



**HAL**  
open science

## Using the Sugeno Integral in Optimal Assignment Problems with Qualitative Utilities

Soufiane Drissi Oudghiri, Patrice Perny, Olivier Spanjaard, Mohamed Hachimi

► **To cite this version:**

Soufiane Drissi Oudghiri, Patrice Perny, Olivier Spanjaard, Mohamed Hachimi. Using the Sugeno Integral in Optimal Assignment Problems with Qualitative Utilities. European Conference on Artificial Intelligence, Aug 2016, The Hague, Netherlands. hal-01355115

**HAL Id: hal-01355115**

**<https://hal.sorbonne-universite.fr/hal-01355115>**

Submitted on 22 Aug 2016

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Using the Sugeno Integral in Optimal Assignment Problems with Qualitative Utilities

Soufiane Drissi Oudghiri<sup>1</sup> and Patrice Perny<sup>2</sup> and Olivier Spanjaard<sup>2</sup> and Mohamed Hachimi<sup>1</sup>

**Abstract.** This paper is devoted to the assignment problem when the preferences of the agents are defined by qualitative utilities. In this setting, it is not possible to compare assignments by summing up individual utilities because the sum operation becomes meaningless. We study here the optimization of a Sugeno integral of the individual utilities. We show that the problem is NP-hard in the general case, but we also identify special cases that are solvable in polynomial time. Furthermore, we provide a mixed integer programming formulation in the general case, which leads to a compact formulation for  $k$ -minitive capacities.

## 1 INTRODUCTION

Assignment problems appear in many AI applications, in various contexts such as task allocation in robot teams [22], auctions with side constraints [29], heuristic search [21]. Furthermore, the assignment problem is of interest in many problems involving agents expressing preferences, such as automated resource allocation, matching people with services, time-slot allocation. The standard assignment problem can be defined as follows: given a set  $N = \{a_1, \dots, a_n\}$  of agents and a set  $T = \{t_1, \dots, t_n\}$  of items, an  $n \times n$  utility matrix  $U = (u_{ij})$ , where  $u_{ij}$  represents the utility of item  $t_j$  for agent  $a_i$ , we want to determine a one-to-one assignment maximizing the sum of individual satisfactions. Formally, denoting by  $x = (x_1, \dots, x_n)$  an assignment, where  $x_i = j$  if item  $t_j$  is assigned to agent  $a_i$ , the utility of  $x$  for agent  $a_i$  is  $u_i(x) = u_{ix_i}$  and we want to maximize  $\sum_{i=1}^n u_i(x)$  (utilitarian criterion). It is well-known that the determination of such an optimal assignment can be performed in polynomial time (e.g., by transforming the problem to a minimization problem and using the Hungarian method [23]).

However the use of the utilitarian criterion requires utilities to be evaluated on a cardinal scale. Cardinal utilities rely on the assumption that the magnitude of increments to satisfaction can be compared across different situations [27]. A cardinal utility scale is a utility index that preserves preference orderings uniquely up to positive affine transformations. Nevertheless, in many situations, we deal with data representing people's opinions, interests, preferences, etc. In this case, data are often ordinal, and therefore should be handled in a *suitable* way unless we have some quite strong arguments making us believe that it can be handled at a higher level (namely as a cardinal scale). As emphasized by Roberts [28], using numerical representations of preference orders may lead people to derive meaningless conclusions when preference aggregation is based on usual

arithmetic operations, even if the utilities  $u_{ij}$  take value in a valuation scale  $\mathbb{L}$  that is linearly ordered according to a binary relation  $\succeq$ . This is illustrated in the following example.

**Example 1** Consider an assignment problem where  $|N| = |T| = 4$  and the utilities  $u_{ij}$  are encoded on an ordinal scale  $\mathbb{L} = \{++, +, 0, -, --\}$ . The scale is endowed with the following linear order:  $++ \succ + \succ 0 \succ - \succ --$ . The utility matrix is:

$$\begin{matrix} & t_1 & t_2 & t_3 & t_4 \\ \begin{matrix} a_1 \\ a_2 \\ a_3 \\ a_4 \end{matrix} & \begin{pmatrix} ++ & - & 0 & - \\ - & + & 0 & + \\ ++ & -- & + & 0 \\ + & -- & - & - \end{pmatrix} \end{matrix}$$

Now, assume that the initial scale is transformed into a numerical scale, using an arbitrary monotonic transformation function  $\phi : \mathbb{L} \rightarrow \mathbb{Z}^+$ . Consider two possible transformation functions  $\phi$  and  $\phi'$  defined by:

$u_{ij}$	++	+	0	-	--
$\phi(u_{ij})$	3	2	1	-2	-3
$\phi'(u_{ij})$	2	1	0	-1	-2

Note that both resulting numerical scales are compatible with  $\mathbb{L}$ , i.e.  $\lambda \succ \lambda' \Rightarrow \phi(\lambda) > \phi(\lambda')$  for all  $(\lambda, \lambda') \in \mathbb{L}^2$ . Let us consider now the two assignments  $x = (3, 2, 4, 1)$  and  $x' = (1, 2, 3, 4)$ . The associated vectors of individual utilities are  $u(x) = (0, +, 0, +)$  and  $u(x') = (++, +, +, -)$ , where the  $i^{\text{th}}$  component is the utility of agent  $a_i$ . With a solver, it can easily be checked that, when using  $\phi$  for numerical encoding, assignment  $x$  is optimal and  $x'$  is suboptimal. Moreover, when using  $\phi'$  for numerical encoding, we obtain the opposite conclusion since  $x'$  is optimal and  $x$  is suboptimal. Thus, one observes that slight changes in the numerical values lead to very different conclusions. This illustrates the need for algorithms specifically dedicated to assignment problems with qualitative scales.

In order to compare assignments in a qualitative setting, we need to be able to compare utility vectors  $y = (y_1, \dots, y_n)$ , where  $y = u(x)$  and  $x$  is an assignment. Several preference models have been proposed to compare solutions when elements are evaluated on a qualitative scale (for simplicity, we present them in the setting of a qualitative assignment problem):

- Some preference models induce a partial order on the feasible solutions. The most popular model in this class is the pairwise dominance relation defined by:  $y \succsim y'$  if there exists a bijection  $\pi : [n] \rightarrow [n]$  such that  $y_i \succeq y_{\pi(i)}$  for all  $i \in [n]$ , where  $[n] = \{1, \dots, n\}$ . The symmetric (resp. asymmetric) part of  $\succsim$ , denoted by  $\sim$  (resp.  $\succ$ ), is the indifference (resp. preference) relation.

