Structured Decomposition for Reversible Boolean Functions

Jiaqing Jiang, Xiaoming Sun, Yuan Sun, Kewen Wu, and Zhiyu Xia

***Abstract*—Reversible Boolean function (RBF) is a one-to-one function which maps n-bit input to n-bit output. Reversible logic synthesis has been widely studied due to its connection with low-energy computation as well as quantum computation. In this paper, we give a structured decomposition for *even* RBFs. Specifically, for $n \geq 6$, any even n-bit RBF can be decomposed to 7 blocks of $(n-1)$-bit RBF, where 7 is a constant independent of n and the positions of these blocks have a large degree of freedom. Moreover, if the $(n-1)$-bit RBFs are required to be even as well, we show for $n \geq 10$, even n-bit RBF can be decomposed to 10 even $(n-1)$-bit RBFs. In short, our decomposition has *block depth* 7 and *even block depth* 10. Our result improves Selinger's work in block depth model, by reducing the constant from 9 to 7 and from 13 to 10, when the blocks are limited to be even. We emphasize that our setting is a bit different from Selinger's work. In Selinger's constructive proof, each block is placed in one of two specific positions and thus the decomposition has an alternating structure. We relax this restriction and allow each block to act on arbitrary $(n-1)$ bits. This relaxation keeps the block structure and provides more candidates when choosing the positions of blocks.**

***Index Terms*—Integrated circuits, logic gates, quantum computation, reversible computation, reversible logic, synthesis method.**

I. Introduction

Reversible Boolean function (RBF) is a one-to-one function which maps n-bit input to n-bit output. Combinatorially, it represents a permutation over $\{0, 1\}^n$. One historical motivation of studying reversible computation is to reduce the energy consumption caused by computation [2]–[4]. According to the Landauer's principle [5], irreversible computation leads to energy dissipation of the order of KT per bit, where K refers to the Boltzmann constant and T is the temperature of the environment. In contrast, if the computing process is reversible, we can in principle use no energy. A classic example of realization of RBF—the billiard ball computer where computation costs no energy—can be found in Nielsen and Chuang's book [6]. In addition, RBFs are widely used in the quantum circuit, such as in the modular exponentiation part of Shor's factoring algorithm [7] or oracles in Grover's search algorithm [8], [9]. Any quantum circuit involving a Boolean function, which is generally irreversible and cannot be implemented in quantum circuit directly, such as quantum arithmetic circuit [10], [11], may benefit from the study of RBF.

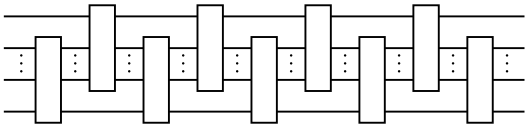


Fig. 1. Alternating structure in [1].

When implementing an n-bit RBF, the intuition is to use induction and divide the problem into smaller cases. That is, we try to decompose an n-bit RBF into a product of several $(n-1)$-bit RBFs. This decomposition is generally impossible, since if the n-bit RBF represents an odd permutation over $\{0, 1\}^n$, it cannot be implemented by $(n-1)$-bit RBFs, which are even when regarded as a permutation on n bits. On the other hand, Selinger [1] found the decomposition does exist for even n-bit RBFs and more remarkably, the number of required $(n-1)$-bit functions is a constant independent of n. More precisely, he proved that an arbitrary even n-bit RBF can be represented by 9 $(n-1)$-bit RBFs with an alternating structure shown in Fig. 1. He also proved that, if we limit the $(n-1)$-bit functions to be even as well, then the number of $(n-1)$-bit functions is at most 13. For simplicity, in the following, we use *block* to refer to the $(n-1)$-bit RBF, and *even block* to refer to the even $(n-1)$-bit RBF.

Our main contributions are: we improve the constant from 9 to 7 for $n \geq 6$ and 13 to 10 for $n \geq 9$ when limiting the blocks to be even. To be concise, our decomposition has block depth 7 and even block depth 10. We should emphasize that our setting is a bit different from Selinger's work. In Selinger's work, the decomposition is restricted to an alternating structure. Instead of fixing two specific positions, we allow blocks to act on arbitrary $(n-1)$-bits. This relaxation keeps the block structure and provides more candidates when choosing the position of blocks. We

Manuscript received January 3, 2019; revised May 11, 2019; accepted June 19, 2019. Date of publication July 16, 2019; date of current version September 18, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61433014, Grant 61832003, Grant 61761136014, Grant 61872334, and Grant 61801459, in part by the 973 Program of China under Grant 2016YFB1000201, in part by the Strategic Priority Research Program of Chinese Academy of Sciences under Grant XDB28000000, and in part by the Anhui Initiative in Quantum Information Technologies under Grant AHY150100. This paper was recommended by Associate Editor J. Cortadella. *(Corresponding author: Xiaoming Sun.)*

J. Jiang, X. Sun, Y. Sun, and Z. Xia are with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China, and also with the School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing 100049, China (e-mail: jiangjiaqing@ict.ac.cn; sunxiaoming@ict.ac.cn; sunyuan2016@ict.ac.cn; xiazhiyu@ict.ac.cn).

K. Wu is with the School of Electronics Engineering and Computer Science, Peking University, Beijing 100000, China (e-mail: shlw_kevin@pku.edu.cn).

Digital Object Identifier 10.1109/TCAD.2019.2928974

believe this relaxation makes the model more flexible in application.

For convenience, we abbreviate reversible Boolean function as RBF. We further say an RBF is controlled RBF (CRBF) if it keeps a certain bit invariant (formal definition is in Section II). Our construction consists of two steps. In the first stage, we prove that an arbitrary even n-bit RBF can be transformed into an even CRBF by 3 $(n-1)$-bit blocks and the positions of those low-level blocks have a lot of freedom. It is worth mentioning that the number 3 is also essentially tight. Then we prove that an arbitrary even CRBF can be substituted with five blocks, where the third and fourth blocks have many choices as well. While putting it together, we can literally merge the last block in the first step with the first block in the second step, thus providing a 7-depth full decomposition. As a partial result during the construction, we show that two different $(n-1)$-bit blocks are sufficient to formulate the cycle pattern of any even n-bit permutation free of 3/5-cycle. We believe this result has some individual interest. Here, cycle pattern is the list $\{c_k\}$, where c_k is the number of cycles of length k; and free of 3/5-cycle means $c_3 = c_5 = 0$. The limitation that cycle pattern is free of 3/5-cycle is indeed inevitable since we can also prove 2 $(n-1)$-bit blocks cannot compose a single 3/5-cycle. The proof of even block depth 10 is similar. Since all the proofs in this paper are constructive in essence, our decomposition can be programmed as an efficient algorithm.

In 2003, Shende *et al.* [9] proved that any even RBF can be decomposed into NOT gates, CNOT gates, and Toffoli gates without using temporary storage. Besides, In 2010, Saeedi *et al.* [12] gave an algorithm which synthesizes a given permutation by seven building blocks. These works focus on decomposing RBFs into smaller pieces, however, their constructions cannot be merged into 7 $(n-1)$-bit blocks, thus they are different from this paper. There are also some related works about decomposing n-bit unitary operator to smaller ones. In 2010, Saeedi *et al.* [13] showed how to decompose an arbitrary n-bit unitary operator down into ℓ-bit unitary operators $(\ell < n)$ using quantum Shannon decomposition [14].

The structured decomposition may have some potential applications. Though not directly improving results in circuit synthesis, the structure of this decomposition implies some interesting results. For instance, in Selinger's construction in Fig. 1, long-distance CNOT, i.e., CNOT between the first and the last bit prohibited by today's quantum devices [15], [16], shall be avoided. Although a similar effect can be realized with SWAP gates [17], this result actually indicates that such gate-costing alternatives will not happen frequently in a proper structure. In our setting, the positions of blocks have certain freedom to choose, which makes the construction even more flexible for different potential physical devices [18], [19].

Organization of This Paper: In Section II, we give formal definitions of the key elements required in expressing problem and formulating proof. Then, in Section III, we list our main results and give a proof sketch. In Sections IV and V, we give detailed proofs to the result of block depth 7. Specifically, in Section IV, we transform an even n-bit RBF to an even CRBF by 3 $(n-1)$-bit blocks. In Section V, we show how to recover an even CRBF by five blocks. In addition, an explicit example of our algorithm is put in Section VI. In Section VII, we give a proof sketch of the result of even block depth. This proof is similar to the proof of block depth but involves a much more sophisticated analysis. Finally, this paper is concluded in Section VIII. Due to the page limit, some proofs are omitted and can be found at [20].

II. Preliminary

In general, this paper aims to implement an even n-bit RBF using $(n-1)$-bit RBF. In order to state our problems and theorems properly, formal definitions are required.

Denote $[n]$ as $\{1, 2, \dots, n\}$ and $\{0, 1\}^n$ as the set of n-bit binary strings. Define $S_{\{0,1\}^n}$ as the permutation group over $\{0, 1\}^n$; and $A_{\{0,1\}^n}$ as the group of even permutations over $\{0, 1\}^n$. For any $\sigma \in S_{\{0,1\}^n}$ and $\boldsymbol{x}, \boldsymbol{y} \in \{0, 1\}^n$, define

$$\mathrm{dist}^{\sigma}(\boldsymbol{x}, \boldsymbol{y}) = \min\{k \in \mathbb{N} | \sigma^k(\boldsymbol{x}) = \boldsymbol{y}\}$$

[if $\boldsymbol{y}$ is not reachable from $\boldsymbol{x}$ under σ, $\mathrm{dist}^{\sigma}(\boldsymbol{x}, \boldsymbol{y}) = +\infty$] and $\mathrm{dist}^{\sigma}_{\min}(\boldsymbol{x}, \boldsymbol{y}) = \min\{\mathrm{dist}^{\sigma}(\boldsymbol{x}, \boldsymbol{y}), \mathrm{dist}^{\sigma}(\boldsymbol{y}, \boldsymbol{x})\}$. We also define the support of σ as $\mathrm{Supp}(\sigma) = \{\boldsymbol{x} | \sigma(\boldsymbol{x}) \neq \boldsymbol{x}\}$.

Recall that every permutation has a unique cycle decomposition. We say σ has a k-cycle if there is a cycle of length k in the cycle decomposition. We say $\boldsymbol{x} \in \{0, 1\}^n$ is a fix-point if $\sigma(\boldsymbol{x}) = \boldsymbol{x}$ and a fix-point is a 1-cycle as well. If σ consists of k_1-cycle, $\dots$, k_t-cycle, we say σ is exactly $k_1, \dots, k_t$-cycle. We may omit k_i if $k_i = 1$. For example, we may abbreviate $1, 3, 4$-cycle as $3, 4$-cycle. In addition, we say σ is free of $l_1/l_2/\cdots/l_s$-cycle if for any $i \in [s], j \in [t], l_i \neq k_j$.

For simplicity, we abbreviate reversible Boolean function as RBF and permutation over $\{0, 1\}^n$ as *n-bit permutation*. Since any n-bit RBF can be viewed as a permutation over $\{0, 1\}^n$, thus the set of all n-bit RBFs is isomorphic to $S_{\{0,1\}^n}$. Moreover, we say an n-bit RBF is *even* if its corresponding permutation is even.

Given $\boldsymbol{x} \in \{0, 1\}^n$, write $\boldsymbol{x}_i$ for the value of its ith bit; and $\boldsymbol{x}^{\oplus i} := \boldsymbol{x}_1 \cdots \boldsymbol{x}_{i-1}(1 - \boldsymbol{x}_i)\boldsymbol{x}_{i+1} \cdots \boldsymbol{x}_n$, i.e., $\boldsymbol{x}^{\oplus i}$ is $\boldsymbol{x}$ flipped the ith bit. Furthermore, define $\boldsymbol{x}^{\oplus i_1, i_2, \dots, i_k}$ recursively as $(\boldsymbol{x}^{\oplus i_1})^{\oplus i_2, \dots, i_k}$.

