Enumerating and categorizing positive Boolean functions separable by a k-additive capacity

Eda Ersek Uyanık[b], Olivier Sobrie[a,b], Vincent Mousseau[a], Marc Pirlot[b,*]

[a] *CentraleSupélec, Grande Voie des Vignes, 92295 Châtenay Malabry, France*
[b] *Université de Mons, Faculté Polytechnique, 9, rue de Houdain, 7000 Mons, Belgium*

Abstract

Motivated by the elicitation or the learning of certain types of models for classifying objects in ordered categories based on several criteria, we categorize the positive Boolean functions up to 6 variables. We list all inequivalent positive Boolean function and we determine the smallest degree k of the k-additive capacity that can be used for separating their true points from their false points. 1-additive Boolean functions are the well-studied threshold functions. Each function is described by its set of minimal true points. The latter correspond to the minimal winning coalitions of simple games. They also correspond to the minimal sufficient coalitions in the multiple criteria classification models we are interested in, namely, the MR-Sort and the noncompensatory sorting model.

Keywords: multiple criteria sorting methods, positive Boolean functions, ordered classification, winning coalitions, k-additive capacity

---

1. Introduction

Recently proposed multiple criteria classification methods (such as MR-Sort and the NCS model) assign objects to predefined ordered categories by using rules that can be expressed as positive (i.e., monotone non-decreasing) Boolean functions. In these classification methods, one considers objects that

*Corresponding author

Email addresses: eda.uyanik@gmail.com (Eda Ersek Uyanık), olivier.sobrie@gmail.com (Olivier Sobrie), vincent.mousseau@centralesupelec.fr (Vincent Mousseau), marc.pirlot@umons.ac.be (Marc Pirlot)

are described by evaluations w.r.t. several criteria. Categories are characterized by their *lower limit profile*. These profiles specify minimal evaluations on each criterion for an object being assigned to a category above the profile. Actually, an object need *not* be at least as good as the profile's value on *all* criteria. In the *majority rule sorting* model (MR-Sort), a weight is assigned to each criterion and a threshold is fixed. The MR-Sort rule [1, 2] assigns an object to a category above the profile if it is at least as good as the profile on a set of criteria and the sum of the weights of these criteria passes the threshold value. Weights and threshold are used to describe which coalitions (i.e., subsets) of criteria are sufficiently important to justify that the object is assigned to a category above the profile. The *noncompensatory sorting* (NCS) model [3, 4, 5] generalizes this type of rule to the case in which sufficient coalitions of criteria cannot necessarily be described by weights and a threshold. In this model the family of sufficient coalitions can be any *upset* of the set of all subsets of criteria. In other words, the only property of the set of *sufficient coalitions* is that any set which contains a sufficient coalition is itself a sufficient coalition[1].

To precisely show the relationship with monotone Boolean functions, we consider the following example involving two categories. A student has to take 4 exams to be admitted in a school. To be successful, she has to take a mark of at least twelve (out of twenty) in each of these exams, with at most one exception. In this case, the lower limit profile b of the category "succeed" is the vector $b = (12, 12, 12, 12)$ and the sufficient coalitions of criteria are all subsets of at least 3 subjects for which the student's mark is at least 12. Denote the student's marks by $a = (a_1, a_2, a_3, a_4)$. The sufficient "coalitions of successful subjects" can be represented by associating a weight to each course, e.g., each exam receives a weight equal to $1/4$, and choosing an appropriate threshold, here $3/4$. The assignment rule then reads: a succeeds iff $|\{i : a_i \geq 12\}| \times 1/4 \geq 3/4$, which is a typical form for a MR-Sort rule. Once the lower limit profile has been set, any student a can be represented

[1]Both the MR-Sort and the NCS models are particular cases of the Electre Tri model, a method for sorting alternatives in ordered categories based on an *outranking* relation (see [6], pp. 389-401 or [7], pp. 381-385). The general principle of outranking is that an object is preferred to another, or to a profile, if it is at least as good as the latter on a sufficient coalition of criteria without being unacceptably worse on any criterion. The latter part is a non-veto condition. In this work, we consider that there are no "unacceptably worse values" and therefore, only the first part of the condition matters.

by a Boolean vector $x = (x_1, \ldots, x_4)$, with $x_i = 1$ iff $a_i \geq b_i = 12$ for all i.

Not all assignment rules based on sufficient coalitions can be represented by additive weights and a threshold. For instance, assume that the exams subjects are French language (1), English language (2), Mathematics (3) and Physics (4). To be successful, a student has to take at least 12 points in one of the first two and in one of the last two. If the weights of the four subjects are respectively denoted w_1, w_2, w_3, w_4 and the threshold is λ and if we aim to represent the rule using these weights and threshold, we see that these parameters have to fulfill the following inequalities:

$$\begin{cases} w_1 + w_3 & \geq & \lambda \\ w_1 + w_4 & \geq & \lambda \\ w_2 + w_3 & \geq & \lambda \\ w_2 + w_4 & \geq & \lambda \\ w_1 + w_2 & < & \lambda \\ w_3 + w_4 & < & \lambda \end{cases}$$

These conditions are contradictory. Indeed, summing up the first four inequalities, we get that $\lambda \leq 1/2 \sum_{i=1}^{4} w_i$, while summing up the last two yields $\lambda > 1/2 \sum_{i=1}^{4} w_i$.

In both these classification examples, we may represent any student a by a Boolean vector $x = (x_1, \ldots, x_4)$, where $x_i = 1$ (resp. 0) iff the student's mark a_i is at least (resp. less than) the profile value $b_i = 12$, for $i = 1, \ldots, 4$. The assignment rule (for a given profile) is a positive Boolean function $f(x_1, \ldots, x_4)$ which takes value 1 iff student a is succeeding (true point). Otherwise it takes value 0 (false point). In the first classification example, f is a threshold Boolean function since the true points and the false points can be linearly separated: x is a true point iff $\sum_{i=1}^{4} w_i x_i \geq 3/4$ with $w_i = 1/4,\ i = 1, \ldots, 4$. In the second classification example, the assignment rule is a positive Boolean function that is no threshold function.

Assignment rules based on sufficient coalitions of criteria which cannot be represented by additive weights and a threshold can always be represented by using a monotone nondecreasing set function, or, more specifically, a *capacity*, and a threshold. To further distinguish among these rules, the concept of k-additive capacity [8] is useful. 1-additive capacities correspond to additive weights as in MR-Sort. Of particular interest (in terms of the number of parameters involved) are the 2-additive capacities. In the language of Boolean functions, a capacity is a monotone non-decreasing pseudo-Boolean function which assigns value 0 to the empty set.

NCS rules that can be represented by a k-additive capacity and a threshold are related to k-additive positive pseudo-Boolean functions. In Boolean functions language, what we aim at is to determine the positive Boolean functions (i.e., NCS rules, with given limit profile) for which the true points can be separated from the false points by means of a k-additive positive pseudo-Boolean function (i.e. a k-additive capacity and a threshold). The $k = 1$ case corresponds to the MR-Sort rule and the separating function is a threshold Boolean function.

The questions dealt with in this paper are also closely related with the theory of simple games. The sufficient coalitions of a NCS model correspond to the *winning coalitions* in the theory of simple games (with the slight difference that a winning coalition cannot be empty). *Weighted simple games* correspond to MR-Sort models (and to threshold Boolean functions): winning coalitions can be represented by weights attached to the agents or players and a *quota*, i.e., a threshold [9]. NCS rules are related to k-additive games [8, 10] since these correspond to k-additive positive pseudo-Boolean functions. In the context of games, a concern similar to ours could be formulated as follows. Assume that we do not know the value of a game, but we know for each coalition whether the value of the game is satisfactory or not. We then want to identify k-additive games, with the smallest possible value of k, such that the satisfactory coalitions are characterized by a value at least as large as a threshold and the others by a value smaller than this threshold.