<sup>1</sup> Laboratoire des sciences de l'ingénieur (LabSi), Université Ibn Zohr, BP 32/S, Agadir, Morocco, email: soufiane.drissioudghiri@edu.uiz.ac.ma, m.hachimi@uiz.ac.ma

<sup>2</sup> Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606, 4 place Jussieu 75005 Paris, France, email: name.surname@lip6.fr

In other words, an assignment  $x$  is preferred to  $x'$  if there exists a bijection  $\pi$  from  $A$  to  $A$  such that: every agent  $a_i$  is at least as satisfied with  $x$  as agent  $a_{\pi(i)}$  with  $x'$ , and there exists at least an agent  $a_i$  that is more satisfied with  $x$  than agent  $a_{\pi(i)}$  with  $x'$ . We call this rule *ordinal dominance* in the following. For example, in the previous instance,  $y \succ y'$  for  $y = u(1, 2, 3, 4) = (++, +, +, -)$  and  $y' = u(4, 3, 2, 1) = (-, 0, --, +)$ . To our knowledge, this ordinal dominance relation was introduced by Barte [1], and revisited by Bossong and Schweigert [2, 31]. Several recent works in a qualitative setting use this relation [3, 4, 5].

– Other preference models induce a complete order on the feasible solutions, namely variations of max ordering, leximax and leximin. The max (resp. min) ordering relation consists in ranking feasible assignments according to the agent whose satisfaction is better off (resp. worse off) among all agents. For example, in the previous instance, if one uses the min ordering relation then  $y = u(3, 2, 4, 1) = (0, +, 0, +)$  is preferred to  $y' = u(1, 2, 3, 4) = (++, +, +, -)$  since  $0 \succ -$ . The leximax (resp. leximin) relation is an enrichment of the max (resp. min) relation, that consists in breaking ties by going down the ranking (i.e., if the worst satisfactions are both equal, one compares the second worsts, and so on...). Finally, one can also consider  $k$ -max (resp  $k$ -min) aggregation operators consisting in evaluating a solution according to its  $k^{\text{th}}$  largest (resp. smallest) element (in the sense of the order relation on  $\mathbb{L}$ ).

In this paper, we are interested in the second type of preference models (those inducing a complete order on the solutions). Finding an optimal assignment according to all the above-mentioned models can be performed in polynomial time [7, 15, 19, 32], and sometimes even faster than in the standard cardinal case. Though appealing from the algorithmic viewpoint, these preference models (inducing a complete order on the solutions) are nevertheless quite simple from the normative and descriptive viewpoints.

Before going on with our contribution, let us mention two related works that do not fit with the above classification. The first work deals with assignment problems with ordinal data viewed as social choice problems, typically with assignment functions based on scoring rules [14]. This amounts to consider ranks instead of costs in the standard Hungarian method, which does not raise new algorithmic issues. The second (more recent) work is devoted to a variant of the covering location problems [34] where the objective function is the Sugeno integral. To our knowledge, this is the first attempt to make use of the Sugeno integral in a combinatorial setting. Yet, their motivation is totally different from ours, as they use it to model fuzziness.

We also study here this sophisticated aggregation operator named *Sugeno integral*, that subsums all the previous models and encompasses a much broader set of decision behaviors. Namely, we investigate the optimization of a Sugeno integral of the individual utilities. The paper is organized as follows. In Section 2, we present the Sugeno integral and some special cases of interest. Then, we show that the determination of an optimal assignment in these special cases can be performed in polynomial time (Section 3) while the problem is NP-hard in the general case (Section 4). Finally, we provide a mixed integer programming formulation for the general case, and we briefly discuss the results of preliminary numerical tests (Section 5).

## 2 THE SUGENO INTEGRAL

In order to compare assignments in a qualitative setting, we need to be able to compare utility vectors  $y = (y_1, \dots, y_n)$ , where  $y_i \in \mathbb{L}$  is the utility of agent  $a_i$  (in Example 1,  $y_i$  corresponds to the grade of the task assigned to agent  $a_i$ ) and  $\mathbb{L}$  is any finite qualitative scale.

The Sugeno integral is precisely an aggregation operator for ordinal scales, that enables to combine several qualitative values into a single representative one. This operator has been introduced by Sugeno [33], and has many applications in qualitative decision theory under uncertainty [6, 12] and multicriteria decision making [11, 17].

Before coming to the Sugeno integral (and to ease its presentation), let us previously introduce the *weighted maximum* [9], that can be seen as a qualitative version of weighted sum, where the sum is replaced by operation max, and the product by operation min. Assume that each agent  $a_i$  has an importance weight  $w_i$ , expressed on the same ordinal scale  $\mathbb{L}$  as the utilities. The weighted maximum of a vector  $y \in \mathbb{L}^n$  then reads as follows:  $\max_{i \in [n]} \min\{w_i, y_i\}$ . For instance, for allocating the different tasks involved in the writing of a research consortium proposal, the head of the consortium may accept to receive a low-rated task if it helps increase the overall satisfaction. This can be modelled by stating that the utility of the head is at most –let’s say– “–” whatever task assigned to her. In other words, the only case in which the satisfaction level  $y_i$  of the head is taken into account should be when a task she rated – – is assigned to her. The weighted maximum aggregation operator makes it possible by setting  $w_i = -$  for the head. Importantly, note that the discriminative power of the weighted maximum can be enhanced by using for the weights a scale  $\mathbb{L}'$  such that  $\mathbb{L}' \supseteq \mathbb{L}$ . The possible aggregate values are indeed then in  $\mathbb{L}'$ , whose range can be much larger than the one of  $\mathbb{L}$ .

The Sugeno integral generalizes the weighted maximum by using the notion of *capacity*. In this respect, it is acknowledged as the qualitative counterpart of the Choquet integral [30]. A capacity is a set function  $v : 2^{[n]} \rightarrow \mathbb{L}$  that represents the importance of a coalition of agents in the present setting. Note that, as for the importance weights in the weighted minimum, the utilities and the capacity values are both expressed on scale  $\mathbb{L}$ . Here also, the discriminative power of the Sugeno integral can be significantly enhanced by using a scale  $\mathbb{L}' \supseteq \mathbb{L}$  for the capacity values. For simplicity, the range of the capacity is assumed to be  $\mathbb{L}$  in the sequel of the paper, but all the presented results are still valid if an enriched scale is used for the capacity values. Another way of breaking ties is to use lexicographic refinements of the Sugeno integral [8, 16]. Most of the algorithms proposed in the remainder of the paper can be adapted to handle such refinements.

**Definition 1** Consider the finite set  $[n] = \{1, \dots, n\}$ . A capacity is a set function  $v : 2^{[n]} \rightarrow \mathbb{L}$  such that:

i)  $v(\emptyset) = \perp, v([n]) = \top,$

ii)  $\forall A, B \subseteq [n], A \subseteq B \Rightarrow v(A) \preceq v(B),$

where  $\top$  (resp.  $\perp$ ) denotes the max (resp. min) element in  $\mathbb{L}$  according to  $\succeq$ .

For any subset  $A \subseteq [n]$ ,  $v(A)$  represents the importance of coalition  $A$  of agents. Let us now recall some definitions about capacities.

