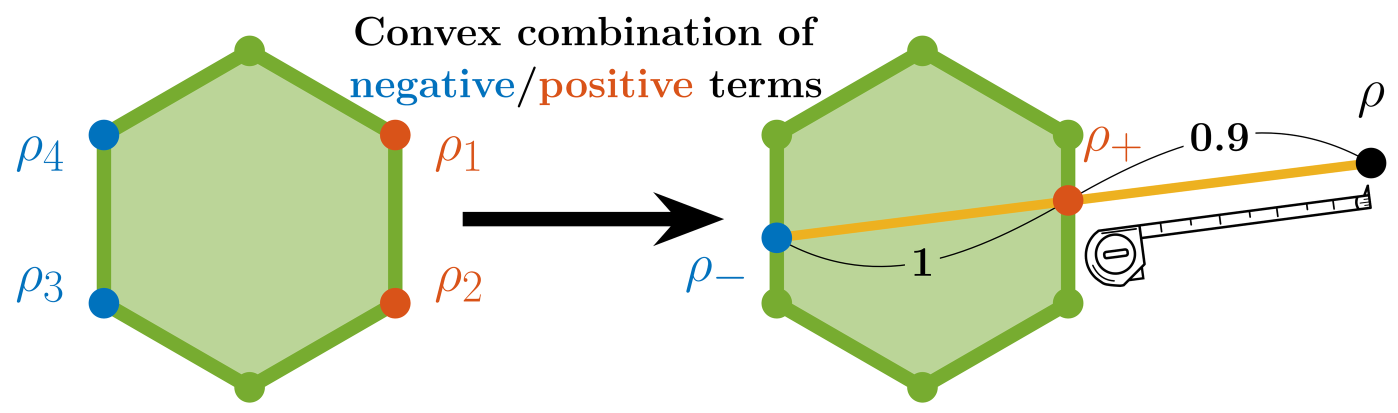
State inside can be readily prepared in FTQC



$$\rho = (1+p)\rho_+ - p\rho_-$$
$$\mathcal{R}(\rho) := 2p + 1$$

(Distance p) = cost to prepare ρ

- Steps for evaluating magic monotones:
  1. Represent the target state as the linear combination of stabilizer states
  2. Minimize the L^1-norm of the linear combination coefficients
- Robustness of Magic (RoM; Howard & Campbell, 2017):
  $$\mathcal{R}(\rho) = \min_{x \in \mathbb{R}^{|\mathcal{S}_n|}} \left\{ \|x\|_1 \;\middle|\; \rho = \textstyle\sum_{\sigma_j \in \mathcal{S}_n} x_j \sigma_j \right\}$$
- Stabilizer Extent (SE; Bravyi et al., 2019):
  $$\xi(|\psi\rangle) = \min_{x \in \mathbb{C}^{|\mathcal{S}_n|}} \left\{ \|x\|_1^2 \;\middle|\; |\psi\rangle = \textstyle\sum_{|\phi_j\rangle \in \mathcal{S}_n} x_j |\phi_j\rangle \right\}$$
- The reason why the L^1-norm $\|x\|_1$ quantifies the distance p:



$$\rho = 1.3\rho_1 + 0.6\rho_2 - 0.6\rho_3 - 0.3\rho_4 = 1.9\rho_+ - 0.9\rho_-$$
$$\mathcal{R}(\rho) = |1.3| + |0.6| + |-0.6| + |-0.3| = 0.9 \times 2 + 1 = \text{(Cost)}$$

L^1-norm of the coefficients $\|x\|_1$ quantifies distance p

- Application: near-Clifford simulators and circuit synthesis
  - RoM and SE quantify the simulation cost of different simulators
- Matrix-based reformulation to convex optimization problems

RoM: $\mathcal{R}(\rho) = \min_{x \in \mathbb{R}^{|\mathcal{S}_n|}} \left\{ \|x\|_1 \;\middle|\; A_n^{\mathrm{RoM}} x = b \right\}$ (Linear Programming)