Our main motivation with this paper is to investigate the expressivity gap between MR-Sort and the NCS model (without veto). In this perspective, we analyze the possible families of sufficient coalitions up to a number of criteria equal to 6. We start by listing all these families, which raises difficulties due to the combinatorial and complex character of this issue. Then we study which families of sufficient coalitions are representable by k-additive capacities (for $k = 1, 2, 3$) and a threshold. These families are counted and listed. This study aims first at an explicit description of the families of sufficient criteria, up to $n = 6$, in order to support further more theoretical investigations and also practical applications in decision analysis. As a by-product, it enables to make simulations by drawing at random a MR-Sort model or a NCS model. This proves useful e.g. for testing the efficiency of algorithms designed for learning a NCS model [5] on the basis of assignment examples.

The rest of this paper is organized as follows. In Section 2, we state the problem more formally, make connections between ordered classification rules, positive Boolean functions and simple games. We define k-additive Boolean functions and we recall combinatorial results related to the enumeration of families of sufficient coalitions. Section 3 describes how the sets of sufficient coalitions were generated. In Section 4, we explain how we partitioned the families of sufficient coalitions; the size of each class of this partition is computed. The next section explains how these results can be exploited for simulation purposes. A short conclusion follows.

2. Sufficient coalitions, winning coalitions and Boolean functions

In this section we describe, in a formal way, the relationships between the sufficient coalitions of a noncompensatory sorting model, positive Boolean functions and the winning coalitions of a simple game. We state the question that motivates us in these different languages.

2.1. Ordered classification

Let $\{1, \ldots, n\}$ be a set of criteria. For assigning an object evaluated on these criteria to either the upper or the lower of two categories, the NCS model (without veto, [3])

- determines the subset of criteria on which the object is at least as good as a reference object called *profile*;
- assigns the object to the upper (resp. lower) category in case the subset of criteria forms a *sufficient* (resp. *insufficient*) *coalition* of criteria.

A NCS rule (for classifying in two categories, without veto) is therefore characterized by a profile and a set of subsets of the set of criteria $\{1, \ldots, n\}$, which are called *sufficient coalitions* (SCs) of criteria. The other subsets of criteria are called insufficient coalitions. The sole condition to be satisfied by the set of SCs in a NCS model is to be an *upset* of the power set of the set of criteria, ordered by inclusion $\subseteq$, i.e., it satisfies the following condition. For every subset of criteria A that is an SC, and for every B such that $A \subseteq B$, we have that B is an SC. The set of insufficient coalitions is the complement of the set of SCs in the power set of the set of criteria. It is a *downset* of this set. Note that the trivial cases in which either all coalitions are SCs or all coalitions are insufficient are not excluded.

2.2. Positive Boolean functions

In the language of Boolean functions, the set of SCs (resp. insufficient coalitions) of a NCS rule is the set of true (resp. false) points of a positive Boolean function (BF). To put it more formally, we introduce some notation. Let $N = \{1, \ldots, n\}$. The set of subsets of N is denoted by $\mathcal{P}(N)$. The set of Boolean vectors $x = (x_1, \ldots, x_n)$ of dimension n ($x_i = 0$ or 1, for $i \in N$) is denoted 2^n. The sets $\mathcal{P}(n)$ and 2^n are in one-to-one correspondence: each $x \in 2^n$ is the characteristic vector 1_A of some subset A of N. Each upset SC of $(\mathcal{P}(N), \subseteq)$ corresponds to a positive BF f defined by $f(x) = 1$ iff $x = 1_A$ for $A \in$ SC.

An MR-Sort rule is a particular case of the NCS model. The set of SCs of an MR-Sort rule can be determined by nonnegative weights $w_i, i \in N$ and a threshold λ. A subset $A \subseteq N$ forms an SC iff

$$\sum_{i \in A} w_i \geq \lambda. \tag{1}$$

We may assume w.l.o.g. that $\sum_{i=1}^{n} w_i = 1$. The positive BF f associated with the set of SCs of an MR-Sort rule thus satisfies $f(x) = 1$ iff $\sum_{i=1}^{n} w_i x_i \geq \lambda$. Such a positive BF is therefore a *threshold* BF: its true points can be separated from its false points by an affine function. Learning a threshold BF on the basis of examples of true and false points can be done efficiently, for instance, by using linear programming. Of course, this transposes to learning an MR-Sort rule, provided the profile is known. Note that the affine separating function is never unique ([11], Theorem 9.4, p. 409).

As was shown by an example in the introduction, there are NCS models the SCs of which cannot be represented using weights and a threshold as in (1). In the perspective of devising learning methods for NCS models we are interested in determining categories of sets of SCs that can be learned efficiently. In the context of BFs, the categories of *polynomial threshold functions of degree k*, for $k \in N$ generalize the threshold functions. Instead of separating the true points from the false points of a BF by an affine function, general separating functions have been considered, namely multilinear polynomials of degree k. These are pseudo-Boolean functions $p(x)$ of the form

$$p(x) = \sum_{A \in \mathcal{P}(N); |A| \leq k} c(A) \prod_{i \in A} x_i, \tag{2}$$

where $c(A)$ are coefficients that vanish for subsets A of N of cardinality superior to k. For any BF f, there is a degree $k \in N$ such that the true points and false points of f can be separated by a multilinear polynomial $p(x)$ of degree k [11, p. 407], in the following sense:

$$f(x) = 1 \quad \text{iff} \quad p(x) \geq 0. \tag{3}$$

The BF f is a *polynomial threshold function of degree* k if k is the smallest possible degree of a function $p(x)$ such that (3) holds (for more on this, see [12, 13, 14, 15]). Polynomial threshold BFs of degree 1 are just threshold BFs. All the latter are monotone. Polynomial threshold BFs of degree $k > 1$ are not necessarily monotone.

In the case f is a positive BF, its true points can be separated from its false points by a *monotone nondecreasing* (pseudo-Boolean) polynomial of degree k, for some $k \in N$. Actually, this is equivalent to separating the true from the false points by a capacity μ and a threshold λ. A capacity is a set function $\mu : \mathcal{P}(N) \to \mathbb{R}_+$ which is monotone w.r.t. to set inclusion, i.e., for all $A, B \subseteq N, A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$ (monotonicity) and $\mu(\emptyset) = 0$. A capacity can be given by means of its Möbius transform m. One has, for all $A \subseteq N$:

$$\mu(A) = \sum_{B \subseteq A} m(B) \tag{4}$$

where m is a set function $\mathcal{P}(N) \to \mathbb{R}$ which satisfies $\sum_{B \subseteq N} m(B) = 1$ and $\sum_{B: i \in B \subseteq A} m(B) \geq 0$, for all $i \in N$ and $A : i \in A \subseteq N$ [16] (see also [10], p. 53). A capacity is k-additive if its Möbius transform vanishes for subsets of size larger than k, i.e., $m(A) = 0$ for all $|A| > k$. We have that a positive BF f is separable in the sense of (3) by a *monotone nondecreasing* (pseudo-Boolean) polynomial p of degree k (for some $k \in N$) iff it is separable by a k-additive capacity μ and a threshold λ in the following sense: for $x = 1_A$,

$$f(x) = 1 \quad \text{iff} \quad \mu(A) \geq \lambda. \tag{5}$$

This result is easy to prove (since we did not find it formally established in the literature, we provide the interested reader with a proof in Appendix A).

Definition 1 (k-additive Boolean function). *A positive Boolean function f is called k-additive if k is the smallest integer such that the true points of f can be separated from its false points by a monotone non-decreasing pseudo-Boolean polynomial of degree k.*

A 1-additive Boolean function is a threshold function.

