Listing the families of Sufficient Coalitions of criteria involved in Sorting procedures

Eda Ersek Uyanık[1,4], **Olivier Sobrie**[2,3,4], **Vincent Mousseau**[3] and **Marc Pirlot**[4]

Abstract. Certain sorting procedures derived from ELECTRE TRI such as MR-Sort or the Non-Compensatory Sorting (NCS model) model rely on a rule of the type: if an object is better than a profile on a "sufficient coalition" of criteria, this object is assigned to a category above this profile. In some cases the strength a coalition can be numerically represented by the sum of weights attached to the criteria and a coalition is sufficient if its strength passes some threshold. This is the type of rule used in the MR-Sort method. In more general models such as Capacitive-MR-Sort or NCS model, criteria are allowed to interact and a capacity is needed to model the strength of a coalition. In this contribution, we want to investigate the gap of expressivity between the two models. In this view, we explicitly generate a list of all possible families of sufficient coalitions for a number of criteria up to 6. We also categorize them according to the degree of additivity of a capacity that can model their strength. Our goal is twofold: being able to draw a sorting rule at random and having at disposal examples in view of supporting a theoretical investigation of the families of sufficient coalitions.

1 Introduction

A *sorting method*, in Multiple Criteria Decision Analysis, is a procedure for assigning objects (or alternatives) described by their evaluation on several criteria to ordered categories. ELECTRE TRI [17, 10] is a sorting method based on an outranking relation. Basically, each category has a lower limit profile which is also the upper limit profile of the category below. An object is assigned to a category if it outranks the lower limit profile of this category but does not outrank its upper limit profile. MR-Sort is a simple version of ELECTRE TRI. MR-Sort assigns an object to a category if its evaluations are better than the value of the lower limit profiles on a majority of criteria and this condition is not fulfilled with respect to the upper limit profile of the category. More precisely, a weight w_i is attached to each criterion $i = 1, 2, \ldots, n$ and the object $a = (a_1, a_2, \ldots, a_n)$ is assigned to a category above profile $b = (b_1, b_2, \ldots, b_n)$ whenever the sum of the weights of the criteria for which $a_i \geq b_i$ passes some threshold λ. Otherwise, it is assigned to a category below b.

[1] email: eda.uyanik@gmail.com
[2] email: olivier.sobrie@gmail.com
[3] École Centrale Paris, Grande Voie des Vignes, 92295 Châtenay Malabry, France, email: vincent.mousseau@ecp.fr
[4] Université de Mons, Faculté Polytechnique, 9, rue de Houdain, 7000 Mons, Belgium, email: marc.pirlot@umons.ac.be

An intermediary sorting method in between ELECTRE TRI and MR-Sort was proposed and characterized by Bouyssou and Marchant [1, 2]. It is known as the *Non Compensatory Sorting* (NCS) model. Consider the simple case in which there are only two categories (e.g. good vs. bad) and no veto. In such a case, an object is assigned to the category "good" if it is better than the lower limit profile of this category on a *sufficient coalition of criteria*. How do they define the "sufficient coalitions of criteria"? Basically, these can be any collection of criteria with the following property: a coalition that contains a sufficient coalition of criteria is itself sufficient.

We claimed that MR-Sort is a particular case of a NCS model. Indeed, with MR-Sort, a set of criteria is a sufficient coalition iff the sum of the weights of the criteria in the set is at least as large as the threshold λ. To fix the ideas consider the following example. A student has to take 4 exams to be admitted in a school. To be successful, he has to take a mark of at least twelve (over twenty) in each of these exams, with at most one exception. In this case the lower limit profile of the category "succeed" is the vector $(12, 12, 12, 12)$ 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 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: x succeeds iff $|\{i : x_i \geq 12\}| \times 1/4 \geq 3/4$, which is indeed the typical form of a MR-Sort rule.

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 want 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 $\lambda > 1/2 \sum_{i=1}^{4} w_i$.

Our goal with this paper is to investigate the gap of expressivity between MR-Sort and 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 an inequality involving weights attached to the criteria, as in MR-Sort. We partition the set of all families of sufficient coalitions according to the type of inequality they fulfill. All 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 practical applications. As a by-product, it also 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 on the basis of assignment examples.

