

Logic Synthesis of EXOR Projected Sum of Products

Anna Bernasconi¹, Valentina Ciriani², and Roberto Cordone²

¹ Department of Computer Science, University of Pisa Italy, annab@di.unipi.it

² DTI, University of Milano Italy, {ciriani,cordone}@dti.unimi.it

Abstract. We define a new algebraic form for Boolean function representation, called *EXOR-Projected Sum of Products (EP-SOP)*, consisting in a four level network that can be easily implemented in practice. Deriving an optimal EP-SOP from an optimal SOP form is a NP^{NP} -hard problem; nevertheless we propose a very efficient approximation algorithm, which returns, in polynomial time, an EP-SOP form whose cost is guaranteed to be near the optimum. Experimental evidence shows that for about 35% of the classical synthesis benchmarks, EP-SOP networks have a smaller area and delay with respect to the optimal SOPs (sometimes gaining even 40-50% of the area). Since the computational times required are extremely short, we recommend the use of the proposed approach as a post-processing step after SOP minimization.

1 Introduction

The classical approach to logic synthesis is the minimization of two-level SOP networks [2, 4, 13]. In this framework the resulting networks have a very low delay, thanks to the fixed number of levels, and the SOP expressions can be computed in a reasonable amount of time. To build networks with a more compact area, multi-level network synthesis has been proposed and widely studied [8, 17]. The drawbacks of this approach are the unbounded number of levels (and therefore the longer delay), as well as the much larger computational time required to synthesize the network. In an attempt to establish an effective trade-off between these two opposite approaches, recent studies have proposed the optimization of networks with a fixed number of levels (typically, three or four levels) [1, 5–7, 12, 14, 16]. Three levels of logic are enough to produce a minimal network for most of the Boolean functions; and in many cases three-level logic is a good compromise between circuit speed, circuit size, and the time needed for the minimization procedure [15]. Three and four-level logic networks are typically more compact than the corresponding SOPs, but the computational time required to compute them can be much longer.

The aim of this paper is to define a network with a bounded number of levels that can be easily implemented in practice and synthesized in a competitive time with respect to two-level synthesis. For this purpose, we propose a four-level network, *EXOR-Projected Sum of Products (EP-SOP)*, which can be built in a very fast post-processing step from an optimal two-level SOP. We first define

the algebraic form of EP-SOP networks, and prove that deriving an optimal EP-SOP from an optimal SOP form is a hard problem (NP^{NP} -hard). We then describe an approximation algorithm, which returns in polynomial time an EP-SOP form whose cost is guaranteed to be near the optimum. Our experimental results show that in about 35% of the classical synthesis benchmarks the EP-SOP obtained has area and delay smaller than those of an optimal SOP form (sometimes gaining even 40-50% of the area). The computational times required are extremely short, thus recommending the use of this approach as a post-processing step after SOP minimization.

Before defining EP-SOP forms, we introduce them informally through an example. Let us consider the Boolean function f shown on the left side of Figure 1. An optimal SOP representation for f is $\phi = \bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + \bar{x}_1\bar{x}_2x_3 + x_1x_2x_3 + x_3\bar{x}_4$. The right side of Figure 1 represents the projections of f onto the two subspaces where $x_1 = x_2$ and $x_1 \neq x_2$, respectively. As described in the Karnaugh maps on the right side of the figure, the projection of f onto the space $x_1 \neq x_2$ is covered by the optimal SOP form $\bar{x}_3 + x_3\bar{x}_4$, while the projection onto the space $x_1 = x_2$ is covered by x_3 . Notice that both SOP forms are much more compact than the original one, because minterms, which were not adjacent in the original Karnaugh map, now merge into new larger prime cubes. For example, the two products $\bar{x}_1x_2\bar{x}_3$ and $x_1\bar{x}_2\bar{x}_3$, which cannot be merged in the original Karnaugh map, correspond to the products $x_2\bar{x}_3$ and $\bar{x}_2\bar{x}_3$, which can be unified into product \bar{x}_3 in the lower Karnaugh map on the right side.

Since the two subspaces, $x_1 = x_2$ and $x_1 \neq x_2$, have characteristic functions equal to $(x_1 \oplus \bar{x}_2)$ and $(x_1 \oplus x_2)$, respectively, f can be expressed as $f \equiv (x_1 \oplus \bar{x}_2)x_3 + (x_1 \oplus x_2)(\bar{x}_3 + \bar{x}_4)$. Figure 2 shows how this form can be easily implemented by using a single 2-fan in EXOR gate and two PLAs.

As the previous example shows, the products of a generic SOP ϕ can be classified into two subsets: those that are entirely included into one of the two subspaces $x_1 = x_2$ and $x_1 \neq x_2$ (for example, in Figure 1 the product $\bar{x}_1x_2\bar{x}_3$ belongs entirely to the subspace $x_1 \neq x_2$) and those that intersect both of them, which we will call *crossing products* (for example, in Figure 1 the product $x_3\bar{x}_4$). In general, it is not always convenient to project a crossing product, since this produces two smaller products, which reside into both subspaces. Therefore, we can choose whether projecting the crossing products or keeping them unprojected. In the second case, the resulting expression also includes a SOP form (called *remainder*) containing all the crossing products. We call the overall form *EP-SOP with remainder*.

Figure 3 reports the same example of Figure 1, in which the only crossing product $x_3\bar{x}_4$ is not projected. In this case, the resulting EP-SOP with remainder form is $f \equiv (x_1 \oplus \bar{x}_2)x_3 + (x_1 \oplus x_2)\bar{x}_3 + x_3\bar{x}_4$.

We can observe that EP-SOP expressions can be seen as Boolean factorized forms. Factorization of literal terms is a widely studied field in multi-level logic [3, 17]. Most of the proposed methods produce disjoint factorization (see [8] for an introduction). In contrast, the factorization of an EP-SOP form is not disjoint since a literal can stay simultaneously in the projected

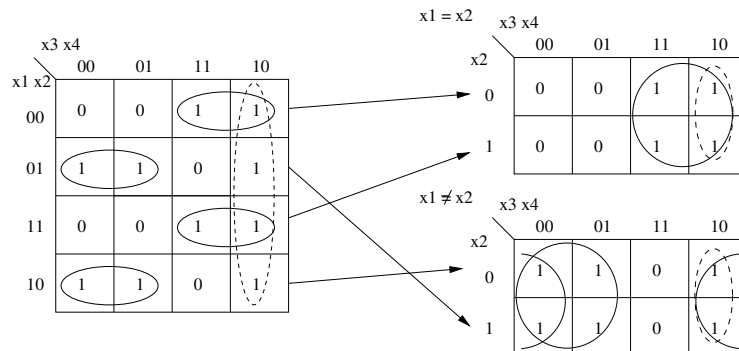


Fig. 1. Karnaugh maps of a function f (left side) and its projections onto $x_1 \oplus \bar{x}_2$ (right side, top) and $x_1 \oplus x_2$ (right side, bottom).