2.3. Simple games

The connection with *simple games* is direct. A simple game v (also called 0-1 capacity [10, p. 42]) is a monotone nondecreasing set function on N, which only takes the values 0 or 1 and is such that $v(\emptyset) = 0$. Simple games correspond to positive BFs which take value 0 on the null binary vector. The sets on which the game v has value 1 (which correspond to the true points of the associated positive BF), are called the *winning coalitions* of the game. They form an upset of $\mathcal{P}(N)$. The set of sufficient coalitions of a NCS model is the set of winning coalitions of a simple game (except in the trivial case of a NCS model for which all coalitions are sufficient). *Weighted* simple games correspond to MR-Sort models and to threshold BFs. [8] has introduced the notion of k-additive games (see also [10, p. 73]). The winning coalitions of k-additive simple games are the coalitions that fulfill $\mu(A) \geq \lambda$ for some k-additive capacity μ and a threshold λ.

2.4. Minimal sufficient coalitions

In the rest of the paper, we use "sufficient coalitions" (SCs) as a generic term referring to subsets of N corresponding to the true points of a n-variables positive BF or to the winning coalitions of a simple game of n players (including a trivial game in which the empty coalition is a winning one). Insufficient coalitions are the subsets of N that are no SC.

From the previous section we know that the set of SC can be specified by an inequality such as (5) with μ, a k-additive capacity for some k. In a preference learning perspective, this representation may be at an advantage since it allows to use powerful optimization techniques (see [17] for the learning of a NCS model on this basis)[2]. As was the case for $k = 1$, the capacity and threshold used for representing a family of SCs are never unique.

In the sequel, we concentrate on parsimonious representations, i.e., representations of a family of SCs as the set of coalitions A satisfying (5), using a k-additive capacity, with as small as possible k. The smaller k, the smaller the number of parameters to identify capacity μ, for instance, in a learning process. If $k = 1$, the family of SCs can be represented by an inequality of type (1), which involves determining the value of $n + 1$ parameters (the weights w_i and the threshold λ). If a family of SCs is representable using a 2-additive capacity, then we can write $\mu(A) = \sum_{i \in A} m_i + \sum_{i,j \in A, i \neq j} m_{ij}$,

[2] In [17], the NCS model without veto is called *capacitive MR-Sort model*.

n	$D(n)$	$R(n)$
0	2	2
1	3	3
2	6	5
3	20	10
4	168	30
5	7581	210
6	7828354	16353
7	2414682040998	490013148
8	56130437228687557907788	?

Table 1: Known values of the Dedekind numbers, $D(n)$, and of the number of inequivalent families of SC, $R(n)$.

where we abuse notation denoting $m(\{i\})$ by m_i and $m(\{i,j\})$ by m_{ij}. In this case, learning μ requires the determination of $\frac{n(n+1)}{2}+1$ parameters.

The set of SCs may be large (typically exponential in n), but one can avoid listing them all. A *minimal sufficient coalition* (MSC) is an SC which is not properly included in another SC. Knowing the set of MSCs allows to determine all SCs since a coalition is sufficient as soon as it contains a MSC. A family of MSCs can be any collection of subsets of N such that none of them is included in another. In other words, a set of MSCs is an *antichain* in the set of subsets of N (partially) ordered by inclusion. It is well-known that the number of antichains in the power set of N is $D(n)$, the nth Dedekind number ([18], sequence A000372). These numbers grow extremely rapidly with n and no exact closed form is known for them. These numbers have been computed up to $n = 8$; these values appear in the second column of Table 1.

Note that MSCs correspond to minimal true points of positive Boolean functions. Since its minimal true points characterize a positive BF [11, Theorems 1.13 and 1.26], the Dedekind numbers $D(n)$ are the numbers of positive Boolean functions in n variables. In game theory, MSCs correspond to minimal winning coalitions. The number of simple games with n players in minimal winning form [19, 20] equals $D(n)-1$ (the empty set is excluded as a winning coalition by definition of a simple game).

One way of simplifying the study of the families of SCs consists in keeping only one representative of each class of equivalent families of SCs. Two families will be considered as equivalent, or isomorphic, if they can be transformed one into the other just by permuting the labels of the elements. Consider e.g. the following family of minimal SCs for $n = 4$: $\{2,4\}, \{2,3\}, \{1,3,4\}$. It consists of 2 subsets of 2 elements and one of 3 elements. There are 12 equivalent families that can be obtained from this one, by permuting the elements' labels (the element which does not show up in the set of 3 can be chosen in 4 different ways and the two elements which distinguish the two pairs can be chosen in 3 different ways). The number $R(n)$ of *inequivalent* families of SCs is known for $n = 0$ to $n = 7$ ([18], sequence A003182). $R(7)$ was only recently computed by Stephen and Yusun [21]. Table 1 lists the known values of $R(n)$.

Finally we recall Sperner's theorem ([22], pp. 116-118), a result that will be useful in the process of generating all possible families of SCs. The maximal size of an antichain in the power set of a set of n elements is $\binom{n}{\lfloor n/2 \rfloor}$. Hence the latter is the maximal number of sets in a family of minimal SCs.

3. Listing the families of minimal sufficient coalitions

For generating all families of MSCs and selecting a representative of each class of equivalent families, we follow a strategy similar to the one used in [21]. We describe it briefly. The families of MSCs can be partitioned according to their *type* (called "profile" in [21]). The type of a family of MSCs is an integer vector $(k_1, k_2, \ldots, k_n)$, where k_i represents the number of coalitions of i elements in the family. For instance, the family $\{\{2,4\}, \{2,3\}, \{1,3,4\}\}$, for $n = 4$, is of the type $(0,2,1,0)$, since it involves two coalitions of 2 elements and one of 3. For any feasible type, $\sum_{i=1}^{n} k_i \leq \binom{n}{\lfloor n/2 \rfloor}$, due to Sperner's theorem.

The listing algorithm roughly proceeds as follows:

1. generate all type vectors $(k_1, k_2, \ldots, k_n)$ in lexicographic increasing order;
2. for each type, generate all families of subsets of N having the right type and eliminate those that are not antichains, i.e. those involving a subset that is included in another subset;
3. for each type and for each family of this type, the list of remaining families is screened for detecting families that are equivalent, counting them and eliminating them from the list of families of the type considered.

This algorithm outputs a list containing a representative of each class of equivalent families of MSCs for each type.

Type	Representative	# equivalent
(1,0,0)	{{1}}	3
(2,0,0)	{{1}, {2}}	3
(3,0,0)	{{1}, {2}, {3}}	1
(0,1,0)	{{1,2}}	3
(1,1,0)	{{1}, {2, 3}}	3
(0,2,0)	{{1, 3}, {2, 3}}	3
(0,3,0)	{{1, 2}, {1, 3}, {2, 3}}	1
(0,0,1)	{{1,2,3}}	1
Total	8	18

Table 2: Inequivalent families of minimal sufficient coalitions for $n = 3$

Example. For $n = 3$, the inequivalent families of MSCs, with their number of equivalent versions, are displayed in Table 2.

Remarks.

1. There exist two additional families which do not appear in Table 2:
   - the empty family, corresponding to the case in which no coalition is sufficient (or winning). For a sorting procedure, this means that all objects are assigned to the "bad" category. This corresponds to the BF $f \equiv 0$.
   - the family of which the sole element is the empty set. This is the only case which does not correspond to a simple game (by definition). It corresponds to the sorting rule that assigns all objects to the "good" category. It also corresponds to the BF $f \equiv 1$.

