Comparison of the Worst and Best Sum-Of-Products Expressions for Multiple-Valued Functions

Tsutomu Sasao
Department of Computer Science and Electronics
Kyushu Institute of Technology
Iizuka 820, Japan

Jon T. Butler
Department of Electrical and Computer Engineering
Naval Postgraduate School
Monterey, CA 93943-5121, U.S.A.

Abstract

Because most practical logic design algorithms produce irredundant sum-of-products (ISOP) expressions, the understanding of ISOPs is crucial. We show a class of functions for which Morel-Minato's ISOP generation algorithm produces worst ISOPs (WSOP), ISOPs with the most product terms. We show this class has the property that the ratio of the number of products in the WSOP to the number in the minimum ISOP (MSOP) is arbitrarily large when the number of variables is unbounded. The ramifications of this are significant; care must be exercised in designing algorithms that produce ISOPs.

We also show that $2^{n-1}$ is a (firm) upper bound on the number of product terms in any ISOP for switching functions on $n$ variables, answering a question that has been open for 30 years. We show experimental data and extend our results to functions of multiple-valued variables.

Index terms: Logic minimization, irredundant sum-of-products, multiple-valued logic.

1 Introduction

The majority of logic minimization algorithms used in practical design produce irredundant sum-of-products expressions (ISOPs) rather than minimum sum-of-products expressions (MSOPs). For example, the PRESTO [2, 21] logic minimizer produces an ISOP as follows:

1) Expand each product into a prime implicant
2) Delete redundant prime implicants.

An ISOP is the OR of prime implicants such that deleting any prime implicant changes the function. For example, two expressions $x_1 \overline{x_2} \overline{x_3} \overline{x_4} \overline{x_5}$ and $x_1 x_2 \overline{x_3} \overline{x_4} \overline{x_5}$ are both ISOPs for the same function. However, only the former is an MSOP. Depending on one's viewpoint, the second ISOP has only one more product than the first, or 33% more products.

These two viewpoints inspire the first question: "Can a logic minimization algorithm yield unreasonably large SOPs?" The answer is yes. We show that there exists an algorithm [13, 14] that produces worst ISOPs (WSOPs), ISOPs with the largest number of product terms, for some class of functions.

The question above and its surprising answer inspire the second question: "To what extent can the number of products in a WSOP exceed the number of products in an MSOP?" The answer to this is also surprising. We show there exist functions in which the ratio of the WSOP product count to MSOP product count is arbitrarily large, when the number of variables is unbounded.

While our results are motivated by the existence of binary logic functions, we show that a similar phenomenon exists for functions of a higher radix. Our results suggest that the disparity between the number of products in multiple-valued WSOPs and MSOPs increases with radix.

2 Definitions and Basic Properties

Definition 2.1 $x$ and $\overline{x}$ are literals of a variable $x$. A logical product that contains at most one literal for each variable is called a product term or a product. Products combined with OR operators form a sum-of-products expression (SOP).

Definition 2.2 A prime implicant (PI) of a function $f$ is a product which implies $f$, such that the deletion of any literal from the product results in a new product that does not imply $f$.

Definition 2.3 An irredundant sum-of-products expression (ISOP) is an SOP, where each product is a PI, and no product can be deleted without changing the function represented by the expression.

Definition 2.4 Among the ISOPs for $f$, one with the maximum number of products is a worst ISOP (WSOP), and one with the minimum number of products is a minimal SOP (MSOP).

Definition 2.5 The number of products in a WSOP for $f$ is denoted by $\tau(WSOP : f)$. The number of products in an MSOP for $f$ is denoted by $\tau(MSOP : f)$.
Because most practical logic design algorithms produce irredundant sum-of-products (ISOP) expressions, the understanding of ISOPs is crucial. We show a class of functions for which Morreale-Minato's ISOP generation algorithm produces worst ISOPs (WSOP), ISOPs with the most product terms. We show this class has the property that the ratio of the number of products in the WSOP to the number in the minimum ISOP (MSOP) is arbitrarily large when the number of variables is unbounded. The ramifications of this are significant; care must be exercised in designing algorithms that produce ISOPs. We also show that $2^{n/2}$ is a (firm) upper bound on the number of product terms in any ISOP for switching functions on $n$ variables, answering a question that has been open for 30 years. We show experimental data and extend our results to functions of multiple-valued variables. Index terms: Logic minimization, irredundant sum-of-products multiple-valued logic.
The following is well known [7, 15]