The rest of the paper is organized as follows. In Section 2, we state the problem more formally, we introduce the notion of capacity 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 and a short conclusion follows.

2 Background

2.1 Numerical representation of the sufficient coalitions

In MR-Sort, the set of sufficient coalitions of criteria can be represented numerically. In other words it is possible to check whether a set of criteria is sufficient by checking whether an inequality is satisfied. More precisely, there is a family of nonnegative weights $w_1, w_2, \ldots, w_n$ and a nonnegative threshold λ such that a set of criteria $A \subseteq \{1, 2, \ldots, n\}$ is sufficient iff

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

We assume w.l.o.g. that $\sum_{i=1}^{n} w_i = 1$. Such a representation is generally not unique. For instance, in the example above involving 4 criteria, the family of sufficient coalitions is formed by all subsets of at least 3 criteria; this family can be represented by assigning equal weights to all criteria and using threshold value 3/4. Alternatively, one could use e.g. $w_1 = .2, w_2 = .2, w_3 = .3, w_4 = .3$ as weights and $\lambda = .70$ as threshold to represent the same family of coalitions.

We saw also above that, in general, not all families of sufficient coalitions can be specified by an inequality such as (1). If this is not the case, is there another kind of inequality that can be used? Actually, any family of sufficient coalitions can be represented using a *capacity* μ and a threshold λ. We briefly recall what is a capacity. A capacity is a set function $\mu : 2^n \to \mathbb{R}_+$ which is monotone w.r.t. to set inclusion, i.e. for all $A, B \subseteq \{1, 2, \ldots, n\}, A \subseteq B \Rightarrow \mu(A) \leq \mu(B)$ (monotonicity) and $\mu(\emptyset) = 0$. We impose w.l.o.g. that $\mu(\{1, 2, \ldots, n\}) = 1$ (normalization). Note that a capacity is not additive, in general, which means that it does not necessarily satisfy the property: $\mu(A \cup B) = \mu(A) + \mu(B)$ whenever $A \cap B = \emptyset$. If it does, then the capacity μ is said to be additive and it is a probability. This means that there are weights $w_1, w_2, \ldots, w_n$ such that $\mu(A) = \sum_{i \in A} w_i$, for all set $A \subseteq \{1, 2, \ldots, n\}$. A (non necessarily additive) capacity can be given by means of an interaction function (or Möbius transform) m. One has, for all $A \subseteq \{1, 2, \ldots, n\}$:

$$\mu(A) = \sum_{B \subseteq A} m(B) \tag{2}$$

where m is a set function $2^n \to \mathbb{R}$ which satisfies $\sum_{B \subseteq \{1,2,\ldots,n\}} m(B) = 1$ and $\sum_{B: i \in B \subseteq A} m(B) \geq 0$, for all $i \in \{1, 2, \ldots, n\}$ and $A \subseteq \{1, 2, \ldots, n\}$. The capacity defined by (2) is a probability iff $m(B) = 0$ whenever $|B| > 1$. A capacity is said to be k-additive when k is the largest cardinality of the subsets for which m is different from 0. Probabilities are 1-additive (or simply "additive") capacities.

Proposition 1 *Any family of sufficient coalitions can be represented as the set of subsets $A \subseteq \{1, 2, \ldots, n\}$ verifying*

$$\mu(A) \geq \lambda, \tag{3}$$

for some capacity μ and threshold $\lambda \geq 0$. Conversely, if μ is a capacity and λ is a nonnegative number, the set of subsets A satisfying the inequality $\mu(A) \geq \lambda$ is a family of sufficient coalitions.

Proof. A family of sufficient coalitions is a family of subsets such that any subset containing a subset of the family is itself in the family. Define a nonnegative set function μ letting $\mu(A) = 1$ if A is a sufficient coalition and 0 otherwise. One can see that μ is monotone, and therefore a capacity, due to the characteristic properties of the families of sufficient coalitions. It is also normalized. Define the threshold $\lambda = .5$. Clearly $\mu(A) \geq .5$ iff A is a sufficient coalition. The proof of the converse is also straightforward.

As a consequence of this result, in a NCS model, the set of sufficient coalitions can be either listed or specified by an inequality such as (3). In a preference learning perspective, the latter representation may be at an advantage since it opens the perspective of using powerful optimization techniques (see [13] for the learning of a NCS model on this basis)[5]. As already observed in the case of weights, the capacity and threshold used for representing a family of SC are generally not unique.