   Adding these two extreme cases to the counts in the last line of Table 2 yields values that are consistent with Table 1.
2. For $n = 3$, every possible class type has a single representative. For larger values of n, this is no longer the case. For instance, for $n = 4$, we have 3 inequivalent representatives for type $(0, 3, 0, 0)$:

Type	Representative	# equivalent
(0,3,0,0)	{{1, 3}, {1, 2}, {3, 4}}	12
(0,3,0,0)	{{2, 4}, {1, 2}, {1, 4}}	4
(0,3,0,0)	{{2, 4}, {3, 4}, {1, 4}}	4

   These three inequivalent families are the three sorts of non-isomorphic 3-edge graphs on 4 vertices displayed in Figure 2.

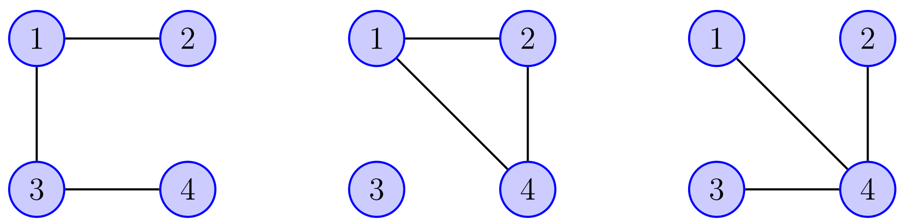

Figure 1: Graph representation of type (0,3,0,0) inequivalent families of SCs on 4 elements

3. In the sequel, in the absence of ambiguity, we drop the brackets around the coalitions and the commas separating the elements of a coalition in order to simplify the description of a family of SCs; for instance, the first family of type (0,3,0,0) above will be denoted by : $\{13, 12, 34\}$ instead of $\{\{1, 3\}, \{1, 2\}, \{3, 4\}\}$.

In order to make the algorithm sketched above more efficient, we implemented the three properties linking the types of MSCs families coined by Stephen and Yusun. For the sake of conciseness we do not describe them, referring the reader to lemma 2.4 in [21].

Using this algorithm on a cluster, we computed the list of all inequivalent families of MSCs for $n = 2$ to $n = 6$. The results, grouped by type, are available at `http://olivier.sobrie.be/shared/mbfs/`. For illustrative purposes, the case $n = 4$ is in Appendix B.

4. Partitioning the families of sufficient coalitions

4.1. Checking representability by a k-additive capacity

Our main goal in this section is to partition the set of families of MSCs, for fixed n, in categories $\mathcal{C}_k$, which are defined as follows.

Definition 2. *A family of sufficient coalitions belongs to class $\mathcal{C}_k$ if*

1. *there is a normalized k-additive capacity μ and a non-negative real number λ such that every coalition A in the family satisfies the inequality $\mu(A) \geq \lambda$, while the other coalitions do not satisfy this condition;*
2. *k is the smallest integer for which the latter holds.*

Note that a family of SCs belongs to class $\mathcal{C}_k$ if and only if it is the set of minimal true points of a k-additive Boolean function. It is clear that all equivalent families of MSCs belong to the same class $\mathcal{C}_k$. Hence it is sufficient to check for *one* representative of each class of equivalent families of MSCs whether or not it belongs to $\mathcal{C}_k$.

The checking strategy is the following. For each inequivalent family of MSCs (listed as explained in Section 3), we iteratively check whether it belongs to class $\mathcal{C}_k$, starting from $k = 1$ and incrementing k until the test is positive (this will occur at the latest for $k = n$). The test can be formulated as a linear program. We have to write constraints imposing that $\mu(A) \geq \lambda$ for each sufficient coalition A and that the same inequality is not satisfied for all other coalitions, i.e., the insufficient ones. It is enough to write these sorts of constraints only for the minimal SC and for the maximal insufficient coalitions. The set of minimal sufficient (resp. maximal insufficient) coalitions will be denoted SCMin (resp. ICMax).

To formulate the problem as a linear program, we use formula (4), which expresses the value of the capacity μ as a linear combination of its associated interaction function m. This enables to control the parameter k which fixes the k-additivity of the capacity. When checking whether a family of MSCs belongs to class $\mathcal{C}_k$, we set the values of the variables $m(B)$ to 0 for all sets B of cardinality greater than k; the remaining values of the interaction function are the main variables in the linear program. The following constitutes the general scheme of the linear programs used for each class $\mathcal{C}_k$:

$$\left\{\begin{array}{rcll} & \max\ \varepsilon & & \\ \mu(A) & \geq & \lambda & \forall A \in \text{SCMin} \\ \mu(A) & \leq & \lambda - \varepsilon & \forall A \in \text{ICMax} \\ \mu(A) & = & \displaystyle\sum_{B \subseteq A} m(B) & \forall A \in \text{SCMin} \cup \text{ICMax} \\ \displaystyle\sum_{B : i \in B \subseteq A} m(B) & \geq & 0 & \forall i \in N \\ & & \text{and} & \forall A \subseteq N \\ \displaystyle\sum_{B \subseteq N} m(B) & = & 1 & \\ \lambda, \varepsilon & \geq & 0 & \end{array}\right. \tag{6}$$

Note that the variables $m(B)$ are not necessarily positive (except for $|B| = 1$). To fix the ideas, we show how to instantiate the third, fourth and fifth constraints in the cases $k = 1$ and $k = 2$.

- $k=1$: 1-additive capacity

  . $\mu(A)=\sum_{i\in A} m_i,\ \forall A\in \text{SCMin}\cup\text{ICMax}$

  . $m_i\geq 0,\ \forall i\in N$

  . $\sum_{i\in N} m_i=1,$

  where m_i stands for $m(\{i\})$

- $k=2$: 2-additive capacity

  . $\mu(A)=\sum_{i\in A} m_i+\sum_{i,j\in A,\ i\neq j} m_{ij},\ \forall A\in \text{SCMin}\cup\text{ICMax}$

  . $m_i+\sum_{j\in A,\ j\neq i} m_{ij}\geq 0,\ \forall i\in N$ and $\forall A\ni i,\ A\subseteq N$

  . $\sum_{i\in N} m_i+\sum_{i,j\in N,\ i\neq j} m_{ij}=1,$

  where m_i stands for $m(\{i\})$ and m_{ij} for $m(\{i,j\})$.

Writing the constraints for the 3-additive case requires the introduction of a third family of variables m_{ijl} for each subset $\{i,j,l\}$ of three different elements of N (in addition to the already introduced variables m_i and m_{ij}).

4.2. Results

For $n=1$ to 6 and for each family in the list of inequivalent families of MSCs, we checked whether this family belongs to $\mathcal{C}_k$, starting with $k=1$ and incrementing its value until a positive answer is reached. The results are displayed in Table 3, regarding the number and proportion of inequivalent families in classes $\mathcal{C}_1$, $\mathcal{C}_2$ and $\mathcal{C}_3$. Up to $n=6$, inclusively, there are no families in classes $\mathcal{C}_4$ or above, which means that all families can be represented using a 3-additive capacity (in the worst case). Up to $n=5$, inclusively, 2-additive capacities are sufficient. Table 4 represents a similar information but each family in the list of inequivalent families is weighted by the size of the equivalence class it represents. In other words, this is the result that would have been obtained by checking all families of MSCs for belonging to class $\mathcal{C}_1$, $\mathcal{C}_2$ or $\mathcal{C}_3$.