Definition 1 (CRBF): Given $n > 0$ and $i \in [n]$, we say π is an n-bit i-CRBF if $\pi \in S^{(i)}_{\{0,1\}^n}$, where

$$S^{(i)}_{\{0,1\}^n} := \left\{\sigma \in S_{\{0,1\}^n} \;\middle|\; \forall \boldsymbol{x} \in \{0, 1\}^n, \sigma(\boldsymbol{x})_i = \boldsymbol{x}_i\right\}.$$

We also define

$$A^{(i)}_{\{0,1\}^n} := \left\{\sigma \in A_{\{0,1\}^n} \;\middle|\; \forall \boldsymbol{x} \in \{0, 1\}^n, \sigma(\boldsymbol{x})_i = \boldsymbol{x}_i\right\}.$$

An i-CRBF keeps the ith bit of any input invariant. For example, if $i = 1$, then there exist $f_0, f_1 \in S_{\{0,1\}^{n-1}}$ such that $\pi(0\boldsymbol{y}) = 0f_0(\boldsymbol{y}), \pi(1\boldsymbol{y}) = 1f_1(\boldsymbol{y})$ for any $\boldsymbol{y} \in \{0, 1\}^{n-1}$. Notice that our definition of controlled-type gate, that is, the CRBF, is a bit different from the standard meaning in reversible circuits. In standard context f_0 must be id, while in our definition, f_0 can be arbitrary permutation in $S_{\{0,1\}^{n-1}}$.

Moreover, we say π is a concurrent CRBF (CCRBF) if $f_0 = f_1$. Further, when f_0 is even, we say π is concurrently even; and concurrently odd when f_0 is odd. The formal definitions are shown below.

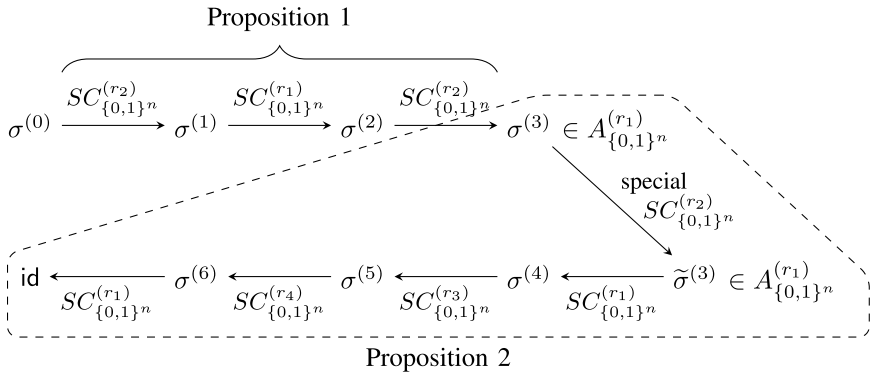

Proposition 1
$\sigma^{(0)} \xrightarrow{SC^{(r_2)}_{\{0,1\}^n}} \sigma^{(1)} \xrightarrow{SC^{(r_1)}_{\{0,1\}^n}} \sigma^{(2)} \xrightarrow{SC^{(r_2)}_{\{0,1\}^n}} \sigma^{(3)} \in A^{(r_1)}_{\{0,1\}^n}$
special $SC^{(r_2)}_{\{0,1\}^n}$
$\mathsf{id} \xleftarrow[SC^{(r_1)}_{\{0,1\}^n}]{} \sigma^{(6)} \xleftarrow[SC^{(r_4)}_{\{0,1\}^n}]{} \sigma^{(5)} \xleftarrow[SC^{(r_3)}_{\{0,1\}^n}]{} \sigma^{(4)} \xleftarrow[SC^{(r_1)}_{\{0,1\}^n}]{} \widetilde{\sigma}^{(3)} \in A^{(r_1)}_{\{0,1\}^n}$
Proposition 2

Fig. 2. Process of the algorithm for Theorem 1.

Definition 2 (CCRBF): Given $n > 0$ and $i \in [n]$, we say π is an n-bit i-CCRBF if $\pi \in SC^{(i)}_{\{0,1\}^n}$, where

$$SC^{(i)}_{\{0,1\}^n} := \Big\{\sigma \in S^{(i)}_{\{0,1\}^n} \;\Big|\; \forall \boldsymbol{x} \in \{0,1\}^n \\ \forall k \in [n]\backslash\{i\}, \sigma(\boldsymbol{x})_k = \sigma(\boldsymbol{x}^{\oplus i})_k\Big\}.$$

Definition 3 (Concurrently Even/Odd): An n-bit i-CCRBF π can be regarded as an $(n-1)$-bit RBF $\sigma|_{-i}$ on bits $[n]/\{i\}$. We say that σ is i-concurrently even/odd if $\sigma|_{-i}$ is even/odd. Define $AC^{(i)}_{\{0,1\}^n}$ as the set of n-bit concurrently even i-CRBF.

When dimension i is clear in the context, we simply use concurrently even/odd. Note that no matter whether $\sigma|_{-i} \in S_{\{0,1\}^{n-1}}$ is odd or even, CCRBF $\sigma \in S_{\{0,1\}^n}$ itself is always even.

Definition 4 Block Depth and Even Block Depth: Given $n \geq 2$ and $\sigma \in S_{\{0,1\}^n}$, we say σ has block depth d if there exist $\sigma_1, \sigma_2, \ldots, \sigma_d \in \bigcup_{j=1}^n SC^{(j)}_{\{0,1\}^n}$ such that $\sigma = \sigma_1\sigma_2\cdots\sigma_d$.

Similarly, we say σ has even block depth d if those $\sigma_i \in \bigcup_{j=1}^n AC^{(j)}_{\{0,1\}^n}$.

Notice that the decomposition problem considered here is a bit different from Selinger's work [1]. In Selinger's work, any σ_i is in one of the two specific positions, thus the decomposition forms an alternating structure as Fig. 1. Here, we relax the restriction and allow blocks acting on arbitrary $(n-1)$ bits. Thus, we consider the block depth instead of alternation depth used in [1].

III. Main Results and Proof Sketch

In the previous work, Selinger [1] proved that an arbitrary even n-bit RBF has alternation depth 9 and even alternation depth 13. Our main contribution is to improve the constants 9 to 7 in block depth model and 13 to 10 in even block depth model. The main theorems are stated as follows.

Theorem 1: For $n \geq 6$, any $\sigma \in A_{\{0,1\}^n}$ has block depth 7.

Theorem 2: For $n \geq 10$, any $\sigma \in A_{\{0,1\}^n}$ has even block depth 10.

Proof of Theorem 1: To prove Theorem 1, we first turn σ into an even CRBF by Proposition 1; then further break the even CRBF down into identity by Proposition 2. We achieve these two steps with 3 and 5 blocks, respectively. By a finer analysis, the last block of the first step and the first block of the second step can be merged. Thus, a 7-block implementation is obtained. The sketch of the whole process is depicted in Fig. 2. ■

The proof of Theorem 2 is similar. Before Section VII, we only focus on the proof of block depth 7.

Proposition 1 states that we can transform an even n-bit RBF to an even CRBF by three CCRBFs with many choices.

Proposition 1: For $n \geq 4, r_1 \in [n]$ and $\sigma \in A_{\{0,1\}^n}$, there exist at least $(n-2)$ different $r_2 \in [n]\backslash\{r_1\}$ such that $\sigma\pi_1\sigma_1\pi_2 \in A^{(r_1)}_{\{0,1\}^n}$ holds for some $\sigma_1 \in SC^{(r_1)}_{\{0,1\}^n}$ and $\pi_1, \pi_2 \in SC^{(r_2)}_{\{0,1\}^n}$.

In addition, we also show the tightness of Proposition 1 by Lemma 5 in Section IV. It is also worth noting that the proof works for $\sigma \in S_{\{0,1\}^n}$ (with $\sigma\pi_1\sigma_1\pi_2 \in S^{(r_1)}_{\{0,1\}^n}$) as well. For our purpose, it is more convenient to state it as Proposition 1.

Proposition 2 states that we can recover any even n-bit CRBF by five CCRBFs.

Proposition 2: For $n \geq 6, r_1 \in [n], r_2, r_3, r_4 \in [n]\backslash\{r_1\}, r_3 \neq r_4$ and $\sigma \in A^{(r_1)}_{\{0,1\}^n}$, there exist $\pi_1 \in SC^{(r_2)}_{\{0,1\}^n}$, $\sigma_1, \sigma_2 \in SC^{(r_1)}_{\{0,1\}^n}, \tau_1 \in SC^{(r_3)}_{\{0,1\}^n}, \tau_2 \in SC^{(r_4)}_{\{0,1\}^n}$ such that $\sigma\pi_1\sigma_1\tau_1\tau_2\sigma_2 = \mathsf{id}$.

The key in the proof of Proposition 2 is the following proposition, which states two n-bit CCRBFs can formulate the cycle pattern of any even n-bit permutation free of 3/5-cycle. We believe this proposition has some individual interest.

Proposition 3: For $n \geq 4$, distinct $r_1, r_2 \in [n]$ and $\sigma \in A_{\{0,1\}^n}$ free of 3/5-cycle, there exist $\pi \in SC^{(r_1)}_{\{0,1\}^n}, \tau \in SC^{(r_2)}_{\{0,1\}^n}$ such that $\pi\tau$ and σ have the same cycle pattern, which is equivalent to that $h\sigma h^{-1} = \pi\tau$ holds for some $h \in S_{\{0,1\}^n}$.

The proof of Proposition 1 is in Section IV and the proof of Propositions 2 and 3 is in Section V.

IV. Transforming Even n-Bit RBF to Controlled RBF

In this section, we give proof of Proposition 1. That is, we transform an even n-bit RBF σ to an even CRBF using three CCRBFs. σ may involve 2^n elements and have a complicated pattern. However, to transform σ to a CRBF, which keeps one bit invariant, the key point is whether the ith bit of $\sigma(\boldsymbol{x})$ equals the ith bit of $\boldsymbol{x}$. So we simplify the representation of an RBF by constructing a black-white cuboid, where the color indicates whether $\sigma(\boldsymbol{x})_i = \boldsymbol{x}_i$. Then proving Proposition 1 is equivalent to transforming the colored cuboid to white. An

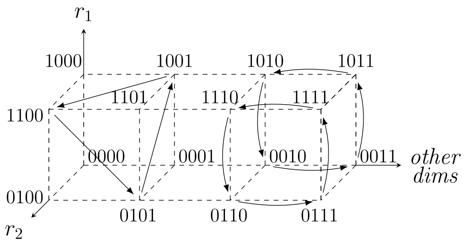


Fig. 3. Visualize σ on a 3-D cuboid.

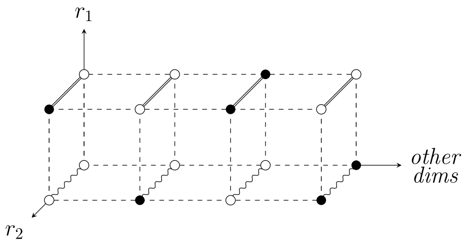


Fig. 4. Visualize σ on a colored cuboid.

explicit example of the whole process of Proposition 1 can be seen in Section VI.

Recall that n-bit RBF is in fact a permutation on $\{0,1\}^n$. Specifically, we visualize the permutation on a $2 \times 2 \times 2^{n-2}$ 3-D cuboid. In Section IV-A, we give the construction for the black-white 3-D cuboid corresponding to σ. After that, in Section IV-B, we give a constructive proof to transform the colored cuboid to a white cuboid.