In the sequel we will be interested in parsimonious representations, i.e. representations of a family of SC as the set of coalitions A satisfying (3), using a k-additive capacity, with k as small as possible. The smaller k, the smaller the number of

[5] In [13], the NCS model without veto is called *capacitive MR-Sort model*. Both models are essentially equivalent

parameters to identify capacity μ, for instance in a learning process. If $k = 1$, the family of SC 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 SC is representable using a 2-additive capacity, then we can write $\mu(A) = \sum_{i \in A} m_i + \sum_{i,j \in A, i \neq j} m_{ij}$, 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.

2.2 Minimal sufficient coalitions

The set of SC may be large (typically exponential in n), but one can avoid listing them all. A *minimal sufficient coalition* (MSC) is a SC which is not properly included in another SC. Knowing the set of MSC allows to determine all SC since a coalition is sufficient as soon as it contains a MSC. A family of MSC is any collection of subsets of $\{1, 2, \ldots, n\}$ such that none of them is included in another. In other words, a set of MSC is an *antichain* in the set of subsets of $\{1, 2, \ldots, n\}$ (partially) ordered by inclusion. It is well-known that the number of antichains in the power set of $\{1, 2, \ldots, n\}$ is $D(n)$, the nth Dedekind number ([15], 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 Table 1.

n	$D(n)$
0	2
1	3
2	6
3	20
4	168
5	7581
6	7828354
7	2414682040998
8	56130437228687557907788

Table 1. Known values of the Dedekind numbers $D(n)$

Remark. The Dedekind numbers are also the number of monotone (more precisely, positive [5]) Boolean functions in n variables. Clearly, the set of sufficient coalitions can be represented as the set of n-dimensional Boolean vectors which give the value 1 to a monotone Boolean function, and conversely. Another application of the Dedekind numbers is in game theory. They are the numbers of simple games with n players in minimal winning form [16, 6].

One way of simplifying the study of the families of sufficient coalitions consists in keeping only one representative of each class of equivalent families of SC. 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 criteria. Consider e.g. the following family of minimal SC for $n = 4$: $\{2, 4\}, \{2, 3\}, \{1, 3, 4\}$. It consists of 2 subsets of 2 criteria and one of 3 criteria. There are 12 equivalent families that can be obtained from this one, by permuting the criteria labels (the criterion which does not show up in the set of 3 criteria can be chosen in 4 different ways and the two criteria which distinguish the two pairs can be chosen in 3 different ways). The number $R(n)$ of *inequivalent* families of SC is known for $n = 0$ to $n = 7$ ([15], sequence A003182). $R(7)$ was only recently computed by Stephen and Yusun [14]. Table 2 lists the known values of $R(n)$.

n	0	1	2	3	4	5	6	7
$R(n)$	2	3	5	10	30	210	16353	490013148

Table 2. Number of inequivalent families of sufficient coalitions of n criteria

Finally we recall Sperner's theorem ([4], p.116-118), a result that will be useful in the process of generating all possible families of SC. 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 SC.

3 Listing the families of minimal sufficient coalitions

For generating all families of MSC and selecting a representative of each class of equivalent families, we follow a strategy similar to the one used in [14]. We describe it briefly. The families of MSC can be partitioned according to their *type* (called "profile" in [14]). The type of a family of MSC is an integer vector $(k_1, k_2, \ldots, k_n)$, where k_i represents the number of coalitions of i criteria 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 criteria and one of 3 criteria. 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 $\{1, 2, \ldots, n\}$ 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 MSC for each type.

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

Remarks:

1. there exist two additional families which do not appear in Table 3:
   - the empty family, corresponding to the case in which no coalition is sufficient, which means, for a sorting procedure, that all objects are assigned to the "bad" category;
   - the family of which the sole element is the empty set; this means that all coalitions are sufficient, even the empty

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 3. Number of inequivalent families of minimal sufficient coalitions

one, and consequently, all objects are sorted in the "good" category.

Adding these two extreme cases to the counts in the last line of Table 3 yields values that are consistent with Tables 2 and 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.

3. in the sequel, in the absence of ambiguity, we shall 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 SC; for instance, the first family of type (0,3,0,0) above will be denoted by : $\{13, 12, 34\}$ instead of $\{\{1, 3\}, \{1, 2\}, \{3, 4\}\}$.