The information displayed in Table 3 (resp. 4) is represented in graphical form in Figure 2 (resp. 3). The cases of 0, 1 and 2 elements need not be represented since all families belong to class $\mathcal{C}_1$. These figures clearly show that the proportion of families that can be represented by means of a 1-additive capacity, i.e. by additive weights, decreases quite rapidly with the number

n	$\mathcal{C}_1$		$\mathcal{C}_2$		$\mathcal{C}_3$		$R(n)$
0	2	(100.0 %)	0	(00.00 %)	0	(00.00 %)	2
1	3	(100.0 %)	0	(00.00 %)	0	(00.00 %)	3
2	5	(100.0 %)	0	(00.00 %)	0	(00.00 %)	5
3	10	(100.0 %)	0	(00.00 %)	0	(00.00 %)	10
4	27	(90.00 %)	3	(10.00 %)	0	(00.00 %)	30
5	119	(56.67 %)	91	(43.33 %)	0	(00.00 %)	210
6	1113	(06.81 %)	14902	(91.13 %)	338	(02.07 %)	16 353

Table 3: Number and proportion of inequivalent families of MSCs that are representable by a 1-, 2- or 3-additive capacity

n	$\mathcal{C}_1$		$\mathcal{C}_2$		$\mathcal{C}_3$		$D(n)$
0	2	(100.0 %)	0	(00.00 %)	0	(00.00 %)	2
1	3	(100.0 %)	0	(00.00 %)	0	(00.00 %)	3
2	6	(100.0 %)	0	(00.00 %)	0	(00.00 %)	6
3	20	(100.0 %)	0	(00.00 %)	0	(00.00 %)	20
4	150	(89.29 %)	18	(10.71 %)	0	(00.00 %)	168
5	3 287	(43.36 %)	4 294	(56.64 %)	0	(00.00 %)	7581
6	244 158	(03.12 %)	7 438 694	(95.02 %)	145 502	(01.86 %)	7 828 354

Table 4: Number and proportion of all families of MSCs that are representable by a 1-, 2- or 3-additive capacity

of criteria. In contrast, the proportion of families that can be represented by a 2-additive capacity grows up to more than 91% from $n = 3$ to $n = 6$. The proportions slightly differ depending on whether only inequivalent families or all families are taken into account. One can observe that the proportion of families in class $\mathcal{C}_2$ is a bit larger when considering all families (Table 4 and Figure 3).

Examples. As a matter of illustration, we describe a few examples for $n = 4$ and $n = 6$. The list of all inequivalent MSCs for $n = 5$, which are not representable by a 1-additive capacity, is displayed in appendix B. The categorization in classes $\mathcal{C}_k$ is available at `http://olivier.sobrie.be/shared/mbfs/` for $n = 4, 5, 6$.

1. Here are the three families of MSCs on 4 elements that cannot be represented using a 1-additive capacity (they can be by a 2-additive capacity).

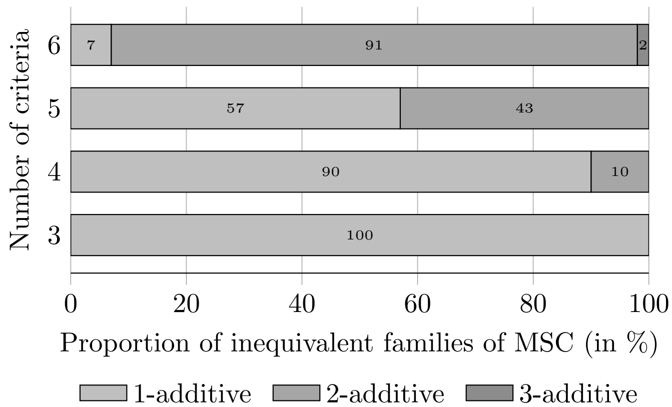

Figure 2: Proportion of inequivalent families of MSCs in classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$

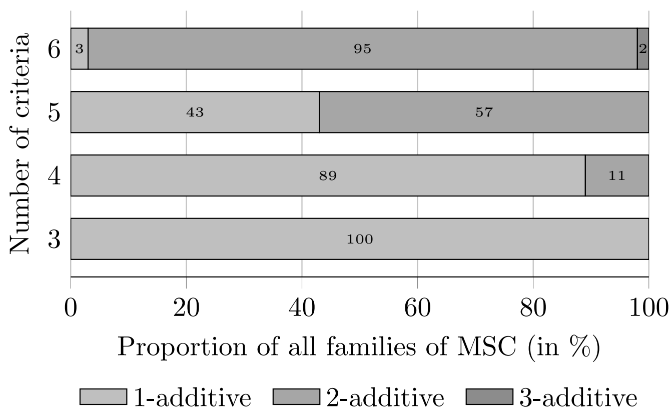

Figure 3: Proportion of all families of MSCs in classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$

Type	Representative	# equivalent
(0,2,0,0)	{12, 34}	3
(0,3,0,0)	{13, 12, 34}	12
(0,4,0,0)	{13, 14, 23, 24}	3

These three inequivalent families yield, by permutations of the elements labels, a total of 18 families that can only be represented using a 2-additive capacity.

The last inequivalent family is precisely the example that we used in Section 1 to show that not all families of SCs can be represented by a 1-additive capacity. In contrast, it can be represented, for instance, by setting $m_1 = m_2 = m_3 = m_4 = 1/6$ and $m_{13} = m_{14} = m_{23} = m_{24} =$

$1/12$, while the other pairwise interactions m_{12} and m_{34} are set to 0. We then have: $\mu(13) = \mu(14) = \mu(23) = \mu(24) = 5/12$ while $\mu(12) = \mu(34) = 4/12$. Setting the threshold λ to $9/24$ allows to separate the sufficient coalitions from the insufficient. Such a representation is never unique. For example, another capacity is obtained by setting $m_1 = m_2 = m_3 = m_4 = 1/3$, $m_{12} = m_{34} = -1/6$ and $m_{13} = m_{14} = m_{23} = m_{24} = 0$. We have: $\mu(13) = \mu(14) = \mu(23) = \mu(24) = 2/3$ while $\mu(12) = \mu(34) = 1/2$. Setting the threshold λ to $7/12$ also separates the sufficient from the insufficient coalitions.

   Note that the second example, a family of type (0,3,0,0), already appeared in Remark 2 after Table 2.

   Note also that the first and the last example are *complementary* in an obvious sense. These families form a bipartition of the set of all pairs of elements. This complementarity is related to the first property allowing to speed up the enumeration of the families of MSCs used in [21] and mentioned at the end of Section 3. In view of the 2-additive capacity used in the third example, it is straightforward to build a 2-additive capacity for the first example. Therefore, both families belong to class $\mathcal{C}_2$.

2. Here are two examples of inequivalent families of MSCs on 6 elements that are not representable by a 2-additive capacity but require a 3-additive capacity. There are 338 such inequivalent families which yield, through permutations, a total of 145 502 families[3]. A simple example of type (0,0,4,0,0,0) is the following family of MSCs: $\{136, 234, 125, 456\}$. There are 30 equivalent families that can be derived from this family through permutations. Another, much more complex example is of the type (0,1,7,1,0,0). The MSCs are $\{135, 256, 345, 36, 234, 456, 1245, 146, 123\}$. There are 360 equivalent families that can be obtained through permutations.

   Among the 338 families belonging to $\mathcal{C}_3$, no MSC consists of a single criterion; none of them involves 5 elements. The largest number of MSCs in such a family is 16, the maximal cardinality of a family of MSCs on 6 elements being the Sperner number 20.