SOPs and in the corresponding EXORs. For example, in the EP-SOP form $(x_1 \oplus \bar{x}_2)(x_2\bar{x}_4 + \bar{x}_3x_4) + (x_1 \oplus x_2)(x_2\bar{x}_3 + \bar{x}_3\bar{x}_4)$ the literal x_2 appears both in the EXORs and in the SOPs.

Finally, EP-SOP forms share some similarities with another model of Boolean function representation: the *Linearly-Transformed BDDs* [10, 11]. LTBDs are binary decision diagrams whose nodes are labeled with EXORs of variables, instead of just single variables. Thus, the node on the first level of a LTBD, if labeled with an EXOR, defines the same kind of decomposition on which EP-SOPs are based.

The remainder of this paper is organized as follows. Section 2 describes the algebraic expression for EP-SOPs with and without remainder. Section 3 characterizes the computational complexity of the problem. Section 4 presents an approximation algorithm for EP-SOP synthesis, and proves that its solution is nearly optimal. In the end, Section 5 discusses the experimental results.

2 EP-SOP representation of Boolean functions

The following two sections formally describe EP-SOP expressions with and without remainder, and show how to derive them from an original optimal SOP form.

2.1 EP-SOP without remainder

Let us consider a SOP form ϕ , and a couple of variables x_i and x_j , where without loss of generality $i < j$. The space $\{0, 1\}^n$ can be partitioned into two disjoint subspaces: the space defined by the characteristic function $\chi_{\bar{\oplus}} = (x_i \oplus \bar{x}_j)$, i.e., the space where $x_i = x_j$, and its complement defined by the function $\chi_{\oplus} = (x_i \oplus x_j)$, i.e., the space where $x_i \neq x_j$.

We can write ϕ as the sum (union) of its two projections, $\phi_{\bar{\oplus}}$ and ϕ_{\oplus} , onto these two spaces. Even if the projections allow us to eliminate a variable ad

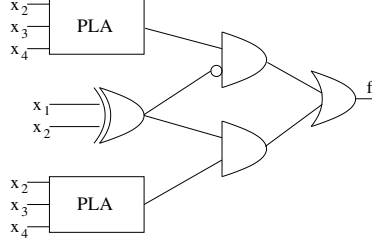


Fig. 2. EP-SOP-network without remainder for the function in Figure 1.

libitum between x_i and x_j , we always remove x_i (the one with lower index). In order to perform the two projections we must project one by one the products $p \in \phi$, considering four cases.

Algorithm 1 (Project. onto $(x_i \oplus \bar{x}_j)$ and $(x_i \oplus x_j)$ for EP-SOP) Given a SOP form $\phi = p_1 + p_2 + \dots + p_m$, for each p in $\{p_1, p_2, \dots, p_m\}$ project p in ϕ_{\oplus} or in ϕ_{\ominus} using the following strategy:

1. If p contains both x_i and x_j (possibly complemented), i.e., $p = x_i x_j q$, where q is a product of literals, p has no projection onto the subspace where $x_i \neq x_j$. Thus, no product will be added to ϕ_{\oplus} . By contrast, the projection of $p = x_i x_j q$ onto the subspace where $x_i = x_j$ gives the product $x_j q$, which will be added to ϕ_{\ominus} . The three other cases ($p = x_i \bar{x}_j q$, $p = \bar{x}_i x_j q$, $p = \bar{x}_i \bar{x}_j q$) can be handled in a similar way.
2. If p contains x_i (possibly complemented) and not x_j , i.e., $p = x_i q$, where q is a product of literals, the projection of p onto the subspace where $x_i \neq x_j$ gives the product $\bar{x}_j q$, which will be added to ϕ_{\oplus} . The projection of p onto the subspace where $x_i = x_j$ gives the product $x_j q$, which will be added to ϕ_{\oplus} . The other case ($p = \bar{x}_i q$) can be handled in a similar way: $x_j q$ will be added to ϕ_{\oplus} , and $\bar{x}_j q$ will be added to ϕ_{\ominus} .
3. If p contains x_j (possibly complemented) and not x_i , i.e., $p = x_j q$, where q is a product of literals, the projections of p onto both subspaces leave the product unchanged, thus $p = x_j q$ will be added to both ϕ_{\oplus} and ϕ_{\ominus} . The other case ($p = \bar{x}_j q$) can be handled in the same way, by adding p to both ϕ_{\oplus} and ϕ_{\ominus} .
4. If p does not contain x_i , \bar{x}_i , x_j , \bar{x}_j , the projections of p onto both subspaces leave the product unchanged, thus p will be added to both ϕ_{\oplus} and ϕ_{\ominus} .

Observe that the last three type of products are indeed crossing products, which are projected onto the two spaces, while the products containing both x_i and x_j are projected only onto one of them.

Example 1. Let us consider the Boolean function f shown on the left side of Figure 1. An optimal SOP representation for f is $\phi = \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + \bar{x}_1 \bar{x}_2 x_3 + x_1 x_2 x_3 + x_3 \bar{x}_4$. Suppose to project ϕ onto the spaces $(x_1 \oplus \bar{x}_2)$ and

$(x_1 \oplus x_2)$. The first product in ϕ contains both x_1 and x_2 , thus it is not a crossing product (strategy 1 of Algorithm 1). Since x_1 is complemented and x_2 is not complemented we project this product onto the space $(x_1 \oplus x_2)$ (in fact, $x_1 \neq x_2$). The projected product is $x_2\bar{x}_3$. The unique crossing product of ϕ is $x_3\bar{x}_4$, since it does not contain x_1 and x_2 . This product will be inserted in both the spaces without removing any literal. The overall projection will return the form $(x_1 \oplus \bar{x}_2)(\bar{x}_2x_3 + x_2x_3 + x_3\bar{x}_4) + (x_1 \oplus x_2)(x_2\bar{x}_3 + \bar{x}_2\bar{x}_3 + x_3\bar{x}_4)$. Note that the SOP forms of the projected spaces are not minimal. Minimizing them we obtain $(x_1 \oplus \bar{x}_2)x_3 + (x_1 \oplus x_2)(\bar{x}_3 + x_3\bar{x}_4)$.

We can now formally define the EP-SOP expressions. These forms can be derived starting from a SOP representation ϕ of a Boolean function f in two steps.

First we project ϕ onto the two subspaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$, as explained before, and we obtain the following expression.

Definition 1. Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$, and let ϕ be a SOP representation of f . Given a couple of variables x_i and x_j , the (i, j) -EP-SOP of f is the expression

$$\xi_{ij} = (x_i \oplus x_j)\phi_{\oplus} + (x_i \oplus \bar{x}_j)\phi_{\bar{\oplus}},$$

where ϕ_{\oplus} and $\phi_{\bar{\oplus}}$ are the projections of ϕ onto the spaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$, respectively.