**Definition 2** Let  $v$  be a capacity on  $[n]$ . We say that  $v$  is:

– *minitive* if for any subsets  $A, B \subseteq [n]$ ,  $v(A \cap B) = v(A) \wedge v(B)$ ;

– *maxitive* if for any subsets  $A, B \subseteq [n]$ ,  $v(A \cup B) = v(A) \vee v(B)$ ;

– *symmetric* if for any subsets  $A, B \subseteq [n]$ ,  $|A| = |B|$  implies  $v(A) = v(B)$ ;

where  $\wedge := \min$  and  $\vee := \max$ .

We are now able to define the Sugeno integral:

**Definition 3** The discrete Sugeno integral of  $y = (y_1, \dots, y_n)$  has

the following equivalent formulas:

$$S_v(y) := \bigvee_{i=1}^n (y_{(i)} \wedge v(A_{(i)})) \quad (1)$$

$$S_v(y) := \bigwedge_{i=1}^n (y_{(i)} \vee v(A_{(i+1)})) \quad (2)$$

where  $(\cdot)$  is a permutation on  $[n]$  such that  $y_{(1)} \preceq \dots \preceq y_{(n)}$ ,  $\vee := \max$ ,  $\wedge := \min$ ,  $A_{(i)} := \{(i), \dots, (n)\}$  and  $A_{(n+1)} := \emptyset$ .

Note that, even if there are several permutations  $\sigma$  such that  $y_{\sigma(1)} \preceq \dots \preceq y_{\sigma(n)}$  (due to possible ex aequos), the value of the Sugeno integral is uniquely defined. In formula 2 above,  $v(A_{(i+1)})$  represents the importance of the coalition of agents that are better off than  $a_i$  in  $y$ . The utility  $y_i$  of agent  $a_i$  will be more taken into account in the evaluation of  $y$  as the importance of this coalition will be low.

**Example 2** Coming back to Example 1, consider assignment  $x = (1, 2, 3, 4)$ . The corresponding utility vector is  $y = (++, +, +, -)$ . Assume that  $v(\{1\}) = 0$ ,  $v(\{1, 2\}) = +$ ,  $v(\{1, 2, 3\}) = ++$ . By definition of a capacity,  $v(\emptyset) = --$  and  $v([4]) = ++$ . The Sugeno integral  $S_v(y)$  of  $y$  reads as follows:

$$\begin{aligned} & \bigvee_{i=1}^4 (y_{(i)} \wedge v(A_{(i)})) \\ &= (y_4 \wedge v([4])) \vee (y_3 \wedge v(\{1, 2, 3\})) \vee (y_2 \wedge v(\{1, 2\})) \vee (y_1 \wedge v(\{1\})) \\ &= (- \wedge ++ ) \vee (+ \wedge ++ ) \vee (+ \wedge + ) \vee (+ \wedge 0) = + \end{aligned}$$

Equivalently, it reads as follows:

$$\begin{aligned} & \bigwedge_{i=1}^4 (y_{(i)} \vee v(A_{(i+1)})) \\ &= (y_4 \vee v(\{1, 2, 3\})) \wedge (y_3 \vee v(\{1, 2\})) \wedge (y_2 \vee v(\{1\})) \wedge (y_1 \vee v(\emptyset)) \\ &= (- \vee ++ ) \wedge (+ \vee + ) \wedge (+ \vee 0) \wedge (+ \vee - - ) = + \end{aligned}$$

A well-known example of Sugeno integral is the so-called h-index, used to measure both the productivity and citation impact of a scientist. If an author has published  $n$  papers, let us define  $(y_1, \dots, y_n)$  as the citation vector of the author, where  $y_i$  is the number of times paper  $i$  is cited. By setting  $v(A_{(i)}) = |A_{(i)}| = n - i + 1$ , the value  $S_v(y)$  is nothing else but the h-index of the author [35].

The Sugeno integral is more expressive than the weighted maximum, which is only a special case obtained for a maxitive capacity  $v$  such that  $v(\{i\}) = w_i$  for all  $i \in [n]$  [18] (in this case,  $v(A) = \max_{i \in A} w_i$ , which makes  $v$  a possibility measure, and weighted max a qualitative expected utility [10]). Actually, Marichal [25] showed that if one restricts to the use of a combination of max and min operators (together with qualitative weights), then the Sugeno integral is the only solution for aggregating qualitative values, provided one adds the natural constraint that the aggregated value for  $(\perp, \dots, \perp)$  should be  $\perp$  and the one for  $(\top, \dots, \top)$  should be  $\top$ .

The Sugeno integral includes other well identified special cases [18]:

- $S_v(y) = \min_i y_i$  iff  $v(A) := \perp$  for all  $A \subsetneq [n]$ ;
- $S_v(y) = \max_i y_i$  iff  $v(A) := \top$  for all  $A \subseteq [n]$ ,  $A \neq \emptyset$ ;
- $S_v(y) = k\text{-min}_i y_i$  iff  $v(A) := \perp$  for  $|A| \leq n - k$  and  $\top$  otherwise;
- $S_v(y)$  is the *ordered weighted maximum* w.r.t.  $w$  defined by  $\text{OWMax}(y) := \bigvee_{i=1}^n (w_i \wedge y_{(i)})$  where  $y_{(1)} \preceq \dots \preceq y_{(n)}$  and  $w_1 \succeq \dots \succeq w_n$ , iff  $v$  is a symmetric capacity such that  $v(A) = w_{n-|A|+1}$  for any  $A \subseteq [n]$ ,  $A \neq \emptyset$ ;

–  $S_v(y)$  is the *ordered weighted minimum* defined by  $\text{OWMin}_w(y) := \bigwedge_{i=1}^n (w_i \vee y_{(i)})$  where  $y_{(1)} \preceq \dots \preceq y_{(n)}$  and  $w_1 \succeq \dots \succeq w_n$ , iff  $v$  is a symmetric capacity such that  $v(A) = w_{n-|A|}$  for any  $A \subsetneq [n]$ .

The ordered weighted minimum encompasses a gradation of aggregation operators ranging from the min operator ( $w_i = \perp$  for all  $i$ ) to the max operator ( $w_n = \perp$  and  $w_i = \top$  for  $i \neq n$ ), including the  $k$ -min operators. Note that the assumption  $w_1 \succeq \dots \succeq w_n$  is not restrictive since  $\bigwedge_{i=1}^n (w_i \vee y_{(i)}) = \bigwedge_{i=1}^n ((\bigwedge_{k=1}^i w_k) \vee y_{(i)})$ .

**Example 3** Coming back to Example 1, assume now that we wish to evaluate utility vector  $y = (++, +, +, -)$  with the ordered weighted minimum and weights  $w_1 = ++$ ,  $w_2 = 0$ ,  $w_3 = -$  and  $w_4 = --$ . The ordered weighted minimum  $\text{OWMin}_w(y)$  is:

$$\begin{aligned} & (w_1 \vee y_{(1)}) \wedge (w_2 \vee y_{(2)}) \wedge (w_3 \vee y_{(3)}) \wedge (w_4 \vee y_{(4)}) \\ &= (w_1 \vee y_4) \wedge (w_2 \vee y_3) \wedge (w_3 \vee y_2) \wedge (w_4 \vee y_1) \\ &= (+ + \vee -) \wedge (0 \vee +) \wedge (- \vee +) \wedge (- - \vee + +) = + \end{aligned}$$

In the next section, we show that an assignment maximizing the  $\text{OWMin}_w$  operator can be determined in polynomial time, as well as one optimizing the weighted minimum.

