アルゴリズム 39－8
（1994. 5. 13）

自己双対正論理関数の分解について

Jan C. Bioch　茨木俊秀
エラスムス大学　京都大学工学部

概要

分散システムの相互排除に用いられるコテリとは、集合族 C で、どの $S, S' \in C$ も比較不能かつ $S \cap S' \neq \emptyset$ を満たすものをいう。集合族 C に対し、正論理関数 f_C を次のように定義する。ベクトル x が C のある部分集合 S の特性ベクトル以上であるとき $f_C(x) = 1$, そうでなければ $f_C(x) = 0$. このとき、C : コテリ $\Longleftrightarrow$ f_C : 劣双対、C : 優コテリ $\Longleftrightarrow$ f_C : 自己双対、という関係が知られている。

本論分では、大きな優コテリを小さな優コテリからどのように構成するかを調べるため、自己双対正関数をより小さな自己双対正関数へ分解する方法を調べる。この目的には、劣双対関数 f をいくつかの自己双対正関数の積 $f = f_1 f_2 \cdots f_k$ に分解することが基本ステップになる。そこで、この分解を可能にする一般的な条件とともに、2個の自己双対正関数への分解 $f = f_1 f_2$ の必要十分条件を与える。さらに、正準分解の概念を導入し、極小正準分解を求めるアルゴリズム、および（極小性は保証できないが）簡単で多項式時間で動作する一つの分解アルゴリズムを提案する。

Decompositions of Positive Self-Dual Boolean Functions

Jan C. Bioch
Erasmus University Rotterdam
bioch@cs.few.eur.nl

Toshihide Ibaraki
Kyoto University
ibaraki@kuamp.kyoto-u.ac.jp

Abstract

A coterie, which is used to realize mutual exclusion in a distributed system, is an incomparable family C of subsets such that any pair of subsets in C has at least one element in common. Associate with a family of subsets C a positive Boolean function f_C such that $f_C(x) = 1$ if the Boolean vector x is equal to or greater than the characteristic vector of some subset in C, and 0 otherwise. It is known that C is a coterie if and only if f_C is dual-minor, and is a non-dominated (ND) coterie if and only if f_C is self-dual. We study in this paper the decomposition of a positive self-dual function into smaller positive self-dual functions, as it explains how to represent and how to construct the corresponding ND coterie. A key step is how to decompose a positive dual-minor function f into a conjunction of positive self-dual functions $f_1, f_2, \ldots, f_k$. In addition to the general condition for this decomposition, we clarify the condition for the decomposition into two functions f_1 and f_2, and introduce the concept of canonical decomposition. Then we present an algorithm that determines a minimal canonical decomposition, and a simple algorithm that usually gives a decomposition close to minimal.

1 Introduction

1.1 Motivation and Results

A coterie C, originally defined in [9, 6], is a family of subsets of an underlying set $\{1,2,\ldots,n\}$, such that each pair of subsets in C has at least one element in common, and no subset in C contains any other subset in C. It is used as a mechanism to realize mutual exclusion [9, 6] in a distributed system; a task can enter the critical section only if it can get agreements from all the members in a subset $S \in C$, where members $1,2,\ldots,n$ represent the sites in the distributed system. By the intersecting property of a coterie C, it is guaranteed that at most one task can enter the critical section at a time if each site is allowed to issue at most one agreement (this property is called mutual exclusion).

Associate with a family of subsets C a positive Boolean function f_C such that $f_C(x) = 1$ if and only if the Boolean vector $x \in \{0,1\}^n$ is equal to or greater than the characteristic vector of some subset in C. It is known [8] that C is a coterie if and only if f_C is dual-minor, and is a non-dominated (ND) coterie if and only if f_C is self-dual (see Section 1.2 for the definitions of these terms). Self-dual functions play an important role also in threshold logic [10], regular Boolean functions [12, 4], circuit theory and other areas of Boolean functions.

The class of positive self-dual functions is closed under compositions. Therefore one of the fundamental questions in this field is how to decompose a given positive self-dual function into smaller positive self-dual functions, as it explains how to represent and how to construct the corresponding coterie by using simpler elements. It was shown in [8] that any positive self-dual function can be decomposed into a set of basic majority functions (the basic majority function is the only self-dual function containing three variables). However, the argument in [8] was only to show the existence of such a decomposition, and the resulting decomposition may be far from the minimum. Other types of decompositions are also found in [11, 8].