**Theorem 2.1** For any switching function of $n$ variables, $	au (\text{MSOP} : f) \leq 2^{n-1}$.

Our first result shows that Theorem 2.1 is also true when we replace MSOP by WSOP. It answers an open question posed by Meo [10] in 1968.

**Theorem 2.2** For any switching function of $n$ variables, $	au (\text{WSOP} : f) \leq 2^{n-1}$.

(Proof) Available from the authors.

There exists a WSOP with $2^{n-1}$ products; it is the SOP of a parity function of $n$ variables. Thus, the upper bound in Theorem 2.2 is tight.

**Lemma 2.1** Let $g(X)$ and $h(Y)$ be functions, where $X$ and $Y$ have no common variables. Let $G(X)$ and $H(Y)$ be ISOPs for $g(X)$ and $h(Y)$, respectively. Then, the SOP $F(X, Y)$ derived from $G(X)H(Y)$ by using distributive laws is an ISOP for $f$.

(Proof) Clearly, $F(X, Y)$ represents $f$. If any product in $F(X, Y)$ is not a PI, then either $G(X)$ or $H(Y)$ or both contain a non-prime product, which contradicts the assumption that $G(X)$ and $H(Y)$ are ISOPs. If any product in $F(X, Y)$ is redundant, then either $G(X)$ or $H(Y)$ or both contain a redundant product which contradicts the assumption that $G(X)$ and $H(Y)$ are irredundant.

Our next result shows that if a function $f(X, Y)$ can be expressed as the AND of two functions, $g(X)$ and $h(Y)$ on disjoint sets of variables, then the number of product terms in a WSOP (MSOP) of $f(X, Y)$ is the product of the number of products in a WSOP (MSOP) of $f(X)$ and $h(Y)$.

**Theorem 2.3** Let $g(X)$ and $h(Y)$ be functions, where $X$ and $Y$ have no common variables. Let $f(X, Y) = g(X)h(Y)$. Then,

1. $\tau (\text{WSOP} : f) = \tau (\text{WSOP} : g)\tau (\text{WSOP} : h)$,
2. $\tau (\text{MSOP} : f) = \tau (\text{MSOP} : g)\tau (\text{MSOP} : h)$.

(Proof) Available from the authors.

This result will be useful later when we demonstrate functions with a large discrepancy between the number of products in the WSOP and in the MSOP.

**3 Comparing the Number of Product Terms in a WSOP to the Number in an MSOP for Specific Functions**

**Definition 3.6** Let $ST(n, k)$ be a symmetric function of $n$-variables $x_1, x_2, \ldots, x_n$ such that

$$ST(n, k) = \begin{cases} 1 & k \leq \sum_{i=1}^{n} x_i \leq n - k \\ 0 & \text{otherwise}, \end{cases}$$

where $\sum$ is ordinary addition in which the value of $x_i$ is viewed as an integer. That is, $\sum_{i=1}^{n} x_i$ is the number of variables that are 1.

**Example 3.1** $ST(n, 0) = 1$, $ST(n, \frac{n}{2})$, for even $n$, is the OR of all minterms with exactly half of the variables complemented.

(End of Example)

**Lemma 3.2** $ST(n, k)$ can be represented as

$$ST(n, k) = S_{\{0, 1, \ldots, n-k\}}S_{\{0, 1, \ldots, n-k\}},$$

where

$$S_A = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} x_i \in A \\ 0 & \text{otherwise}, \end{cases}$$

where $A \subseteq \{0, 1, 2, \ldots, n\}$.