Proof of Proposition 1: First, we choose arbitrary two different $r_1, r_2 \in [n]$ and construct a black-white cuboid. Then we transform the colored cuboid to a canonical form by $SC^{(r_2)}_{\{0,1\}^n}$ using Lemma 2. We also prove in most cases, by Lemma 1, the canonical form can be transformed to a white cuboid by $SC^{(r_2)}_{\{0,1\}^n}$, $SC^{(r_1)}_{\{0,1\}^n}$, and $SC^{(r_2)}_{\{0,1\}^n}$. Finally, if the canonical form falls into a bad case, we prove that for any $r_3 \in [n]\backslash\{r_1, r_2\}$, by checking the new canonical form based on r_1 and r_3, this case can be tackled with $SC^{(r_3)}_{\{0,1\}^n}$, $SC^{(r_1)}_{\{0,1\}^n}$, and $SC^{(r_3)}_{\{0,1\}^n}$ using Lemma 3. ■

A. Visualizing Permutation on 3-D Cuboid

Given permutation $\sigma \in S_{\{0,1\}^n}$, in this section, we construct a 3-D black-white cuboid for σ and discuss the effect of transformation, that is, the new colored cuboid for $\sigma\tau$, $\tau \in S_{\{0,1\}^n}$.

Recall that σ is a permutation over 2^n elements. Fixing $r_1, r_2 \in [n]$ and compressing the other $(n-2)$ dimensions, we get a 3-D cuboid. For example, if $n = 4$, $r_1 = 1$, and $r_2 = 2$, then we compress the remaining two dimensions into one by letting the coordinates to be 00, 01, 10, and 11. We visualize σ in Fig. 3, where

$$\begin{aligned}\sigma := &(1001, 1100, 0101)(1110, 0110, 0111, 1111)\\ &(1010, 0010, 0011, 1011).\end{aligned}$$

As an example, 1100 is labeled on $(1, 1, 00)$, where 00 represents the third coordinate. The arrows in the figure stand for permutation σ. In this case, $\sigma(1100) = 0101$, so we draw an arrow from 1100 to 0101.

The graph reflects both pattern and structure of the permutation. If we exert a CCRBF

$$\begin{aligned}\tau = &(1010, 1011, 0011, 0010)\\ &(1110, 1111, 0111, 0110) \in SC^{(2)}_{\{0,1\}^4}\end{aligned}$$

on σ, it will have the same effect on the front and back face of the cuboid, eliminating the two 4-cycles. That is, the 3-D cuboid corresponding to $\sigma\tau$ will only have a 3-cycle.

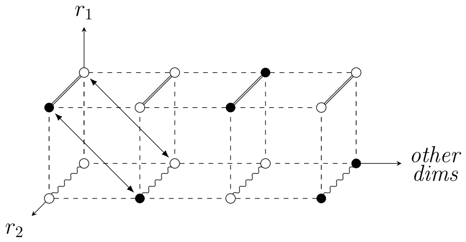


Fig. 5. Colored cuboid for σ. Arrows refer to π.

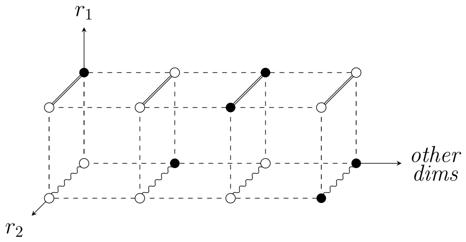
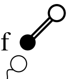
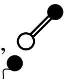
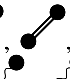
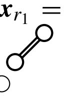
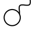
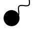
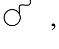


Fig. 6. Colored cuboid for $\sigma\pi$.

Back to Proposition 1, here, we aim to eliminate cycles which have overlap with both top and bottom face. To further simplify the notation, we transform the cuboid with arrow pattern into a cuboid with black-white colored nodes. That is, we paint coordinate $\boldsymbol{x} \in \{0,1\}^n$ black if $\sigma(\boldsymbol{x})_{r_1} \neq \boldsymbol{x}_{r_1}$ as shown in Fig. 4. Intuitively, the black node means that $\sigma(\boldsymbol{x})$ is in a wrong face.

Now, we consider the cuboid of $\sigma\pi$ with some permutation π. For example, if π pushes $\boldsymbol{x}$ to the opposite face, the color of $\boldsymbol{x}$ in cuboid for $\sigma\pi$ will be the opposite of original $\boldsymbol{\pi}(x)$'s in cuboid for σ. That is, assuming $\pi(\boldsymbol{x}) = \boldsymbol{x}'$ and $\boldsymbol{x}_{r_1} \neq \boldsymbol{x}'_{r_1}$, if $\sigma(\boldsymbol{x}')_{r_1} \neq \boldsymbol{x}'_{r_1}$, then $\sigma(\boldsymbol{x}')_{r_1} = \boldsymbol{x}_{r_1}$ (i.e., $\sigma(\pi(\boldsymbol{x}))_{r_1} = \boldsymbol{x}_{r_1}$), vice versa. An example is in Figs. 5 and 6 for $\pi = (1100, 0101)(1000, 0001) \in SC^{(2)}_{\{0,1\}^4}$.

Using colored cuboid, for some π', the cuboid for $\sigma\pi'$ is white if and only if $\sigma\pi' \in S^{(r_1)}_{\{0,1\}^n}$. To prove Proposition 1, it suffices to show that we can transform any black-white cuboid into a white cuboid, using CCRBFs.

For simplicity, as shown in Fig. 6, we use a double line to connect $\boldsymbol{x}$ and $\boldsymbol{x}^{\oplus r_2}$ for all $\boldsymbol{x}$ with $\boldsymbol{x}_{r_1} = 1$; and zigzag line to connect $\boldsymbol{x}$ and $\boldsymbol{x}^{\oplus r_2}$ for all $\boldsymbol{x}$ with $\boldsymbol{x}_{r_1} = 0$. Let a_1, a_2, a_3, a_4 be the number of , , ,  and b_1, b_2, b_3, b_4 be the number of , , , , respectively.

Algorithm 1: Canonical Form (CANONICAL)

```
Input: $\sigma \in S_{\{0,1\}^n}$ and its colored cuboid
Output: Canonical form of the cuboid
1 Swap ●═● with ●∿● until $a_3$ or $b_3$ reaches zero
2 if $a_3 = 0$ then
3     Swap ○═● (or ●═○ ) with ●∿● until $b_3$ reaches zero
4 end
5 else
6     Swap ●═● with ○∿● (or ●∿○ ) until $a_3$ reaches zero
7 end
8 Swap ○═● with ○∿● until $a_2$ reaches zero
```

B. Transforming σ to Controlled Permutation

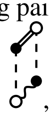
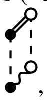
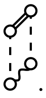
In this section, we transform the given permutation to CRBF. Following the previous section, we construct a colored cuboid for $\sigma \in A_{\{0,1\}^n}$ and calculate corresponding a_i's, b_i's. According to a_i's and b_i's, we transform σ to $A^{(r_1)}_{\{0,1\}^n}$ using Lemma 1 or Lemma 3. We also show the tightness of three steps by Lemma 5.

First, we prove Lemma 1 to show most cases are solvable by $SC^{(r_2)}_{\{0,1\}^n}$, $SC^{(r_1)}_{\{0,1\}^n}$, and $SC^{(r_2)}_{\{0,1\}^n}$. Since the number of black nodes in lower and upper faces is the same, it is easy to see $a_3 + a_4 + b_3 + b_4$ is even.

Lemma 1: There exist $\sigma_1 \in SC^{(r_1)}_{\{0,1\}^n}$ and $\pi_1, \pi_2 \in SC^{(r_2)}_{\{0,1\}^n}$ such that $\sigma\pi_1\sigma_1\pi_2 \in A^{(r_1)}_{\{0,1\}^n}$ if:

1) $a_3 + a_4 + b_3 + b_4 > 2$ holds; or
2) $a_3 + a_4 + b_3 + b_4 = 2$ and $\min\{b_1 + a_2, a_1 + b_2\} > 0$ hold; or
3) $a_3 + a_4 + b_3 + b_4 = 0$ and $b_1 + a_2$ is even (equivalently $a_1 + b_2$ is even) hold.

To give specific constructions, we first transform the colored cuboid to a *canonical form* by Lemma 2. Then we classify them into different cases and solve case by case.

A *canonical form* is a colored cuboid only containing three kinds of matching pairs ("cards") along the compressed dimensions, which are , , . We call them A-card, B-card, and C-card; and the numbers of these three kinds are α, β, and γ, respectively.

If $a_2 + a_3 \leq b_2 + b_3$, we can use Lemma 2 to transform the colored cuboid to a canonical form.

Lemma 2: If $a_2 + a_3 \leq b_2 + b_3$, there exists $\pi \in SC^{(r_2)}_{\{0,1\}^n}$ such that the colored cuboid for $\sigma\pi$ is of canonical form.

Proof: Recall that the color of a node $\boldsymbol{x}$ refers to whether $\sigma(\boldsymbol{x})$ is in the correct face. So if coordinate $\boldsymbol{x}'$ is black and $\boldsymbol{x}'_{r_1} \neq \boldsymbol{x}_{r_1}$, then coordinate $\boldsymbol{x}$ will be white after swapping $\boldsymbol{x}$ and $\boldsymbol{x}'$, vice versa. See Fig. 6 as an example.

We first apply $\tau \in SC^{(r_2)}_{\{0,1\}^n}$ such that the cuboid for $\sigma\tau$ satisfies $a'_4 = b'_4$, $a'_2 = a'_3 = b'_3 = 0$. Then we use $\tau' \in SC^{(r_2)}_{\{0,1\}^n}$ to rearrange the nodes, such that the cuboid for $\sigma\tau\tau'$ is a canonical form. τ is achieved by the following algorithm.

The correctness comes from the following observation. Since the number of black nodes is the same in the top and bottom face, if there is a black node in one face, the opposite face has one as well. Therefore, in line 3, the number of ○═● and ●═○ is no fewer than ●∿●; in line 6, the number of ○∿● and ●∿○ is no fewer than ●═●. Since $a_2 + a_3 \leq b_2 + b_3$, it can be verified when algorithm executes in line 8, the number of ○═● is no more than ○∿●. After performing this algorithm, we have $a'_2 = a'_3 = b'_3 = 0$ and $a'_4 = b'_4$.

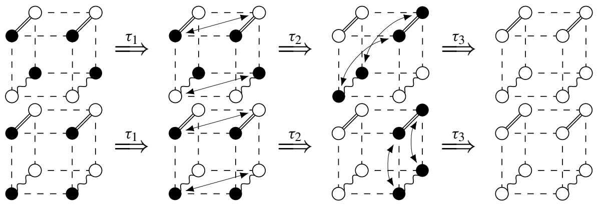
Then we rearrange the nodes to form A-, B-, C-cards. Since $a'_4 = b'_4$, by some permutation $\tau' \in SC^{(r_2)}_{\{0,1\}^n}$, we can assure that the colored cuboid corresponding to $\sigma\tau\tau'$ only has these three kind of cards. Thus, let $\pi = \tau\tau'$, then the colored cuboid for $\sigma\pi$ is of canonical form.

Since the number of ○═● and ●∿○ is invariant in Algorithm 1, as well as ●═○ and ○∿●, we have $\alpha = (1/2)(a_1 - a_2 + b_2 - b_1)$, $\beta = b_1 + a_2$, and $\gamma = (1/2)(a_3 + a_4 + b_3 + b_4)$. ■

Now, we give the proof of Lemma 1.

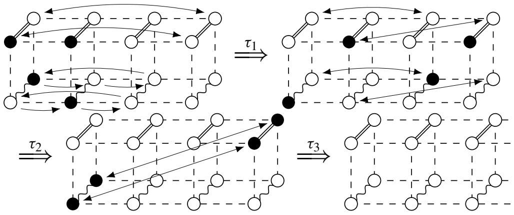
Proof of Lemma 1: Without loss of generality, assume $a_2 + a_3 \leq b_2 + b_3$. Using Lemma 2, we transform the colored cuboid to a canonical form with $\pi' \in SC^{(r_2)}_{\{0,1\}^n}$. Record the number of the three kind of cards, i.e., α, β, and γ.