In this paper, which is an abbreviated version of [1], we systematically study the decomposition of a positive self-dual function. A key step is how to decompose a given positive dual-minor function f into a conjunction of positive self-dual functions:

$$f = f_1 f_2 \cdots f_k.$$

We first derive in Section 2.2 the general condition for this decomposition, and then give in Section 2.3 a necessary and sufficient condition for the decomposition into two functions f_1 and f_2. The concept of canonical decomposition is then introduced in Section 3.1, and an algorithm to find a minimal canonical decomposition is given in Section 3.2.

The complexity issues related to these problems are mentioned in the last section. All the above algorithms are of polynomial time in the length of input and output if dualization of a positive Boolean function can be done in time polynomial in the length of input and output. However, the latter problem is still not solved, and it is related to other well-known open problems such as deciding the NDness of a coterie [6] and 2-coloring a simple hypergraph [5]. More detailed discussion can be found in the accompanying paper [2].

1.2 Definitions and Basic Properties

Coteries

Let C be a non-empty family of subsets of the non-empty finite set $\{1,2,\ldots,n\}$. Then C is called a *coterie* if the following conditions are satisfied for all $S, S' \in C$:

(i) $S \not\subseteq S'$ (minimality)
(ii) $S \cap S' \neq \emptyset$ (intersection property).

A coterie C *dominates* a coterie D if $C \neq D$ and for each $S \in D$ there exists a $S' \in C$ such that $S' \subseteq S$. A coterie C is called *non-dominated* (ND) if no coterie dominates C. ND coteries are important in practice, since those are the coteries with maximal efficiency when implemented to realize mutual exclusion.

Positive Boolean functions

A *Boolean function*, or in short a *function* is a mapping $f : \{0,1\}^n \mapsto \{0,1\}$. Let $x \in \{0,1\}^n$ be a *Boolean vector*, or in short a *vector*. If $f(x) = 1$

(resp. 0), then x is called a *true* (resp. *false*) vector of f. The set of all true vectors is denoted by $T(f)$. For a function f, the minimal elements in $T(f)$ are called the minimal true vectors of f, and the set of all minimal true vectors is denoted by $\mathrm{minT}(f)$. A function f is called *positive* or *monotone* if $x \leq y$ always implies $f(x) \leq f(y)$. It is known that a positive function f is uniquely determined by $\mathrm{minT}(f)$, and that f has the unique minimal disjunctive form (MDF) consisting of all the prime implicants of f, in which all the literals of each prime-implicant are uncomplemented. There is a one-to-one correspondence between $\mathrm{minT}(f)$ and the set of all prime implicants of f, such that a vector v corresponds to the monomial m_v defined by $m_v = x_{i_1}x_{i_2}\cdots x_{i_k}$ if $v_{i_j} = 1, j = 1, 2, \ldots, k$ and $v_i = 0$ otherwise. For example, vector (101) corresponds to monomial x_1x_3. The MDF of a positive function such as $f = x_1x_2 + x_2x_3 + x_3x_1$ is usually represented by a simplified form $f = 12 + 23 + 31$, by using only the subscripts of literals, where the operator $+$ is used as an alias for the Boolean-or operator $\vee$. The functions f with $\mathrm{T}(f) = \emptyset$ and $\mathrm{F}(f) = \emptyset$ are respectively denoted $\bot$ and $\top$ in this paper.

Dual-comparable functions

The *dual* of f, denoted f^d, is defined by

$$f^d(x) = \bar{f}(\bar{x}),$$

where $\bar{f}$ and $\bar{x}$ denote the complement of f and x, respectively. As is well-known, the MDF expression defining f^d is obtained from that of f by exchanging $+$ (or) and $\cdot$ (and), as well as the constants 0 and 1. Denote $f \leq g$ if these functions satisfy $f(x) \leq g(x)$ for all $x \in \{0,1\}^n$. It is easy to see that $(f+g)^d = f^d g^d$, $(fg)^d = f^d + g^d$, $f \leq g$ implies $f^d \geq g^d$, and so on. A function is called *dual-minor* if $f \leq f^d$ and *dual-major* if $f \geq f^d$. f is called *dual-comparable* if $f \leq f^d$ or $f \geq f^d$ holds, and *self-dual* if $f^d = f$.