SE: $\xi(|\psi\rangle) = \min_{x \in \mathbb{C}^{|\mathcal{S}_n|}} \left\{ \|x\|_1^2 \;\middle|\; A_n^{\mathrm{SE}} x = b \right\}$ (Second-Order Cone Programming)

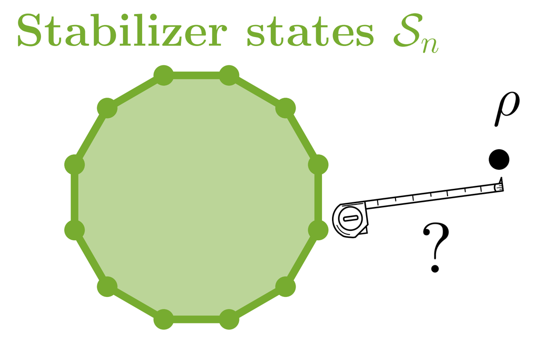
3. Result

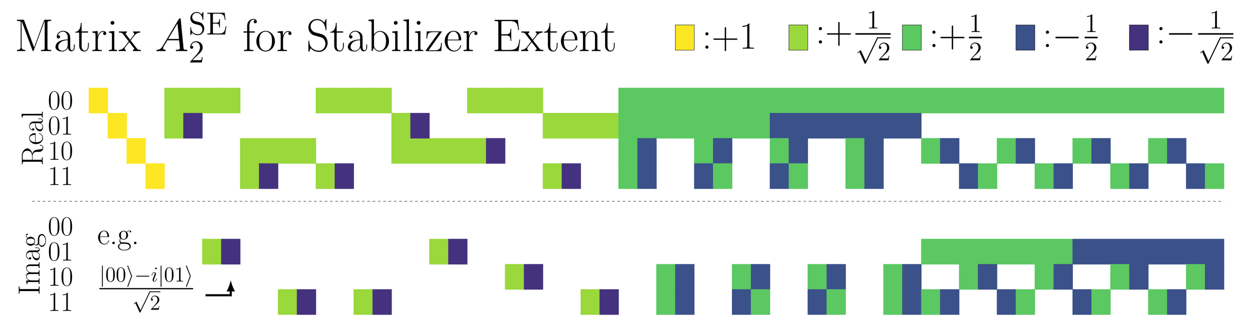
- The minimization problems are super-exponentially large: $|\mathcal{S}_n| = 2^{\mathcal{O}(n^2)}$
- Heinrich & Gross (2019) scaled up RoM computation for $|H\rangle^{\otimes n}$ and $|F\rangle^{\otimes n}$.
- We propose an improvement strategy for **general** states.

	qubit count n	5	6	7	8	9
	states $\lvert\mathcal{S}_n\rvert$	2.4×10^6	3.2×10^8	8.1×10^{10}	4.2×10^{13}	4.3×10^{16}
RoM	size of A_n^{RoM}	379 MiB	95 GiB	86 TiB	86 PiB	172 EiB
RoM	naive time	2 min	×	×	×	×
RoM	our time	2.3 s	7.0 min	1.6 h	2.0 d	×
SE	size of A_n^{SE}	1011 MiB	254 GiB	153 TiB	153 PiB	305 EiB
SE	naive time	7.7 min	×	×	×	×
SE	our time	1.5 s	3.8 s	12.9 s	8.8 min	19.2 h