**Example 3.2**

$$ST(n, 1) = S_{\{1, 2, \ldots, n\}}S_{\{0, 1, \ldots, n-1\}} = (x_1 \lor x_2 \lor \cdots \lor x_n)(x_1 \lor x_2 \lor \cdots \lor x_n)$$

and

$$ST(n, 2) = S_{\{2, 3, \ldots, n\}}S_{\{0, 1, \ldots, n-2\}} = (x_1 x_2 \lor x_1 x_2 \lor \cdots \lor x_{n-1} x_n) \lor (x_1 x_2 \lor x_1 x_2 \lor \cdots \lor x_{n-1} x_n).$$

(End of Example)

We are interested in the total number of PIs in $ST(n, k)$, the number of PIs in an MSOP for $ST(n, k)$, and the number of PIs in a WSOP for $ST(n, k)$. We can derive these as follows.

**Theorem 3.4**

1) $ST(n, k)$ has $\binom{n}{k} \frac{n!}{k!(n-2k)!}$ PIs.

2) $\tau (\text{MSOP} : ST(n, k)) = \binom{n}{k} \frac{n!}{k!(n-2k)!}$.

3) $\tau (\text{WSOP} : ST(n, k)) = 2^{\binom{n}{k}} - \binom{n}{k}$.

(Proof) Available from the authors.

A special case of Theorem 3.4 occurs when $n = 3$ and $k = 1$.

**Corollary 3.1** $ST(3, 1)(x_1, x_2, x_3) = (x_1 \lor x_2 \lor x_3)(x_1 \lor x_2 \lor x_3)$ has the following properties:

1) $ST(3, 1)$ has 6 PIs.

2) $\tau (\text{MSOP} : ST(3, 1)) = 3$.

3) $\tau (\text{WSOP} : ST(3, 1)) = 4$.

Fig. 1 (a) and (b) suggest how the MSOP and WSOP are formed. With the MSOP, product terms cover as many minterms as possible. With the WSOP, product terms are chosen so that as much overlap occurs as possible. Notice that any product term added to Fig. 1(a) or (b) is redundant. We now consider the questions posed in the introduction.

**Definition 3.7** A set of true minterms $S$ for $f$ is called independent if no implicant of $f$ contains a pair of minterms in $S$. 56
Lemma 3.3 Let $S$ be an independent set of minterms for $f$. Then $\tau(\text{MSOP} : f) \geq |S|$.

Example 3.3 Consider the Karnaugh map of $ST(4,1)$ in Fig. 2. Independent sets for $ST(4,1)$ include $S_1 = \{m_7, m_{11}, m_{13}, m_{14}\}$ and $S_2 = \{m_1, m_2, m_4, m_8\}$. From Lemma 3.3, $\tau(\text{MSOP} : ST(4,1)) \geq 4$. (End of Example)

Definition 3.8 The redundancy ratio of a function $f$ is

$$\rho(f) = \frac{\tau(\text{WSOP} : f)}{\tau(\text{MSOP} : f)}.$$ 

The normalized redundancy ratio of an $n$-variable function $f$ is

$$\sigma(f) = \sqrt[n]{\rho(f)}.$$ 

The redundancy ratio is a measure of the discrepancy between the number of product terms in WSOPs and MSOPs. A small ratio suggests that any logic minimization algorithm will do well, while a large ratio suggests that care should be exercised. The normalized redundancy ratio is normalized with respect to the number of variables. It is a convenience; it allows one to compare the redundancy ratio of two functions with a different number of variables.

Example 3.4

$$\rho(ST(3,1)) = \frac{4}{3}, \quad \sigma(ST(3,1)) = \sqrt[4]{\frac{4}{3}} \approx 1.0906$$
$$\rho(ST(4,1)) = \frac{6}{4}, \quad \sigma(ST(4,1)) = \sqrt[6]{\frac{6}{4}} \approx 1.0866$$
$$\rho(ST(5,1)) = \frac{8}{5}, \quad \sigma(ST(5,1)) = \sqrt[8]{\frac{8}{5}} \approx 1.0986$$
$$\rho(ST(6,1)) = \frac{10}{6}, \quad \sigma(ST(6,1)) = \sqrt[10]{\frac{10}{6}} \approx 1.0889$$
$$\rho(ST(7,1)) = \frac{12}{7}, \quad \sigma(ST(7,1)) = \sqrt[12]{\frac{12}{7}} \approx 1.0800$$
$$\rho(ST(8,1)) = \frac{14}{8}, \quad \sigma(ST(8,1)) = \sqrt[14]{\frac{14}{8}} \approx 1.0725.$$ 

For these $ST(n,1)$ functions, $\sigma$ is the largest when $n = 4$. That is, as $n$ increases above 3, $\sigma$ first increases peaking at 4, and then it continually decreases.