For example, $f = 123$ is dual-minor since $f^d = 1 + 2 + 3$ satisfies $f \leq f^d$. Similarly, the dual of $f = 12 + 23 + 31$ is

$$f^d = (1+2)(2+3)(3+1) = 12 + 23 + 31.$$

This function f is self-dual, and is called the *basic majority function*. This is known to be the only positive self-dual function containing three variables. The basic majority function of three variables x, y, z is sometimes denoted by $[x, y, z]$ in the subsequent discussion. There is no positive self-dual function of two variables. However, each function $f = x_i$ is a positive self-dual function of one variable.

Coteries and Boolean functions

Let C be a family of subsets of $\{1, 2, \ldots, n\}$ satisfying the minimality condition (i). With C we associate a positive function f_C defined by $f_C(x) = 1$ if and only if there exists a subset $S \in C$ such that $cv(S) \leq x$, where the characteristic vector $y = cv(S)$ of S is defined by $y_i = 1$ if $i \in S$ and 0 if $i \notin S$. The correspondence $C \leftrightarrow \mathrm{minT}(f_C)$, where $S \in C$ corresponds to $y = cv(S) \in \mathrm{minT}(f_C)$, is one-to-one. Furthermore it is known [8] that C is a coterie if and only if f_C is dual-minor, and that C is an ND-coterie if and only if f_C is self-dual.

Contra-dual functions

The *contra-dual* f^* of f is defined by

$$f^*(x) = f(\bar{x}).$$

For example, the contra-dual of $f = 12 + 23 + 31$ is $f^* = \bar{1}\bar{2} + \bar{2}\bar{3} + \bar{3}\bar{1}$, where $\bar{i}$ stands for literal $\bar{x}_i$. It is known [7] that the four operations: identity, d, $*$ and complementation $^-$ form Klein's four-group. This means that these operations commute, are idempotent and satisfy the relation $\alpha\beta = \gamma$, where α, β, γ are three different operations: $(\bar{f})^d = \overline{(f^d)} = f^*$, $(\bar{f})^* = \overline{(f^*)} = f^d$, $(f^d)^* = (f^*)^d = \bar{f}$ and so on.

The dual-minority can be checked in polynomial time by the following lemma, though there is no counterpart known for the dual-majority.

Lemma 1 Let f be a positive function. Then f is dual-minor if and only if every pair of prime implicants m_i and m_j in its MDF has at least one literal in common. □

2 Decomposition of Positive Self-Dual Functions

2.1 Shannon's Decomposition of Self-Dual Functions

Shannon's decomposition expands a given function f on a variable x_i as follows:

$$f = f(x_i=0)\bar{x}_i + f(x_i=1)x_i. \quad (1)$$

If f is positive and self-dual, this becomes

$$f = g + g^d x_i, \quad (2)$$

where $g = f(x_i = 0)$ is positive and dual-minor, and $g^d = (f(x_i=0))^d = f(x_i=1)$ is positive and dual-major [8]. It is also known that any positive dual-minor function g is the conjunction of positive self-dual functions $f_1, f_2, \ldots, f_k$:

$$g = f_1 f_2 \cdots f_k. \quad (3)$$

In this case, g^d can be given by

$$g^d = f_1 + f_2 + \cdots + f_k. \quad (4)$$

Let $[x, y, z]$ denote the basic majority function of three variables x, y, z. Then decomposition (2) can be expressed by

$$\begin{aligned} f &= f_1 f_2 \cdots f_k + (f_1 + f_2 + \cdots + f_k)x \\ &= [x, f_1, [x, f_2, [\cdots[x, f_{k-1}, f_k]\cdots]]], \end{aligned} \quad (5)$$

as easily proved by induction starting from the case of $k = 2$:

$$f_1 f_2 + (f_1 + f_2)x = [x, f_1, f_2].$$

Since the functions $f_1, f_2, \ldots, f_k$ do not contain the variable x, and are positive and self-dual, these decompositions can be repeatedly applied to the generated functions until only functions of one variable remain. If we interpret each such decomposition as (5), the entire process yields a tree shaped decomposition of the original positive self-dual function f into basic majority functions. This is called the B-*decomposition* of f in [8], where B stands for "binary tree".

