Minimizing ESCT Forms for Two-Variable Multiple-Valued Input Binary Output Functions*

Takaaki Mizuki Daizo Mikami Hideaki Sone

Tohoku University
tm-paper+minesctweb[atmark]g-mail.tohoku-university.jp

Abstract

As EXOR (Exclusive-OR) expansions of binary output functions, the ESOP (EXOR Sum of Products) form and its extension, the ESCT (EXOR Sum of Complex Terms) form, have been studied extensively in the literature. An efficient algorithm for minimizing ESOP forms is known for two-variable multiple-valued input functions. On the other hand, no ESCT minimization algorithm is known for such functions. In this paper, we give an efficient algorithm for minimizing ESCT forms of two-variable multiple-valued input functions, showing that the number of terms can be reduced by at most one.

1 Introduction

This paper deals with EXOR (Exclusive-OR) expansions of binary output functions, especially those whose inputs are two-variable and multiple-valued. Namely, this paper addresses EXOR expansions of functions $f : \mathbb{Z}_m \times \mathbb{Z}_m \to \{0, 1\}$ with $m \geq 2$ and $\mathbb{Z}_m \stackrel{\text{def}}{=} \{0, 1, \ldots, m-1\}$. Below, we simply call such a function an *m-valued input function* or just a 'function.' For example, the truth table in Table 1 specifies a 5-valued input function $g(a, b)$.

		b				
		0	1	2	3	4
a	0	1	1	1	0	0
	1	0	1	0	1	0
	2	0	1	0	0	1
	3	1	0	1	0	1
	4	1	0	1	1	0

Table 1: A truth table for the 5-valued input function $g(a, b)$.

As an expansion of EXOR, we begin with the *ESOP (EXOR Sum of Products) form*, which combines products $a^V \wedge b^W$ of literals a^V, b^W by EXORs $\oplus$.

*NOTICE: this is the author's version of a work that was accepted for publication in Discrete Applied Mathematics. Changes resulting from the publishing process, such as peer review, editing, corrections, structural formatting, and other quality control mechanisms may not be reflected in this document. Changes may have been made to this work since it was submitted for publication. A definitive version is subsequently published in Discrete Applied Mathematics, DOI: 10.1016/j.dam.2013.12.023.

1.1 ESOP forms

The following is an ESOP form of the 5-valued input function g given in Table 1 (omitting the conjunction symbol $\wedge$):

$$g(a,b) = a^{\{0,3,4\}}b^{\{0,1,2\}} \oplus a^{\{1,3,4\}}b^{\{1,3\}} \oplus a^{\{2\}}b^{\{1,4\}} \oplus a^{\{3\}}b^{\{3,4\}}, \tag{1}$$

where *literals* a^V and b^W with $V, W \subseteq \mathbb{Z}_m$ are defined as

$$a^V = a^V(a) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } a \in V; \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and

$$b^W = b^W(b) \stackrel{\text{def}}{=} \begin{cases} 1 & \text{if } b \in W; \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

For instance, the leading literal $a^{\{0,3,4\}}$ of the ESOP form in Eq. (1) becomes 1 if $a \in \{0,3,4\}$; and it becomes 0 if $a \in \{1,2\}$. One can verify the equality in Eq. (1) (as can also later be seen in Eq. (11) of Section 2.1).

Note that while the ESOP form in Eq. (1) above has 4 product terms, the following ESOP form of g has only 3 terms:

$$g(a,b) = a^{\{0,3,4\}}b^{\{0,1,2\}} \oplus a^{\{1,2,3,4\}}b^{\{1,3\}} \oplus a^{\{2,3\}}b^{\{3,4\}}. \tag{4}$$

Thus, there exists a minimization problem regarding the ESOP form. For example, in this case of g there is no ESOP form with less than 3 terms (as seen below), and hence the ESOP form in Eq. (4) is a minimal form of g. Generally, a *minimal ESOP form* of a function f is one having the minimum number of terms among all possible ESOP forms representing f. We denote by $\tau_{\text{ESOP}}(f)$ the number of terms in any minimal ESOP form of f. For example, we can write $\tau_{\text{ESOP}}(g) = 3$ for the function g above.

The problem of minimization of ESOP forms has been investigated in the literature for many decades (e.g. [1, 6, 8, 9, 10]). For the addressed case of two-variable and multiple-valued input, a recently developed efficient algorithm [3] showed that the minimum number $\tau_{\text{ESOP}}(f)$ for a function f (which is not identically zero) is equal to the rank of the binary matrix M_f, denoted by $\text{rank}(M_f)$, which corresponds to the truth table of f. In other words,

$$\tau_{\text{ESOP}}(f) = \text{rank}(M_f). \tag{5}$$

For example, the binary matrix M_g for the function g given in Table 1 is

$$M_g = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}, \tag{6}$$

and since one can calculate $\text{rank}(M_g) = 3$ (computed in modulo 2), we surely have

$$\tau_{\text{ESOP}}(g) = 3.$$

(Throughout this paper, given an m-valued input function f, the corresponding $m \times m$ binary matrix M_f is defined as above. That is, the (i,j)-entry of M_f for every i and j with $1 \leq i, j \leq m$ is set equal to the value of $f(i,j)$. Refer to [11] for matrix theory terminology.)