After the projection we can further minimize the two SOPs ϕ_{\oplus} and $\phi_{\bar{\oplus}}$ in order to minimize the EP-SOP ξ_{ij} .

Definition 2. Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$, and let ϕ be a SOP representation of f . Given a couple of variables x_i and x_j , the minimal (i, j) -EP-SOP of f is the expression

$$\xi_{ij}^{(min)} = (x_i \oplus x_j)\phi_{\oplus}^{(min)} + (x_i \oplus \bar{x}_j)\phi_{\bar{\oplus}}^{(min)},$$

where $\phi_{\oplus}^{(min)}$ and $\phi_{\bar{\oplus}}^{(min)}$ are two minimal SOP forms representing the projections of ϕ onto the spaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$, respectively.

In the previous definitions we have fixed a single couple of variables, but we are interested in finding the minimal EP-SOP representation of a Boolean function, i.e., the expression containing the minimum number of products among all possible minimal EP-SOP with respect to *any* couple of variables.

Let $|\phi|$ denote the number of products in a SOP ϕ , and $|\xi| = |\phi_{\oplus}| + |\phi_{\bar{\oplus}}|$ the overall number of products in an EP-SOP ξ .

Definition 3. The minimal EP-SOP representation of a Boolean function f is given by the EP-SOP expression ξ_{MIN} such that

$$|\xi_{MIN}| = \min_{i,j} |\xi_{ij}^{(min)}|.$$

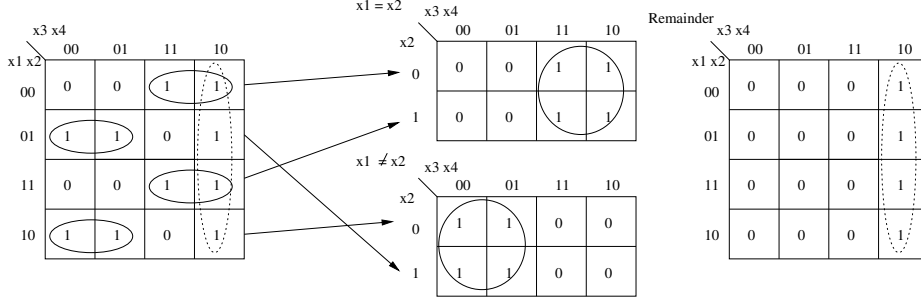


Fig. 3. Karnaugh maps of a function f (left side), its projections onto $x_1 \oplus \bar{x}_2$ (center, top) and $x_1 \oplus x_2$ (center, bottom), and the remainder (right side).

2.2 EP-SOP with remainder

As already noted, when we project a SOP form onto the two spaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$, some products will appear only once in the final expression, precisely the products containing the two literals defining the projection spaces, while the other products (crossing products) will appear twice, one in each projected SOP.

In order to keep the number of products as small as possible, we introduce the notion of *EP-SOP with remainder*.

Algorithm 2 (Proj. onto $(x_i \oplus \bar{x}_j)$ and $(x_i \oplus x_j)$ for EP-SOP with rem.)

Given a SOP form $\phi = p_1 + p_2 + \dots + p_m$, for each p in $\{p_1, p_2, \dots, p_m\}$ project p in ϕ_{\oplus} or in $\phi_{\oplus\bar{}}$, or insert it in the remainder ρ using the following strategy:

1. If p contains both x_i and x_j (possibly complemented), i.e., $p = x_i x_j q$, where q is a product of literals, p has no projection onto the subspace where $x_i \neq x_j$. Thus, no product will be added to ϕ_{\oplus} . By contrast, the projection of $p = x_i x_j q$ onto the subspace where $x_i = x_j$ gives the product $x_j q$, which will be added to ϕ_{\oplus} . The three other cases ($p = x_i \bar{x}_j q$, $p = \bar{x}_i x_j q$, $p = \bar{x}_i \bar{x}_j q$) can be handled in a similar way.
2. Otherwise (p is a crossing product) insert p in the remainder.

For example, let us consider the Boolean function f shown on the left side of Figure 3. The unique crossing product of ϕ is $x_3 \bar{x}_4$ since it does not contain x_1 and x_2 . This product will be inserted now in the remainder. The overall projection will return the form: $(x_1 \oplus \bar{x}_2)(\bar{x}_2 x_3 + x_2 x_3) + (x_1 \oplus x_2)(x_2 \bar{x}_3 + \bar{x}_2 \bar{x}_3) + x_3 \bar{x}_4$. Minimizing the projected SOPs we obtain $(x_1 \oplus \bar{x}_2)x_3 + (x_1 \oplus x_2)\bar{x}_3 + x_3 \bar{x}_4$.

Formally we have:

Definition 4. Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$, and let ϕ be a SOP representation of f . Given a couple of variable x_i and x_j , the (i, j) -EP-SOP with remainder of f is the expression

$$\psi_{ij} = (x_i \oplus x_j)\phi'_{\oplus} + (x_i \oplus \bar{x}_j)\phi'_{\oplus\bar{}} + \rho,$$

where ϕ'_{\oplus} and ϕ'_{\ominus} are the two projections of the products of ϕ containing both x_i and x_j (possibly complemented) onto the spaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$, respectively, and ρ is the sum of all crossing products of ϕ .

In other words we project onto the subspaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$ only the products that entirely reside in them, while the crossing products are not projected, but are inserted in the remainder ρ . Again for this form, we can further minimize the projected SOPs ϕ'_{\oplus} and ϕ'_{\ominus} in order to obtain a more compact expression:

Definition 5. Let $f : \{0, 1\}^n \rightarrow \{0, 1\}$, and let ϕ be a SOP representation of f . Given a couple of variable x_i and x_j , the minimal (i, j) -EP-SOP with remainder of f is the expression

$$\psi_{ij}^{(min)} = (x_i \oplus x_j)\phi'_{\oplus}{}^{(min)} + (x_i \oplus \bar{x}_j)\phi'_{\ominus}{}^{(min)} + \rho^{(min)},$$

where $\phi'_{\oplus}{}^{(min)}$ and $\phi'_{\ominus}{}^{(min)}$ are two minimal SOP forms representing the projections of the products of ϕ containing both x_i and x_j (possibly complemented) onto the spaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$, respectively, and $\rho^{(min)}$ is the optimal sum of all other products of ϕ .

The overall minimal form (with respect to any possible couple of variables) is described as follows. Let $|\psi|$ denote the number of products in an EP-SOP with remainder, i.e., $|\psi| = |\phi'_{\oplus}| + |\phi'_{\ominus}| + |\rho|$.

Definition 6. The minimal EP-SOP with remainder representation of a Boolean function f is given by the EP-SOP expression ψ_{MIN} such that

$$|\psi_{MIN}| = \min_{i,j} |\psi_{ij}^{(min)}|.$$

Note that if we start from a minimal SOP, the remainder is already minimal, i.e., the number of its products cannot be further reduced: $|\rho^{(min)}| = |\rho|$.