To understand how poorly an ISOP generator can do, we investigate functions with large $\rho$. For such functions, the choice of algorithm is important; a poorly designed algorithm can produce ISOPs with many products. When $\rho$ is small, there is less concern; any ISOP generator algorithm will do well. Our first result below shows that the $ST(n,k)$ functions have reasonably small $\rho$.

Theorem 3.5

$$1 \leq \rho(ST(n,k)) < 2.$$ 

(Proof) The first inequality follows from the fact that a WSOP has at least as many product terms as a MSOP. The second inequality is proved as follows.

$$\rho(ST(n,k)) = \frac{\tau(ST(n,k) : WSOP)}{\tau(ST(n,k) : MSOP)} = \frac{2^{|S|} - \binom{|S|}{k}}{\binom{n}{k}} = 2 - \binom{k}{k}.$$ 

Since $\binom{k}{k} > 0$, we have the theorem. \hfill \Box

Note that the largest $\rho$ is achieved when $n$ is much larger than $k$, and this value can never be more than 2. However, one can use $ST(n,k)$ functions to construct functions with large $\rho$, as follows.

Definition 3.9 Let $ST(n,k)^r$ be the $n \cdot r$-variable function

$$ST(n,k)^r(x_1, x_2, \ldots, x_{nr}) = \bigwedge_{i=1}^{r} ST(n,k)(x_{n(i-1)+1}, x_{n(i-1)+2}, \ldots, x_{ni}).$$ 

Lemma 3.4 Let $g_i(X)$ be a function with redundancy ratio $\rho(g_i)$. Let $f(x_1, x_2, \ldots, x_r) = g_1(x_1)g_2(x_2)\cdots g_r(x_r)$, where $x_1, x_2, \ldots, x_r$ are pairwise disjoint. Then, $\rho(f) = \prod_{i=1}^{r} \rho(g_i)$.

Theorem 3.6 $ST(n,k)^r$ has the following properties:

1) $ST(n,k)^r$ has $\binom{n}{k-r} \binom{r}{n-k}$ PIs.
2) $\tau(\text{MSOP} : ST(n,k)^r) = \binom{n}{r} r!$.
3) $\tau(\text{WSOP} : ST(n,k)^r) = \binom{n}{k} - \binom{k}{k} r!$.

Example 3.5 For $n = 3$ and $k = 1$, $ST(3,1)^r$ has $6^r$ PIs, $\tau(\text{MSOP} : ST(3,1)^r) = 3^r$, and $\tau(\text{WSOP} : ST(3,1)^r) = 4^r$. (End of Example)

The ISOP generator developed by Minato [13] is quite fast. Unfortunately, it produces WSOPs instead of MSOPs for $ST(3,1)^6$. This heuristic logic minimizer produces ISOPs that have many more products than MSOPs. This answers the first question posed in the introduction.

We have
Theorem 3.7
\[ p(ST(n, k)^\prime) = \left[ 2 - \frac{(2k)}{k} \right]^n. \]

Example 3.6 For \( n = 4 \) and \( k = 1 \) we have
\[ p(ST(4, 1)^\prime) = (1.5)^4. \]

(End of Example)

From this, it can be seen that \( p \) can be arbitrarily large. This answers the second question posed in the introduction.

4 Extension to Multiple-Valued Functions

4.1 Multiple-Valued Input Two-valued Output Functions

Definition 4.10 A multi-valued input two-valued output function is \( f: \prod_i P_i \rightarrow \{0, 1\} \), where \( P_i = \{0, 1, \ldots, p_i - 1\} \), \( p_i \geq 2 \).