4. Our Method

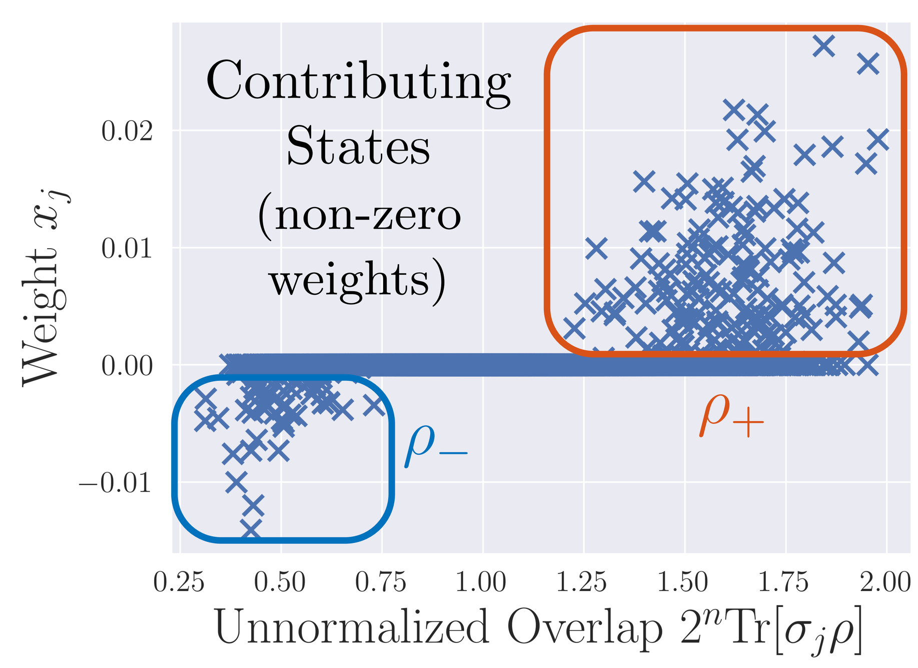
Q1. Reducing the size is inevitable. Can we predict contributing states?