We cannot decide in advance which one of the two EP-SOP expressions (with or without remainder) is the more compact. On one hand, if we project the crossing products in the two spaces we could further minimize them. On the other hand it could be more convenient kipping them in the remainder.

For example, consider the minimal SOP form $\phi = \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 + x_3 \bar{x}_4$ and the couple x_1 and x_2 . The minimal $(1, 2)$ -EP-SOP without remainder is $(x_1 \oplus \bar{x}_2) x_3 \bar{x}_4 + (x_1 \oplus x_2) (\bar{x}_3 + x_3 \bar{x}_4)$, while the minimal $(1, 2)$ -EP-SOP with remainder is $(x_1 \oplus x_2) \bar{x}_3 + x_3 \bar{x}_4$. In this case the form with remainder is clearly more compact.

Alternatively consider the minimal SOP form $\phi = \bar{x}_1 \bar{x}_2 x_3 \bar{x}_4 + x_1 x_2 x_3 \bar{x}_4 + \bar{x}_1 x_2 \bar{x}_3 + x_1 \bar{x}_2 \bar{x}_3 \bar{x}_4 + \bar{x}_1 \bar{x}_3 x_4 + x_2 \bar{x}_3 x_4$ and the couple x_1 and x_2 . The minimal $(1, 2)$ -EP-SOP without remainder is $(x_1 \oplus \bar{x}_2) (x_3 \bar{x}_4 + \bar{x}_3 x_4) + (x_1 \oplus x_2) (x_2 \bar{x}_3 + \bar{x}_3 \bar{x}_4)$, while the minimal $(1, 2)$ -EP-SOP with remainder is $(x_1 \oplus \bar{x}_2) (x_3 \bar{x}_4) + (x_1 \oplus x_2) (x_2 \bar{x}_3 + \bar{x}_3 \bar{x}_4) + \bar{x}_1 \bar{x}_3 x_4 + x_2 \bar{x}_3 x_4$. In this case the form without remainder is more convenient.

3 Computational Complexity

In this section we analyze the computational complexity of the following problem: given a *minimal* SOP form ϕ for a Boolean function f and a couple of variables x_i and x_j , find a minimal (i, j) -EP-SOP of f .

Since projecting ϕ is easy (polynomial), as shown in Section 2, the core of the problem is the minimization of the two projected SOPs. In general $\phi_{\oplus}^{(min)}$ and $\phi_{\ominus}^{(min)}$ are different from the projections ϕ_{\oplus} and ϕ_{\ominus} , even if ϕ is already in minimal form. Indeed, projecting the single products of ϕ , we have no guarantee that the resulting SOP forms ϕ_{\oplus} and ϕ_{\ominus} are still prime and irredundant.

Notice that the more common projections of a minimal SOP form ϕ onto the spaces x_i and \bar{x}_i (Shannon projections) are guaranteed to be minimal. For instance, the projection (Shannon decomposition) with respect to x_1 and \bar{x}_1 of the minimal SOP form $\phi = \bar{x}_1x_2\bar{x}_3 + x_1\bar{x}_2\bar{x}_3 + x_1x_3\bar{x}_4$ is $\bar{x}_1(x_2\bar{x}_3) + x_1(\bar{x}_2\bar{x}_3 + x_3\bar{x}_4)$, and the two projected SOP forms are already minimal.

In [19], the decision version of the problem of finding a minimal SOP representation of a Boolean function f starting from any SOP for f (*SOP-2-MIN SOP*) has been proved to be NP^{NP} -complete. Finding $\phi_{\oplus}^{(min)}$ and $\phi_{\ominus}^{(min)}$ from ϕ_{\oplus} and ϕ_{\ominus} when the starting SOP ϕ is *minimal*, could nevertheless be an easy (polynomial) problem? We show here that the answer to this question is negative, since the problem under study turns out to be at least as difficult as *SOP-2-MIN SOP*. Let us first formally define the two problems.

Problem 1 (SOP-2-MIN SOP).

INSTANCE: A SOP formula ϕ and an integer k .

QUESTION: Is there a SOP ϕ' with at most k products and for which $\phi' \equiv \phi$?

Problem 2 (MIN SOP-2-MIN (i, j) -EP-SOP).

INSTANCE: Minimal SOP formula ϕ , a couple of variables x_i and x_j .

QUESTION: Find the minimal (i, j) -EP-SOP $\xi_{ij}^{(min)}$:

$$\xi_{ij}^{(min)} = (x_i \oplus x_j)\phi_{\oplus}^{(min)} + (x_i \oplus \bar{x}_j)\phi_{\ominus}^{(min)}.$$

The proof of the hardness of *MIN SOP-2-MIN (i, j) -EP-SOP* is based on the concept of *polynomial time Turing reduction*, defined as follows.

Definition 7. A problem Π is Turing-reducible to a problem Π' , $\Pi \preceq_T \Pi'$, if there is an algorithm A that solves Π by using a hypothetical subroutine S for solving Π' such that, if S were a polynomial time algorithm for Π' , then A would be a polynomial time algorithm for Π .

Let us consider the complexity of the following problem:

Problem 3 (MIN SOP+PRODUCT-2-MIN SOP).

INSTANCE: A minimal SOP formula ϕ for a Boolean function f and a product p .

QUESTION: Find a minimal SOP formula for the function $f + p$

Lemma 1. *MIN SOP+PRODUCT-2-MIN SOP is a NP^{NP} -hard problem.*

Proof. We show that the problem *SOP-2-MIN SOP* is Turing-reducible to *MIN SOP+PRODUCT-2-MIN SOP*.

Consider a SOP $\phi = p_1 + p_2 + \dots + p_m$ for a function f depending on n variables. In order to find a minimal SOP ϕ' for f , we can proceed iteratively as follows.

First we compute a minimal SOP for the function $p_1 + p_2$. Note that this corresponds to deriving a minimal SOP $\phi^{(1)}$ for the union of the minimal SOP p_1 and a product p_2 .

In general, step i of this iterative process would consist in computing a minimal SOP $\phi^{(i)}$ for the function $\phi^{(i-1)} + p_i$, defined as the union of a minimal SOP, $\phi^{(i-1)}$, and the product p_i .

If we could perform this step in polynomial time, then we could solve the problem *SOP-2-MIN SOP* in polynomial time, performing $m - 1$ iterations. ■

Based on the previous lemma, we can now prove our main complexity result.

Theorem 1. *MIN SOP-2-MIN (i, j) -EP-SOP is NP^{NP} -hard.*

Proof. To prove the thesis, it is enough to show that the *MIN SOP+PRODUCT-2-MIN SOP* problem is Turing-reducible to *MIN SOP-2-MIN (i, j) -EP-SOP*.