The algorithm sketched above can be made more efficient by implementing the following properties (see [14], lemma 2.4 for a proof) linking the families of MSC.

1. There is a one-to-one correspondence between families consisting exclusively of k_i MSC of cardinality i and families consisting exclusively of $\binom{n}{i} - k_i$ MSC of cardinality i. In other terms, there is a bijection between the families of the type $(0, \ldots, 0, k_i, 0, \ldots, 0)$ and these of the type $(0, \ldots, 0, \binom{n}{i} - k_i, 0, \ldots, 0)$. For instance, in Table 3, generating family $\{12\}$ of type (0,1,0), automatically yields family $\{13, 23\}$ of type (0,2,0). The number of representatives in both types are identical (three, in the latter example).
2. If a family of MSC on n criteria contains at least one singleton, then the remaining MSC of the family do not involve this criterion and hence belong to a type of family of MSC on $n - 1$ criteria. In the example of $n = 3$, knowing the families of MSC on 2 criteria allows to generate the families on three criteria for which one criterion alone is a sufficient coalition. For instance, if criterion 1 alone is sufficient, one can build all families in which 1 is a MSC by putting together with 1 each family of MSC on criteria 2 and 3, i.e.: {}, {2},{3},$\{2, 3\}$ and $\{23\}$. This, however, will not allow to directly compute the number of representatives of each type, since some families, involving more than one singleton as MSC, can be generated in several ways. For instance, $\{1, 2\}$ will be obtained both when starting from the singleton 1 as a MSC and completing this family by MSC included in $\{2, 3\}$, and, starting from the singleton 2 and completing this family by MSC extracted from $\{1, 3\}$.
3. There is a one-to-one correspondence between families of MSC belonging to type $(k_1, k_2, \ldots, k_{n-1}, 0)$ and these belonging to the "reverse" type $(k_{n-1}, \ldots, k_2, k_1, 0)$. For instance, starting from the family $\{1, 2\}$ belonging to type (2,0,0) and taking the complement of each MSC, one obtains the family $\{23, 13\}$, which belongs to (0,2,0). This correspondence allows to halve the computations for $D(n)$ and $R(n)$.

Using this algorithm on a cluster, we have computed the list of all inequivalent families of MSC 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 A.

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 MSC, for fixed n, in categories $\mathcal{C}_k$, which are defined as follows.

Definition 1 *A family of sufficient coalitions belongs to class $\mathcal{C}_k$ if*

1. *it is the set of all subsets A of $\{1, 2, \ldots, n\}$ satisfying an inequality of the type: $\mu(A) \geq \lambda$, where μ is a normalized k-additive capacity and λ a non-negative real number;*
2. *k is the smallest integer for which such an inequality is valid.*

It is clear that all equivalent families of MSC belong to the same class $\mathcal{C}_k$. Hence it is sufficient to check for *one* representative of each class of equivalent families of MSC whether or not it belongs to $\mathcal{C}_k$.

The checking strategy is the following. For each inequivalent family of MSC (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 (we know, by proposition 1, that this will occur at the latest for $k = n$). The test can be formulated as a linear program. Basically, 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 , which will be called *insufficient* coalitions. It is enough to write these sorts of constraints only for the minimal sufficient coalitions and for the maximal insufficient coalitions. The set of minimal sufficient (resp. maximal insufficient) coalitions will be denoted SCMin (resp. SIMax).

To formulate the problem as a linear program, we use formula (2), 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 MSC belongs to class $\mathcal{C}_k$, we set the values of the variables $m(B)$ to 0 for all sets B of cardinality superior to 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 $\mathcal{C}_k$:

$$\begin{cases} \max\ \varepsilon & & & \\ \mu(A) & \geq & \lambda & \forall A \in \text{SCMin} \\ \mu(A) & \leq & \lambda - \varepsilon & \forall A \in \text{SIMax} \\ \mu(A) & = & \sum_{B \subseteq A} m(B) & \forall A \in \text{SCMin} \cup \text{SIMax} \\ \sum_{B: i \in B \subseteq A} m(B) & \geq & 0 & \forall i \in \{1,2,\dots,n\} \\ & & \text{and} & \forall A \subseteq \{1,2,\dots,n\} \\ \sum_{B \subseteq \{1,2,\dots,n\}} m(B) & = & 1 & \\ \lambda, \varepsilon & \geq & 0 & \end{cases} \quad (4)$$