A key step in the B-decomposition is decomposition (3) of a positive dual-minor function into positive self-dual functions. Call the number k in (3) the *size* of the decomposition. If the size of each decomposition is small, the resulting B-decomposition will become small. In the following, therefore, we carry out a more careful study so that decompositions (3) with small sizes can be realized in a systematic manner.

2.2 Decomposition of a Dual-Minor Function

For functions f and g, define the *extension of f with respect to g* by

$$f \uparrow g = f + f^d g. \quad (6)$$

If g is self-dual and f is dual-minor then $f \uparrow g$ is self-dual, since

$$(f \uparrow g)^d = f^d(f + g) = f + f^d g = f \uparrow g.$$

Expression (6) may be considered as an extension of Shannon's decomposition (2) in the sense that the positive self-dual function x_i in (2) is now replaced by a general positive self-dual function g. It is also easy to see that if g is self-dual, then the function $f \uparrow g$ is always dual-major, and that if f is dual-major, then $f \uparrow g = f$. Obviously $f \uparrow g$ is positive if so are both f and g.

Theorem 2 Let f be a positive dual-minor function. Then f can be decomposed into k positive self-dual functions $(f \uparrow g_i), i = 1, 2, \ldots, k$:

$$f = (f \uparrow g_1)(f \uparrow g_2) \cdots (f \uparrow g_k), \quad (7)$$

where $g_1, g_2, \ldots, g_k$ are given positive self-dual functions, if and only if

$$g_1 g_2 \cdots g_k \leq f + f^*. \quad (8)$$

□

Example 1

$$\begin{aligned} f &= 123 + 124 + 134 + 234 \\ f^d &= 12 + 13 + 14 + 23 + 24 + 34 \\ f^* &= \bar{1}\bar{2}\bar{3} + \bar{1}\bar{2}\bar{4} + \bar{1}\bar{3}\bar{4} + \bar{2}\bar{3}\bar{4}, \end{aligned}$$

where this f is positive and dual-minor, as easily checked by Lemma 1. A set of positive self-dual functions $g_1, g_2, \ldots, g_k$ satisfying condition (8) is for example given by

$$g_1 = 12 + 23 + 31$$
$$g_2 = 4.$$

In fact,

$$f_1 = f \uparrow g_1 = f + f^d g_1 = 12 + 23 + 31$$
$$f_2 = f \uparrow g_2 = 14 + 24 + 34 + 123$$

are both positive and self-dual, and it is immediate to see that $f = f_1 f_2$. □

2.3 Decomposition into Two Positive Self-Dual Functions

Theorem 3 A positive dual-minor function f has a decomposition $f = f_1 f_2$ into two positive self-dual functions f_1 and f_2 if and only if set

$$M = \min\mathrm{T}(f^d) \setminus \min\mathrm{T}(f). \qquad (9)$$

has a partition into M_1 and M_2 such that neither of them contains a pair of vectors x and y such that $x_i y_i = 0$ for all i. □

The existence of the above partition M_1 and M_2 can be found by constructing an undirected graph $G_f = (V, E)$ such that

$$V = M$$
$$E = \{(x,y) \mid x, y \in M,\ x_i y_i = 0 \text{ for all } i\}.$$

Corollary 4 A positive dual-minor function f has a decomposition M_1 and M_2 of Theorem 3 if and only if G_f is bipartite. □

Example 2 Consider the following f:

$$f = 123 + 125 + 134 + 145 + 2345$$
$$f^d = 12 + 13 + 14 + 15 + 24 + 35.$$

Its G_f is shown in Fig. 1, in which each vector $x \in M$ is represented by the corresponding monomial. This G_f is bipartite, and M has a partition into the following two independent sets:

$$M_1 = \{12, 14, 24\}, \quad M_2 = \{13, 15, 35\}.$$

Therefore, f has a decomposition $f = f_1 f_2$ into two positive self-dual functions

$$f_1 = 12 + 14 + 24$$
$$f_2 = 13 + 15 + 35. \ \square$$

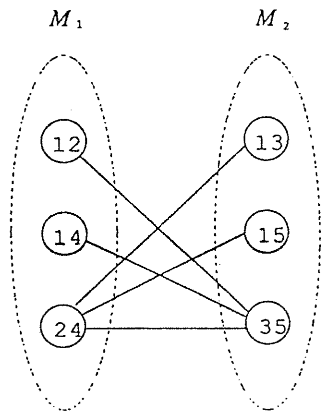

Fig. 1 Graph G_f of Example 2.