1.2 ESCT forms

We next visit the *ESCT (EXOR Sum of Complex Terms) form*, which is an extension of the ESOP form. While an ESOP form consists of product terms $a^V b^W$ as seen in the previous subsection, an ESCT form consists of *complex terms* $a^{U,V}(b^W)$ defined as

$$a^{U,V}(b^W) \stackrel{\text{def}}{=} \begin{cases} a^U & \text{if } b^W = 0; \\ a^V & \text{if } b^W = 1, \end{cases}$$

where a^U, a^V and b^W are literals defined in Eqs. (2) and (3) above. For example,

$$g(a,b) = a^{\{2,3\},\{1,4\}}(b^{\{1,3\}}) \oplus a^{\emptyset,\{0,2,4\}}(b^{\{0,1,2\}}) \tag{7}$$

is an ESCT form of the function g given in Table 1. The first complex term $a^{\{2,3\},\{1,4\}}(b^{\{1,3\}})$ becomes 1 only when either (i) $b^{\{1,3\}} = 0$ and $a \in \{2,3\}$ or (ii) $b^{\{1,3\}} = 1$ and $a \in \{1,4\}$ (the equality in Eq. (7) can be verified later in Section 3.2).

Since a literal $a^{\emptyset}$ is just a constant 0 (recall the definition of a literal shown in Eq. (2)), the second term $a^{\emptyset,\{0,2,4\}}(b^{\{0,1,2\}})$ of the ESCT form in Eq. (7) becomes 0 whenever $b^{\{0,1,2\}} = 0$ (or $b \notin \{0,1,2\}$). On the other hand, it becomes $a^{\{0,2,4\}}$ if $b^{\{0,1,2\}} = 1$ (or $b \in \{0,1,2\}$). Thus, note that the complex term $a^{\emptyset,\{0,2,4\}}(b^{\{0,1,2\}})$ can be regarded as exactly a product term $a^{\{0,2,4\}}b^{\{0,1,2\}}$. Generalizing this, for any $V, W \subseteq \mathbb{Z}_m$, we have

$$a^V b^W = a^{\emptyset,V}(b^W), \tag{8}$$

which means that the complex term is a generalization of the product term, and hence the ESOP form is a special case of the ESCT form.

Since the ESCT form is a more generic expression than the ESOP form, there is a possibility that the number of terms can be reduced by using ESCT forms instead of ESOP forms. This has been one motivation for the existing research on minimization or simplification of ESCT forms (e.g. [7, 12, 13]). Actually, while $\tau_{\text{ESOP}}(g) = 3$ as seen before, the ESCT form in Eq. (7) has only 2 terms and hence $\tau_{\text{ESCT}}(g) \leq 2$, where $\tau_{\text{ESCT}}(f)$ for a function f is defined similarly as $\tau_{\text{ESOP}}(f)$, that is, $\tau_{\text{ESCT}}(f)$ denotes the number of terms in any *minimal ESCT form* of f.

Thus, the definitions of ESOP and ESCT forms immediately imply that $\tau_{\text{ESCT}}(f) \leq \tau_{\text{ESOP}}(f)$ for any function f. This raises the question, how much smaller is $\tau_{\text{ESCT}}(f)$ than $\tau_{\text{ESOP}}(f)$?

1.3 Our results

This paper answers the question above, in the form of a polynomial-time algorithm to find a minimal ESCT form of a given function f. Specifically, we show the following theorem. Hereafter, $\vec{1}$ denotes an all-one column vector, M_f^{T} denotes the transpose of M_f, and $\mathbf{C}(M_f^{\text{T}})$ denotes the row space of M_f (or the column space of M_f^{T}), namely

$$\mathbf{C}\left(\left(\,\vec{p}_1\,\vec{p}_2\cdots\vec{p}_m\,\right)\right) \stackrel{\text{def}}{=} \left\{ \bigoplus_{i=1}^{m} r_i\vec{p}_i \;\middle|\; r_i \in \{0,1\},\ 1 \leq i \leq m \right\}$$

for column vectors $\vec{p}_1, \vec{p}_2, \ldots, \vec{p}_m$; for example, $\vec{1} \in \mathbf{C}(M_g^{\text{T}})$ for the matrix M_g given in Eq. (6) because the third and fifth rows in M_g satisfy

$$\left(\,0\,1\,0\,0\,1\,\right)^{\text{T}} \oplus \left(\,1\,0\,1\,1\,0\,\right)^{\text{T}} = \vec{1}.$$

Theorem 1. *For any (two-variable) m-valued input function f with* $\mathrm{rank}(M_f) \geq 2$*,*

$$\tau_{\mathrm{ESCT}}(f) = \begin{cases} \mathrm{rank}(M_f) - 1 & \textit{if } \vec{1} \in \mathbf{C}(M_f^{\mathrm{T}}); \\ \mathrm{rank}(M_f) & \textit{otherwise.} \end{cases}$$

Thus, the minimum number of complex terms $\tau_{\mathrm{ESCT}}(f)$ is equal to $\mathrm{rank}(M_f) - 1$ or $\mathrm{rank}(M_f)$. Note that we ruled out the case of $\mathrm{rank}(M_f) \leq 1$ in Theorem 1 because any function with $\mathrm{rank}(M_f) \leq 1$ is obviously expressed as a single term.