Consider a minimal SOP $\phi^{(min)}$ for a function f , depending on n variables x_1, x_2, \dots, x_n , and a product p . Then consider the SOP

$$\Phi = x_{n+1}\phi^{(min)} + x_{n+2}p,$$

where x_{n+1} and x_{n+2} are two additional variables. Suppose that $\phi^{(min)}$ contains t products.

First of all observe that Φ is a minimal SOP form. Indeed, $x_{n+1}\phi^{(min)}$ is minimal and does not cover the points of the cube described by $\bar{x}_{n+1}x_{n+2}p$. Thus we need at least a product to cover these points. This means that a minimal SOP must contain at least $t + 1$ products, and this immediately implies that Φ is minimal.

Now, let us derive an EP-SOP from Φ with respect to the couple of additional variables x_{n+1} and x_{n+2} . We get the following expression:

$$\xi_{n+1,n+2} = (x_{n+1} \oplus x_{n+2})(\bar{x}_{n+2}\phi^{(min)} + x_{n+2}p) + (x_{n+1} \oplus \bar{x}_{n+2})(x_{n+2}\phi^{(min)} + x_{n+2}p).$$

If we could derive $\xi_{n+1,n+2}^{(min)}$ in polynomial time, then we would be able to minimize the two expressions $(\bar{x}_{n+2}\phi^{(min)} + x_{n+2}p)$ and $(x_{n+2}\phi^{(min)} + x_{n+2}p)$ in polynomial time. This implies that we could solve in polynomial time an instance of *MIN SOP+PRODUCT-2-MIN SOP* since, from the second expression, we have $(x_{n+2}\phi^{(min)} + x_{n+2}p)^{(min)} = x_{n+2} \cdot (\phi^{(min)} + p)^{(min)}$. ■

4 Polynomial time approximation algorithms

In the previous section we have shown that, even if we start from a minimal SOP form and we fix a couple of variables x_i and x_j , finding a *minimal* (i, j) -EP-SOP is a hard problem. In this section we will show how it is possible to find a good solution to the latter problem in polynomial time.

In a minimization framework, a p -approximation algorithm (i.e., an algorithm with approximation ratio p) guarantees that the cost C of its solution is such that $C/C^* \leq p$, where C^* is the cost of an optimal solution [9]. Both heuristics and approximation algorithms do not guarantee the minimality of their solution, but while we cannot perform any evaluation on the result of a heuristic, an approximation algorithm gives guaranteed near-optimum solutions.

We now describe a *polynomial approximation algorithm* for the problem of finding the minimal EP-SOP (minimal EP-SOP with remainder) representation of a function f starting from a *minimal SOP* ϕ for f that guarantees an approximation ratio of 4 (2). The main idea is to select the most frequent couple of variables in the minimal SOP representation, and project the expression with respect to this couple. The two projected SOPs will be further synthesized with a SOP polynomial heuristic. The overall algorithm is described below.

Algorithm 3 (Approximation Algorithm) *Given a minimal SOP expression ϕ :*

Step 1 *Select the couple of variables x_i and x_j simultaneously appearing (possibly complemented) with the highest frequency in the products of ϕ .*

Step 2 *Project ϕ onto the spaces $(x_i \oplus x_j)$ and $(x_i \oplus \bar{x}_j)$ as described in Algorithms 1 or 2.*

Step 3 *Minimize the two projected SOPs using a polynomial time heuristic (e.g., ESPRESSO NOT EXACT).*

Notice that the two versions (with and without remainder) differ only in the projection Step 2 discussed in Section 2. The three steps can be performed in polynomial time.

We now prove that the proposed synthesis strategy is indeed an approximation algorithm for the two EP-SOP minimization problems.

Consider first the problem without remainder. In order to prove that the cost $|\xi_{ij}^{(min)}|$ of our solution is such that $|\xi_{ij}^{(min)}|/|\xi_{MIN}|$ is upper bounded by a constant, where $|\xi_{MIN}|$ is the cost of an optimal solution, we first find a lower bound for $|\xi_{MIN}|$, as shown in Lemma 2 and then an upper bound for $|\xi_{ij}^{(min)}|$, as shown in Theorem 2. We follow a similar strategy for EP-SOPs with remainder using Lemma 3 and Theorem 3.

Let us consider a minimal SOP form ϕ for a Boolean function f and a minimal EP-SOP without remainder ξ_{MIN} .

Lemma 2.

$$|\xi_{MIN}| \geq \frac{1}{2}|\phi|.$$

Proof. Let us suppose that the variables x_h and x_k are such that

$$\xi_{hk}^{(min)} = (x_h \oplus x_k)\phi_{\oplus}^{(min)} + (x_h \oplus \bar{x}_k)\phi_{\ominus}^{(min)} = \xi_{MIN}.$$

We build a SOP ϕ_{hk} starting from $\xi_{hk}^{(min)}$. Let $\phi_{\oplus}^{(min)} = \sum_{i=1}^{|\phi_{\oplus}^{(min)}|} p_i$ and $\phi_{\ominus}^{(min)} = \sum_{i=1}^{|\phi_{\ominus}^{(min)}|} q_i$. Thus

$$\begin{aligned} \phi_{hk} &= (x_h \oplus x_k) \sum_{i=1}^{|\phi_{\oplus}^{(min)}|} p_i + (x_h \oplus \bar{x}_k) \sum_{i=1}^{|\phi_{\ominus}^{(min)}|} q_i \\ &= x_h \bar{x}_k \sum_{i=1}^{|\phi_{\oplus}^{(min)}|} p_i + \bar{x}_h x_k \sum_{i=1}^{|\phi_{\oplus}^{(min)}|} p_i + x_h x_k \sum_{i=1}^{|\phi_{\oplus}^{(min)}|} q_i + \bar{x}_h \bar{x}_k \sum_{i=1}^{|\phi_{\ominus}^{(min)}|} q_i. \end{aligned}$$

Since ϕ is minimal, we have that $|\phi_{hk}| \geq |\phi|$. Moreover, since

$$|\phi_{hk}| = 2|\xi_{hk}^{(min)}| = 2(|\phi_{\oplus}^{(min)}| + |\phi_{\ominus}^{(min)}|) \geq |\phi|,$$

the thesis immediately follows:

$$|\xi_{MIN}| = |\xi_{hk}^{(min)}| = |\phi_{\oplus}^{(min)}| + |\phi_{\ominus}^{(min)}| \geq \frac{1}{2}|\phi|.$$

■

A similar result holds for the EP-SOPs with remainder.

Lemma 3.

$$|\psi_{MIN}| \geq \frac{1}{2}(|\phi| + |\rho|),$$

where ρ is the remainder of ψ_{MIN} .

Proof. Let us suppose that the EP-SOP form

$$\psi_{hk}^{(min)} = (x_h \oplus x_k)\phi_{\oplus}'^{(min)} + (x_h \oplus \bar{x}_k)\phi_{\ominus}'^{(min)} + \rho$$

is minimal, with respect to the overall number of products, among all other EP-SOPs with remainder, i.e., $\psi_{hk}^{(min)} = \psi_{MIN}$.