### 3 POLYNOMIAL CASES

This section is devoted to the presentation of polynomial time solution procedures for the determination of optimal assignments according to special cases of the Sugeno integral.

#### 3.1 Weighted min and weighted max

To warm up, we begin with the weighted minimum and the weighted maximum because it is very simple to show the polynomial time complexity in these cases. The solution procedures are derived from the threshold method for the max min assignment problem.

The max min assignment problem aims at determining an assignment  $x$  that maximizes  $\min_i u_i(x)$ , where  $u_i(x)$  is the (qualitative) utility of agent  $a_i$  in assignment  $x$ . We recall that the threshold algorithm to compute a max min assignment works as follows: it consists in determining  $\max\{\lambda \in \mathbb{L} : \exists x \forall i \ u_i(x) \succeq \lambda\}$ . In order to know if the max min value is greater or equal to a given  $\lambda \in \mathbb{L}$ , one uses a max flow algorithm to compute a maximal matching in the bipartite graph  $G_\lambda = (V_1, V_2, E)$  where  $V_1 = \{a_1, \dots, a_n\}$  (resp.  $V_2 = \{t_1, \dots, t_n\}$ ) is the set of agents (resp. tasks) and there is an edge between  $a_i$  and  $t_j$  if  $u_{ij} \succeq \lambda$ . We recall that a maximal matching is a set of non-adjacent edges of maximal cardinality. It is said to be perfect when the cardinality is  $n$ . The maximal matching is perfect in  $G_\lambda$  iff the value of a max min assignment is at least  $\lambda$ . One can use a bisection search on  $\mathbb{L}$  to speed up the determination of the minimal  $\lambda$  such that a feasible assignment exists. This assignment is then a max min assignment.

The adaptation of this method to compute a maximal assignment according to a weighted minimum  $\bigwedge_{i=1}^n (w_i \vee u_i(x))$  simply consists in replacing  $u_{ij}$  by  $u_{ij} \vee w_i$  in the matrix, and then applying the threshold method used for the max min case. The correctness of this approach is obvious. The generation of the transformed matrix is in  $O(n^2)$ , and then there are at most  $\log |\mathbb{L}|$  calls to a max flow algorithm in  $O(n^3)$  [24]. The overall complexity is therefore  $O(n^3 \log |\mathbb{L}|)$ .

This positive result also holds for the determination of an assignment maximizing the weighted maximum  $\bigvee_{i=1}^n (w_i \wedge u_i(x))$ . Such

an assignment can indeed be simply computed as follows. Let  $i^*, j^*$  denote indices such that  $u_{i^* j^*} = \max_i \max_j (u_{ij} \wedge w_i)$ . Any assignment where task  $t_{j^*}$  is assigned to agent  $a_{i^*}$  is optimal for the weighted maximum.

### 3.2 OWMin and OWMax

The procedure to determine a maximal assignment according to the  $\text{OWMin}_w$  operator is also a threshold method, based on the following proposition:

**Proposition 1** *Let  $y \in \mathbb{L}^n$ . The following property holds:*

$$|\{i \in [n] : y_i \succeq \lambda\}| \geq |\{i \in [n] : w_i \prec \lambda\}| \Leftrightarrow \text{OWMin}_w(y) \succeq \lambda.$$

**Proof** Assume that  $|\{i \in [n] : y_i \succeq \lambda\}| \geq |\{i \in [n] : w_i \prec \lambda\}|$ . By definition of  $\text{OWMin}_w(y)$ , this implies that all values  $y_{(i)} \prec \lambda$  are weighted by  $w_i \succeq \lambda$ . Consequently,  $\text{OWMin}_w(y) \succeq \lambda$ .

Conversely, assume that  $|\{i \in [n] : y_i \succeq \lambda\}| < |\{i \in [n] : w_i \prec \lambda\}|$ . By definition of  $\text{OWMin}_w(y)$ , this implies there exists a rank  $i$  for which  $y_{(i)} \prec \lambda$  and  $w_i \prec \lambda$ . Consequently,  $\text{OWMin}_w(y) \prec \lambda$ . ■

Proposition 1 may be interpreted in the following way in the setting of ordinal assignment problems: the value of the ordered weighted min of an assignment is greater than or equal to  $\lambda$  if and only if there are at least  $g(\lambda)$  agents whose (qualitative) utility is at least  $\lambda$ , where  $g(\lambda) = |\{i \in [n] : w_i \prec \lambda\}|$ . A threshold method can therefore be used to determine  $\min\{\lambda \in \mathbb{L} : \exists x \text{ s.t. } |\{i \in [n] : u_i(x) \succeq \lambda\}| \geq g(\lambda)\}$ . In order to know if the optimal value is greater than or equal to a given  $\lambda \in \mathbb{L}$ , one uses a max flow algorithm to compute a maximal matching in the same bipartite graph  $G_\lambda$  as above, where the set of edges is now defined by  $E = \{\{a_i, t_j\} : u_{ij} \succeq \lambda\}$ . If the cardinality of a maximal matching is greater than or equal to  $g(\lambda)$ , then the optimal value is greater than or equal to  $\lambda$ , otherwise it is the opposite. By using here also a binary search to determine the minimal  $\lambda$ , the complexity of the whole threshold method is of course the same as for the max min case ( $O(n^3 \log |\mathbb{L}|)$ ).

**Example 4** *Coming back to Example 1, assume that we wish to determine an optimal assignment for  $\text{OWMin}_w$  with  $w = (++, 0, -, --)$ . The matrix with the utilities of the agents is recalled on the top of Figure 1. Since we have  $|\{i \in [n] : w_i \prec ++\}| = 3$  and there does not exist a matching of cardinality 3 in  $G_{++}$  (graph in the bottom left corner of Figure 1), the optimal value is necessarily strictly smaller than  $++$ . However, we have  $|\{i \in [n] : w_i \prec +\}| = 3$  and there exists a matching of cardinality 3 in  $G_+$  (see the dotted edges in the bottom right graph of Figure 1). By completing this matching in a complete assignment (by assigning task  $t_4$  to agent  $a_4$  here), one obtains an assignment  $x$  such that  $\text{OWMin}_w(u(x)) = +$ . This assignment is optimal for  $\text{OWMin}_w$ .*

A similar approach can be used for maximizing an  $\text{OWMax}$  operator: the value of the ordered weighted max of an assignment is greater than or equal to  $\lambda$  if and only if there are at least  $g(\lambda) + 1$  agents whose qualitative utility is at least  $\lambda$ .