As seen in Eq. (5) of Section 1.1, we already know that $\tau_{\mathrm{ESOP}}(f) = \mathrm{rank}(M_f)$ [3]. Therefore, our result in Theorem 1 implies that $\tau_{\mathrm{ESCT}}(f) = \tau_{\mathrm{ESOP}}(f) - 1$ or $\tau_{\mathrm{ESCT}}(f) = \tau_{\mathrm{ESOP}}(f)$. In other words, using the ESCT form instead of the ESOP form can reduce the number of terms by at most one. For example, since $\tau_{\mathrm{ESOP}}(g) = 3$ and $\tau_{\mathrm{ESCT}}(g) \leq 2$ for g in Table 1 as seen in Sections 1.1 and 1.2, we have $\tau_{\mathrm{ESCT}}(g) = 2$ and hence the ESCT form in Eq. (7) is a minimal one of g.

Although this paper proves that the ESCT form has only slightly fewer terms than the ESOP form has, one does not need to be too pessimistic for the following reason. The original motivation for the research [3] on minimizing ESOP forms of two-variable multiple-valued input functions comes from trying to improve a cryptographic protocol [2]. The cost (communication complexity) of the cryptographic protocol developed in [2] to securely compute a function $f(a, b)$ is proportional to $\tau_{\mathrm{ESOP}}(f)$. Specifically, in that protocol, Alice with a secret bit a sends a $2\tau_{\mathrm{ESOP}}(f)$-bit message to Carol, and Bob with a secret bit b sends a $(\tau_{\mathrm{ESOP}}(f) + 1)$-bit message to Carol, where Alice and Bob have shared a $3\tau_{\mathrm{ESOP}}(f)$-bit random string, so only Carol can learn the value of $f(a, b)$. Sampson et al. [5] carried the idea behind the protocol further, showing that the ESOP form used in the protocol can just be replaced with the ESCT form. As a result, one obtains a protocol whose cost is given by replacing $\tau_{\mathrm{ESOP}}(f)$ above with just $\tau_{\mathrm{ESCT}}(f)$. Thus, even reducing only one term contributes to enhancing the efficiency of secure computations, especially when $\tau_{\mathrm{ESOP}}(f)$ $(= \mathrm{rank}(M_f))$ is small. For example, for the function g above (which satisfies $\tau_{\mathrm{ESOP}}(g) = 3$ and $\tau_{\mathrm{ESCT}}(g) = 2$), the former protocol (using the ESOP form) needs communication cost of 19 $(= 6 + 4 + 9)$ bits in total while the latter protocol (using the ESCT form) needs only 13 $(= 4 + 3 + 6)$ bits, and hence a minimal ESCT form can reduce the cost by approximately 32%.

As seen, this paper fixes the number of variables to two although the input size m of each variable can be any value. Actually, all of the existing efficient *exact* algorithms for ESOP/ESCT minimization (e.g. [9, 12, 13]) have limitations in the input size, the number of variables or the number of terms. Even an efficient exact ESOP minimization algorithm for n-variable two-valued input functions has not been known (where n takes any number). Note that one can convert a two-valued minimal ESOP form (with multiple variables) into a two-variable multiple-valued ESOP form, but it is not necessarily minimal. Take a 4-valued input function h such that

$$M_h = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 \end{pmatrix},$$

for example. The function h can be written as a 4-variable 2-valued minimal ESOP form like

$$h(x_1, x_2, x_3, x_4) = x_1 x_3 \oplus \bar{x}_1 x_2 x_3,$$

which can be converted into a 2-variable 4-valued ESOP form

$$h(a,b) = a^{\{2,3\}}b^{\{2,3\}} \oplus a^{\{1\}}b^{\{2,3\}}.$$

However, the ESOP form is not minimal because $\tau_{\mathrm{ESOP}}(h) = \mathrm{rank}(M_h) = 1$, and indeed, $h(a,b) = a^{\{1,2,3\}}b^{\{2,3\}}$.

The remainder of the paper is organized as follows. In Section 2, we introduce a method to express ESOP in a matrix multiplication form, which enables us to easily describe our algorithm. In Section 3, we present our efficient algorithm to find a minimal ESCT form of any function. In Section 4, we prove the correctness of our algorithm. This paper concludes in Section 5.

2 Minimal ESOP Forms as Full Rank Decompositions

This section gives a new view of how to express ESOP forms. Namely, we demonstrate that any ESOP form can be regarded as a product of two matrices. This matrix-multiplication-based expression helps us to intuitively understand the minimization of ESOP forms, and enables us to easily describe the ESCT minimization algorithm in Section 3.