As in the proof of Lemma 2, we derive a SOP representation ϕ_{hk} for f from $\psi_{hk}^{(min)}$, and we get

$$|\phi_{hk}| = 2(|\phi_{\oplus}'^{(min)}| + |\phi_{\ominus}'^{(min)}|) + |\rho| \geq |\phi|.$$

Thus

$$|\phi_{\oplus}'^{(min)}| + |\phi_{\ominus}'^{(min)}| + \frac{1}{2}|\rho| \geq \frac{1}{2}|\phi|,$$

and we immediately derive

$$|\psi_{MIN}| = |\psi_{hk}^{(min)}| = |\phi_{\oplus}'^{(min)}| + |\phi_{\ominus}'^{(min)}| + |\rho| \geq \frac{1}{2}(|\phi| + |\rho|).$$

■

We now prove that if we project the starting minimal SOP ϕ with respect to the couple of variables x_i and x_j simultaneously appearing (possibly complemented) with the highest frequency in its products, we get a solution whose approximation ratio in the worst case is bounded by 4 for the EP-SOP without remainder, and by 2 for the EP-SOP with remainder.

For a couple of variables x_i and x_j , let us denote with ν_{ij} the number of products in ϕ containing both x_i and x_j , possibly complemented.

Theorem 2. *Let ξ_{MIN} be a minimal EP-SOP of a Boolean function f , and ϕ a minimal SOP form for f . Let $\xi_{ij}^{(min)}$ be the minimal (i, j) -EP-SOP derived with respect to the couple of variables $(x_i$ and $x_j)$ appearing with the highest frequency in the products of ϕ . Then*

$$\frac{|\xi_{ij}^{(min)}|}{|\xi_{MIN}|} \leq \frac{|\xi_{ij}|}{|\xi_{MIN}|} \leq 4 - \frac{2\nu_{ij}}{|\phi|}.$$

Proof. Observe that

$$|\xi_{ij}^{(min)}| \leq |\xi_{ij}| \leq 2|\phi| - \nu_{ij},$$

since the ν_{ij} products containing the two variables x_i and x_j appear only once in ξ_{ij} , while all other products appear twice. The thesis follows since Lemma 2 implies that $|\xi_{MIN}| \geq \frac{|\phi|}{2}$. \blacksquare

Observe that in the best case $\nu_{ij} = |\phi|$, thus the bound becomes

$$\frac{|\xi_{ij}^{(min)}|}{|\xi_{MIN}|} \leq \frac{|\xi_{ij}|}{|\xi_{MIN}|} \leq 2,$$

while in the worst case $\nu_{ij} = 1$ and we have

$$\frac{|\xi_{ij}^{(min)}|}{|\xi_{MIN}|} \leq \frac{|\xi_{ij}|}{|\xi_{MIN}|} \leq 4 - \frac{2}{|\phi|} \leq 4.$$

Theorem 3 shows a similar result for the EP-SOPs with remainder.

Theorem 3. *Let ψ_{MIN} be a minimal EP-SOP with remainder of a Boolean function f , and ϕ be a minimal SOP form for f .*

Let $\psi_{ij}^{(min)}$ be the minimal (i, j) -EP-SOP with remainder derived with respect to the couple of variables x_i and x_j appearing with the highest frequency in the products of ϕ . Then

$$\frac{|\psi_{ij}^{(min)}|}{|\psi_{MIN}|} \leq \frac{|\psi_{ij}|}{|\psi_{MIN}|} \leq 2.$$

Proof. First observe that

$$|\psi_{ij}^{(min)}| \leq |\psi_{ij}| = |\phi|,$$

since each product of ϕ appears only once in ψ_{ij} , in one of the two factors ϕ'_{\oplus} and ϕ'_{\ominus} , or in the remainder ρ , and the two expressions ϕ'_{\oplus} and ϕ'_{\ominus} are further minimized. Moreover, Lemma 3 implies that

$$|\psi_{MIN}| \geq \frac{1}{2}(|\phi| + |\rho|).$$

Now suppose that the projections in ψ_{MIN} are performed with respect to the variables x_h and x_k . Thus, since $|\rho| = |\phi| - \nu_{hk}$, with $\nu_{hk} \leq \nu_{ij}$ and $\nu_{ij} \leq |\phi|$, we get

$$\frac{|\psi_{ij}^{(min)}|}{|\psi_{MIN}|} \leq \frac{|\psi_{ij}|}{|\psi_{MIN}|} \leq \frac{|\phi|}{|\phi| - \nu_{ij}/2} \leq \frac{|\phi|}{|\phi| - |\phi|/2} \leq 2.$$

■

Note that Theorem 2 and Theorem 3 show that the approximation ratios hold even if the factors ϕ_{\oplus} , ϕ_{\ominus} , ϕ'_{\oplus} and ϕ'_{\ominus} are not minimized. Therefore, the algorithms proposed are indeed polynomial approximation algorithms for the given problems. The resulting EP-SOP without remainder ξ_{ij} has a size that can be upper bounded by $(4 - 2\nu_{ij}/|\phi|)|\xi_{MIN}|$, i.e., in the worst case by $4|\xi_{MIN}|$, while the EP-SOP with remainder ψ_{ij} has a size that can be upper bounded by $2|\psi_{MIN}|$.

As a final observation, we would like to point out that the couple of variables, say x_i and x_j , with the highest frequency in general does not guarantee that $\xi_{MIN} = \xi_{ij}^{(min)}$ and $\psi_{MIN} = \psi_{ij}^{(min)}$, as the following counterexample shows.

Example 2. Let us consider the minimal SOP $\phi = x_1x_2x_3x_4 + x_1x_2\bar{x}_3\bar{x}_4 + \bar{x}_1x_2$. We want to find the two minimal EP-SOP forms. The couple of variables with the highest frequency is x_1 and x_2 .

The approximation algorithm computes the following form without remainder:

$$\xi_{12}^{(min)} = (x_1 \oplus x_2)x_2 + (x_1 \oplus \bar{x}_2)(x_2x_3x_4 + x_2\bar{x}_3\bar{x}_4)$$

and the following form with remainder:

$$\psi_{12}^{(min)} = (x_1 \oplus x_2)x_2 + (x_1 \oplus \bar{x}_2)(x_2x_3x_4 + x_2\bar{x}_3\bar{x}_4),$$

while the minimal solutions are $\xi_{MIN} = \xi_{34}^{(min)} = (x_3 \oplus \bar{x}_4)x_2 + (x_3 \oplus x_4)(\bar{x}_1x_2)$ and $\psi_{MIN} = \psi_{34}^{(min)} = (x_3 \oplus \bar{x}_4)(x_1x_2) + \bar{x}_1x_2$, respectively.