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
  - . $\mu(A) = \sum_{i \in A} m_i,\ \forall A \in \text{SCMin} \cup \text{SIMax}$
  - . $m_i \geq 0,\ \forall i \in \{1, 2, \dots, n\}$
  - . $\sum_{i \in \{1,2,\dots,n\}} m_i = 1$,

  where m_i stands for $m(\{i\})$
- $k = 2$: 2-additive capacity
  - . $\mu(A) = \sum_{i \in A} m_i + \sum_{i,j \in A,\ i \neq j} m_{ij},\ \forall A \in \text{SCMin} \cup \text{SIMax}$
  - . $m_i + \sum_{j \in A,\ j \neq i} m_{ij} \geq 0,\ \forall i \in \{1, 2, \dots, n\}$ and $\forall A \ni i,\ A \subseteq \{1, 2, \dots, n\}$
  - . $\sum_{i \in \{1,2,\dots,n\}} m_i + \sum_{i,j \in \{1,2,\dots,n\},\ i \neq j} m_{ij} = 1$,

  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 criteria (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 MSC, we checked whether this family belongs to $\mathcal{C}_k$, starting with $k = 1$ and incrementing its value until the check is positive. The results are presented in Table 4 for the number and proportion of inequivalent families in classes $\mathcal{C}_2$ and $\mathcal{C}_3$. The families that are not in these classes belong to class $\mathcal{C}_1$. 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 5 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 MSC for belonging to class $\mathcal{C}_1$, $\mathcal{C}_2$ or $\mathcal{C}_3$.

n	$R(n)$	$\mathcal{C}_2$	$\mathcal{C}_3$
0	2	0 (00.00 %)	0 (00.00 %)
1	3	0 (00.00 %)	0 (00.00 %)
2	5	0 (00.00 %)	0 (00.00 %)
3	10	0 (00.00 %)	0 (00.00 %)
4	30	3 (10.00 %)	0 (00.00 %)
5	210	91 (43.33 %)	0 (00.00 %)
6	16 353	15 240 (93.19 %)	338 (02.07 %)

Table 4. Number and proportion of inequivalent families of MSC that are representable by a 2- or 3-additive capacity

n	$D(n)$	$\mathcal{C}_2$	$\mathcal{C}_3$
0	2	0 (00.00 %)	0 (00.00 %)
1	3	0 (00.00 %)	0 (00.00 %)
2	6	0 (00.00 %)	0 (00.00 %)
3	20	0 (00.00 %)	0 (00.00 %)
4	168	18 (10.71 %)	0 (00.00 %)
5	7 581	4 294 (56.64 %)	0 (00.00 %)
6	7 828 354	7 584 196 (96.88 %)	145 502 (01.86 %)

Table 5. Number and proportion of all families of MSC that are representable by a 2- or 3-additive capacity

The information displayed in Table 4 (resp. 5) is represented in graphical form in Figure 1 (resp. 2). The cases of 0, 1 and 2 criteria are not represented since all families can be represented by a 1-additive capacity. 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 of criteria. In contrast, the proportion of families that can be represented by a 2-additive capacity grows up to more than 93% 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 5 and Figure 2).
Examples. As a matter of illustration, we describe a few examples for $n = 4$ and $n = 6$. The list of all inequivalent MSC for $n = 5$, which are not representable by a 1-additive capacity, is 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 MSC on 4 criteria that cannot be represented using a 1-additive capacity (they can be by a 2-additive capacity).