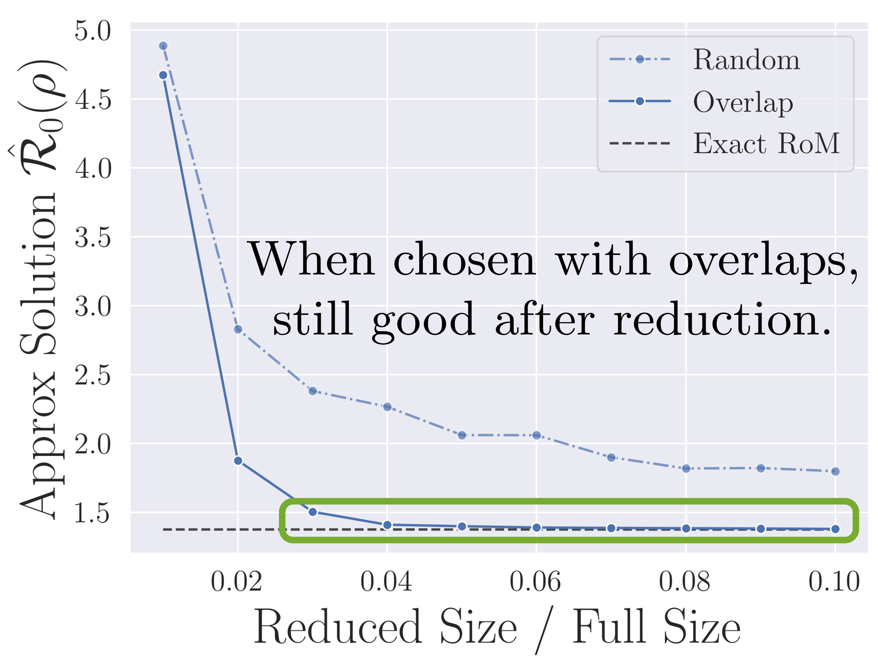




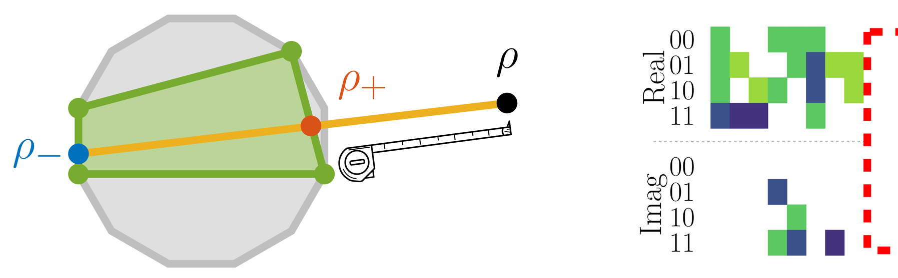
A1. Reduction by Overlaps

- **Overlap** ($\approx$ **Stabilizer Fidelity** $\approx$ closeness between states) is a good indicator for the prediction. The more extreme the overlaps, the more important.
- overlap $= a_j^\top b = \begin{cases} 2^n \operatorname{Tr}[\sigma_j \rho] & \text{(RoM)} \\ \langle \phi_j | \psi \rangle & \text{(SE)} \end{cases}$ where a_j is the j-th column of $A_n^{\mathrm{RoM}}, A_n^{\mathrm{SE}}$.





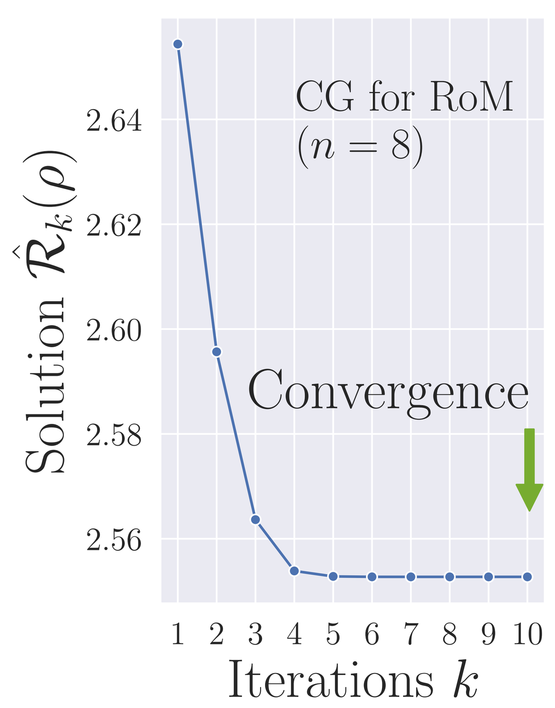
Q2. We got an approximate solution. Can we guarantee the optimality?



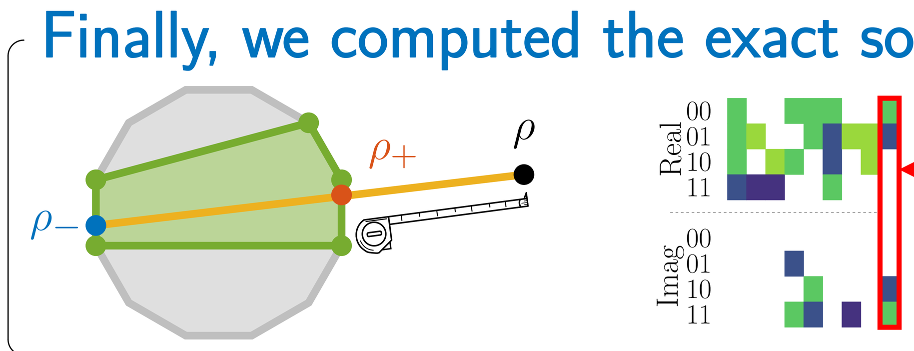
Solve with the reduced matrix
$\rightarrow$ Approximate solution

A2. Column Generation (CG) method

- **Dual problems** are equivalent to primal problems.
  - RoM: $\max_{y \in \mathbb{R}^{4^n}} \left\{ b^\top y \;\middle|\; \left\| A_n^{\mathrm{RoM}\top} y \right\|_\infty \leq 1 \right\}$
  - SE: $\max_{y \in \mathbb{C}^{2^n}} \left\{ \operatorname{Re}(b^\dagger y)^2 \;\middle|\; \left\| A_n^{\mathrm{SE}\dagger} y \right\|_\infty \leq 1 \right\}$
- CG regenerate removed but contributing columns.
- Repeatedly add columns a_j s.t. $|a_j^\dagger y| > 1$.
- No more restoration $\Longleftrightarrow$ Exactness of the solution



Finally, we computed the exact solution!