2.1 ESOP forms as products of matrices

We begin with an example. Consider the following ESOP form of the 5-valued input function g given in Table 1 (already seen in Eq. (1)):

$$g(a,b) = a^{\{0,3,4\}}b^{\{0,1,2\}} \oplus a^{\{1,3,4\}}b^{\{1,3\}} \oplus a^{\{2\}}b^{\{1,4\}} \oplus a^{\{3\}}b^{\{3,4\}}. \tag{9}$$

Given such a 5-valued ESOP form having 4 terms, we construct the product of a 5×4 matrix and a 4×5 matrix as follows:

$$\begin{array}{c} \overset{\curvearrowright a^{\{0,3,4\}}}{\downarrow} \\ \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \\ \underset{\curvearrowright a^{\{1,3,4\}}}{\uparrow} \end{array} \begin{array}{c} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} \\ \begin{matrix} \leftarrow b^{\{0,1,2\}} \\ \leftarrow b^{\{1,3\}} \end{matrix} \end{array}, \tag{10}$$

which represents the ESOP form above. The first column of the left matrix corresponds to the left literal $a^{\{0,3,4\}}$ of the first term in the ESOP form, the first row of the right matrix corresponds to the right literal $b^{\{0,1,2\}}$ of the first term, the second column of the left matrix corresponds to $a^{\{1,3,4\}}$ of the second term, and so on. (Note that each literal is described by a bit pattern of length 5.) Multiplying these two matrices, we obtain a truth table of g, namely, the same matrix as M_g:

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \ (= M_g). \tag{11}$$

Generally, given an ESOP form $\bigoplus_{\ell=1}^{t} a^{V_\ell} b^{W_\ell}$ of an m-valued input function f, we have the corresponding product of an $m \times t$ matrix and a $t \times m$ matrix:

$$\begin{pmatrix} a^{V_1}(0) & a^{V_2}(0) & \dots & a^{V_t}(0) \\ a^{V_1}(1) & a^{V_2}(1) & \dots & a^{V_t}(1) \\ \vdots & \vdots & \ddots & \vdots \\ a^{V_1}(m) & a^{V_2}(m) & \dots & a^{V_t}(m) \end{pmatrix} \times \begin{pmatrix} b^{W_1}(0) & b^{W_1}(1) & \dots & b^{W_1}(m) \\ b^{W_2}(0) & b^{W_2}(1) & \dots & b^{W_2}(m) \\ \vdots & \vdots & \ddots & \vdots \\ b^{W_t}(0) & b^{W_t}(1) & \dots & b^{W_t}(m) \end{pmatrix},$$

which is equal to the matrix M_f. The equality can be easily verified: the (i, j)-entry of the resulting matrix from multiplying these two matrices is

$$a^{V_1}(i)b^{W_1}(j) \oplus a^{V_2}(i)b^{W_2}(j) \oplus \cdots \oplus a^{V_t}(i)b^{W_t}(j),$$

which equals the value of $f(i, j)$ because of the ESOP form $f(a, b) = \bigoplus_{\ell=1}^{t} a^{V_\ell} b^{W_\ell}$.

Thus, Eq. (11) ensures that the function g has the ESOP form seen in Eq. (1) (or Eq. (9)). For another example, the following equation guarantees that the function g also has the ESOP form (of three terms) seen in Eq. (4) of Section 1.1:

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix}. \tag{12}$$

Note that there has been a matrix-based expression of ESOP forms before, in which two block matrices are aligned side by side [3]. This paper carries the idea further (just in representation), showing that a simple product of matrices can entirely exhibit an ESOP form, as seen in this subsection.

2.2 Appearance of minimal ESOP forms

We now consider how a minimal ESOP form appears in the matrix-multiplication-based expression.

As shown in the previous subsection, given an m-valued input function f, if there are an $m \times t$ matrix A and a $t \times m$ matrix B such that $M_f = AB$, then there correspondingly exists an ESOP form consisting of exactly t product terms. Therefore, the number t of columns in A (or, equivalently, rows in B) directly specifies the number of terms in the ESOP form. Hence, minimizing ESOP forms can be regarded as minimizing the number t of columns in such an A. That is, finding a matrix product $M_f = AB$ with the minimum number of columns in A is equivalent to minimizing ESOP forms of f. Such a matrix product is known as a "full rank decomposition" (also called a "full rank factorization") in matrix theory (e.g. [4]).

Specifically, given an $m \times m$ matrix Z with $\mathrm{rank}(Z) \geq 1$, $Z = XY$ is called a *full rank decomposition* if X is an $m \times \mathrm{rank}(Z)$ matrix and Y is a $\mathrm{rank}(Z) \times m$ matrix. It should be noted that in this case $\mathrm{rank}(X) = \mathrm{rank}(Y) = \mathrm{rank}(Z)$, and that one could not have X with less than $\mathrm{rank}(Z)$ columns. For example, since $\mathrm{rank}(M_g) = 3$ for M_g given in Eq. (6) of Section 1.1, Eq. (11) in Section 2.1 is not a full rank decomposition, while Eq. (12) is a full rank decomposition. Thus, a full rank decomposition corresponds to a minimal ESOP form. This also conforms to the correctness of the known result $\tau_{\mathrm{ESOP}}(f) = \mathrm{rank}(M_f)$ [3].

A full rank decomposition of a given binary matrix Z can be efficiently found, say by the use of the following Fact 2.