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

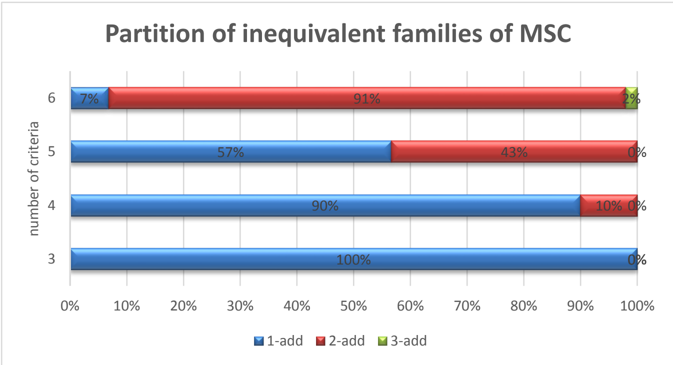

Figure 1. Proportion of inequivalent families of MSC in classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$

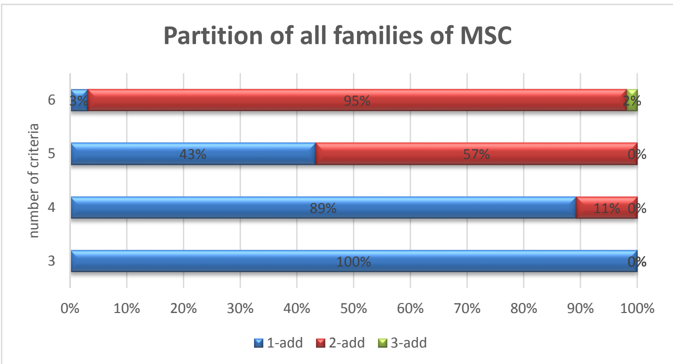

Figure 2. Proportion of all families of MSC in classes $\mathcal{C}_1$, $\mathcal{C}_2$, $\mathcal{C}_3$

These three inequivalent families yield, by permutations of the criteria 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 SC 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. This representation is by no means unique. We construct another capacity 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 3.

Note also that the first and the last example are *complementary* in the sense of the first property allowing to speed up the enumeration of the families of MSC described at the end of Section 3. Both these families are composed of pairs of criteria; the two pairs in the first family are disjoint from the four in the third family and all pairs are either in one or the other family. In such a situation, it is clear that both families belong to the same class $\mathcal{C}_k$.

2. Here are two examples of inequivalent families of MSC on 6 criteria 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[6]. A simple example is of the type (0,0,4,0,0,0). The MSC are $\{136, 234, 125, 456\}$. There are 30 equivalent families that can be derived from this family by permutation. Another, much more complex example is of the type (0,1,7,1,0,0). The MSC are $\{135, 256, 345, 36, 234, 456, 1245, 146, 123\}$. There are 360 families that are equivalent to this one through permutations.

   In the 338 families, no MSC consists of a single criterion; none of them involves 5 criteria. The largest number of MSC in a family is 16, the maximal cardinality of a family of MSC on 6 criteria being the Sperner number 20.

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

5 Usefulness of this analysis

5.1 Applications

The above results, although limited to 6 criteria, maybe useful for different purposes, mainly related to the choice of a sorting model and to simulation.

5.1.1 *Choice of a sorting model*

In the introduction, we argued that the MR-Sort model might not be sufficiently flexible to accommodate certain assignment rules of interest. The quick decrease with n (illustrated by Figures 1 and 2) of the proportion of rules that can be represented by an inequality comparing a sum of weights to a threshold (corresponding to families of MSC in class $\mathcal{C}_1$) shows that it may indeed be useful to consider more general rules. For $n = 4$, only 18 rules in a total of 168 cannot be represented by a 1-additive capacity. For 5 criteria, there is no need to consider more complicated models than these using a 2-additive capacity. And for $n = 6$, in most of the cases (93% in terms of inequivalent families of MSC and more than 96% if we consider all families of MSC), a 2-additive capacity is enough. These considerations are important in the case one wants to learn a Capacitive-MR-Sort model (i.e. a NCS model without veto) as in [13]. Knowing the minimal value of k enabling to represent the set of MSC on n criteria allows to limit the number of parameters (the interaction function m) that have to be elicited or learned on the basis of examples.

Obviously, in many applications, the number of criteria may exceed 6 and it would therefore be useful to extend the analysis for $n > 6$. Using the same methods as we did, it could be possible to solve the case $n = 7$. But from $n = 8$ on, methods based on enumeration become impracticable: the number $R(8)$ of inequivalent families of MSC is not even known. Alternative approaches would consist in trying to find bounds on the cardinal of the classes $\mathcal{C}_k$ or to obtain characterizations of the families in the different classes and use these to generate examples, whenever they exist, in the various categories.