[3]If all permutations of the elements labels were yielding different families, the total number of families would be $338 \times 720 = 243\,360$

4.3. What about $n > 6$?

Stephen and Yusun [21] have computed the number of inequivalent families of MSCs up to $n = 7$. Categorizing the families of MSCs in classes $\mathcal{C}_k$ for $n = 7$, with the same method we used in the case $n \leq 6$ cannot be envisaged. We do not only have to generate and store the inequivalent families of MSCs. In addition, for each family, we have to compute the maximal insufficient coalitions, write and solve usually more than one linear program in order to determine the minimal value of k for which the family belongs to $\mathcal{C}_k$. The order of magnitude of the time needed for categorizing each family out of the 16 353 families for $n = 6$ is about 1 second, which amounts to a total computing time of more than 4 hours. Since, the number of inequivalent families of MSCs for $n = 7$, $R(7)$, is more than 30 000 times $R(6)$, using the same method is too time-consuming. Establishing properties that allow to categorize some families of MSCs without solving linear programs is an interesting research question.

As early as 1970, Muroga, Tsuboi and Baugh [23] managed to compute the number of inequivalent threshold functions, i.e., the cardinal of the class $\mathcal{C}_1$ of inequivalent families of MSCs, up to $n = 8$. There are 29 375 (resp. 2 730 166) threshold functions for $n = 7$ (resp. $n = 8$). The authors strategy consists of identifying threshold functions in a (relatively) small class of BF's, which is known to contain them all. Linear programming was used to identify threshold functions in this class (see [23, 24] for more detail). Enumerating the threshold functions up to $n = 8$ was made possible by a detailed knowledge of the properties of these functions (a very active topic in switching theory and threshold logic in the 60's). Such a detailed knowledge, for the class $\mathcal{C}_2$, for instance, is not available.

While extending the classification of families of MSCs for $n \geq 7$ is not straightforward, it is worth noting that asymptotic formulas are known both for the number of monotone BF's [25] (see also [26], p. 933) and the number of threshold functions [27] (see also [12], p. 561). We are not aware of asymptotic results for the sizes of classes $\mathcal{C}_k$, for $k \geq 2$.

5. Applications

The above results, although limited to 6 elements or criteria, can be used for different purposes. We focus on two applications in ordered classification based on multiple criteria.

5.1. Selection of a sorting model in view of learning a classification

Classifying objects described by Boolean vectors in two categories in a monotone nondecreasing way is equivalent to determine a positive Boolean function. Efficient learning or elicitation by queries of a BF has been extensively studied (see e.g., [12, 28]). The MR-Sort or the NCS rules classify objects represented by their evaluations on n criteria, which usually are not binary values. Therefore, eliciting or learning a MR-Sort or a NCS model (without veto) involves the determination of an additional element, namely the lower limit profile of the upper category. In these models, the limit profile allows to binarize the evaluations depending on whether each of them is above or below the corresponding value of the limit profile. The object is subsequently assigned to a category by means of a positive BF applied to the Boolean representation of the object w.r.t. the limit profile.

The algorithm proposed in [2, 5] to learn a MR-Sort or a NCS model on the basis of large sets of assignment examples proceeds by iterating two phases. Starting from an initial estimate of the profile, a k- additive capacity which maximizes the number of examples correctly assigned is determined by solving a linear program. Then the profile is modified, without changing the capacity, in order to improve the rate of correct classification. The process is then iterated with the new profile. Choosing an appropriate value of k has important consequences in this process. Large values of k lead to linear programs involving many variables (as shown in Section 4.1). This study shows that 1-additive capacities (corresponding to a MR-Sort model) are enough up to $n = 4$ criteria. Up to $n = 6$, in most of the cases (93% in terms of inequivalent families of MSCs and more than 96% if we consider all families of MSCs), a 2-additive capacity is enough. Obviously, it would be useful to know which is the most frequent class of k-additive BFs for $n \geq 7$ and have an estimation of the distribution of positive BFs among the k-additivity classes.

5.1.1. Simulation

Recently, methods have been proposed to learn variants of the Electre Tri sorting model on the basis of assignment examples [1, 29, 2, 17]. It has also been done [30] for a ranking method based on *reference points* proposed by Rolland [31, 32]. Consider e.g. a learning algorithm designed to learn a MR-Sort model, as in [2]. Real data sets can be used to test the performance of the algorithm. But for learning algorithms which aim at selecting a rule in a specific family of sorting rules, it is also needed to perform the following test, with artificial data. When a set of assignment examples is generated by a known MR-Sort model, we want to check whether the algorithm, when applied to these examples, learns a model similar to the original one. If some noise is added to the learning set, one expects that the algorithm remains robust. In order to design such tests, we have to draw at random a MR-Sort model, i.e. the profiles, the criteria weights and a threshold. Drawing the profiles and the threshold at random does not raise major problems. An algorithm for drawing weights summing up to 1 in a uniform way is also well-known [33].

In order to perform the same type of tests in the case of the NCS model (without veto), we are facing a difficulty. How to draw a capacity at random, or more particularly, a k-additive capacity? How can one define a uniform distribution on the set of capacities? On second thought, we moved to another formulation of this question. What we have to do is to draw at random, uniformly (in some sense), an MR-Sort rule or a NCS rule (without veto), not a capacity. And this makes a difference, since the representation of a NCS rule by an inequality involving a capacity and a threshold is not unique (as observed previously), hence there is a representation bias in this approach. Note that this remark also applies to drawing at random an MR-Sort model. The alternative is thus to *select a rule at random*, i.e. a family of MSCs. That's what our results allow to do, up to $n = 6$. There is no need to test the algorithm for several equivalent versions of the same rule (i.e. for families of MSCs that only differ by a permutation of the criteria labels). We can thus sample the set of inequivalent families (each weighted proportionally to the size of its equivalence class). Note that the lists of inequivalent families also permit to consider non-uniform distributions and to draw at random from them according to an arbitrary probability distribution on the families.

6. Conclusion

In this work, we explored the families of minimal sufficient coalitions as they appear in sorting models such as MR-Sort and NCS. These families are in one-to-one correspondence with positive BFs, of which they are the sets of minimal true points. This exploration is limited to small numbers of criteria because of the huge number of such models. Our goal was at least twofold:

1. to provide a detailed picture of the possible families of SCs for as many criteria as we could; this information could help further investigations

related in particular to the characterization of the families of SCs that can be separated from the insufficient ones by an inequality involving a k-additive capacity. In other words, this could help in the study of k-additive BFs.

2. to make available a list of the possible sorting rules in the NCS model, in order to enable to draw a rule at random according to any specified probability distribution and use it in simulations. The space needed to store these lists and the time to scan them can be reduced, to some extent, by retaining only inequivalent rules.

Further efforts in the future could lead to obtain the list of inequivalent families of SCs for $n = 7$, probably requiring a theoretical study of the different classes $\mathcal{C}_k$. Alternatively, other approaches to subdividing the set of all families of SCs could be of practical and theoretical interest.

Acknowledgements. We are grateful to Yves Crama and Endre Boros for providing us with many references to the theory of Boolean functions in relation with the topic of this work. We thank two anonymous referees for their useful comments which helped us improving the initial manuscript.

References

[1] A. Leroy, V. Mousseau, M. Pirlot, Learning the parameters of a multiple criteria sorting method, in: R. Brafman, F. Roberts, A. Tsoukiàs (Eds.), Algorithmic Decision Theory, Vol. 6992 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2011, pp. 219–233.

[2] O. Sobrie, V. Mousseau, M. Pirlot, Learning a majority rule model from large sets of assignment examples, in: P. Perny, M. Pirlot, A. Tsoukiás (Eds.), Algorithmic Decision Theory, Vol. 8176 of Lecture Notes in Artificial Intelligence, Springer, 2013, pp. 336–350.

[3] D. Bouyssou, T. Marchant, An axiomatic approach to noncompensatory sorting methods in MCDM, I: The case of two categories, European Journal of Operational Research 178 (1) (2007) 217–245.

[4] D. Bouyssou, T. Marchant, An axiomatic approach to noncompensatory sorting methods in MCDM, II: More than two categories, European Journal of Operational Research 178 (1) (2007) 246–276.