Fact 2. *Let Z be an $m \times m$ binary matrix such that*

$$Z = \begin{pmatrix} x_{11} \cdots x_{1i} \cdots x_{1j} \cdots x_{1t} \\ x_{21} \cdots x_{2i} \cdots x_{2j} \cdots x_{2t} \\ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \\ x_{m1} \cdots x_{mi} \cdots x_{mj} \cdots x_{mt} \end{pmatrix} \begin{pmatrix} y_{11} \ y_{12} \cdots y_{1m} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{i1} \ y_{i2} \cdots y_{im} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{j1} \ y_{j2} \cdots y_{jm} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{t1} \ y_{t2} \cdots y_{tm} \end{pmatrix}$$

for some t ($\leq m$). Then, it holds that

$$Z = \begin{pmatrix} x_{11} \cdots x_{1j} \cdots x_{1i} \cdots x_{1t} \\ x_{21} \cdots x_{2j} \cdots x_{2i} \cdots x_{2t} \\ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \ \ddots \ \vdots \\ x_{m1} \cdots x_{mj} \cdots x_{mi} \cdots x_{mt} \end{pmatrix} \begin{pmatrix} y_{11} \ y_{12} \cdots y_{1m} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{j1} \ y_{j2} \cdots y_{jm} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{i1} \ y_{i2} \cdots y_{im} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{t1} \ y_{t2} \cdots y_{tm} \end{pmatrix}$$

and

$$Z = \begin{pmatrix} x_{11} \cdots x_{1i} \oplus x_{1j} \cdots x_{1j} \cdots x_{1t} \\ x_{21} \cdots x_{2i} \oplus x_{2j} \cdots x_{2j} \cdots x_{2t} \\ \vdots \ \ddots \ \ \vdots \ \ \ddots \ \vdots \ \ddots \ \vdots \\ x_{m1} \cdots x_{mi} \oplus x_{mj} \cdots x_{mj} \cdots x_{mt} \end{pmatrix} \times \begin{pmatrix} y_{11} \ \ y_{12} \ \cdots \ y_{1m} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{i1} \ \ y_{i2} \ \cdots \ y_{im} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{i1} \oplus y_{j1} \ \ y_{i2} \oplus y_{j2} \cdots y_{im} \oplus y_{jm} \\ \vdots \ \ \vdots \ \ddots \ \vdots \\ y_{t1} \ \ y_{t2} \ \cdots \ y_{tm} \end{pmatrix}.$$

Thus, given $Z = XY$, one can transform the matrix product XY into another one without changing the original matrix Z: (i) one can exchange the ith and jth columns in X as long as the ith and jth rows in Y are also exchanged, and (ii) one can add the jth column to the ith one in X as long as the ith row in Y is added to the jth one.

Note that for a binary matrix the two equations in Fact 2 can be regarded as elementary column or row operations. Therefore, based on Fact 2, a known algorithm, say the Gaussian elimination algorithm, immediately yields a full rank decomposition. More specifically, given an $m \times m$ binary matrix Z, starting with $Z = IZ$ where I is an identity matrix, we transform the right matrix of the product into a matrix in row echelon form using Fact 2 as elementary row operations. Then, we have

$$Z = X' \begin{pmatrix} * \\ O \end{pmatrix} \begin{matrix} \} \text{rank}(Z) \\ \} m - \text{rank}(Z) \end{matrix}$$

for some X', that is, Z is written as a product of X' and a matrix whose last $m - \mathrm{rank}(Z)$ rows are all-zero. After removing the all-zero rows in the right matrix and the corresponding columns, namely the last $m - \mathrm{rank}(Z)$ columns, in the left matrix X', we obtain a full rank decomposition.

Take the matrix M_g in Eq. (6) as an example again. Starting with IM_g, the following is a possible transformation of the right matrix into an echelon one:

$$\begin{aligned}
&\begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix} \\
&= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}.
\end{aligned}$$

Remove the fourth and fifth all-zero rows in the right matrix and the corresponding fourth and fifth columns in the left matrix, then we surely have a full rank decomposition in Eq. (12).

Note that the transformation in Fact 2 never changes the row space of the right matrix, i.e., the space filled by all linear combinations of the rows there, because the transformation is just either to exchange two rows, or to add one row to another row. Therefore, any full rank decomposition $M_f = AB$ for a function f satisfies $\mathbf{C}({M_f}^{\mathrm{T}}) = \mathbf{C}(B^{\mathrm{T}})$ (where X^{T} denotes the transpose of a matrix X, and $\mathbf{C}(X^{\mathrm{T}})$ denotes the row space of a binary matrix X). It also holds that $\mathbf{C}(M_f) = \mathbf{C}(A)$.

3 Our Algorithm for Minimizing ESCT Forms

In this section, we give an efficient algorithm for minimizing ESCT forms. As mentioned in Section 1.3, using the ESCT form instead of the ESOP form can reduce the number of terms by at most one. That is, given a function f, our algorithm produces its ESCT form consisting of $\mathrm{rank}(M_f) - 1$ or $\mathrm{rank}(M_f)$ terms. (Remember that the number of terms $\tau_{\mathrm{ESOP}}(f)$ in a minimal ESOP form equals $\mathrm{rank}(M_f)$).

We first exhibit a simplification rule to reduce the number of terms in Section 3.1. We then present the general idea behind our ESCT minimization algorithm in Section 3.2, and a complete description of the algorithm in Section 3.3.

3.1 A simplification rule

As mentioned before, the ESCT form is a generalization of the ESOP form, and hence the former is a more generic expression than the latter. Actually, for example, an ESOP form $a^{\{1,2,3,4\}}b^{\{1,3\}} \oplus a^{\{2,3\}}$ consisting of a product term and a literal can be converted into a single complex term:

$$a^{\{1,2,3,4\}}b^{\{1,3\}} \oplus a^{\{2,3\}} = a^{\{2,3\},\{1,4\}}(b^{\{1,3\}}).$$

More generally, we have the following Lemma 3.