**Remark 1** *Note that the ordered weighted operators  $f : \mathbb{L}^n \rightarrow \mathbb{L}$  suffer from the drowning effect, i.e. one can have  $f(y) = f(y')$  while  $y' \succ y$  (ordinal dominance). This is due to the fact that all these operators use max and min operations. In order to guarantee that the returned assignment  $x$  is non-dominated (i.e.,  $\forall x'$*

*not( $u(x') \succ u(x)$ )), one can adapt the threshold methods as follows. One replaces the max flow algorithm by a max flow min cost algorithm from  $s$  to  $t$  in the network  $G = (V, E, c)$  with unit capacities, where  $V = \{a_1, \dots, a_n\} \cup \{t_1, \dots, t_n\} \cup \{s, t\}$ ,  $E = \{(s, a_i) : i \in [n]\} \cup \{(a_i, t_j) : i \in [n], j \in [n]\} \cup \{(t_j, t) : j \in [n]\}$  and the costs of edges (with unit capacities) are defined by  $c(a_i, t_j) = \epsilon_\lambda$  (resp.  $1 + \epsilon_\lambda$ ) if  $u_{ij} \succeq \lambda$  (resp.  $u_{ij} \prec \lambda$ ),  $c(s, a_i) = 0$ ,  $c(t_j, t) = 0$ . Furthermore, we assume that  $\epsilon_\lambda < \epsilon_{\lambda'}$  for  $\lambda \succ \lambda'$  and that  $\epsilon_\lambda \ll 1 \forall \lambda \in \mathbb{L}$ . The reader can easily verify that the min cost of a max  $s$ - $t$  flow in  $G$  is strictly smaller than 1 (resp.  $n - g(\lambda) + 1$ ) iff there are at least  $n$  (resp.  $g(\lambda)$ ) edges of costs  $\epsilon_\lambda$  ( $\lambda \in \mathbb{L}$ ) with a flow 1. We claim that the assignment corresponding to a max flow of min cost for the optimal value of  $\lambda$  necessarily is non-dominated. This comes from the fact that  $y' \succ y \Rightarrow \sum_{i \in [n]} \epsilon_{y'_i} < \sum_{i \in [n]} \epsilon_{y_i}$ .*

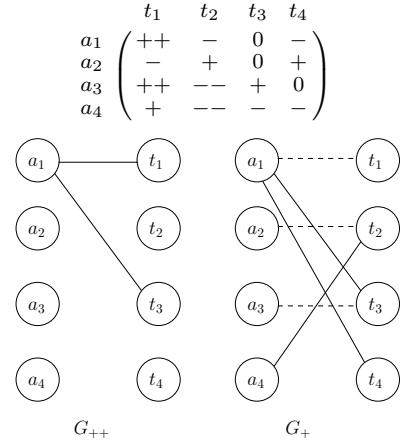


Figure 1: Illustration of Example 4.

### 3.3 Sugeno with $k$ -maxitive capacities

We now show how to extend the polynomiality result for the weighted maximum to a wider class of Sugeno integrals. In order to define this class, let us recall the definition of the Möbius transform of a capacity  $v$  in a qualitative setting and the formulation of the Sugeno integral in terms of ordinal Möbius masses. Note that the Möbius transform is not uniquely defined in the ordinal case [18]. Nevertheless, we use in this paper a specific Möbius transform that we call *canonical ordinal Möbius transform*.

**Definition 4** *The canonical ordinal Möbius transform of a capacity  $v$  is the set function  $m$  on  $[n]$  defined by:*

$$m(A) := \begin{cases} v(A) & \text{if } v(A) \succ v(A \setminus \{i\}), \forall i \in A \\ \perp & \text{otherwise,} \end{cases}$$

Note that if  $v(A) \in \mathbb{L}$  then  $m(A) \in \mathbb{L}$ . Coefficients  $m(A)$ , for  $A \subseteq [n]$ , are called *ordinal Möbius masses*. Conversely, capacity  $v$  can be recovered from  $m$  using the following equation:

$$v(E) = \bigvee_{A \subseteq E} m(A)$$

Interestingly enough, any Sugeno integral can be rewritten using ordinal Möbius masses as follows [18]:

$$S_v(y) = \bigvee_{A \subseteq [n]} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \quad (3)$$

The computation of  $S_v(y)$  with Equation 3 involves  $2^n$  Möbius masses. However only a polynomial number of them must be considered in some particular cases. For example, this is the case for  $k$ -maxitive capacities that can be seen as ordinal counterparts of  $k$ -additive capacities, see [26].

**Definition 5** A capacity  $v$  defined on an ordinal scale  $\mathbb{L}$  endowed with a smallest (resp. greatest) element denoted by  $\perp$  (resp.  $\top$ ) is said to be  $k$ -maxitive if the canonical Möbius transform  $m$  satisfies:

- (i)  $\forall A \subseteq [n], |A| > k \Rightarrow m(A) = \perp$
- (ii)  $\exists A \subseteq [n], |A| = k$  s.t.  $m(A) \neq \perp$

Given a capacity  $v$ , there is no need to explicitly compute all ordinal Möbius masses to decide whether it is  $k$ -maxitive or not. The recognition can be made directly on  $v$  using the following characterization result:

**Proposition 2**  $v$  is a  $k$ -maxitive capacity if and only if  $v(A) = \top \forall A \subseteq [n]$  such that  $|A| \geq k$ .

**Proof** Assume that  $v$  is  $k$ -maxitive. By definition,  $\forall A \subseteq [n], |A| > k \Rightarrow m(A) = \perp$  which implies that  $\forall A \subseteq [n], |A| > k \Rightarrow v(A) = v(A \setminus \{i\}) \forall i \in A$ . Since  $v([n]) = \top$  we obtain the desired result by induction. Conversely, for any  $A$  such that  $|A| > k$  and any  $i \in A$ , we have  $v(A) = v(A \setminus \{i\}) = \top$ . Hence  $m(A) = \perp$  by Definition 4, which completes the proof. ■

Interestingly enough, the Sugeno integral can be reformulated as follows for  $k$ -maxitive capacities:

$$\begin{aligned} S_v(y) &= \bigvee_{A \subseteq [n]} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \\ &= \left( \bigvee_{A \subseteq [n], |A| \leq k} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \right) \vee \left( \bigvee_{A \subseteq [n], |A| > k} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \right) \\ &= \left( \bigvee_{A \subseteq [n], |A| > k} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \right) \vee \left( \bigvee_{A \subseteq [n], |A| > k} \left( \bigwedge_{i \in A} y_i \wedge \perp \right) \right) \\ &= \left( \bigvee_{A \subseteq [n], |A| \leq k} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \right) \vee \left( \bigvee_{A \subseteq [n], |A| > k} \left( \perp \right) \right) \\ &= \bigvee_{A \subseteq [n], |A| \leq k} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \end{aligned}$$

This compact formulation of the Sugeno integral, involving only  $\binom{n}{k}$  coefficients, enables to identify a new polynomial case:

**Theorem 1** The determination of an assignment  $x$  maximizing  $S_v(u(x))$  can be performed in polynomial time if the capacity is  $k$ -maxitive for a fixed  $k$ .