5 Experimental results

In this section we discuss the computational results obtained by applying the polynomial approximation algorithm presented above to the standard ESPRESSO benchmark suite [20]. We consider four different variants of our algorithm. In fact, we address the minimization of EP-SOP forms both with and without remainder, in order to estimate the practical utility of either form. Moreover, as

most benchmarks have multiple outputs, the definition of the most frequent couple of variables can be referred either to the whole set of outputs (*global frequency*) or to each single output (*local frequency*). In the former case, we will determine a single EP-SOP form, projecting the original minimal SOP form ϕ with respect to the couple of variables appearing in the largest number of products of ϕ . In the latter case, we will find the most frequent couple of variables for each different output and perform independent projections, obtaining separate EP-SOP forms for the outputs which have been projected onto different couples of subspaces. In both cases all the SOP forms are synthesized together with multi-output synthesis. Combining the two approaches related to the use of the remainder and the two approaches related to the global and local frequency, we obtain four different algorithms, respectively denoted as NG (no remainder and global frequency), NL (no remainder and local frequency), RG (remainder and global frequency), RL (remainder and local frequency).

All computational experiments were performed on a Pentium 1.6 GHz processor with 1 GB RAM. We report in the following a significant subset of the experiments.

Table 1 reports a cost-oriented comparison among the original optimal SOP form determined by ESPRESSO EXACT and the EP-SOP forms yielded by the four algorithms: the first column reports the name of the instance, the following five triples of columns report the computational time in seconds, together with the area and the delay of physical implementations for the five expressions. These were evaluated using a technology mapping (mcnc.genlib) provided by the SIS [18] tool.

The computational time for the EP-SOP forms does not include the time required to compute the optimal SOP form (which is shown in the second column), but only the time to factorize it and to heuristically minimize its projections. As the results show, the overhead added by the last two steps is quite limited.

Of course, the physical implementation of the EP-SOP forms also includes one or more EXOR gates, whose cost cannot be neglected, as our results clearly show. First of all, the EXOR part of the network can be expensive, depending on the technology adopted. Second, some functions benefit from the multi-output minimization: common products can be shared, thus reducing the overall area. Comparing the performances of the four algorithms one to another, we can note how this fact particularly affects the performance of the algorithms NL and RL referring to the local definition of frequency, while the algorithm performing better seems to be the RG algorithm.

It should be noticed, however, that the gain obtained by the EP-SOP form is on about the 35% of instances, and can be quite striking: the gain on instance *adr4* exceeds 50% and for many other instances (e.g., *root*, *z4*) it exceeds 40%.

Apart from algorithm NL, which only equals some best result, never hitting one alone, even the less effective of the other three algorithms, that is RL, improves by 45% the cost of instance *f51m*.

Given that the time required to obtain such improvements is rather limited, evaluating the EP-SOP forms as a possible alternative to the optimal SOP form

appears to be an advisable *post-processing strategy*. We have further investigated

Bench.	min SOP			min EP-SOP											
	CPU	area	delay	NG		NL		RG		RL					
	CPU	area	delay	CPU	area	delay	CPU	area	delay	CPU	area	delay			
addm4	0.14	1172	47.9	0.06	1291	52.5	0.06	975	40.4	0.04	1101	48.5	0.07	906	38.3
adr4	0.04	224	19.2	0.03	174	15.2	0.03	155	16.0	0.03	105	11.1	0.04	141	13.5
amd	0.06	1171	46.7	0.03	1082	43.5	0.05	1040	39.1	0.03	1046	42.4	0.06	1022	38.0
b2	0.23	3876	79.8	0.06	4113	81.3	0.06	4180	81.3	0.04	4169	82.6	0.04	4242	82.6
b4	3.45	645	30.5	0.01	802	33.3	0.01	841	33.1	0.01	717	34.4	0.01	779	32.8
br1	0.01	446	32.5	0.02	353	24.5	0.02	381	25.7	0.02	353	24.5	0.02	381	25.7
br2	0.01	352	26.6	0.01	292	25.5	0.01	314	30.0	0.01	292	25.5	0.01	314	30.0
chkn	0.48	717	43.6	0.04	832	42.2	0.06	777	39.2	0.01	758	36.1	0.01	764	46.7
dc2	0.04	253	23.1	0.01	286	22.4	0.01	236	19.7	0.01	263	21.7	0.01	236	19.7
exps	0.50	3932	114.5	0.06	3778	114.8	0.06	3900	104.6	0.08	3760	112.6	0.09	3877	106.4
f51m	0.09	501	31.5	0.04	413	26.2	0.04	339	26.4	0.04	311	20.5	0.04	273	19.1
in0	0.10	1214	48.3	0.03	1056	48.1	0.05	1015	42.5	0.05	1019	48.0	0.06	989	44.9
in1	0.23	3876	79.8	0.06	4113	81.3	0.06	4180	81.3	0.06	4169	82.6	0.06	4242	82.6
in2	0.09	1112	41.4	0.03	1000	36.7	0.01	1041	37.3	0.03	1002	37.3	0.03	1039	37.9
in5	0.14	905	38.5	0.01	976	39.2	0.01	1040	37.2	0.01	923	40.9	0.01	993	39.7
intb	2.96	2170	57.3	0.44	3392	75.5	0.83	2693	63.2	0.34	2466	57.6	0.67	2526	61.6
luc	0.01	806	41.0	0.01	779	52.8	0.01	883	51.8	0.01	758	52.4	0.01	862	50.6
m1	0.01	208	19.6	0.03	304	21.0	0.03	352	21.2	0.03	308	22.8	0.03	356	22.8
m2	0.01	710	37.8	0.01	833	40.9	0.01	893	40.5	0.01	861	42.5	0.01	921	41.9
m3	0.04	839	38.3	0.01	1286	48.4	0.01	1283	52.2	0.01	1172	51.7	0.01	1235	54.4
m181	0.60	166	18.4	0.01	327	22.4	0.03	311	24.9	0.01	240	22.5	0.01	267	19.8
max128	0.09	1292	58.0	0.09	2055	71.6	0.09	2194	77.6	0.07	2098	71.5	0.07	1975	72.4
mlp4	0.31	734	36.4	0.03	983	43.0	0.04	891	40.1	0.03	839	40.5	0.03	857	40.1
mp2d	0.25	362	26.0	0.01	428	25.3	0.01	420	28.9	0.01	333	23.7	0.01	360	25.5
newcond	0.01	114	17.4	0.01	132	18.6	0.01	124	18.6	0.01	119	18.2	0.01	124	18.6
p82	0.01	239	18.4	0.01	239	25.8	0.01	302	23.9	0.01	241	25.0	0.01	309	24.7
radd	0.39	183	15.7	0.01	196	18.9	0.01	181	19.5	0.01	120	15.1	0.01	158	16.8
rckl	0.04	341	49.7	0.01	495	72.3	0.01	519	72.3	0.01	495	72.3	0.01	519	72.3
rd73	0.03	220	25.6	0.03	389	27.6	0.03	308	28.4	0.03	339	26.9	0.03	264	24.1
risc	0.01	228	18.7	0.02	312	29.0	0.02	435	32.7	0.03	310	29.0	0.02	434	32.5
root	0.35	592	35.5	0.02	367	27.7	0.02	380	25.3	0.03	349	26.5	0.03	350	25.7
sqr6	0.06	278	25.5	0.01	397	27.0	0.01	462	26.2	0.01	330	24.9	0.01	405	26.2
t3	0.40	186	21.5	0.02	193	16.2	0.03	213	15.8	0.02	180	19.8	0.02	206	19.7
tms	0.03	587	35.4	0.01	675	35.2	0.01	754	35.5	0.01	675	35.2	0.01	754	35.5
vg2	0.53	341	18.6	0.04	628	25.7	0.06	581	26.0	0.03	468	22.5	0.04	500	21.4
vtx1	0.17	324	21.3	0.01	441	25.5	0.01	497	21.1	0.01	365	23.4	0.01	465	20.7
x6dn	0.18	1054	36.8	0.01	854	34.9	0.01	870	34.9	0.01	817	34.8	0.01	834	34.8
x9dn	0.20	384	23.0	0.04	496	25.4	0.06	560	24.2	0.04	424	24.7	0.03	528	22.6
z4	0.01	171	18.3	0.01	159	18.6	0.01	165	20.6	0.01	99	14.2	0.01	132	17.9