First notice that if we pair two A-cards or two B-cards, the paired A-cards and B-cards can be transformed to C-cards by the following permutations, where $\tau_1 \in SC^{(r_2)}_{\{0,1\}^n}$, $\tau_2 \in SC^{(r_1)}_{\{0,1\}^n}$, and $\tau_3 \in SC^{(r_2)}_{\{0,1\}^n}$:

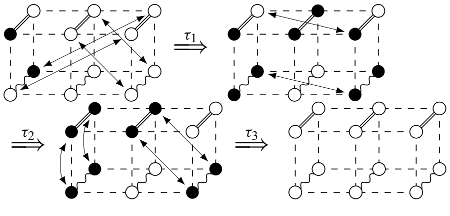


This approach solves the third case directly and reduces the first case to the following three subcases. Since these card groups can be tackled in parallel, in final construction, $\pi_1 = \pi'\tau_1$, $\sigma_1 = \tau_2$, and $\pi_2 = \tau_3$.

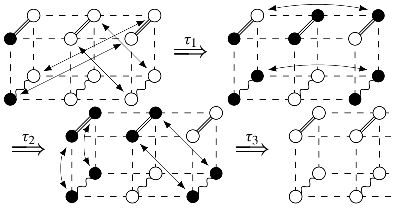
1) $\alpha = 1, \beta = 1$, and $\gamma \geq 2$: This graph shows how to tackle 1 A-card and 1 B-card with 2 C-cards.



2) $\alpha = 1, \beta = 0$, and $\gamma \geq 2$: This graph shows how to tackle 1 A-card with 2 C-cards.

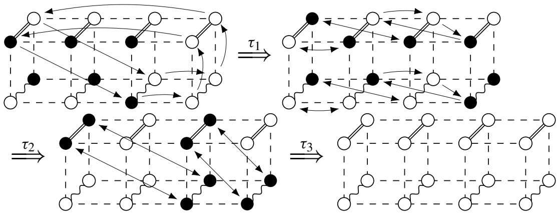


3) $\alpha = 0, \beta = 1$, and $\gamma \geq 2$: This graph shows how to tackle 1 B-card with 2 C-card.

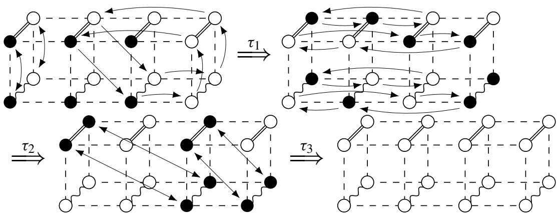


For the second case, we reduce it to the following.

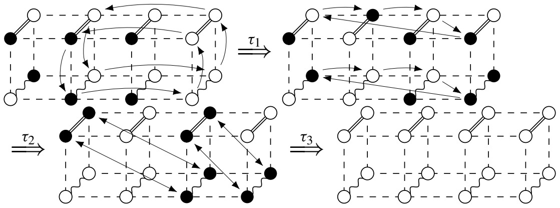
1) $\alpha = 2, \beta = 1$, and $\gamma \geq 1$: This graph shows how to tackle 2 A-cards and 1 B-card with 1 C-card.



2) $\alpha = 0, \beta = 3$, and $\gamma \geq 1$: This graph shows how to tackle 3 B-cards with 1 C-card.



3) $\alpha = 1, \beta = 2$, and $\gamma \geq 1$: This graph shows how to tackle 1 A-card and 2 B-cards with 1 C-card.



■

For the other cases, which cannot be solved by Lemma 1, can in turn be dealt with Lemma 3.

Lemma 3: For any $r_3 \in [n]\backslash\{r_1, r_2\}$, there exist $\sigma_1 \in SC^{(r_1)}_{\{0,1\}^n}$ and $\pi_1, \pi_2 \in SC^{(r_3)}_{\{0,1\}^n}$ such that $\sigma\pi_1\sigma_1\pi_2 \in A^{(r_1)}_{\{0,1\}^n}$ if:

1) $a_3 + a_4 + b_3 + b_4 = 2$ and $\min\{b_1 + a_2, a_1 + b_2\} = 0$ hold or
2) $a_3 + a_4 + b_3 + b_4 = 0$ and $b_1 + a_2$ is odd (equivalently $a_1 + b_2$ is odd) hold.

Fixing r_1, if for some r_2, the corresponding canonical form falls into Lemma 3. Then for any $r_3 \in [n]\backslash\{r_1, r_2\}$, the canonical form corresponding with r_1, r_3 will fall into 3-step solvable cases, that is, it can be solved by Lemma 1 with $r_2' = r_3$.

Before the proof, we show how to switch dimensions. We visualize the permutation on a black-white 4-D cuboid as two 3-D cuboids. When r_1 and r_2 are fixed, pick $r_3 \in [n]\backslash\{r_1, r_2\}$ and compress all the other $(n-3)$ dimensions. As before, paint $\boldsymbol{x}$ black if $\sigma(\boldsymbol{x})_{r_1} \neq \boldsymbol{x}_{r_1}$ for all $\boldsymbol{x} \in \{0,1\}^n$. An example of $n = 4, r_1 = 1, r_2 = 2$, and $r_3 = 4$ is Fig. 7. The left and right 3-D cuboids corresponding to $r_3 = 0$ and $r_3 = 1$.

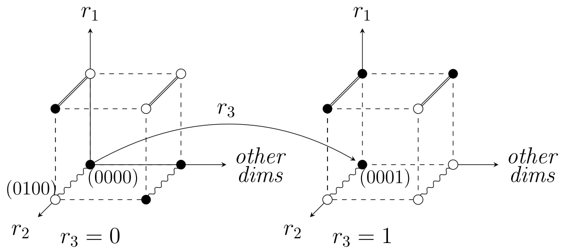


Fig. 7. 4-D cuboid for $n = 4, r_1 = 1, r_2 = 2$, and $r_3 = 4$.

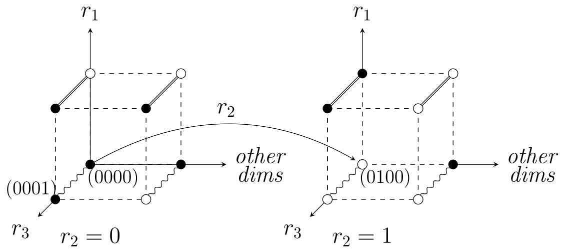


Fig. 8. Switching from r_2 to r_3.

In Fig. 7, let a_1, a_2, a_3, a_4 be the number of ●⫽○, ○⫽●, ●⫽●, ○⫽○ and b_1, b_2, b_3, b_4 be the number of ●∿○, ○∿●, ●∿●, ○∿○, respectively. When we switching dimensions r_2 and r_3, Fig. 7 changes to Fig. 8. Similarly, in Fig. 8, denote $\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{a}_4$ to be the number of ●⫽○, ○⫽●, ●⫽●, ○⫽○ and $\hat{b}_1, \hat{b}_2, \hat{b}_3, \hat{b}_4$ to be the number of ●∿○, ○∿●, ●∿●, ○∿○, respectively.

Proof of Lemma 3: For the first case in Lemma 3, without loss of generality, assume $b_1 + a_2 = 0$. And we have the following four cases.

1) $a_3 + a_4 = 2$: Thus, all $\boldsymbol{x} \in \{0,1\}^n, \boldsymbol{x}_{r_1} = \boldsymbol{x}_{r_2} = 0$ are black; and all $\boldsymbol{x} \in \{0,1\}^n, \boldsymbol{x}_{r_1} = 0, \boldsymbol{x}_{r_2} = 1$ are white. Therefore, $\hat{b}_3 = \hat{b}_4 = 2^{n-3}$, which is 3-step solvable in the first case of Lemma 1.
2) $b_3 + b_4 = 2$: Similar with case $a_3 + a_4 = 2$.
3) $b_3 = 1$: Thus, all $\boldsymbol{x} \in \{0,1\}^n, \boldsymbol{x}_{r_1} = \boldsymbol{x}_{r_2} = 0$ are black; and all $\boldsymbol{x} \in \{0,1\}^n, \boldsymbol{x}_{r_1} = 0, \boldsymbol{x}_{r_2} = 1$ are white except one. Therefore, $\hat{b}_3 = 2^{n-3}, \hat{b}_4 = 2^{n-3} - 1$, which is 3-step solvable in the first case of Lemma 1.
4) $b_4 = 1$: Similar with case $b_3 = 1$.

For the second case in Lemma 3, since $a_3 + a_4 + b_3 + b_4 = 0$, then for any $\boldsymbol{x} \in \{0,1\}^n$, the color of $\boldsymbol{x}$ is different from the color of $\boldsymbol{x}^{\oplus r_2}$. Define

$$
\begin{aligned}
u_b &= \left|\left\{\text{black } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_1} = 1, \boldsymbol{x}_{r_2} = \boldsymbol{x}_{r_3} = 0\right\}\right| \\
u_w &= \left|\left\{\text{white } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_1} = 1, \boldsymbol{x}_{r_2} = \boldsymbol{x}_{r_3} = 0\right\}\right| \\
l_b &= \left|\left\{\text{black } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_1} = 0, \boldsymbol{x}_{r_2} = \boldsymbol{x}_{r_3} = 0\right\}\right| \\
l_w &= \left|\left\{\text{white } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_1} = 0, \boldsymbol{x}_{r_2} = \boldsymbol{x}_{r_3} = 0\right\}\right|
\end{aligned}
$$

and

$$
\begin{aligned}
u_b' &= \left|\left\{\text{black } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_2} = 0, \boldsymbol{x}_{r_1} = \boldsymbol{x}_{r_3} = 1\right\}\right| \\
u_w' &= \left|\left\{\text{white } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_2} = 0, \boldsymbol{x}_{r_1} = \boldsymbol{x}_{r_3} = 1\right\}\right| \\
l_b' &= \left|\left\{\text{black } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_1} = \boldsymbol{x}_{r_2} = 0, \boldsymbol{x}_{r_3} = 1\right\}\right| \\
l_w' &= \left|\left\{\text{white } \boldsymbol{x} \in \{0,1\}^n \middle| \boldsymbol{x}_{r_1} = \boldsymbol{x}_{r_2} = 0, \boldsymbol{x}_{r_3} = 1\right\}\right|.
\end{aligned}
$$

By assumption, $a_1+b_2=u_w+u'_w+l_b+l'_b$ and $b_1+a_2=u_b+u'_b+l_w+l'_w$. And $u_w+u_b=u'_w+u'_b=l_w+l_b=l'_w+l'_b=2^{n-3}$. Thus

$$\begin{aligned}u_b+u'_b+l_b+l'_b=(u_b+u'_b+l_w+l'_w)+(l_w+l_b)\\+(l'_w+l'_b)-2(l_w+l'_w)\end{aligned}$$

is odd. On the other hand, $|\{\boldsymbol{x}\mid \boldsymbol{x}_{r_2}=0,\boldsymbol{x}_{r_3}=0\}|=|\{\boldsymbol{x}\mid \boldsymbol{x}_{r_2}=0,\boldsymbol{x}_{r_3}=1\}|=2^{n-2}$ is even. Therefore, there exists $\boldsymbol{x}\in\{0,1\}^n,\boldsymbol{x}_{r_2}=0$ such that the color of $\boldsymbol{x}$ is the same with the color of $\boldsymbol{x}^{\oplus r_3}$. Thus, $\hat{a}_3+\hat{a}_4+\hat{b}_3+\hat{b}_4>0$.