Lemma 3. *For any product term $a^V b^W$ and any literal $a^{V'}$, it holds that*

$$a^V b^W \oplus a^{V'} = a^{V',V\oplus V'}(b^W).$$

Proof. If $b^W = 0$, then $a^V b^W \oplus a^{V'} = a^{V'}$ and $a^{V',V\oplus V'}(b^W) = a^{V'}$. If $b^W = 1$, then $a^V b^W \oplus a^{V'} = a^V \oplus a^{V'}$ and $a^{V',V\oplus V'}(b^W) = a^{V\oplus V'} = a^V \oplus a^{V'}$. □

Our algorithm utilizes this transformation rule to reduce the number of terms.

3.2 The idea behind our algorithm

To apply Lemma 3 to an ESOP form, it must contain a term consisting of a single literal $a^{V'}$ (for some V'). For example, for a minimal ESOP form

$$g(a,b) = a^{\{0,2,4\}}b^{\{0,1,2\}} \oplus a^{\{1,2,3,4\}}b^{\{1,3\}} \oplus a^{\{2,3\}} \tag{13}$$

of the function g given in Table 1, the simplification rule in Lemma 3 together with Eq. (8) (which says that any ESOP form can be expressed as an ESCT form) converts the ESOP form into the following ESCT form, already seen in Eq. (7) though the order of the terms is different:

$$g(a,b) = a^{\emptyset,\{0,2,4\}}(b^{\{0,1,2\}}) \oplus a^{\{2,3\},\{1,4\}}(b^{\{1,3\}}). \tag{14}$$

Based on this idea, in our algorithm we first find a minimal ESOP form of a given function f, and then, if possible, transform it so that a single literal appears. Finally, we convert the minimal ESOP form into an ESCT form using Eq. (8) and Lemma 3.

Again taking the function g as an example, and recalling that $\mathrm{rank}(M_g) = 3$, its full rank decomposition was seen in Eq. (12). Note that for the full rank decomposition (12), adding the first row in the right matrix to the third row yields an all-one row, that is, we can obtain another full rank decomposition

$$\begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 0 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix}, \tag{15}$$

whose third row in the right matrix is all ones. This all-one row together with the third column in the left matrix represents a single literal $a^{\{2,3\}}$ in the minimal ESOP form because $b^{\{0,1,2,3,4\}}$ is just a constant 1. Therefore, the full rank decomposition in Eq. (15) corresponds to the minimal ESOP form seen in Eq. (13). Now, we need only to convert it into the ESCT form, the number of whose terms is $\mathrm{rank}(M_g) - 1 = 2$, as demonstrated in Eq. (14) above.

Hence, when a full rank decomposition $M_f = AB$ satisfies $\vec{1} \in \mathbf{C}(B^{\mathrm{T}})$, we have an ESCT form consisting of $\mathrm{rank}(M_f) - 1$ terms (where $\vec{1}$ denotes an all-one column vector). To determine whether $\vec{1} \in \mathbf{C}(B^{\mathrm{T}})$ or not, it suffices to solve the linear equation $B^{\mathrm{T}}\vec{x} = \vec{1}$. This can be efficiently done using known algorithms, say the Gaussian elimination algorithm. When $\vec{1} \in \mathbf{C}(B^{\mathrm{T}})$, one can also efficiently transform $M_f = AB$ into a full rank decomposition $M_f = A'B'$ such that the last row in B' is all ones.

3.3 The description of our algorithm

We are now ready to present a complete description of our algorithm. Given an m-valued input function f with $\mathrm{rank}(M_f) \geq 2$, our algorithm proceeds as follows.

1. Find a full rank decomposition $M_f = AB$.
2. Determine whether $\vec{1} \in \mathbf{C}(B^{\mathrm{T}})$.
3. **Case 1:** $\vec{1} \notin \mathbf{C}(B^{\mathrm{T}})$.
   Let $\bigoplus_{\ell=1}^{t} a^{V_\ell} b^{W_\ell}$ be the minimal ESOP form corresponding to the full rank decomposition $M_f = AB$. Using Eq. (8), convert the ESOP form into an ESCT form
   $$\bigoplus_{\ell=1}^{t} a^{\emptyset, V_\ell}(b^{W_\ell}),$$
   and output it. Note that the number of its terms is $t = \mathrm{rank}(M_f)$.

   Case 2: $\vec{1} \in \mathbf{C}(B^{\mathrm{T}})$.
   Transform the full rank decomposition $M_f = AB$ into $M_f = A'B'$ such that the last row in B' is all ones. Let
   $$(\bigoplus_{\ell=1}^{t-2} a^{V_\ell} b^{W_\ell}) \oplus a^{V_{t-1}} b^{W_{t-1}} \oplus a^{V_t}$$
   be the minimal ESOP form corresponding to $M_f = A'B'$. Using Eq. (8) and Lemma 3, convert the ESOP form into an ESCT form
   $$\left(\bigoplus_{\ell=1}^{t-2} a^{\emptyset, V_\ell}(b^{W_\ell})\right) \oplus a^{V_t, V_{t-1} \oplus V_t}(b^{W_{t-1}}),$$
   and output it. Note that the number of its terms is $t - 1 = \mathrm{rank}(M_f) - 1$.