[5] O. Sobrie, V. Mousseau, M. Pirlot, Learning the parameters of a non compensatory sorting model, in: T. Walsh (Ed.), Algorithmic Decision Theory, Vol. 9346 of Lecture Notes in Artificial Intelligence, Springer, 2015, pp. 153–170.

[6] B. Roy, D. Bouyssou, Aide Multicritère à la Décision : Méthodes et Cas, Economica, Paris, 1993.

[7] D. Bouyssou, T. Marchant, M. Pirlot, A. Tsoukiàs, P. Vincke, Evaluation and decision models with multiple criteria: Stepping stones for the analyst, International Series in Operations Research and Management Science, Volume 86, Springer, Boston, 2006.

[8] M. Grabisch, k-order additive discrete fuzzy measures and their representation, Fuzzy Sets and Systems 92 (2) (1997) 167 – 189.

[9] A. D. Taylor, W. S. Zwicker, A characterization of weighted voting, Proceedings of the American Mathematical Society 115 (1992) 1089–1094.

[10] M. Grabisch, Set Functions, Games and Capacities in Decision Making, Springer, 2016.

[11] Y. Crama, P. L. Hammer, Boolean Functions - Theory, Algorithms, and Applications, Vol. 142 of Encyclopedia of mathematics and its applications, Cambridge University Press, New York, USA, 2011.

[12] M. Anthony, Neural networks and Boolean functions, in: Y. Crama, P. L. Hammer (Eds.), Boolean Models and Methods in Mathematics, Computer Science, and Engineering, Vol. 134 of Encyclopedia of mathematics and its applications, Cambridge University Press, New York, USA, 2010, pp. 554–576.

[13] M. Anthony, Discrete mathematics of neural networks: selected topics, SIAM monographs on discrete mathematics and applications, Society for industrial and applied mathematics, New York, 2001.

[14] J. Bruck, Fourier transforms and threshold circuit complexity, in: Y. Crama, P. L. Hammer (Eds.), Boolean Models and Methods in Mathematics, Computer Science, and Engineering, Vol. 134 of Encyclopedia

of mathematics and its applications, Cambridge University Press, New York, USA, 2010, pp. 531–553.

[15] M. E. Saks, Slicing the hypercube, in: K. Walker (Ed.), Surveys in Combinatorics, Cambridge University Press, New York, USA, 1993, pp. 211–255.

[16] A. Chateauneuf, J. Jaffray, Some characterizations of lower probabilities and other monotone capacities through the use of Möbius inversion, Mathematical Social Sciences 17 (1989) 263–283.

[17] O. Sobrie, V. Mousseau, M. Pirlot, Learning the parameters of a majority rule sorting model taking attribute interactions into account, in: Proceedings of the DA2PL 2014 Workshop, November 20-21, 2014, Paris, France, 2014, pp. 22–30.

[18] The OEIS Foundation Inc., On-line encyclopedia of integer sequences, `https://oeis.org`, [Online; accessed April 12, 2016].

[19] J. von Neumann, O. Morgenstern, Theory of Games and Economic Behavior, 3rd Edition, Princeton University Press, NJ, USA, 1972.

[20] S. Kurz, N. Tautenhahn, On Dedekind´s problem for complete simple games, International Journal of Game Theory 42 (2) (2013) 411–437.

[21] T. Stephen, T. Yusun, Counting inequivalent monotone boolean functions, Discrete Applied Mathematics 167 (2014) 15 – 24.

[22] N. Caspard, B. Leclerc, B. Monjardet, Finite Ordered Sets : Concepts, results and uses, Encyclopedia of Mathematics and its applications, Cambridge University Press, New York, USA, 2012.

[23] S. Muroga, T. Tsuboi, C. Baugh, Enumeration of threshold functions of eight variables, IEEE Transactions on Computers C-19.

[24] R. O. Winder, Enumeration of seven-argument threshold functions, IEEE Transactions on Electronic Computers EC-14.

[25] A. Korshunov, On the number of monotone boolean functions, Problemy Kibernetiki 38 (1981) 5–108, in Russian.

[26] A. D. Korshunov, Monotone boolean functions, Russian Mathematical Surveys 58.

[27] Y. A. Zuev, Asymptotics of the logarithm of the number of threshold functions of the algebra of logic, Soviet Mathematics Doklady 39 (1989) 512–513.

[28] R. H. Sloan, B. Szörényi, G. Turán, Learning Boolean Functions with Queries, Vol. 134 of Encyclopedia of Mathematics and its Applications, Cambridge University Press, New York, USA, 2010, pp. 221–256.

[29] J. Zheng, Preference elicitation for reference based aggregation models: Algorithms and Procedures, Phd thesis, Ecole Centrale Paris (2012).

[30] J. Liu, V. Mousseau, W. Ouerdane, Preference elicitation from inconsistent pairwise comparisons for multi-criteria ranking with multiple reference points, in: Proceedings of the 14th International Conference on Informatics and Semiotics in Organisations (ICISO 14), Stockholm (Sweden), 2013, pp. 120–132.

[31] A. Rolland, Reference-based preferences aggregation procedures in multi-criteria decision making, European Journal of Operational Research 225 (3) (2013) 479 – 486.

[32] D. Bouyssou, T. Marchant, Multiattribute preference models with reference points, European Journal of Operational Research 229 (2) (2013) 470 – 481.

[33] N. A. Smith, R. W. Tromble, Sampling uniformly from the unit simplex, Department of Computer Science, Center for Language and Speech Processing, Johns Hopkins University (August 2004).

Appendix A. Separability

Let f be a positive BF. We establish the following equivalence: for some $k \in N$,

1. f is separable by a *monotone nondecreasing* (pseudo-Boolean) polynomial p of degree k, in the sense that $f(x) = 1$ iff $p(x) \geq 0$;
2. f is separable by a k-additive capacity μ and a threshold λ, in the sense that, for $x = 1_A$, $f(x) = 1$ iff $\mu(A) \geq \lambda$.

Assume the latter. We have $\mu(A) = \sum_{B\subseteq A;|B|\leq k} m(B)$. The k-additive capacity μ corresponds to the monotone pseudo-Boolean polynomial $p(x)$ of degree k defined by $p(x) = \sum_{B\subseteq N;|B|\leq k} m(B) \prod_{i\in B} x_i$. For $x = 1_A$, $p(x) = \mu(A)$ and we have $\mu(A) \geq \lambda$ iff $p(x) \geq \lambda$. It is easy to modify the polynomial p into another polynomial p' in order to have $p(x) \geq \lambda$ iff $p'(x) \geq 0$. To do this, we translate p by adding $-\lambda$. Actually, we change the value $m(\emptyset) = 0$ into $m'(\emptyset) = -\lambda$. The obtained polynomial p' is also monotone nondecreasing.

Starting from the hypothesis that $f(x) = 1$ iff $p(x) \geq 0$, with p a monotone non decreasing polynomial of degree k, we add $\lambda = -p(0)$, the value of the characteristic function of the emptyset, to both sides of the latter inequality, getting $p(x) - p(0) \geq \lambda$. The righthand side is a monotone non-decreasing pseudo-Boolean function which vanishes on the null vector. The set function μ defined by $\mu(A) = p(1_A) - p(0)$ is a capacity. Since $p(x) - p(0)$ has a representation as a multilinear polynomial of degree k, the corresponding capacity is k-additive.

Note that the trivial cases in which all coalitions are sufficient or none of them is are separable by a polynomial of degree 0 (i.e., a constant) or a 0-additive capacity (i.e., the null constant).