1) $\hat{a}_3+\hat{a}_4+\hat{b}_3+\hat{b}_4>2$: It is 3-step solvable in the first case of Lemma 1.
2) $\hat{a}_3+\hat{a}_4+\hat{b}_3+\hat{b}_4=2$: Thus, there exists $\boldsymbol{x}\in\{0,1\}^n,\boldsymbol{x}_{r_1}=0$, such that $\boldsymbol{x}$ is white; then $\boldsymbol{x}^{\oplus r_3}$ and $\boldsymbol{x}^{\oplus r_2}$ are all black; and $\boldsymbol{x}^{\oplus r_2,r_3}$ is white. Thus, when r_2 is swapped with r_3, $\boldsymbol{x}$ with $\boldsymbol{x}^{\oplus r_3}$ and $\boldsymbol{x}^{\oplus r_2}$ with $\boldsymbol{x}^{\oplus r_2,r_3}$ form ●–○ and ○–●. Therefore, $\hat{b}_1,\hat{b}_2>0$, which is 3-step solvable in the second case of Lemma 1. ■

For completeness, in Lemma 4, we show that cases in Lemma 3 cannot be solved in the order r_2,r_1,r_2. The proof can be found at the Appendix of [20].

Lemma 4: For any $\sigma_1\in SC^{(r_1)}_{\{0,1\}^n},\pi_1,\pi_2\in SC^{(r_2)}_{\{0,1\}^n}$, $\sigma\pi_1\sigma_1\pi_2\notin A^{(r_1)}_{\{0,1\}^n}$ if:

1) $a_3+a_4+b_3+b_4=2$ and $\min\{b_1+a_2,a_1+b_2\}=0$ hold or
2) $a_3+a_4+b_3+b_4=0$ and b_1+a_2 is odd (equivalently a_1+b_2 is odd) hold.

Lemma 5 shows that three steps is tight for transforming arbitrary permutation into a CRBF. The proof can be found at the Appendix of [20].

Lemma 5: For all even number $n\geq 4$, there exists $\sigma\in A_{\{0,1\}^n}$ such that $\sigma\tau\pi\notin S^{(r_3)}_{\{0,1\}^n}$ for any $r_1,r_2,r_3\in[n],\tau\in SC^{(r_1)}_{\{0,1\}^n},\pi\in SC^{(r_2)}_{\{0,1\}^n}$.

V. Transforming CRBF to Identity

In this section, we transform an even CRBF to id through five CCRBFs, where the first block can be merged with the last block of Proposition 1.

Recall that given $\sigma\in S^{(1)}_{\{0,1\}^n}$, there exist $f,g\in S_{\{0,1\}^{n-1}}$ such that for all $\boldsymbol{y}\in\{0,1\}^{n-1}$, $\sigma(0\boldsymbol{y})=0f(\boldsymbol{y}),\sigma(1\boldsymbol{y})=1g(\boldsymbol{y})$. We represent σ by $2^n\times 2^n$ matrix and f,g by $2^{n-1}\times 2^{n-1}$ matrix. For example, if $\tau=(00,01)(10,11)\in SC^{(1)}_{\{0,1\}^2}$, the basis is 00, 01, 10, and 11, then

$$\tau=\begin{bmatrix}0&1&0&0\\1&0&0&0\\0&0&0&1\\0&0&1&0\end{bmatrix}$$

where f_τ and g_τ are

$$f_\tau=g_\tau=\begin{bmatrix}0&1\\1&0\end{bmatrix}=(0,1)\in S_{\{0,1\}^1}.$$

The proof in this section is based on the following two observations. The first observation is that, for any $h\in S_{\{0,1\}^{n-1}}$

$$\sigma=\begin{bmatrix}f&0\\0&g\end{bmatrix}=\begin{bmatrix}fh^{-1}&0\\0&fh^{-1}\end{bmatrix}\begin{bmatrix}\mathrm{id}&0\\0&hf^{-1}gh^{-1}\end{bmatrix}\begin{bmatrix}h&0\\0&h\end{bmatrix}.$$

The second observation is that, for any $q\in S_{\{0,1\}^{n-2}}$, the following $\pi\in S^{(1)}_{\{0,1\}^n}$ is actually in $SC^{(2)}_{\{0,1\}^n}$

$$\pi=\begin{bmatrix}\mathrm{id}&0&0&0\\0&\mathrm{id}&0&0\\0&0&q&0\\0&0&0&q\end{bmatrix}.$$

Notice that $hf^{-1}gh^{-1}$ shares same cycle pattern with $f^{-1}g$. If we aim to prove σ can be decomposed to identity in four steps, it suffices to show there exist $\sigma_1\in SC^{(j)}_{\{0,1\}^{n-1}}$ and $\sigma_2\in SC^{(k)}_{\{0,1\}^{n-1}}$ such that $\sigma_1\sigma_2$ has same cycle pattern with $f^{-1}g\in S_{\{0,1\}^{n-1}}$.

However, Lemma 7 indicates $\sigma_1\sigma_2$ cannot formulate a single 3/5-cycle. In contrast, we show that $\sigma_1\sigma_2$ can indeed achieve any cycle pattern free of 3/5-cycle by Proposition 3. To reduce 3/5-cycles, we develop a cycle elimination algorithm as Lemma 6, which can be absorbed into the last block of Proposition 1.

Lemma 6: For $n\geq 5$, $r_1\in[n]$, and $\sigma\in A_{\{0,1\}^n}$, there exists $\pi\in SC^{(r_1)}_{\{0,1\}^n}$ such that $\sigma\pi$ is free of 3/5-cycles.

Proof: This π is constructed in several rounds. In round-i, $\pi_i\in SC^{(r_1)}_{\{0,1\}^n}$ is performed. Let $S_{i,c}$ be the set of c-cycles in σ_{i-1} ($\sigma_0=\sigma$ and $\sigma_t=\sigma\pi_1\pi_2\cdots\pi_t$).

Denote $\zeta_i=|S_{i,1}|+|S_{i,2}|+|S_{i,3}|+|S_{i,4}|+|S_{i,5}|$. If $S_{i-1,3}\cup S_{i-1,5}\neq\emptyset$, pick an arbitrary cycle $\mathscr{C}_1$ from it. Since $\mathscr{C}_1$ is an odd cycle, there exists $\boldsymbol{u}\in\mathscr{C}_1$ such that $\boldsymbol{v}{:=}\boldsymbol{u}^{\oplus r_1}\notin\mathscr{C}_1$. Let $\mathscr{C}_2$ be the cycle where $\boldsymbol{v}$ belongs. Define

$$T=\mathscr{C}_1\cup\left\{\boldsymbol{w}\in\mathscr{C}_2\mid \mathrm{dist}^{\sigma_{i-1}}_{\min}(\boldsymbol{v},\boldsymbol{w})\leq 5\right\}.$$

Note that $|T|\leq 5+11$. Since $n\geq 5$ and $2^{n-1}>|T|-1$, there must exist $t\notin T$ such that $\boldsymbol{u}_{r_1}=\boldsymbol{t}_{r_1}$ and $\boldsymbol{s}{:=}\boldsymbol{t}^{\oplus r_1}\notin T$. Then, let $\pi_i=(\boldsymbol{u},\boldsymbol{t})(\boldsymbol{v},\boldsymbol{s})\in SC^{(r_1)}_{\{0,1\}^n}$. We will prove $\zeta_{\sigma_i}<\zeta_{\sigma_{i-1}}$, by checking the following cases.

1) $\boldsymbol{t},\boldsymbol{s}\notin\mathscr{C}_2$: Swapping $\boldsymbol{u},\boldsymbol{t}$ merges $\mathscr{C}_1$ with another cycle and similarly when swapping $\boldsymbol{v},\boldsymbol{s}$.
2) $\boldsymbol{t}\notin\mathscr{C}_2,\boldsymbol{s}\in\mathscr{C}_2$: Swapping $\boldsymbol{u},\boldsymbol{t}$ merges $\mathscr{C}_1$ with another cycle. Then swapping $\boldsymbol{v},\boldsymbol{s}$ splits new $\mathscr{C}_2$ into two cycles; and the length of neither is smaller than 6, which will not increase the number of short cycles.
3) $\boldsymbol{t}\in\mathscr{C}_2,\boldsymbol{s}\notin\mathscr{C}_2$: Swapping $\boldsymbol{u},\boldsymbol{t}$ merges $\mathscr{C}_1$ with $\mathscr{C}_2$. Then swapping $\boldsymbol{v},\boldsymbol{s}$ merges new $\mathscr{C}_2$ with another cycle.
4) $\boldsymbol{t},\boldsymbol{s}\in\mathscr{C}_2$: Swapping $\boldsymbol{u},\boldsymbol{t}$ merges $\mathscr{C}_1$ with $\mathscr{C}_2$. Then swapping $\boldsymbol{v},\boldsymbol{s}$ splits new $\mathscr{C}_2$ into two cycles; and the length of neither is smaller than 6, which will not increase the number of short cycles.

Repeat until $S_{i,3}\cup S_{i,5}=\emptyset$. Suppose this process has k rounds, then the desired permutation π is $\pi_1\pi_2\cdots\pi_k$. ■

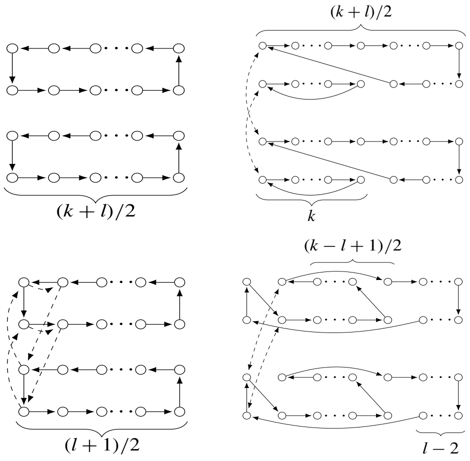
Given $r_1,r_2\in[n]$, for any $\boldsymbol{x}\in\{0,1\}^n$, define $\boldsymbol{x}_{\mathrm{out}}$ as the binary string of $\boldsymbol{x}$ throwing away the r_1th and r_2th bit; then for any $S\subseteq\{0,1\}^n$ and $a,b\in\{0,1\}$, define

$$S_{ab}=\{\boldsymbol{x}_{\mathrm{out}}\mid \boldsymbol{x}\in S,\boldsymbol{x}_{r_1}=a,\boldsymbol{x}_{r_2}=b\}.$$

Now, we present two algorithms (RPack and TPack) to generate desired cycle patterns. RPack in Algorithm 2 performs two inplace concurrent permutations to obtain a,b-cycle. For example, let $r_1=1,r_2=2$

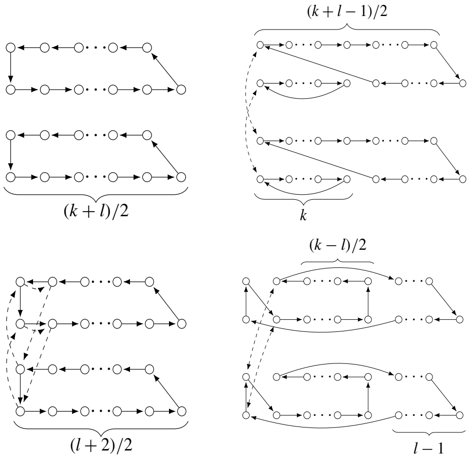
Algorithm 2: *a*, *b*-Cycle in Rectangles (RPACK)

```
Input: $r_1, r_2, a, b, S$ $(0 < a \le b)$
Output: $\pi \in SC^{(r_1)}_{\{0,1\}^n}, \tau \in SC^{(r_2)}_{\{0,1\}^n}$
/* $\pi\tau$ is $a, b$-cycle, Supp($\pi$), Supp($\tau$) $\subseteq S$ */
if ($|S| \not\equiv 0 \mod 4$) or ($|S| \ne a + b$) then
    return Error                          /* Invalid pattern */
end
if not ($S_{00} = S_{01} = S_{10} = S_{11}$) then
    return Error                          /* Invalid support */
end
$k \leftarrow \lfloor a/2 \rfloor, l \leftarrow \lfloor b/2 \rfloor$
switch $a, b$ do
    /* Fall into the first satisfied */
    case $a = b$ Top left case
    case $a$ is even Top right case
    case $a = 1, b \ge 7$ Bottom left case
    case $a$ is odd, $a, b \ge 5$ Bottom right case
    otherwise return Error
end
$\pi \leftarrow$ solid arrows, $\tau \leftarrow$ dashed arrows
return $\pi, \tau$
/* For the meaning of following figures, see
   Figure 3 and Example x */
```