**Proof** Given a feasible assignment  $x$ , the Sugeno value of  $u(x)$  for a  $k$ -maxitive capacity reads:

$$S_v(u(x)) = \bigvee_{A \subseteq [n], |A| \leq k} \left( \bigwedge_{i \in A} u_{ix_i} \wedge m(A) \right)$$

Thus, we easily deduce the following:

$$S_v(u(x)) \succeq \lambda \Leftrightarrow \left[ \exists A \subseteq [n], |A| \leq k \text{ s.t. } \left( \bigwedge_{i \in A} u_{ix_i} \wedge m(A) \right) \succeq \lambda \right]$$

To check if an assignment  $x$  s.t.  $S_v(u(x)) \succeq \lambda$  exists, we propose the following procedure. For every  $A \subseteq [n], |A| \leq k$  s.t.  $m(A) \succeq \lambda$ , compute a maximal matching  $M_\lambda(A)$  in the bipartite graph  $G_\lambda^A = (A, T, E_\lambda)$ , where  $E_\lambda = \{(a_i, t_j) \mid u_{ij} \succeq \lambda\}$ . The procedure can be stopped as soon as a set  $A$  such that  $|M_\lambda(A)| = |A|$  is found. Matching  $M_\lambda$  can then be arbitrarily completed into a feasible assignment, without downgrading the Sugeno value. If no such set  $A$  is found, then  $\max_x S_v(u(x)) \prec \lambda$ . By using a standard bisection algorithm, we can determine the maximal  $\lambda$  such that  $S_v(u(x)) \succeq \lambda$  for some feasible assignment  $x$ . The corresponding assignment  $x$  is, by construction, an optimal assignment. For each value of  $\lambda$ , there are at most  $O(n^k)$  calls to a max flow algorithm in  $O(n^3)$ . The number of examined values for  $\lambda$  is in  $O(\log |\mathbb{L}|)$ . The overall complexity is therefore  $O(n^{3+k} \log |\mathbb{L}|)$ . ■

## 4 COMPLEXITY IN THE GENERAL CASE

In this section we prove that the determination of a Sugeno optimal assignment is an NP-hard problem. We assume that the size function of an instance of the ordinal assignment problem is  $n$ . Consequently, we also assume that the capacity function is not defined in extension (which would take up  $O(2^n)$  space), but rather that the determination of  $v(A)$  for a given  $A \subseteq [n]$  can be carried out in polynomial time in  $n$ . The following result holds:

**Theorem 2** The determination of an assignment  $x$  maximizing  $S_v(u(x))$  is NP-hard, even if  $|\mathbb{L}| = 2$ .

**Proof** The proof relies on a polynomial reduction from the monotone 1-in-3-SAT problem defined as follows:

**Instance:** A collection of clauses  $C = \{C^1, \dots, C^m\}$ , each clause consists of exactly three variables, the set of boolean variables is denoted by  $B = \{b_1, \dots, b_n\}$ , and all literals are positive.

**Question:** Does there exists a truth assignment of boolean variables in  $B$  such that each clause has exactly one true variable ?

From an instance of 1-in-3-SAT, we define an instance of the ordinal assignment problem as follows:

- the set of agents is  $N = \{b_{i_1}^1, b_{j_1}^1, b_{k_1}^1, \dots, b_{i_m}^m, b_{j_m}^m, b_{k_m}^m\}$  where  $i_p$  (resp.  $j_p, k_p$ ) denotes the index of the first (resp. second, third) variable in  $C^p$ ;
- the set of tasks is  $T = \{c_1, \dots, c_m\} \cup \{f_1, \dots, f_{2m}\}$  where task  $c_p$  represents clause  $C^p$  and tasks  $f_1, \dots, f_{2m}$  are added to have  $|N| = |T| = 3m$ ;
- the utility of assigning task  $c_p$  to  $b_p^p$  ( $\forall p \in [m]$ ) is  $\top$ , otherwise it is  $\perp$ .

– the capacity  $v$  in the Sugeno integral is:

$$v(A) = \begin{cases} \top & \text{if } |A| > m \\ \top & \text{if } |A| = m \text{ and } \forall i \in [m], A \cap N_i \neq \emptyset \Rightarrow N_i \subseteq A \\ \perp & \text{if } |A| = m \text{ and } \text{not}(\forall i \in [m], A \cap N_i \neq \emptyset \Rightarrow N_i \subseteq A) \\ \perp & \text{if } |A| < m \end{cases}$$

where  $N_i = \{b_p^i : p \in [m] \text{ and } b_p^i \in N\}$  is the set of occurrences of variable  $b_i$  in  $N$  (it is easy to check that the capacity is well defined). For illustration, the matrix obtained from formula  $(b_1 \vee b_2 \vee b_3) \wedge$

$(b_1 \vee b_2 \vee b_5) \wedge (b_2 \vee b_3 \vee b_4)$  is (all empty components are  $\perp$ ):

$$\begin{array}{c} b_1^1 \\ b_2^1 \\ b_3^1 \\ b_1^2 \\ b_2^2 \\ b_3^2 \\ b_1^3 \\ b_2^3 \\ b_3^3 \\ b_4^3 \end{array} \begin{pmatrix} c_1 & c_2 & c_3 & f_1 & f_2 & f_3 & f_4 & f_5 & f_6 \\ \top & & & & & & & & \\ \top & & & & & & & & \\ \top & & & & & & & & \\ & \top & & & & & & & \\ & \top & & & & & & & \\ & \top & & & & & & & \\ & & \top & & & & & & \\ & & \top & & & & & & \\ & & \top & & & & & & \\ & & & \top & & & & & \\ & & & \top & & & & & \\ & & & \top & & & & & \end{pmatrix}.$$

We claim that the value of an optimal assignment w.r.t.  $S_v$  is  $\top$  iff the answer to the 1-in-3-SAT instance is yes.

First, assume that the value of an optimal assignment  $x$  w.r.t.  $S_v$  is  $S_v(u(x)) = \top$ . Let  $y = u(x)$  denote the vector of individual utilities ( $3m$  components) for  $x$ . We recall that our objective function is  $S_v(y) = \bigwedge_{i=1}^{3m} (y_{(i)} \vee v(A_{(i+1)}))$ . By construction of the instance of the ordinal assignment problem, there are at most  $m$  components of  $y$  that are equal to  $\top$ . By definition of  $v$ ,  $v(A_{(i+1)}) = \perp$  for  $i = 2m + 1, \dots, 3m$  (because  $|A_{(i+1)}| < m$ ). Consequently, we necessarily have  $y_{(i)} = \top$  for  $i = 2m + 1, \dots, 3m$  (otherwise  $S_v(y) \neq \top$ ). By looking at  $y_{(2m+1)}, \dots, y_{(3m)}$ , for each clause we have therefore one variable (belonging to the clause) assigned to it. By setting these variables to true and the other variables to false, one obtains a feasible truth assignment for the 1-in-3-SAT instance if the following property holds: if a variable belongs to several clauses, if it is assigned to one of them, then it is assigned to all of them. This is ensured by the way capacity  $v$  is defined. As shown previously, we have indeed necessarily  $y_{(2m)} = \perp$ , which implies  $v(A_{(2m+1)}) = \top$  (otherwise  $S_v(y) \neq \top$ ). Note that  $|A_{(2m+1)}| = m$ . By definition of  $v$ , it means that  $\forall i \in [n] \ A \cap N_i \neq \emptyset \Rightarrow N_i \subseteq A$ , which is precisely the required property.