Note that $\vec{1} \in \mathbf{C}(B^{\mathrm{T}})$ in step 2 of the algorithm if and only if $\vec{1} \in \mathbf{C}(M_f^{\mathrm{T}})$ because the row space never changes as mentioned in Section 2.2. Therefore, our algorithm above implies the following Lemma 4.

Lemma 4. *For any (two-variable) m-valued input function f with* $\mathrm{rank}(M_f) \geq 2$,

$$\tau_{\mathrm{ESCT}}(f) \leq \begin{cases} \mathrm{rank}(M_f) - 1 & \textit{if } \vec{1} \in \mathbf{C}(M_f^{\mathrm{T}}); \\ \mathrm{rank}(M_f) & \textit{otherwise.} \end{cases}$$

In the next section, we show that there is no ESCT form with fewer terms than that in an ESCT form produced by our algorithm.

4 Correctness of Our Algorithm

In this section, we show that our algorithm always produces a minimal ESCT form. That is, we prove the following lower bound lemma.

Lemma 5. *For any (two-variable) m-valued input function f with* $\mathrm{rank}(M_f) \geq 2$*,*

$$\tau_{\mathrm{ESCT}}(f) \geq \begin{cases} \mathrm{rank}(M_f) - 1 & \textit{if } \vec{1} \in \mathbf{C}(M_f^{\mathrm{T}}); \\ \mathrm{rank}(M_f) & \textit{otherwise.} \end{cases}$$

To prove Lemma 5, we use the following Lemma 6.

Lemma 6. *For any complex term $a^{U,V}(b^W)$ and any literal $a^{V'}$, it holds that*

$$a^{U,V}(b^W) \oplus a^{V'} = a^{\emptyset,U\oplus V}(b^W) \oplus a^{U\oplus V'}.$$

Proof. If $b^W = 0$, then $a^{U,V}(b^W) \oplus a^{V'} = a^U \oplus a^{V'}$ and $a^{\emptyset,U\oplus V}(b^W) \oplus a^{U\oplus V'} = a^{U\oplus V'} = a^U \oplus a^{V'}$, as desired. The proof for $b^W = 1$ is similar. □

We are now ready to give a proof of Lemma 5.

Proof of Lemma 5. Let f be an m-valued input function with $\mathrm{rank}(M_f) \geq 2$, and let

$$f(a,b) = a^{U_1,V_1}(b^{W_1}) \oplus a^{U_2,V_2}(b^{W_2}) \oplus \cdots \oplus a^{U_t,V_t}(b^{W_t})$$

be an arbitrary minimal ESCT form of f; note that $\tau_{\mathrm{ESCT}}(f) = t$. Then, add a constant $a^{\emptyset}(= 0)$ to the ESCT form above:

$$f(a,b) = a^{U_1,V_1}(b^{W_1}) \oplus a^{U_2,V_2}(b^{W_2}) \oplus \cdots \oplus a^{U_t,V_t}(b^{W_t}) \oplus a^{\emptyset}.$$

Using Lemma 6, we have

$$f(a,b) = a^{\emptyset,U_1\oplus V_1}(b^{W_1}) \oplus a^{\emptyset,U_2\oplus V_2}(b^{W_2}) \oplus \cdots \oplus a^{\emptyset,U_t\oplus V_t}(b^{W_t}) \oplus a^{U}$$

where $U = U_1 \oplus U_2 \oplus \cdots \oplus U_t$. Remembering Eq. (8), convert the complex terms above into product terms:

$$f(a,b) = a^{U_1\oplus V_1}b^{W_1} \oplus a^{U_2\oplus V_2}b^{W_2} \oplus \cdots \oplus a^{U_t\oplus V_t}b^{W_t} \oplus a^{U}. \tag{16}$$

Since the right side of Eq. (16) is an ESOP form consisting of $t+1$ product terms, we have

$$\tau_{\mathrm{ESOP}}(f) \leq t + 1.$$

Therefore, since $\tau_{\mathrm{ESOP}}(f) = \mathrm{rank}(M_f)$ (by Eq. (5)) and $t = \tau_{\mathrm{ESCT}}(f)$, we have

$$\tau_{\mathrm{ESCT}}(f) \geq \mathrm{rank}(M_f) - 1.$$

Thus, to complete the proof it suffices to show that if $\vec{1} \notin \mathbf{C}(M_f^{\mathrm{T}})$, then $\tau_{\mathrm{ESCT}}(f) \geq \mathrm{rank}(M_f)$.

Assume that $\vec{1} \notin \mathbf{C}(M_f^{\mathrm{T}})$. Express the ESOP form in Eq. (16) as a product of an $m \times (t+1)$ matrix and a $(t+1) \times m$ matrix:

$$M_f = \begin{pmatrix} a^{U_1 \oplus V_1}(0) & \dots & a^{U_t \oplus V_t}(0) & a^{U}(0) \\ \vdots & \ddots & \vdots & \vdots \\ a^{U_1 \oplus V_1}(m) & \dots & a^{U_t \oplus V_t}(m) & a^{U}(m) \end{pmatrix} \times \begin{pmatrix} b^{W_1}(0) & \dots & b^{W_1}(m) \\ \vdots & \ddots & \vdots \\ b^{W_t}(0) & \dots & b^{W_t}(m) \\ 1 & \dots & 1 \end{pmatrix}. \tag{17}$$