3 Canonical Decomposition of Positive Self-Dual Functions

3.1 Canonical Decomposition

Recall that every variable x_j itself is a positive self-dual function. If $g_i = x_{j_i}$ are used for all $i = 1, 2, \ldots, k$, the decomposition (7) of f is called a *canonical decomposition*. A canonical decomposition is called *minimal* if none of its components $(f \uparrow x_i)$ can be deleted.

Let f be a positive function. Any prime implicant $x_{j_1} x_{j_2} \cdots x_{j_k}$ of f satisfies

$$x_{j_1} x_{j_2} \cdots x_{j_k} \leq f + f^*,$$

and this leads to the following theorem.

Theorem 5 Let f be positive and dual-minor, and let $m = x_{j_1} x_{j_2} \cdots x_{j_k}$ be one of its prime implicants. Then there is the corresponding canonical decomposition:

$$f = f_{j_1} f_{j_2} \cdots f_{j_k}$$
$$f_{j_i} = f \uparrow x_{j_i}, \quad i = 1, 2, \ldots, k. \quad \square$$

Example 3 Consider a positive dual-minor function and its dual:

$$f = 123 + 234 + 235 + 145$$
$$f^d = 12 + 13 + 24 + 25 + 34 + 35 + 145$$

Then, by Theorem 5, we have canonical decompositions

$$f = f_1 f_2 f_3 = f_2 f_3 f_4 = f_2 f_3 f_5 = f_1 f_4 f_5,$$

where $f_j = f \uparrow x_j$. However some of these are not minimal, since there is another canonical decomposition $f = f_2 f_3$, as easily checked (see also Example 4). □

3.2 Minimal Canonical Decomposition

In order to derive a condition for minimal canonical decompositions, we examine the condition (8) of Theorem 2 more carefully. Since the function $f + f^*$ in (8) is not positive, we define the *positive core* of f by

$$\check{f} = \vee\{h \mid h \leq f + f^*,\ h : \text{positive}\}. \quad (10)$$

This $\check{f}$ is the unique maximal positive function contained in $f + f^*$. The dual of $\check{f}$ is denoted by $\hat{f}$, and is called the *positive closure* of f.

Theorem 6 Let f be a positive dual-minor function. Then $f = f_{j_1} f_{j_2} \cdots f_{j_k}$ is a minimal decomposition of f if and only if $x_{j_1} x_{j_2} \cdots x_{j_k}$ is a prime implicant of $\check{f}$. □

To make use of this theorem, we now turn our attention to how to compute $\check{f}$.

Theorem 7 Let f be a positive dual-minor function. Then its positive closure $\hat{f}$ satisfies

$$\mathrm{minT}(\hat{f}) = \mathrm{minT}(f^d) \setminus \mathrm{minT}(f). \quad \square$$

Noting that $\check{f}$ is the dual of $\hat{f}$, we now have the following algorithm.

Algorithm POSITIVE-CORE

Input: A positive dual-minor function f.
Output: All prime implicants of $\check{f}$.

1. Dualize f to compute all prime implicants of f^d.
2. Remove all prime implicants of f^d that are also prime implicants of f (by Theorem 7, the resulting set gives all prime implicants of $\hat{f}$).
3. Dualize $\hat{f}$. This yields all prime implicants of $\check{f}$. □

Example 4 We apply this algorithm to the positive dual-minor function f of Example 3:

$$f = 123 + 234 + 235 + 145$$
$$f^d = 12 + 13 + 24 + 25 + 34 + 35 + 145$$
$$\hat{f} = 12 + 13 + 24 + 25 + 34 + 35$$
$$\check{f} = 23 + 145.$$

Therefore, by Theorem 6, f has the following two minimal canonical decompositions and no others.

$$f = f_2 f_3 \text{ and } f = f_1 f_4 f_5. \quad \square$$

Before concluding this subsection, we apply Algorithm POSITIVE-CORE to the function f of Example 2. Then $\hat{f} = f^d$, and hence $\check{f} = f$. Therefore, any minimal canonical decomposition has at least three components. However, as we have seen in Example 2, this f has a decomposition into two components, showing that canonical decompositions do not generally contain a decomposition into the smallest number of components. The problem of finding a decomposition with the smallest number of components appears to be very difficult, except for the case of two components, which was discussed in Section 2.3.