Conversely, assume that the answer to the 1-in-3-SAT instance is yes. By assigning  $c_p$  to  $b_i^p$  iff variable  $b_i$  is true in the solution of the 1-in-3-SAT instance, one obtains an assignment  $x$  such that  $S_v(u(x)) = \top$  (this is proved by similar arguments as above). ■

Since a polynomial procedure is unlikely to exist in the general case (unless  $P = NP$ ), we looked for a MIP formulation. This is the topic of the next section.

## 5 A MIP FORMULATION

To model our problem as a mixed integer program (MIP), we need to find a linear reformulation of the Sugeno integral. In this respect, Equations 1 and 2 are not very useful because they are based on a sorting of the components of  $y$ , therefore their linearization would require to introduce binary permutation variables. Equation 3 presents a more interesting form since the ordered aggregation is achieved by an aggregation over all subsets of  $N$ . This type of formulation may appear as intractable as the number of agents increase, but we are going to show that it can be efficiently used to minimize the Sugeno integral provided that the capacity is  $k$ -maxitive. Note that the maximization case has been treated in Section 3 dedicated to the polynomial cases. Let us consider the following minimization problem, that can be of interest when variables  $y_i$  represent disutilities:

$$\begin{aligned} \min \quad & \bigvee_{A \subseteq [n]} \left( \bigwedge_{i \in A} y_i \wedge m(A) \right) \\ \text{s.t.} \quad & y \in Y \end{aligned}$$

where  $Y \subseteq [0, 1]^n$  denotes the set of feasible utility vectors. It is equivalent to the MIP:

$$\begin{aligned} \min \quad & w \\ \text{s.t.} \quad & \begin{cases} w \geq t_A \quad \forall A \subseteq [n] \\ t_A = \sum_{i \in A} b_i^A y_i + b_0^A m(A) \quad \forall A \subseteq [n] \\ \sum_{i \in A} b_i^A + b_0^A = 1 \quad \forall A \subseteq [n] \end{cases} \end{aligned} \quad (4)$$

where  $b_i^A \in [0, 1]$  and  $t_A = \sum_{i \in A} b_i^A y_i + b_0^A m(A)$  represents the value  $\bigwedge_{i \in A} (y_i \wedge m(A))$  at optimum. Note that constraint 5 is quadratic, but the quadratic terms can be removed by introducing product variables  $p_i^A = b_i^A y_i$  and the following constraints:

$$p_i^A \leq b_i^A \quad (6)$$

$$p_i^A \leq y_i \quad (7)$$

$$p_i^A \geq y_i + b_i^A - 1 \quad (8)$$

This usual reformulation through the linearization constraints 6, 7 and 8 is due to Fortet [13, 20]. Every triple of constraints insures that  $b_i^A = 0 \Rightarrow p_i^A = 0$  and  $b_i^A = 1 \Rightarrow p_i^A = y_i$ . Hence we obtain the following mathematical program:

$$\begin{aligned} \min \quad & w \\ \text{s.t.} \quad & \begin{cases} w \geq t_A \quad \forall A \subseteq [n] \\ t_A = \sum_{i \in A} p_i^A + b_0^A m(A) \quad \forall A \subseteq [n] \\ \sum_{i \in A} b_i^A + b_0^A = 1 \quad \forall A \subseteq [n] \\ p_i^A \leq b_i^A \quad \forall A \subseteq [n] \quad \forall i \in A \\ p_i^A \leq y_i \quad \forall A \subseteq [n] \quad \forall i \in A \\ p_i^A \geq y_i + b_i^A - 1 \quad \forall A \subseteq [n] \quad \forall i \in A \end{cases} \end{aligned}$$

This formulation is not compact for general capacities since the number of variables and constraints grows exponentially with  $n$ . However the formulation becomes compact for  $k$ -maxitive capacities. Under the  $k$ -maxitivity assumption indeed, we have  $\bigwedge_{i \in A} (y_i \wedge m(A)) = \perp$  for  $|A| > k$ . The constraints and variables that were needed for all subsets of  $[n]$  become therefore only necessary for all subsets of size not greater than  $k$ . Hence the number of variables and constraints becomes polynomial in  $n$  (assuming that  $k$  is a constant).

Unfortunately the linearization technique presented in the case of a minimization does not apply directly for maximizing the Sugeno integral. To overcome the problem, we now use another formulation of the Sugeno integral based on ordinal co-Möbius masses [18]. We use here the canonical ordinal co-Möbius transform  $\overline{m}$  defined by:

$$\overline{m}(A) := \begin{cases} v([n] \setminus A) & \text{if } v([n] \setminus A) < v(( [n] \setminus A) \cup \{i\}), \forall i \in A \\ \top & \text{otherwise,} \end{cases}$$

Note that, here also,  $\overline{m}(A) \in \mathbb{L}$ . The Sugeno integral can be expressed as a function of  $\overline{m}$  as follows [18]:

$$S_v(y) = \bigwedge_{A \subseteq [n]} \left( \bigvee_{i \in A} y_{(i)} \vee \overline{m}(A) \right) \quad (9)$$

Let us consider now the maximization problem:

$$\begin{aligned} \max \quad & \bigwedge_{A \subseteq [n]} \left( \bigvee_{i \in A} y_{(i)} \vee \overline{m}(A) \right) \\ \text{s.t.} \quad & y \in Y \end{aligned}$$

This problem can be reformulated as:

$$\max \quad w \quad (10)$$

$$\text{s.t.} \quad \begin{cases} w \leq d_A \quad \forall A \subseteq [n] \\ d_A = \sum_{i \in A} b_i^A y_i + b_0^A \overline{m}(A) \quad \forall A \subseteq [n] \\ \sum_{i \in A} b_i^A + b_0^A = 1 \quad \forall A \subseteq [n] \end{cases} \quad (11)$$

where  $b_i^A \in [0, 1]$  and  $d_A = \sum_{i \in A} b_i^A y_i + b_0^A \overline{m}(A)$  represents the value  $\bigvee_{i \in A} y(i) \vee \overline{m}(A)$  at optimum. Using the Fortet linearization recalled above, the maximization problem can be reformulated as follows:

$$\max w \quad (12)$$

$$\text{s.t.} \begin{cases} w \leq d_A \quad \forall A \subseteq [n] \\ d_A = \sum_{i \in A} p_i^A + b_0^A \overline{m}(A) \quad \forall A \subseteq [n] \\ \sum_{i \in A} b_i^A + b_0^A = 1 \quad \forall A \subseteq [n] \\ p_i^A \leq b_i^A \quad \forall A \subseteq [n] \quad \forall i \in A \\ p_i^A \leq y_i \quad \forall A \subseteq [n] \quad \forall i \in A \\ p_i^A \geq y_i + b_i^A - 1 \quad \forall A \subseteq [n] \quad \forall i \in A \end{cases} \quad (13)$$