Algorithm 3: *a*, *b*, *c*, *d*-Cycle in Trapezoids (TPACK)

```
Input: $r_1, r_2, a, b, c, d, S$ $(0 < a \le b, 0 < c \le d)$
Output: $\pi \in SC^{(r_1)}_{\{0,1\}^n}, \tau \in SC^{(r_2)}_{\{0,1\}^n}$
/* $\pi\tau$ is $a, b, c, d$-cycle, Supp($\pi$), Supp($\tau$) $\subseteq S$ */
if ($|S| \not\equiv 0 \mod 4$) or ($|S| \ne a + b + c + d$) then
    return Error                          /* Invalid pattern */
end
if not ($S_{00} = S_{01} = S_{10} = S_{11}$) then
    return Error                          /* Invalid support */
end
if $a + b \not\equiv 2 \mod 4$ then
    return Error                          /* Invalid pattern */
end
Pick $T \subseteq S_{00}, |T| = \lfloor (a + b)/4 \rfloor$ and $\boldsymbol{t} \in S_{00} \backslash T$
$X_0 \leftarrow \{\boldsymbol{x} \in S \mid (\boldsymbol{x}_{\text{out}} \in T_0) \vee (\boldsymbol{x}_{\text{out}} = \boldsymbol{t} \wedge \boldsymbol{x}_{r_2} = 1)\}$
$X_1 \leftarrow S \backslash X_0$
$\pi \leftarrow \text{id}, \tau \leftarrow \text{id}$
foreach $(u, v, i) \in \{(a, b, 0), (c, d, 1)\}$ do
    /* Supp($\pi_i$), Supp($\tau_i$) $\subseteq X_i$ */
    if $u = v = 1$ then Skip the following $k \leftarrow \lfloor u/2 \rfloor, l \leftarrow \lfloor v/2 \rfloor$
    switch $u, v$ do
        /* Fall into the first satisfied */
        case $u = v$ Top left case
        case $u$ is even Top right case
        case $u = 1, v \ge 7$ Bottom left case
        case $u$ is odd, $u, v \ge 5$ Bottom right case
        otherwise return Error
    end
    $\pi_i \leftarrow$ solid arrows, $\tau_i \leftarrow$ dashed arrows
    $\pi \leftarrow \pi\pi_i, \tau \leftarrow \tau\tau_i$
end
return $\pi, \tau$
/* For the meaning of following figures, see
   Figure 3 and Example x */
```



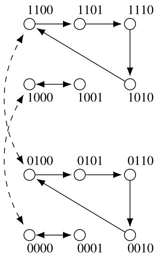


Fig. 9. Example of Algorithm 2.

and $a = 4, b = 6$

$$S = \{0000, 0001, 0010, 0100, 0101, 0110 \\ 1000, 1001, 1010, 1100, 1101, 1110\}.$$

As in Fig. 9, RPACK(r_1, r_2, a, b, S) returns

$$\tau = (1100, 0100)(1000, 0000)$$
$$\pi = (1100, 1101, 1110, 1010)(1000, 1001) \\ \times (0100, 0101, 0110, 0010)(0000, 0001).$$

The aim of TPACK in Algorithm 3 is to obtain a, b, c, d-cycle. It first divides the general rectangle shaped S into two trapezoid shaped X_0 and X_1, then performs two inplace concurrent permutations on X_0 and X_1 to obtain a, b-cycle and c, d-cycle, respectively. Since a, b-cycle and c, d-cycle are generated separately on X_0 and X_1, these two parts can be performed simultaneously, thus can be combined together.

Now, we give the proof of Proposition 3, which states two CCRBFs can compose most of the patterns.

Proof of Proposition 3: Without loss of generality, assume $r_1 = 1$ and $r_2 = 2$. Let c_k be the number of k-cycles in σ and c_1 is the number of fix-points.

Now, we initialize $\pi = \tau = \text{id}, T = \{0, 1\}^{n-2}$ and construct them in two stages.

Stage I (Pairing): Initialize the set of pairs as $P = \emptyset$.

1) Pick i with $c_i > 0$ and update $c_i \leftarrow c_i - 1$.

2) Pick j with $c_j > 0, i + j \equiv 0 \mod 2$ and update $c_j \leftarrow c_j - 1$.
3) Swap i, j if $i > j$. Then add (i, j) to P.

Repeat the procedure until $c_i = 0$ for any i.

Since σ is even, we have $\sum_i c_{2i} \equiv 0 \mod 2$. Meanwhile, $\sum_i c_{2i-1} \equiv \sum_k kc_k \equiv 2^n \equiv 0 \mod 2$. Thus, as long as the first step succeeds, the second step will not fail.

Stage II (Construct): Now, we construct π, τ.

1) Pick $(a, b) \in P$ and remove it from P.
2) If $a + b \equiv 0 \mod 4$, select $S \subseteq T, |S| = (a + b)/4$. Let
$$\pi', \tau' \leftarrow \text{RPACK}\Big(r_1, r_2, a, b, \{0, 1\}^2 \times T\Big).$$
3) If $a + b \equiv 2 \mod 4$, pick $(c, d) \in P, c + d \equiv 2 \mod 4$ and remove it from P. Select $S \subseteq T, |S| = (a + b + c + d)/4$. Let
$$\pi', \tau' \leftarrow \text{TPACK}\Big(r_1, r_2, a, b, c, d, \{0, 1\}^2 \times T\Big).$$
4) Update $T \leftarrow T \backslash S, \pi \leftarrow \pi\pi', \tau \leftarrow \tau\tau'$.

Repeat the procedure until $P = \emptyset$.

Since $\sum_{(a,b)\in P} a + b = 2^n$ and $n \geq 4$, if there is $a + b \equiv 2 \mod 4$ then there must be another pair $c+d \equiv 2 \mod 4$. Also, σ is free of 3/5-cycle, thus, RPACK and TPACK will not err.

Since π', τ''s are inplace and separate, and π, τ is the desired permutation. ■

Combining these results, finally, we are able to prove Proposition 2.

Proof of Proposition 2: Without loss of generality, we assume $r_1 = 1$ and $r_2 = 2$. Since $\sigma \in A^{(r_1)}_{\{0,1\}^n}$, there exist $f, g \in S_{\{0,1\}^{n-1}}$ such that
$$\sigma = \begin{bmatrix} f & 0 \\ 0 & g \end{bmatrix}.$$
Let $\pi_1 = \begin{bmatrix} \mathsf{id} & 0 \\ 0 & g' \end{bmatrix}$, we have
$$\begin{aligned}\sigma\pi_1 &= \begin{bmatrix} f & 0 \\ 0 & g \end{bmatrix}\begin{bmatrix} \mathsf{id} & 0 \\ 0 & g' \end{bmatrix} \\ &= \begin{bmatrix} fh^{-1} & 0 \\ 0 & fh^{-1} \end{bmatrix}\begin{bmatrix} \mathsf{id} & 0 \\ 0 & hf^{-1}gg'h^{-1} \end{bmatrix}\begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix}\end{aligned}$$
where $f, g', g, h \in S_{\{0,1\}^{n-1}}$, and g', h shall be determined later.

Since $f^{-1}g$ is even, by Lemma 6, there exists $g' \in SC^{(r_2)}_{\{0,1\}^{n-1}}$ such that $f^{-1}gg'$ is free of 3/5-cycle. Then by Proposition 3, there exist $\rho_1 \in SC^{(r_4)}_{\{0,1\}^{n-1}}$ and $\rho_2 \in SC^{(r_3)}_{\{0,1\}^{n-1}}$ such that $\rho_1\rho_2$ has the same cycle pattern as $f^{-1}gg'$. This condition is equal to that there exists $h \in S_{\{0,1\}^{n-1}}$ such that $hf^{-1}gg'h^{-1} = \rho_1\rho_2$. Therefore
$$\sigma\pi_1 = \begin{bmatrix} fh^{-1} & 0 \\ 0 & fh^{-1} \end{bmatrix}\begin{bmatrix} \mathsf{id} & 0 \\ 0 & \rho_1 \end{bmatrix}\begin{bmatrix} \mathsf{id} & 0 \\ 0 & \rho_2 \end{bmatrix}\begin{bmatrix} h & 0 \\ 0 & h \end{bmatrix}.$$
Then setting
$$\pi_1 = \begin{bmatrix} \mathsf{id} & 0 \\ 0 & g' \end{bmatrix}, \sigma_1 = \begin{bmatrix} h^{-1} & 0 \\ 0 & h^{-1} \end{bmatrix}, \tau_1 = \begin{bmatrix} \mathsf{id} & 0 \\ 0 & \rho_2^{-1} \end{bmatrix}$$
$$\tau_2 = \begin{bmatrix} \mathsf{id} & 0 \\ 0 & \rho_1^{-1} \end{bmatrix}, \sigma_2 = \begin{bmatrix} hf^{-1} & 0 \\ 0 & hf^{-1} \end{bmatrix}$$
will do. ■

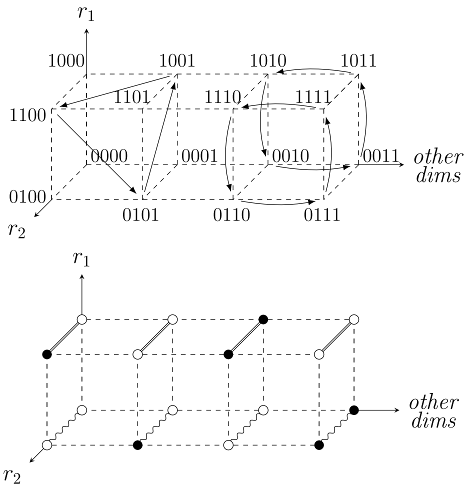


Fig. 10. Visualize σ on a colored cube.

For completeness, we show in Lemma 7 that the restriction that the cycle pattern contains no 3/5-cycle is inevitable. The proof can be found at the Appendix of [20].

Lemma 7: For any $\sigma_1 \in SC^{(r_1)}_{\{0,1\}^n}$ and $\sigma_2 \in SC^{(r_2)}_{\{0,1\}^n}$, $\sigma_1\sigma_2$ cannot be a permutation that is merely a 3-cycle or a 5-cycle.

VI. Explicit Example of Our Algorithm

In this section, we decompose a specified $\sigma \in A_{\{0,1\}^4}$ to 7 blocks of 3-bit RBFs by our algorithm. Here
$$\begin{aligned}\sigma = (1001, 1100, 0101)(1110, 0110, 0111, 1111) \\ (1010, 0010, 0011, 1011).\end{aligned}$$

A. Transform σ to CRBF

Step 1: Choose $r_1 = 1$ and $r_2 = 2$. Using the method in Section IV-A, we construct colored cube for σ as Fig. 10.

Read the colored cube, we get $a_1 = 1, a_2 = 0, a_3 = 1, a_4 = 2$; and $b_1 = 1, b_2 = 0, b_3 = 1, b_4 = 2$.

Step 2: Check Lemmas 1 and 3, we find this case falls into Lemma 1. we can transform σ to $S^{(1)}_{\{0,1\}^4}$ by $SC^{(2)}_{\{0,1\}^4}$, and $SC^{(1)}_{\{0,1\}^4}$ and $SC^{(2)}_{\{0,1\}^4}$ by Lemma 1. Specific constructions are as follows.