3.3 A Simple B-Decomposition Algorithm

Given a positive self-dual function f, one of its B-decomposition can be obtained by recursively applying canonical decompositions. The entire algorithm is described by two procedures SD(f) and DM(f). SD(f) outputs a positive dual-minor function g obtained by Shannon's decomposition (2) applied to f, where f is assumed without loss of generality to contain at least three variables. Given a positive dual-minor function g, DM(g) then computes a canonical decomposition $g = f_{j_1} f_{j_2} \cdots f_{j_k}$, obtains positive dual-minor functions $g_{j_i} = f_{j_i}(x_{j_i}=0) = g(x_{j_i}=0), i = 1, 2, \ldots, k$, and then recursively calls DM(g_{j_i}) if $g_{j_i} \neq \perp$.

Algorithm SD(f)

1. Choose a variable x of f.
2. Apply Shannon's decomposition $f = g + g^d x$, where $g = f(x=0)$.
3. Call DM(g). (Note that g^d in step 2 is not explicitly required, since only g is used in this step.) □

Algorithm DM(g)

1. Find a monomial $m = x_{j_1} x_{j_2} \cdots x_{j_k}$ such that $m \leq \breve{g}$, and compute

$$g = f_{j_1} f_{j_2} \cdots f_{j_k}$$

$$f_{j_i} = g \uparrow x_{j_i}, \quad i = 1, 2, , \ldots, k.$$

2. For each of the positive dual-minor functions

$$g_{j_i} = f_{j_i}(x_{j_i}=0) = g(x_{j_i}=0), \quad i = 1, 2, , \ldots, k,$$

call DM(g_{j_i}) if $g_{j_i} \neq \perp$. □

Example 5 We compute a B-decomposition of

$$\begin{aligned} f &= 123 + 234 + 235 + 145 + 126 + 136 \\ & \quad +246 + 256 + 346 + 356, \end{aligned}$$

which is positive and self-dual. First execute SD(f). In step 1 of SD(f), choose variable $x = x_6$, and we have

$$g = f(x_6 = 0) = 123 + 234 + 235 + 145.$$

Then we execute DM(g). In step 1 of DM(g), choose monomial $m = 23$ since it satisfies $23 \leq \breve{g}$, as discussed in Example 4. This gives the canonical decomposition

$$g = f_2 f_3$$

$$f_i = g \uparrow x_i, \quad i = 2, 3$$

and

$$\begin{aligned} g_2 &= g(x_2=0) = 145 \\ g_3 &= g(x_3=0) = 145. \end{aligned}$$

In step 2 of DM(g), we first call DM(g_2). In step 1 of DM(g_2), choose $m_2 = 145$, i.e., $g_2 = f_{21} f_{24} f_{25}$ with $f_{2i} = g_2 \uparrow x_i$. Then

$$g_2(x_1=0) = g_2(x_4=0) = g_2(x_5=0) = \perp,$$

and no new call to DM is necessary. Since $g_3 = g_2$, the call to DM(g_3) gives the same result, and the entire computation halts. The resulting decomposition into basic majority functions is

$$[[1, 2, [4, 2, 5]], 6, [1, 3, [4, 3, 5]]]. \; \square$$

In the above description, the selection rules in step 1 of SD and DM are not specified. We may employ the following heuristic rules.

(i) In step 1 of SD(f), choose a variable x that is not contained in all the shortest prime implicants of f. (As a result of this rule, $g = f(x=0)$ contains a shortest prime implicant of f, and it may then be chosen in DM(g) to decompose.)

(ii) In step 1 of DM(g), choose one of the shortest prime implicants m of g.

The rule (ii) is attractive for its simplicity, since the computation of a minimal canonical decomposition, as described in Section 3.2, requires the dualization operation twice. On the contrary, computation with rule (ii) uses no dualization operation but repeats the following two operations.