5.1.2 *Simulation*

Recently, methods have been proposed to learn variants of the ELECTRE TRI sorting model on the basis of assignment examples [7, 18, 12, 13]. It has also been done [8] for a ranking method based on *reference points* proposed by Rolland [9, 3]. Consider e.g. a learning algorithm designed to learn a MR-Sort model, as in [12]. 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 would like to verify that 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 [11].

In order to perform the same type of tests in the case of the Capacitive-MR-Sort model (or the NCS model without veto), we are facing a difficulty. How can one draw at random a capacity, 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), a MR-Sort rule or a Capacitive-MR-Sort rule, not a capacity. And this makes a difference, since the representation of a Capacitive-MR-Sort rule by an inequality involving a capacity and a threshold is not unique (as observed previously), hence there is a representation bias in this way of proceeding. 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 MSC. 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 MSC 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). To draw a rule uniformly at random from the set of all Capacitive-MR-Sort rules on n criteria (for $n \leq 6$), proceed as follows:

1. prepare a file in which all inequivalent families of MSC on *criteria* are listed together with the size of their equivalence class; let y_l denote the lth family and s_l the size of its equivalence class, for $l = 1, \dots R(n)$;
2. scan this list and sequentially assign to each family y_l an interval of s_l consecutive integer numbers: y_l is assigned the interval $[N_l, N_l + s_l - 1]$, where $N_l = \sum_{j=1}^{l-1} s_j + 1$;
3. draw uniformly at random an integer number N between 1 and $N_{R(n)}$;
4. find l such that N belongs to the interval $[N_l, N_l + s_l - 1]$ and retrieve the representative of the family of MSC that occupies the lth position in the list.

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 Capacitive-MR-Sort. 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 have at disposal and make generally available a detailed picture of the possible families of sufficient coalitions for as large as possible numbers of criteria; this information could help further investigations related in particular to the characterization of the families of sufficient coalitions that

can be separated from the insufficient ones by an inequality involving a k-additive capacity.
2. to have at disposal and make generally 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, at least somewhat, by retaining only inequivalent rules.

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

REFERENCES

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

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

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

[4] Caspard, N., Leclerc, B., Monjardet, B.: Finite Ordered Sets : Concepts, results and uses. Encyclopedia of Mathematics and its applications, Cambridge University Press (2012)

[5] Crama, Y., Hammer, P.L.: Boolean Functions - Theory, Algorithms, and Applications, Encyclopedia of mathematics and its applications, vol. 142. Cambridge University Press (2011)

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

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

[8] Liu, J., Mousseau, V., Ouerdane, W.: 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). pp. 120–132. Stockholm (Sweden) (2013)

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

[10] Roy, B., Bouyssou, D.: Aide multicritère à la décision: méthodes et cas. Economica Paris (1993)

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

[12] Sobrie, O., Mousseau, V., Pirlot, M.: Learning a majority rule model from large sets of assignment examples. In: Perny, P., Pirlot, M., Tsoukiás, A. (eds.) Algorithmic Decision Theory. pp. 336–350. Springer (2013)

[13] Sobrie, O., Mousseau, V., Pirlot, M.: Learning the parameters of a majority rule sorting model taking attribute interactions into account. Working paper, University of Mons, Belgium and Ecole Centrale Paris, France (2014), submitted for presentation at DA2PL 2014 Workshop, November 20-21, 2014, Paris, France

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

[15] The OEIS Foundation Inc.: On-line encyclopedia of integer sequences. `https://oeis.org`, [Online; accessed September 13, 2014]

[16] von Neumann, J., Morgenstern, O.: Theory of Games and Economic Behavior. Princeton University Press, NJ, USA, 3rd edn. (1972)

[17] Yu, W.: Aide multicritère à la décision dans le cadre de la problématique du tri: méthodes et applications. Ph.D. thesis, LAMSADE, Université Paris Dauphine, Paris (1992)

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

Appendix

A List of inequivalent families of MSC for $n = 4$

The families are grouped by type. There are 25 possible types, 29 inequivalent families of MSC (plus the trivial case in which all coalitions are sufficient) and 167 families of MSC (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 MSC	# 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	

B List of inequivalent families of MSC of class $\mathcal{C}_2$ for $n = 5$

We list below the 91 inequivalent families of MSC 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 MSC	# 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
(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

Type	Family of MSC	# eq.
(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
(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