Step 2.1: Using Lemma 2, we transform σ to canonical form by $\pi = \pi_1\pi_2$. Let
$$\pi_1 = (1110, 0111)(1010, 0011)$$
which transforms the colored cube to a cube with $a_3 = b_3 = 0, a_2 = 0$. Setting $\pi_2 = (0100, 0101)(0000, 0001)$, it rearranges the cube to canonical form. The process is pictured as Figs. 11 and 12.

Step 2.2: Using Lemma 1, we construct the following CCRBFs:
$$\begin{aligned}\pi_3 &= (0100, 1110)(0000, 1010) \\ &\quad (1101, 0110)(1001, 0010) \\ \pi_4 &= (1000, 1010)(0000, 0010) \\ \pi_5 &= (1100, 0100)(1000, 0000) \\ &\quad (1101, 0110)(1001, 0010).\end{aligned}$$

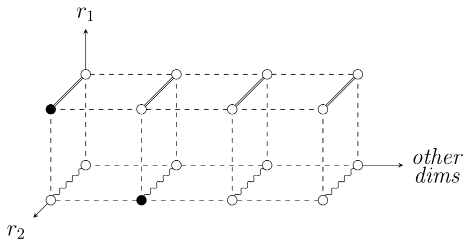


Fig. 11. Colored cube for $\sigma\pi_1$.

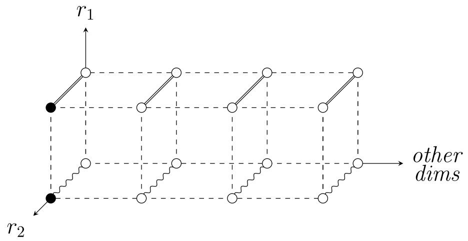


Fig. 12. Colored cube for $\sigma\pi_1\pi_2$.

It is easy to verify $\pi_1, \pi_2, \pi_3, \pi_5 \in SC^{(2)}_{\{0,1\}^4}$,$\pi_4 \in SC^{(1)}_{\{0,1\}^4}$, and

$$\begin{aligned}\pi_1\pi_2\pi_3 = {} & (0000, 0011, 1010, 0001)\\ & (0100, 0111, 1110, 0101)(0010, 1001)(0110, 1101).\end{aligned}$$

And, finally, we transform the colored cube for σ to a white cube by verifying

$$\begin{aligned}\sigma^{(1)} &= \sigma(\pi_1\pi_2\pi_3)\pi_4\pi_5\\ &= (0000, 0001)(0010, 0011)(0100, 0101)\\ &\quad (0110, 0111)(1000, 1100, 1111, 1110\\ &\quad 1001, 1011, 1010).\end{aligned}$$

B. Transform $\sigma^{(1)}$ to Identity

We can use two 3-bit RBFs to represent $\sigma^{(1)}$. That is

$$\begin{aligned}f &= (000, 001)(010, 011)(100, 101)(110, 111)\\ g &= (000, 100, 111, 110, 001, 011, 010)\end{aligned}$$

such that

$$\sigma^{(1)} : (0, \boldsymbol{x}) \to (0, f(\boldsymbol{x})), (1, \boldsymbol{x}) \to (1, g(\boldsymbol{x})).$$

Step 1: Determine whether $f^{-1}g$ has 3/5-cycle. By directly calculating $f^{-1}g$, we know the answer is no. So we can jump the process for eliminating 3/5-cycles

$$f^{-1}g = (000, 101, 100, 110)(001, 010).$$

Step 2: First, we construct a $\sigma_1 \in SC^{(1)}_{\{0,1\}^3}$, and a $\sigma_2 \in SC^{(2)}_{\{0,1\}^3}$ to generate a 2, 4-cycle pattern like $f^{-1}g$. Based on the TPACK algorithm

$$\begin{aligned}\sigma_1 &= (000, 011)(100, 111)\\ \sigma_2 &= (010, 110)(000, 100).\end{aligned}$$

Step 3: Find $h \in S_{\{0,1\}^3}$ such that $h(f^{-1}g)h^{-1} = \sigma_1\sigma_2$. By group theory, we know that if $\tau = (i_1, i_2, \ldots, i_k)$, then $h\tau h^{-1} = (h(i_1), h(i_2), \ldots, h(i_k))$. So we can construct

$$h = (101, 111)(001, 010, 110, 011).$$

Step 4: Now, we verify $h(f^{-1}g)h^{-1} = \sigma_1\sigma_2$. Thus

$$\begin{aligned}\sigma^{(1)} &= \begin{bmatrix} f & \\ & g \end{bmatrix}\\ &= \begin{bmatrix} fh^{-1} & \\ & fh^{-1} \end{bmatrix}\begin{bmatrix} \mathrm{id} & \\ & \sigma_1 \end{bmatrix}\begin{bmatrix} \mathrm{id} & \\ & \sigma_2 \end{bmatrix}\begin{bmatrix} h & \\ & h \end{bmatrix}\\ &\triangleq \pi_6\pi_7\pi_8\pi_9.\end{aligned}$$

Written in the form of permutation cycle pattern

$$\begin{aligned}\pi_6 &= (0000, 0001, 0010)(0011, 0111, 0100, 0101, 0110)\\ &\quad \times (1000, 1001, 1010)(1011, 1111, 1100, 1101, 1110)\\ \pi_7 &= (1000, 1011)(1100, 1111)\\ \pi_8 &= (1010, 1110)(1000, 1100)\\ \pi_9 &= (0101, 0111)(0001, 0010, 0110, 0011)\\ &\quad \times (1101, 1111)(1001, 1010, 1110, 1011).\end{aligned}$$

C. Summary

In a word, $\sigma = \pi_6\pi_7\pi_8\pi_9\pi_5^{-1}\pi_4^{-1}(\pi_1\pi_2\pi_3)^{-1}$, where

$$\begin{aligned}&\pi_6, \pi_9, \pi_4^{-1} \in SC^{(1)}_{\{0,1\}^4}\\ &\pi_7, \pi_5^{-1}, (\pi_1\pi_2\pi_3)^{-1} \in SC^{(2)}_{\{0,1\}^4}\\ &\pi_8 \in SC^{(3)}_{\{0,1\}^4}.\end{aligned}$$

VII. Even Block Depth

In the previous sections, we prove that for any $\sigma \in A_{\{0,1\}^n}$, $n \geq 6$, σ has block depth 7. However, the block itself may be an odd permutation which resists further decomposition. In this section, we address this concern and show that any $\sigma \in A_{\{0,1\}^n}$, with $n \geq 10$, has even block depth 10, which is stated as Theorem 2. This is proven by some modification of the framework in the previous sections. The idea is similar, but the analysis is much more complicated. Here, we only sketch the proof and leave the detail into the Appendix of [20].

We prove Theorem 2 by the modified versions of Propositions 1 and 2. Specifically, we prove that arbitrary even n-bit permutation can be transformed to even CRBF by three even blocks; arbitrary even CRBF can be transformed to identity by eight even blocks. Choosing carefully, we can merge some of them and finally decompose even n-bit permutation to identity using ten even blocks. The results are summarized as the following two propositions.

Proposition 4: For $n \geq 4, \sigma \in A_{\{0,1\}^n}$, and $r_1 \in [n]$, there exist at least $(n-2)$ different $r_2 \in [n]\backslash\{r_1\}$ such that there exist $\sigma_1 \in AC^{(r_1)}_{\{0,1\}^n}, \pi_1, \pi_2 \in AC^{(r_2)}_{\{0,1\}^n}$ satisfying $\sigma\pi_1\sigma_1\pi_2 \in A^{(r_1)}_{\{0,1\}^n}$.

Here, we only give the intuition. The key observation in the proof of Lemma 1 is that we can always swap some nodes without changing color in cuboid. For example, if we swap two nodes who has the same color and lie in the same face, then the corresponding colored cuboid will not change. This observation can be used to modify the permutation to be concurrently even.

For example, we can transform two B-cards to white cube by the following two methods.

Proposition 5 states that we can recover any even n-bit CRBF by eight concurrently even CCRBFs.

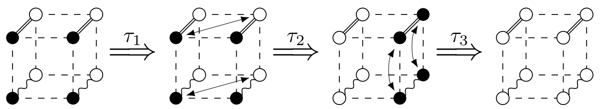


Fig. 13. Transform two B-cards to identity where τ_1 is concurrently even, τ_2 and τ_3 are concurrently odd.

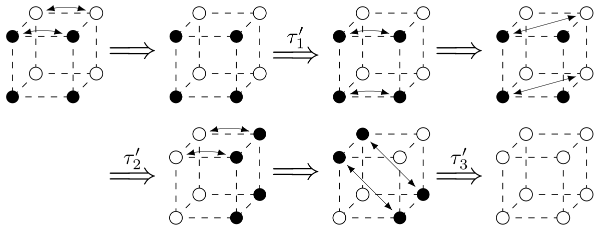


Fig. 14. Transform two B-cards to identity where τ_1' is concurrently odd, τ_2' and τ_3' are concurrently even.

Proposition 5: For $n \geq 10$, $r_1 \in [n]$, $\sigma \in A^{(r_1)}_{\{0,1\}^n}$, and distinct $r_2, r_3, r_4 \in [n]/\{r_1\}$. There exist $\sigma_1, \sigma_4, \sigma_7 \in AC^{(r_1)}_{\{0,1\}^n}$, $\sigma_6, \sigma_8 \in AC^{(r_2)}_{\{0,1\}^n}$, $\sigma_2, \sigma_5 \in AC^{(r_3)}_{\{0,1\}^n}$, $\sigma_3 \in AC^{(r_4)}_{\{0,1\}^n}$ such that $\sigma = \sigma_1 \circ \cdots \circ \sigma_8$.

Similar to the proof of Proposition 2, here, we first construct a concurrently even CCRBF π such that $\sigma\pi$ is free of 3/5-cycle and $\sigma\pi$ has an even cycle. Then we use concurrently even CCRBFs to formulate cycles. Besides, we need to solve some special cases. Those proofs are similar to the corresponding ones and can be found at the Appendix of [20].

Here is the new lemma for eliminating cycles.

Lemma 8: For $n \geq 8$, $r_1 \in [n]$, and $\sigma \in A_{\{0,1\}^n}$, there exists $\pi \in AC^{(r_1)}_{\{0,1\}^n}$ such that $\sigma\pi$ is free of 3/5-cycle, and $\sigma\pi$ has at least an even cycle.

The additional demand for an even cycle comes from the following lemma.

Lemma 9: For $\sigma, \pi \in S_{\{0,1\}^n}$. σ, π have the same cycle pattern and σ has an even cycle. Then there exists $h \in A_{\{0,1\}^n}$ such that $h\sigma h^{-1} = \pi$.

Lemmas 8 and 9 ensure that cycle pattern can be constructed by two concurrently even CCRBFs on different dimensions *under some restrictions*.

Lemma 10: For $\sigma \in A_{\{0,1\}^n}$ which is free of 3/5-cycle and contains at least 12 cycles with the length of at least 2, there exist $\pi \in AC^{(r_1)}_{\{0,1\}^n}$ and $\tau \in AC^{(r_2)}_{\{0,1\}^n}$ such that $\pi\tau$ has the same cycle pattern with σ.

Lemma 11: For $\sigma \in A_{\{0,1\}^n}$ which is free of 3/5-cycle and contains a cycle with the length of at least 12, there exist $\pi \in AC^{(r_1)}_{\{0,1\}^n}$ and $\tau \in AC^{(r_2)}_{\{0,1\}^n}$ such that $\pi\tau$ has the same cycle pattern with σ.

The last preparation is to construct a concurrently odd CCRBF by four concurrently even CCRBFs.