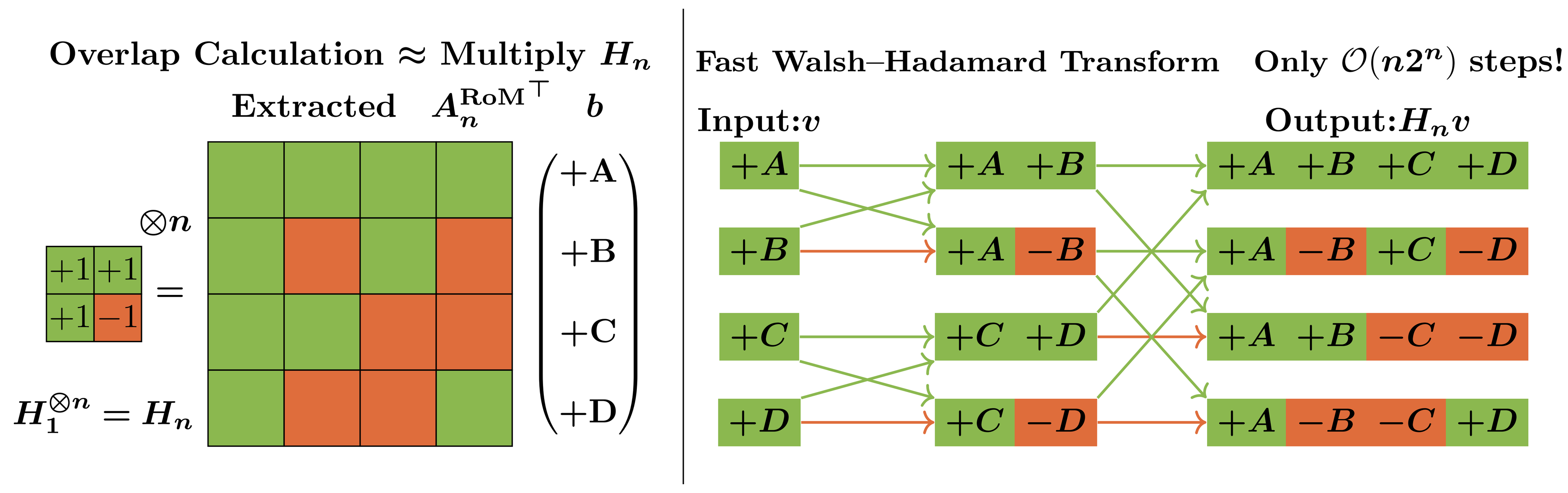
Restore contributing states
$\rightarrow$ Exact solution 🎉

5. Subroutines for Computing Overlaps

Calculating overlaps is the bottle-neck. We speed up by the methods below:

RoM: Fast Walsh–Hadamard Transform (FWHT)

- In-place algorithm for multiplying a vector by a tensor product of matrices
- Variant: Pauli decomposition in $\mathcal{O}(n4^n)$ time and $\mathcal{O}(4^n)$ space



SE: Stabilizer Pruning

- We propose Stabilizer Pruning based on **Branch and Bound method**.
- It is in the same spirit as FWHT, but even faster.
- It finds $\max_{\phi_j \in \mathcal{S}_n} |\langle \phi_j | \psi \rangle|$ among 4.3×10^{16} states ($n = 9$) within 26 min.