Definition 4.11 \( X^S \) is a literal of \( p \)-valued variable \( X \), where \( S \subseteq \{0, 1, \ldots, p - 1\} \). \( X^S = 1 \) if \( X = a \in S \) and \( X^S = 0 \), otherwise. A logical product of literals that contains at most one literal for each variable is a product term or a product. Products combined with OR operators form a sum-of-products expression (SOP). Prime implicants (PI), irredundant sum-of-products expression (ISOP), worst ISOP (WSOP), and minimum SOP (MSOP) are defined in a manner similar to the binary case.

Theorem 4.8 ([17]) Let \( f \) be the function defined in Definition 4.11. Then,
\[ \tau(MSOP: f) \leq B = \frac{p_1 \cdots p_n}{\max p_i}. \]

When \( p_i = 2(i = 1, 2, \ldots, n) \), i.e., for switching functions, \( B = 2^{n-1} \). In this case, \( \tau(WSOP: f) \leq B \) (Theorem 2.1), and \( \tau(MSOP: f) \leq B \) (Theorem 2.2). However, for some functions on multiple-valued variables, an ISOP (e.g. a WSOP) can have more than \( B \) products, i.e., \( \tau(WSOP: f) > B \).

Example 4.7 Consider the function \( f: P_1 \times P_2 \times P_3 \rightarrow \{0, 1\} \), where \( P_1 = P_2 = P_3 = \{0, 1\} \) and \( P_3 = \{0, 1, 2, 3\} \). The map is shown in Fig. 3. Since all the implicants are prime and irredundant, the SOP shown in the map is an ISOP. Note that this ISOP requires 17 products. On the other hand, Theorem 4.8 gives \( B = 16 \). Thus, \( \tau(WSOP: f) > B = 16 \). (End of Example)

Example 4.8 Consider the function MV04: \( P_1 \times P_2 \times P_3 \rightarrow \{0, 1\} \), where \( P_1 = P_2 = P_3 = \{0, 1\} \) and \( P_3 = \{0, 1, 2, 3\} \). Fig. 4 shows the positional cube notation of the expression. The cubes are prime and irredundant, so this figure represents a WSOP with 16 products. Note that the MSOP has only 8 products. (End of Example)

Figure 3. Multiple-valued function.

<table>
<thead>
<tr>
<th>WSOP</th>
<th>MSOP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 10 10 10</td>
<td>10 11 11 11</td>
</tr>
<tr>
<td>10 10 10 01</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>10 10 01 10</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>10 01 10 10</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>01 01 10 10</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>01 01 01 10</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>01 01 01 01</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>01 01 01 01</td>
<td>11 11 11 11</td>
</tr>
<tr>
<td>01 01 01 01</td>
<td>11 11 11 11</td>
</tr>
</tbody>
</table>

Figure 4. Arrays for the WSOP and MSOP for MVO4.

By generalizing the above example, we have the following:

Theorem 4.9 There exists a function MVOn: \( P_1 \times P_2 \times \cdots \times P_n \times P_{n+1} \rightarrow \{0, 1\} \), where \( P_1 = P_2 = \cdots = P_n = \{0, 1\} \) and \( P_{n+1} = \{0, 1, \ldots, 2^{n-1}\} \), such that a WSOP requires \( 2^n \) products, while an MSOP requires \( 2^n \) products.

58
Definition 4.12 The function \( MV2 : \{0,1,2\}^2 \rightarrow \{0,1\} \) is defined as follows:

\[
MV2(X_1, X_2) = X_1^{(0,1)} X_2^{(0)} \lor X_1^{(1,2)} X_2^{(1)} \lor X_1^{(0,2)} X_2^{(2)}.
\]

From Fig 5, we have the following:

Lemma 4.5 \( MV2 \) has the following properties:
1) \( MV2 \) has 6 PIS.
2) \( \tau(\text{MSOP} : MV2) = 3 \).
3) \( \tau(\text{WSOP} : MV2) = 4 \).