Remember that any full rank decomposition $M_f = AB$ satisfies $\mathbf{C}({M_f}^{\mathrm{T}}) = \mathbf{C}(B^{\mathrm{T}})$. Since $\vec{1} \notin \mathbf{C}(M_f^{\mathrm{T}})$, Eq. (17) is not a full rank decomposition. Therefore, we have $\mathrm{rank}(M_f) < t+1$ and hence $\mathrm{rank}(M_f) \leq t$. Since $t = \tau_{\mathrm{ESCT}}(f)$, we have $\tau_{\mathrm{ESCT}}(f) \geq \mathrm{rank}(M_f)$. □

5 Conclusions

For a two-variable multiple-valued input function f, the number of terms in any minimal ESOP form $\tau_{\mathrm{ESOP}}(f)$ was already known, while the number of terms in any minimal ESCT form $\tau_{\mathrm{ESCT}}(f)$ has not been previously reported. To solve the open question, determining how much smaller $\tau_{\mathrm{ESCT}}(f)$ is than $\tau_{\mathrm{ESOP}}(f)$, this paper gave an efficient algorithm to find a minimal ESCT form of a given two-variable m-valued input function f. That is, we proved Lemmas 4 and 5, both of which together constitute Theorem 1, which claims that

$$\tau_{\mathrm{ESCT}}(f) = \begin{cases} \mathrm{rank}(M_f) - 1 & \text{if } \vec{1} \in \mathbf{C}(M_f^{\mathrm{T}}); \\ \mathrm{rank}(M_f) & \text{otherwise.} \end{cases}$$

Therefore, since $\tau_{\mathrm{ESOP}}(f) = \mathrm{rank}(M_f)$ [3], using the ESCT form instead of the ESOP form can reduce the number of terms by at most one.

The complexity of our algorithm is dominated by finding a full rank decomposition (in step 1) and solving a linear equation (in step 2). Therefore, if we use the Gaussian elimination algorithm in steps 1 and 2, then our algorithm runs in $O(m^3)$ time.

This paper together with the previous research [3] presents the properties of the ESOP and ESCT forms of two-variable functions. It is an interesting open problem to find the exact number of terms in minimal ESOP and ESCT forms of multiple-valued input functions having three or more variables.

Acknowledgments

We thank Mr. Naoki Katagami for his valuable comments on the expression in Theorem 1. We thank the associate editor and the anonymous referees whose comments helped us improve the presentation of the paper. This work was supported by JSPS KAKENHI Grant Number 23700007.

References

[1] T. Hirayama, Y. Nishitani, and T. Sato, "A faster algorithm of minimizing AND-EXOR expressions," IEICE Trans. Fundamentals, vol. E85-A, no. 12, pp. 2708–2714, 2002.

[2] T. Mizuki, T. Otagiri, and H. Sone, "An application of ESOP expressions to secure computations," Journal of Circuits, Systems, and Computers, vol. 16, no. 2, pp. 191–198, 2007.

[3] T. Mizuki, H. Tsubata, and T. Nishizeki, "Minimizing AND-EXOR expressions for two-variable multiple-valued input binary output functions," Journal of Multiple-Valued Logic and Soft Computing, vol. 16, pp. 197–208, 2010.

[4] S. Puntanen, G. P. H. Styan, and J. Isotalo, "Matrix Tricks for Linear Statistical Models: Our Personal Top Twenty," Springer-Verlag, Berlin Heidelberg, 2011.

[5] M. Sampson, D. Voudouris, and G. Papakonstantinou, "Using simple disjoint decomposition to perform secure computations," Journal of Circuits, Systems, and Computers, vol. 19, no. 7, pp. 1559–1569, 2010.

[6] T. Sasao, "Switching Theory for Logic Synthesis," Kluwer Academic Publishers, Boston, MA, 1999.

[7] N. Song and M. Perkowski, "Minimization of exclusive sums of multi-valued complex terms for logic cell arrays," 28th IEEE International Symposium on Multiple-Valued Logic, pp. 32–37, 1998.

[8] N. Song and M. A. Perkowski, "Minimization of exclusive sum-of-products expressions for multiple-valued input, incompletely specified functions," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, vol. 15, no. 4, pp. 385–395, 1996.

[9] S. Stergiou and G. Papakonstantinou, "Exact minimization of ESOP expressions with less than eight product terms," Journal of Circuits, Systems and Computers, vol. 13, no. 1, pp. 1–15, 2004.

[10] S. Stergiou, D. Voudouris, and G. Papakonstantinou, "Multiple-value exclusive-or sum-of-products minimization algorithms," IEICE Trans. Fundamentals, vol. E87-A, no. 5, pp. 1226–1234, 2004.

[11] G. Strang, "Introduction to Linear Algebra, Fourth Edition," Wellesley-Cambridge Press, Wellesley, 2009.

[12] D. Voudouris, S. Stergiou, and G. Papakonstantinou, "Minimization of reversible wave cascades," IEICE Trans. Fundamentals, vol. E88-A, no. 4, pp. 1015–1023, 2005.

[13] D. Voudouris, M. Sampson, and G. Papakonstantinou, "Exact ESCT minimization for functions of up to six input variables," Integration, the VLSI Journal, vol. 41, no. 1, pp. 87–105, 2008.