Appendix B. List of inequivalent families of MSCs for $n = 4$

The families are grouped by type. There are 25 possible types, 29 inequivalent families of MSCs (plus the trivial case in which all coalitions are sufficient) and 167 families of MSCs (plus the same trivial case). Each inequivalent family in the list is associated the size of its equivalence class. All inequivalent families, except three of them, can be represented by a 1-additive capacity. The three other families can be represented by a 2-additive capacity. They are marked in the last column by $\mathcal{C}_2$.

Type	Family of MSCs	# eq.	$\mathcal{C}_k$
(0,0,0,0)	{}	1	
(0,0,0,1)	{1234}	1	
(0,0,1,0)	{124}	4	
(0,0,2,0)	{234, 124}	6	
(0,0,3,0)	{134, 123, 124}	4	
(0,0,4,0)	{134, 123, 234, 124}	1	
(0,1,0,0)	{24}	6	
(0,1,1,0)	{14, 123}	12	
(0,1,2,0)	{24, 134, 123}	6	
(0,2,0,0)	{12, 23}	12	
	{23, 14}	3	$\mathcal{C}_2$
(0,2,1,0)	{24, 134, 23}	12	
(0,3,0,0)	{13, 12, 34}	12	$\mathcal{C}_2$
	{24, 12, 14}	4	
	{24, 34, 14}	4	
(0,3,1,0)	{13, 34, 23, 124}	4	
(0,4,0,0)	{24, 12, 13, 34}	3	$\mathcal{C}_2$
	{24, 12, 14, 23}	12	
(0,5,0,0)	{24, 12, 14, 13, 34}	6	
(0,6,0,0)	{24, 12, 14, 34, 23, 13}	1	
(1,0,0,0)	{1}	4	
(1,0,1,0)	{234, 1}	4	
(1,1,0,0)	{14, 2}	12	
(1,2,0,0)	{13, 34, 2}	12	
(1,3,0,0)	{24, 34, 23, 1}	4	
(2,0,0,0)	{4, 3}	6	
(2,1,0,0)	{4, 23, 1}	6	
(3,0,0,0)	{4, 2, 1}	4	
(4,0,0,0)	{4, 2, 3, 1}	1	

Appendix C. List of inequivalent families of MSCs of class $\mathcal{C}_2$ for $n = 5$

We list below the 91 inequivalent families of MSCs that cannot be represented by a 1-additive capacity. They can all be represented using a 2-additive capacity. The families are grouped by type. Each inequivalent family in the list is associated the size of its equivalence class.

Type	Family of MSCs	# eq.
(0,0,2,0,0)	{135, 234}	15
(0,0,2,1,0)	{234, 125, 1345}	15
(0,0,3,0,0)	{145, 123, 345}	30
	{235, 234, 125}	60
(0,0,3,1,0)	{134, 135, 2345, 124}	60
(0,0,4,0,0)	{145, 234, 345, 124}	15
	{135, 245, 234, 125}	60
	{235, 145, 135, 123}	60
	{134, 345, 234, 125}	10
(0,0,4,1,0)	{245, 123, 234, 125, 1345}	15
(0,0,5,0,0)	{235, 134, 135, 345, 125}	60
	{235, 134, 135, 245, 124}	12
	{235, 145, 134, 245, 124}	60
	{145, 134, 123, 234, 125}	60
(0,0,6,0,0)	{135, 235, 234, 125, 145, 123}	15
	{135, 345, 234, 125, 245, 123}	10
	{345, 235, 234, 125, 124, 134}	60
	{135, 345, 235, 125, 124, 145}	60
(0,0,7,0,0)	{345, 234, 125, 145, 134, 245, 123}	30
	{135, 235, 125, 124, 145, 134, 245}	60
(0,0,8,0,0)	{135, 345, 234, 125, 124, 145, 245, 123}	15
(0,1,1,0,0)	{123, 45}	10
(0,1,2,0,0)	{15, 123, 345}	60
	{12, 134, 345}	60
(0,1,3,0,0)	{235, 14, 123, 125}	60
	{13, 235, 145, 124}	60
	{235, 14, 123, 245}	60
	{24, 134, 135, 123}	30

Type	Family of MSCs	# eq.
(0,1,4,0,0)	{235, 15, 245, 123, 234}	120
	{135, 123, 25, 345, 124}	60
	{235, 34, 145, 125, 124}	60
	{24, 235, 135, 123, 125}	20
(0,1,5,0,0)	{345, 235, 15, 234, 134, 123}	30
	{235, 125, 124, 145, 34, 123}	60
	{24, 135, 345, 235, 125, 123}	60
(0,1,6,0,0)	{24, 135, 345, 235, 145, 134, 123}	60
(0,2,0,0,0)	{34, 15}	15
(0,2,1,0,0)	{12, 35, 234}	60
	{145, 23, 25}	60
(0,2,2,0,0)	{24, 13, 125, 345}	30
	{24, 12, 135, 345}	30
	{134, 23, 35, 124}	60
	{13, 12, 245, 234}	120
	{12, 245, 35, 234}	60
(0,2,3,0,0)	{15, 23, 134, 345, 124}	60
	{45, 134, 135, 234, 25}	120
	{135, 123, 45, 125, 14}	60
	{24, 235, 14, 345, 135}	30
	{24, 34, 135, 123, 125}	60
(0,2,4,0,0)	{135, 235, 14, 234, 123, 45}	60
	{14, 35, 234, 125, 245, 123}	15
	{24, 135, 235, 125, 34, 123}	30
(0,3,0,0,0)	{12, 14, 45}	60
	{12, 34, 45}	30
(0,3,1,0,0)	{24, 145, 23, 25}	60
	{34, 14, 35, 125}	60
	{34, 245, 23, 14}	120
	{34, 14, 123, 25}	60
(0,3,2,0,0)	{15, 14, 123, 25, 345}	60
	{24, 12, 134, 35, 145}	30
	{13, 23, 245, 125, 14}	120
	{15, 45, 123, 234, 25}	60
(0,3,3,0,0)	{24, 135, 145, 134, 23, 25}	20
	{12, 35, 234, 145, 13, 245}	60

Type	Family of MSCs	# eq.
(0,4,0,0,0)	{34, 15, 14, 35}	15
	{24, 15, 23, 25}	60
	{24, 34, 15, 23}	10
	{24, 34, 15, 35}	60
(0,4,1,0,0)	{13, 34, 35, 25, 145}	60
	{24, 13, 15, 25, 345}	60
	{13, 15, 23, 25, 345}	30
	{34, 14, 45, 125, 23}	60
(0,4,2,0,0)	{24, 12, 35, 145, 134, 23}	60
	{24, 35, 145, 34, 25, 123}	15
(0,5,0,0,0)	{24, 13, 15, 23, 14}	60
	{24, 12, 15, 35, 25}	60
	{24, 12, 15, 35, 34}	12
	{12, 15, 34, 25, 45}	60
(0,5,1,0,0)	{135, 12, 14, 34, 23, 25}	60
	{15, 35, 124, 23, 13, 45}	60
(0,6,0,0,0)	{24, 12, 23, 25, 13, 45}	15
	{24, 12, 35, 34, 25, 13}	10
	{24, 12, 34, 23, 13, 45}	60
	{15, 14, 34, 23, 25, 45}	60
(0,6,1,0,0)	{24, 12, 35, 145, 34, 25, 13}	10
(0,7,0,0,0)	{12, 14, 34, 23, 25, 13, 45}	30
	{24, 12, 15, 14, 35, 34, 45}	60
(0,8,0,0,0)	{24, 12, 15, 34, 23, 25, 13, 45}	15
(1,2,0,0,0)	{34, 15, 2}	15
(1,3,0,0,0)	{24, 15, 3, 25}	60
(1,4,0,0,0)	{13, 2, 14, 35, 45}	15