Definition 4.13 A 2\( k \)-variable function \( MV2^k \) is defined as follows:

\[
MV2^k(X_1, X_2, \ldots, X_{2k}) = \bigwedge_{i=1}^{k} MV2(X_{2i-1}, X_{2i}).
\]

Theorem 4.10 \( MV2^k \) has the following properties:
1) \( MV2^k \) has \( 6^k \) PIS.
2) \( \tau(\text{MSOP} : MV2^k) = 3^k \).
3) \( \tau(\text{WSOP} : MV2^k) = 4^k \).

Thus, \( \rho(MV2) = 4/3 \), \( \sigma(MV2) = 2/\sqrt{3} = 1.154 \).

4.2 Multiple-Valued Logic Functions

Definition 4.14 A multiple-valued logic function is \( f : P^n \rightarrow P \), where \( P = \{0,1,\ldots,p-1\} \) and \( p \geq 3 \).

In the case of multiple-valued logic functions, two different "sum" operators exist: "MAX" and "truncated sum." When the expression uses Max operators, we need only consider the prime implicants. (We assume that the literal takes only two values. If the literal can be any function of one-variable, then minimum SOP may contain non-prime implicant [12]). However, when the expression uses truncated sum, minimum SOPs may contain non-prime implicants, as well as prime implicants [5].

Example 4.9 Consider the following expressions:

\[
F_1 = X_1^{(1,2)}X_2^{(1,3)} + X_1^{(2,3)}X_2^{(2,2)} \quad (1)
\]

\[
F_2 = X_1^{(1)}X_2^{(1,3)} + X_1^{(2)}X_2^{(2,3)} + X_1^{(3)}X_2^{(1,2)} + 2X_1^{(0)}X_2^{(0)} \quad (2)
\]

Figure 6. Expression using truncated sum.

Fig. 6(a) and (b) are maps for (1) and (2), respectively. Note that both \( F_1 \) and \( F_2 \) are irredundant, and represent the same function. Also, note that \( F_2 \) consists of non-prime implicants. In this case, \( \tau(F_1) = 2 \) and \( \tau(F_2) = 4 \). Thus, \( \rho(f) = 4/2 = 2 \) and \( \sigma(f) = \sqrt{2} \). (End of Example)

For expressions using truncated sum operators, \( \rho(f) \) can be larger than in the binary case. For expressions using MAX operators, the MSOP can be obtained by minimizing expressions for multiple-valued input functions with don't cares [18, 20].

5 Experimental Results and Observations

5.1 Two-valued Case

We generated \( ST(n,k) \) for different \( n \) and \( k \), and obtained their ISOPs. To generate ISOPs, we used Minato's method [13] which is based on Morreale's algorithm [14]. Minato's methods produced WSOPs for all the functions in Table 1.

The 9SYM [6, 19, 22] function shown in page 165 of [1] is identical to \( ST(9,3) \). It has 1680 PIS, \( \tau(\text{WSOP} : 9SYM) = 148 \), and \( \tau(\text{MSOP} : 9SYM) = 84 \). POP [3], a PRESTO [2, 21] type logic minimization algorithm, produced a solution with 148 products. Thus, POP produced a WSOP.

5.2 Multiple-Valued Case

For multiple-valued functions, we generated \( MVO4^2 \). This function has 6400 PIs. To obtain an ISOP, we used the following method [4, 8, 20]: (note that Minato's method does not apply to the multiple-valued case).

1) Generate the set \( S \) of PIs of \( f \).
2) For each cube \( c \) in \( S \), do the following: if \( c \) is contained by \( S - c \), then \( S \leftarrow S - c \).

It produced an ISOP with 256 products, which is the WSOP. The MSOP has only 64 products. So, \( \rho(MVO4^2) = 4 \).

6 Conclusions and Comments

The analysis of ISOPs is important because ISOPs are so often used in logic synthesis. Their importance, however, was recognized 30 years ago when Meo [10] conjectured that...
The largest number of products. Such functions are useful for
representation, showing indeed that
variable functions. In this paper, we settle this open ques-
tion algorithm
arbitrarily. These two results clearly show that it is impor-
tant to develop good minimization algorithms [1, 3, 5, 7].