Table 1. Synthesis time, area and delay of EP-SOP and SOP forms (computed in SIS after the technology mapping).

whether the Boolean factorization proposed in the present paper actually differs from similar techniques already known in the literature and applied in synthesis tools. We have applied the multilevel synthesis routines (script.rugged) of SIS to the optimal SOP forms and to the four EP-SOP forms, in order to find out whether they end up with a similar final structure or not. The first remark that can be done is that in some cases (e.g., *b2*, *exps* and *in1*), SIS was unable to process the optimal SOP form (in a limit time of 12 hours). Starting from the EP-SOP forms, this happened only for instance *in1*, and only for the two EP-SOP forms with remainder. Only few times the final results were identical (10%), and half of the times the final result obtained starting from an EP-SOP form

was better than the one obtained from the optimal SOP form, ranging from 30% better to 30% worse.

6 Conclusion

Although deriving an optimal EP-SOP form from an optimal SOP form is an NP^{NP} -hard problem, in this paper we have described a polynomial time approximation algorithm which guarantees a near-optimal solution. We propose this algorithm as a post-processing step after the SOP synthesis, in order to possibly reduce the area of the resulting networks. Our experiments show that in about 35% of the considered benchmarks the area obtained is smaller, sometimes even by 40-50%.

It could be an interesting development to study different kinds of projection, such as dividing the Boolean space into subspaces whose characteristic functions are represented by EXORs with more than two literals. Given the similar nature of the problem, it could also be interesting to study the relationship between Linear Transformed BDDs [10] and EP-SOP forms.

References

1. A. Bernasconi, V. Ciriani, F. Luccio, and L. Pagli. Three-Level Logic Minimization Based on Function Regularities. *IEEE Transactions on TCAD*, 22(8):1005–1016, 2003.
2. R. Brayton, G. Hachtel, C. McMullen, and A. Sangiovanni-Vincentelli. *Logic Minimization Algorithms for VLSI Synthesis*. Kluwer Ac. Pub., 1984.
3. G. Caruso. Near Optimal Factorization of Boolean Functions. *IEEE Transactions on CAD*, 10(8):1072–1078, 1991.
4. O. Coudert. Two-Level Logic Minimization: an Overview. *INTEGRATION*, 17:97–140, 1994.
5. D. Debnath and T. Sasao. A Heuristic Algorithm to Design AND-OR-EXOR Three-Level Networks. In *Asia and South Pacific Design Automation Conference*, pages 69–74, 1998.
6. D. Debnath and Z. Vranesic. A Fast Algorithm for OR-AND-OR Synthesis. *IEEE Transactions on CAD*, 22(9):1166–1176, 2003.
7. E. Dubrova, D. Miller, and J. Muzio. AOXMIN-MV: A Heuristic Algorithm for AND-OR-XOR Minimization. In *4th Int. Workshop on the Applications of the Reed Muller Expansion in circuit Design*, pages 37–54, 1999.
8. M. Fujita, Y. Matsunaga, and M. Ciesielski. Multi-Level Logic Optimization. In S. Hassoun and T. Sasao, editors, *Logic Synthesis and Verification*, pages 29–63. Kluwer Academic Publishers, 2002.
9. M. Garey and D. Johnson. *Computer and Intractability: A Guide to the Theory of NP-completeness*. W.H. Freeman and Company, 1979.
10. W. Günther and R. Drechsler. On the Computational Power of Linearly-Transformed BDDs. *Information Processing Letters*, 75(3):119–125, 2000.
11. W. Günther and R. Drechsler. Efficient Minimization and Manipulation of Linearly-Transformed Binary Decision Diagrams. *IEEE Transaction on Computers*, 52(9):1196–1209, 2003.

12. F. Luccio and L. Pagli. On a New Boolean Function with Applications. *IEEE Transactions on Computers*, 48(3):296–310, 1999.
13. P. McGeer, J. Sanghavi, R. Brayton, and A. Sangiovanni-Vincentelli. ESPRESSO-SIGNATURE: A New Exact Minimizer for Logic Functions. *IEEE Transactions on VLSI*, 1(4):432–440, 1993.
14. M. Perkowski. A New Representation of Strongly Unspecified Switching Functions and its Application to Multi-Level AND/OR/EXOR Synthesis. In *IFIP WG 10.5 Workshop on Applications of the Reed-Muller Expansion*, pages 143–151, 1995.
15. T. Sasao. On the Complexity of Three-Level Logic Circuits. In *Int. Workshop on Logic Synthesis*, 1989.
16. T. Sasao. A Design Method for AND-OR-EXOR Three Level Networks. In *Int. Workshop on Logic Synthesis*, pages 8:11–8:20, 1995.
17. T. Sasao. *Switching Theory for Logic Synthesis*. Kluwer Academic Publishers, 1999.
18. E. Sentovich, K. Singh, L. Lavagno, C. Moon, A. S. R. Murgai, H. Savoj, P. Stephan, R. Brayton, and A. Sangiovanni-Vincentelli. SIS: A system for sequential circuit synthesis. Technical report, 1992.
19. C. Umans, T. Villa, and A. Sangiovanni-Vincentelli. Complexity of Two-Level Logic Minimization. *IEEE Transactions on Computer-Aided Design*, to appear.
20. S. Yang. Logic synthesis and optimization benchmarks user guide version 3.0. User guide, Microelectronic Center, 1991.