Lemma 12: For $n \geq 3$, distinct $r_1, r_2, r_3 \in [n]$, there exists concurrently odd $\pi \in SC^{(r_1)}_{\{0,1\}^n}$, such that $\pi = \tau_1\tau_2\tau_3\tau_4$, where $\tau_1 \in AC^{(r_3)}_{\{0,1\}^n}$, $\tau_2, \tau_4 \in AC^{(r_2)}_{\{0,1\}^n}$, and $\tau_3 \in AC^{(r_1)}_{\{0,1\}^n}$.
Finally, we give proof of Proposition 5.

Proof of Proposition 5: Without loss of generality, assume $r_1 = 1$ and $r_2 = 2$. Similar to the proof of Proposition 2, since $\sigma \in A^{(r_1)}_{\{0,1\}^n}$, there exist $f, g \in S_{\{0,1\}^{n-1}}$ such that $\sigma = \begin{bmatrix} f & \\ & g \end{bmatrix}$. Observe that for any $g', s, h \in S_{\{0,1\}^{n-1}}$, let $\pi_9 = \begin{bmatrix} \mathsf{id} & \\ & g' \end{bmatrix}$, we have

$$\sigma\pi_9 = \begin{bmatrix} fsh & \\ & fsh \end{bmatrix} \begin{bmatrix} \mathsf{id} & \\ & h^{-1}(fs)^{-1}(gg's)h \end{bmatrix} \\ \times \begin{bmatrix} h^{-1} & \\ & h^{-1} \end{bmatrix} \begin{bmatrix} s^{-1} & \\ & s^{-1} \end{bmatrix}.$$

We first use Lemma 8 to choose $g' \in AC^{(r_2)}_{\{0,1\}^{n-1}}$, such that $f^{-1}gg'$ is free of 3/5-cycle and has an even cycle. For convenience, we perform another preprocessing. Technically, if $f^{-1}gg'$ has a cycle of length ≥ 12 or has at least 12 cycles, we do nothing. Otherwise, there are at least 13 fix-point pairs $(\boldsymbol{x}_1, \boldsymbol{y}_1), \ldots, (\boldsymbol{x}_{13}, \boldsymbol{y}_{13})$ in $f^{-1}gg'$ satisfying $(\boldsymbol{x}_i)_{r_1} = (\boldsymbol{y}_i)_{r_1} = 1$ and $\boldsymbol{x}_i = \boldsymbol{y}_i^{\oplus r_2}$ for all $i \in [13]$ since $n \geq 10$. Thus, we can perform $g'' \in AC^{(r_2)}_{\{0,1\}^{n-1}}$ to add two 13-cycles without affecting other cycle in $f^{-1}gg'$. For simplicity, we update g' as $g'g''$.

Since σ, π_9 are even, and f, gg' are either both even or both odd. If f, gg' are both even, we choose $s = \mathsf{id}$. If otherwise, using Lemma 12, we choose concurrently odd $\begin{bmatrix} s^{-1} & \\ & s^{-1} \end{bmatrix} \in SC^{(r_1)}_{\{0,1\}^n}$, where s^{-1} is odd, and construct it with four even blocks in order r_3, r_2, r_1, r_2 (i.e., $\pi_5, \pi_6, \pi_7, \pi_8$). Then $fs, gg's$ will be both even.

Next, we synthesize $\begin{bmatrix} \mathsf{id} & \\ & h^{-1}(fs)^{-1}(gg's)h \end{bmatrix}$. Note that $f^{-1}gg'$ either contains at least 12 cycles, or contains a long cycle of length at least 12. According to Lemmas 10 and 11, there exist $\tau_1 \in AC^{(r_3)}_{\{0,1\}^{n-1}}$ and $\tau_2 \in AC^{(r_4)}_{\{0,1\}^{n-1}}$ such that $\tau_1\tau_2$ has the same cycle pattern with $f^{-1}gg'$ and $(fs)^{-1}gg's$. Furthermore, since $f^{-1}gg'$ has an even cycle, by Lemma 9, there exists $h \in A_{\{0,1\}^{n-1}}$ such that $\tau_1\tau_2 = h^{-1}(fs)^{-1}gg'sh$.

To sum up, let $\pi_1 = \begin{bmatrix} (fs)h & \\ & (fs)h \end{bmatrix}$, $\pi_2 = \begin{bmatrix} id & \\ & \tau_1 \end{bmatrix}$, $\pi_3 = \begin{bmatrix} id & \\ & \tau_2 \end{bmatrix}$, and $\pi_3 = \begin{bmatrix} h^{-1} & \\ & h^{-1} \end{bmatrix}$. Then $\pi_1, \pi_4, \pi_7 \in AC^{(r_1)}_{\{0,1\}^n}$, $\pi_6, \pi_8, \pi_9 \in AC^{(r_2)}_{\{0,1\}^n}$, $\pi_2, \pi_5 \in AC^{(r_3)}_{\{0,1\}^n}$, $\pi_3 \in AC^{(r_4)}_{\{0,1\}^n}$, and

$$\sigma = \pi_1\pi_2\pi_3\pi_4\pi_5\pi_6\pi_7(\pi_8\pi_9^{-1}).$$

■

VIII. Conclusion and Open Questions

In this paper, we offer a method to decompose arbitrary even n-bit RBF into seven blocks of $(n-1)$-bit RBFs for $n \geq 6$, or into ten blocks of even $(n-1)$-bit RBFs for $n \geq 10$, where the blocks have certain freedom to choose. Technically, we first transform even RBF to an even CRBF by three blocks. Then we transform the even CRBF to identity by five blocks. In addition, the last block of the first step can be merged with the first block of the second step, thus providing a 7-depth decomposition. The road map of even block depth is similar but much more complicated.

One direct open question is whether the constants 7 and 10 are optimal or can be further improved. Besides, one practical issue is trying to relax the conditions that $n \geq 6$ and $n \geq 10$.

Another interesting question is, given an even n-bit RBF, if we are allowed to use general unitary blocks to synthesize it, can we use strictly fewer blocks than only using RBF blocks?

REFERENCES

[1] P. Selinger, “A finite alternation result for reversible Boolean circuits,” *Sci. Comput. Program.*, vol. 151, pp. 2–17, Jan. 2018.

[2] C. H. Bennett, “Notes on the history of reversible computation,” *IBM J. Res. Develop.*, vol. 32, no. 1, pp. 16–23, Jan. 1988.

[3] M. Saeedi and I. L. Markov, “Synthesis and optimization of reversible circuits—A survey,” *ACM Comput. Surveys*, vol. 45, no. 2, p. 21, 2013.

[4] M. Arabzadeh, M. Saeedi, and M. S. Zamani, “Rule-based optimization of reversible circuits,” in *Proc. Asia South Pac. Design Autom. Conf.*, 2010, pp. 849–854.

[5] R. Landauer, “Irreversibility and heat generation in the computing process,” *IBM J. Res. Develop.*, vol. 5, nos. 1–2, pp. 261–269, 1961.

[6] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge, U.K.: Cambridge Univ. Press, 2010.

[7] P. W. Shor, “Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer,” *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[8] L. K. Grover, “A fast quantum mechanical algorithm for database search,” in *Proc. 28th ACM Symp. Theory Comput.*, 1996, pp. 212–219.

[9] V. V. Shende, A. K. Prasad, I. L. Markov, and J. P. Hayes, “Synthesis of reversible logic circuits,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 22, no. 6, pp. 710–722, Jun. 2003.

[10] D. Maslov, G. W. Dueck, D. M. Miller, and C. Negrevergne, “Quantum circuit simplification and level compaction,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 27, no. 3, pp. 436–444, Mar. 2008.

[11] Y. Takahashi, S. Tani, and N. Kunihiro, “Quantum addition circuits and unbounded fan-out,” *Quantum Inf. Comput.*, vol. 10, no. 9, pp. 872–890, 2010.

[12] M. Saeedi, M. S. Zamani, M. Sedighi, and Z. Sasanian, “Reversible circuit synthesis using a cycle-based approach,” *ACM J. Emerg. Technol. Comput. Syst.*, vol. 6, no. 4, p. 13, 2010.

[13] M. Saeedi, M. Arabzadeh, M. S. Zamani, and M. Sedighi, “Block-based quantum-logic synthesis,” *Quantum Inf. Comput.*, vol. 11, no. 3, pp. 262–277, 2011.

[14] V. V. Shende, S. S. Bullock, and I. L. Markov, “Synthesis of quantum-logic circuits,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 25, no. 6, pp. 1000–1010, Jun. 2006.

[15] S. J. Devitt, “Performing quantum computing experiments in the cloud,” *Phys. Rev. A*, vol. 94, no. 3, 2016, Art. no. 032329.

[16] D. P. DiVincenzo, “The physical implementation of quantum computation,” *Fortschritte Der Physik Progr. Phys.*, vol. 48, nos. 9–11, pp. 771–783, 2000.

[17] A. Zulehner, A. Paler, and R. Wille, “An efficient methodology for mapping quantum circuits to the IBM QX architectures,” *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 38, no. 7, pp. 1226–1236, Jul. 2019.

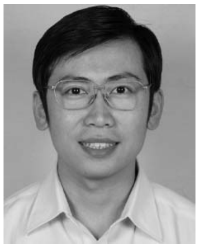
[18] C. G. Almudever *et al.*, “The engineering challenges in quantum computing,” in *Proc. Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2017, pp. 836–845.

[19] M. Veldhorst, H. G. J. Eenink, C. H. Yang, and A. S. Dzurak, “Silicon CMOS architecture for a spin-based quantum computer,” *Nat. Commun.*, vol. 8, no. 1, p. 1766, 2017.

[20] J. Jiang, X. Sun, Y. Sun, K. Wu, and Z. Xia, “Structured decomposition for reversible Boolean functions,” *arXiv preprint arXiv:1810.04279*, 2018.

Jiaqing Jiang received the B.S. degree in mathematical science from Nankai University, Tianjin, China, in 2017. She is currently pursuing the master’s degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences, Beijing.

Under the guidance of Prof. X. Sun, Prof. J. Zhang, and Prof. G. Tian, she has published and coauthored two papers in *Physical Review A* and one journal in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. Her current research interests include quantum circuit synthesis and quantum algorithms.

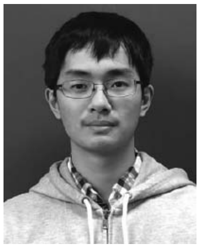


Xiaoming Sun received the B.S. and Ph.D. degrees from Tsinghua University, Beijing, China, in 2001 and 2005, respectively.

He is a Professor with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, and the University of Chinese Academy of Sciences, Beijing. His current research interests include quantum computing, decision tree complexity, communication complexity, and combinatorics.

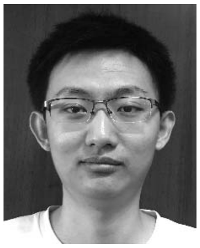
Yuan Sun received the B.E. degree in mathematics from the University of Science and Technology of China, Hefei, China, in 2017. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences, Beijing, under the guidance of Dr. X. Sun.

He has published and coauthored two papers in International Computing and Combinatorics Conference and one paper in the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. His current research interests include Boolean function analysis and combinatorics.



Kewen Wu is currently pursuing the undergraduation degree in computer science (doubles in mathematics) with Peking University, Beijing, China.

With the guidance of Prof. X. Sun from the Chinese Academy of Sciences, Beijing, he developed broad interests in theoretical computer science and also cryptography. Cooperating with Prof. X. Sun and senior students, he posted several manuscripts on Arxiv and published some of them to conferences and journals, like the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS.



Zhiyu Xia received the B.S. degree in software engineering from Sichuan University, Chengdu, China, in 2016. He is currently pursuing the Ph.D. degree with the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, and the University of Chinese Academy of Sciences, Beijing, with the guidance of Prof. X. Sun.

His current research interests include decision tree complexity, analysis of Boolean function, and combinatorial algorithms.