We also show a class of functions in which an ISOP gen-
erates a WSOP, an ISOP with the
redundancy ratio:

\[ \text{ISOP: number of products in ISOP (Minato's method).} \]

<table>
<thead>
<tr>
<th>n</th>
<th>MSOP</th>
<th>WSOP</th>
<th>ISOP</th>
<th>PI</th>
<th>( \rho )</th>
<th>( \sigma )</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST(3, 1)</td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>1.33333</td>
</tr>
<tr>
<td>ST(3, 2)</td>
<td>6</td>
<td>9</td>
<td>16</td>
<td>16</td>
<td>16</td>
<td>1.77778</td>
</tr>
<tr>
<td>ST(3, 3)</td>
<td>9</td>
<td>27</td>
<td>64</td>
<td>64</td>
<td>64</td>
<td>2.12307</td>
</tr>
<tr>
<td>ST(3, 4)</td>
<td>12</td>
<td>81</td>
<td>256</td>
<td>256</td>
<td>256</td>
<td>3.10049</td>
</tr>
<tr>
<td>ST(3, 5)</td>
<td>15</td>
<td>243</td>
<td>1024</td>
<td>1024</td>
<td>1024</td>
<td>4.21399</td>
</tr>
<tr>
<td>ST(3, 6)</td>
<td>18</td>
<td>729</td>
<td>4096</td>
<td>4096</td>
<td>4096</td>
<td>6.18866</td>
</tr>
<tr>
<td>ST(4, 1)</td>
<td>4</td>
<td>4</td>
<td>6</td>
<td>6</td>
<td>6</td>
<td>1.50000</td>
</tr>
<tr>
<td>ST(4, 2)</td>
<td>8</td>
<td>16</td>
<td>36</td>
<td>36</td>
<td>36</td>
<td>1.42857</td>
</tr>
<tr>
<td>ST(4, 3)</td>
<td>12</td>
<td>24</td>
<td>126</td>
<td>126</td>
<td>126</td>
<td>1.81250</td>
</tr>
<tr>
<td>ST(4, 4)</td>
<td>16</td>
<td>256</td>
<td>1206</td>
<td>1206</td>
<td>1206</td>
<td>3.08333</td>
</tr>
<tr>
<td>ST(4, 5)</td>
<td>20</td>
<td>1024</td>
<td>7776</td>
<td>7776</td>
<td>7776</td>
<td>2.59375</td>
</tr>
</tbody>
</table>

n: number of input variables.
MSOP: number of products in MSOP.
WSOP: number of products in WSOP.
ISOP: number of products in ISOP (Minato's method).
PI: number of prime implicants.
\( \rho \): redundancy ratio: \( \frac{\text{WSOP}(f)}{\text{MSOP}(f)} \)
\( \sigma \): normalized redundancy ratio: \( \frac{\rho}{\bar{\rho}} \).

\( 2^{n-1} \) was the largest number of products in an ISOP of \( n \)-variable functions. In this paper, we settle this open question, showing indeed that \( 2^{n-1} \) is firm upper bound.

We also show a class of functions in which an ISOP generation algorithm [13] produces a WSOP, an ISOP with the largest number of products. Such functions are useful for comparing the performance of two-level logic minimizers. We also show, for this class, that as the number of variables increases, \( \rho \), the ratio of the number of products in the WSOP to the number of products in the MSOP increases arbitrarily. These two results clearly show that it is important to develop good minimization algorithms [1, 4, 8, 16]. Specifically, it shows that 1) reasonable algorithms can produce poor results and 2) these poor results can be far from minimum.

A key part of our results is a theorem that allows us to magnify small values of \( \rho \). That is, we compose functions on many variables with large \( \rho \) from functions on few variables with small \( \rho \).

We present experimental results and we extend our analysis to multiple-valued logic. For example, we show a multiple-valued function whose WSOP has four times the products than the MSOP.

Acknowledgments

This work was supported in part by a Grant in Aid for Scientific Research of the Ministry of Education, Science, Culture and Sports of Japan. Discussions with Dr. Shin-ichi Minato were quite useful.

References