1. Find a shortest prime implicant $m = x_{j_1} x_{j_2} \cdots x_{j_k}$ of a given function g (instead of $\breve{g}$).
2. Compute $g(x_{j_i} = 0), i = 1, 2, , \ldots, k$.

Therefore, algorithm SD with rules (i) and (ii) runs in polynomial time, and can be used as an efficient heuristic algorithm for the decomposition.

4 Discussion

We have not discussed so far the complexity issues of the following problems introduced in this paper.

1. To decide if a positive function f is self-dual (i.e., if a coterie is non-dominated),
2. To compute the extension $f \uparrow g$ of a positive dual-minor function f with respect to a positive self-dual function g,

3. To construct graph G_f of a positive dual-minor function f, defined in Section 2.3 (i.e., to decide if there is a decomposition into two positive self-dual functions),

4. To compute the positive core $\check{f}$ of a given positive dual-minor function (i.e., Algorithm POSITIVE-CORE) in order to obtain a minimal canonical decomposition.

It is obvious that these problems can be solved in time polynomial in the length of input and output, if dualization of a positive function is possible in polynomial time with respect to the output length $|f^d|$ as well as the input length $|f|$, where $|\cdot|$ denotes the length of its MDF form. Unfortunately it is not known yet whether this dualization can be done in polynomial time or not. It is known however that the dualization of a general Boolean function is NP-hard, and that some special classes of positive functions have polynomial time dualization algorithms (e.g., [4, 12]). Problem 1 above is also a well-known open problem, first stated in [6]. The reader may find many related topics on these problems in such references as [2, 3, 4, 5, 6, 8, 12].

It is known, however, that deciding if a positive function f is dual-minor can be done in polynomial time by using Lemma 1, and that deciding if a positive function f is dual-major is coNP-complete (e.g., [2]; equivalent results can also be found in [6, 5] and others). Also, for general Boolean functions, it is proved in [2] that the problem of deciding whether a function is self-dual is coNP-complete, and that the computation of $\check{f}$ is NP-hard.

Acknowledgments

The authors appreciate the discussion with Hiroshi Nagamochi and Kazuhisa Makino of Kyoto University. This work was partially supported by a Grant in Aid by the Ministry of Education, Science and Culture of Japan.

References

[1] J.C. Bioch and T. Ibaraki, Decompositions of positive self-dual Boolean functions, Discrete Mathematics, to appear.

[2] J.C. Bioch and T. Ibaraki, Complexity of dualization and identification of positive Boolean functions, *RUTCOR Research Report*, RRR 25-93, Rutgers University, 1993.

[3] E. Boros, P.L. Hammer, T. Ibaraki and K. Kawakami, Identifying 2-monotonic positive Boolean functions in polynomial time, *ISA '91 Algorithms*, edited by W. L. Hsu and R.C.T. Lee, Springer Lecture Notes in Computer Science 557 (1991) 104-115.

[4] Y. Crama, Dualization of regular Boolean functions, *Discrete Applied Mathematics*, 16 (1987) 79-85.

[5] T. Eiter and G. Gottlob, Identifying the minimal transversals of a hypergraph and related problems, *Technical Report CD-TR 91/16*, Christial Doppler Labor für Expertensysteme, Technische Universität Wien, January 1991.

[6] H. Garcia-Molina and D. Barbara, How to assign votes in a distributed system, *J. ACM*, 32 (1985) 841-860.

[7] P.R. Halmos, *Lectures on Boolean Algebras*, Princeton, NJ, Van Nostrand, 1963.

[8] T. Ibaraki and T. Kameda, Theory of coteries: Mutual exclusion in distributed systems, *IEEE Trans. on Parallel and Distributed Systems*, 4 (1993) 779-794.

[9] L. Lamport, Time, clocks, and the ordering of events in a distributed system, *Comm. of the ACM*, 21 (1978) 558-565.

[10] S. Muroga, *Threshold Logic and Its Applications*, Wiley-Interscience, 1971.

[11] M.L. Neilsen and M. Mizuno, Coterie join algorithm, *IEEE Trans. on Parallel and Distributed Systems*, 3 (1992) 582-590.

[12] U.N. Peled and B. Simeone, Polynomial-time algorithm for regular set-covering and threshold synthesis, *Discrete Applied Mathematics*, 12 (1985) 57-69.