Here also, a more compact formulation can be obtained for  $k$ -minitive capacities:

**Definition 6** A capacity  $v$  defined on an ordinal scale  $\mathbb{L}$  endowed with a smallest (resp. greatest) element denoted by  $\perp$  (resp.  $\top$ ) is said to be  $k$ -minitive if its ordinal co-Möbius transform is equal to  $\top$  for any  $A \subseteq [n]$  such that  $|A| > k$ , and there exists at least one subset  $A$  of exactly  $k$  elements such that  $\overline{m}(A) \neq \top$ . Formally:

- (i)  $\forall A \subseteq [n], |A| > k \Rightarrow \overline{m}(A) = \top$
- (ii)  $\exists A \subseteq [n], |A| = k$  s.t.  $\overline{m}(A) \neq \top$

Given a capacity  $v$ , there is no need to explicitly compute all ordinal co-Möbius masses to decide whether it is  $k$ -maxitive or not. The recognition can be made directly on  $v$  by using the following result:

**Proposition 3**  $v$  is a  $k$ -minitive capacity if and only if  $v([n] \setminus A) = \perp$  for all  $A \subseteq [n]$  such that  $|A| \geq k$ .

**Proof** Assume that  $v$  is  $k$ -minitive. By definition,  $\forall A \subseteq [n], |A| > k \Rightarrow \overline{m}(A) = \top$  which implies that  $\forall A \subseteq [n], |A| > k \Rightarrow v([n] \setminus A) = v((\{[n] \setminus A\} \cup \{i\})) \forall i \in A$ . Since  $v(\emptyset) = \perp$  we obtain the desired result by induction. Conversely, let  $A$  be a subset such that  $|A| = k$ . We have  $v([n] \setminus A) = v([n] \setminus A) \cup \{i\} = \perp$  for all  $i \in A$ . Hence  $\overline{m}(A) = \top$  by definition of  $\overline{m}$ , which completes the proof.  $\blacksquare$

Under the  $k$ -minitivity assumption, the constraints and variables that were needed for all subsets of  $[n]$  become therefore only necessary for all subsets of size not greater than  $k$ .

In order to apply this mathematical programming technique to the Sugeno assignment problem (maximization case) where utilities are expressed on a qualitative scale  $\mathbb{L}$ , we need first to convert  $\mathbb{L}$  to the following numeric scale  $C = \{\frac{1}{|\mathbb{L}|}, \frac{2}{|\mathbb{L}|}, \dots, 1\}$ . Then we consider boolean variables  $z_{ij}$  representing assignment decisions. We have  $z_{ij} = 1$  whenever task  $j$  is assigned to agent  $i$  and  $z_{ij} = 0$  otherwise. Hence, assignment constraints are expressed as follows:

$$\sum_{i=1}^n z_{ij} = 1, j = 1, \dots, n \quad (14)$$

$$\sum_{j=1}^n z_{ij} = 1, i = 1, \dots, n \quad (15)$$

The utility of agent  $i$  is therefore defined by  $y_i = \sum_{j=1}^n u_{ij} z_{ij}$ . Note that, due to constraint (15), this definition of  $y_i$  ensures that  $y_i \in [0, 1]$ . Hence, the vector of individual utilities is given by  $y = (y_1, \dots, y_n) \in [0, 1]^n$ . These variables  $y_i$  can be used in combi-

nation with Program 12-13, which yields the following overall MIP:

$\max w$

$$\text{s.t.} \begin{cases} \sum_{i=1}^n z_{ij} = 1 \quad j = 1, \dots, n \\ \sum_{j=1}^n z_{ij} = 1 \quad i = 1, \dots, n \\ y_i = \sum_{j=1}^n u_{ij} z_{ij} \quad i = 1, \dots, n \\ w \leq d_A \quad \forall A \subseteq [n] \\ d_A = \sum_{i \in A} p_i^A + b_0^A \overline{m}(A) \quad \forall A \subseteq [n] \\ \sum_{i \in A} b_i^A + b_0^A = 1 \quad \forall A \subseteq [n] \\ p_i^A \leq b_i^A \quad \forall A \subseteq [n] \quad \forall i \in A \\ p_i^A \leq y_i \quad \forall A \subseteq [n] \quad \forall i \in A \\ p_i^A \geq y_i + b_i^A - 1 \quad \forall A \subseteq [n] \quad \forall i \in A \end{cases}$$

$z_{ij} \in \{0, 1\} \quad \forall i, \forall j, \quad w \in \mathbb{R}^+, \quad y_i \in \mathbb{R}^+ \quad \forall i$   
 $d_A \in \mathbb{R}^+ \quad \forall A \subseteq [n], \quad b_i^A \in \mathbb{R}^+ \quad \forall A \subseteq [n], \quad \forall i \in \{0\} \cup A$

Some numerical tests have been carried out using IBM ILOG CPLEX Optimization Studio 12.4 on a computer with 3.8 Gb of memory and an Intel Core 2 Quad 2.40 Ghz processor. We used an ordinal scale  $\mathbb{L}$  with  $|\mathbb{L}| = 5$ . Möbius masses corresponding to 2-minitive capacities were randomly drawn in  $\mathbb{L}$ , and the average computation times over 20 instances (for each size) were observed. The results showed that instances up to size  $n = 20$  can be solved in a few seconds, while the solution of instances of size 25 requires more than 600 seconds (value of the timeout in our tests). This fast growth of the solution times goes with the  $O(n^4)$  growth of the number of constraints in the MIP formulations: from 20 to 30 agents, the number of constraints increases from  $\sim 160,000$  to  $\sim 1,000,000$ . Interestingly, for Möbius masses defined such that  $S_v(y) = \min_i y_i$ , instances up to 100 agents were solved in a few seconds. This tends to confort the intuition that the more the Sugeno integral differs from the min, the less effective the formulation becomes.

## 6 CONCLUSION

In this paper, we have studied the optimization of the Sugeno integral in ordinal assignment problems. Table 1 summarizes our contributions in the maximization case. We have identified special cases (weighted min, ordered weighted min) that admit a polynomial time solution procedure, and we have shown that it is NP-hard in the general case, even if the cardinality of the scale is only two. Finally, we have provided the first MIP formulation for the optimization of the Sugeno integral under linear constraints. The preliminary numerical tests carried out showed that the operability of this latter approach is limited to instances of modest size. Nevertheless, for future works, it would be interesting to study the possibility of other MIP formulations. It is indeed worth noting that an efficient linearization of the Sugeno integral would impact on a number of combinatorial problems other than the assignment problem studied here (e.g., minimum spanning tree problem, traveling salesman problem).

Capacity	Operator	Complexity	Algorithm
general	Sugeno	NP-hard	MIP
minitive	weighted min	$O(n^3 \log  \mathbb{L} )$	Threshold
maxitive	weighted max	$O(n^2)$	$\max_{i,j} (u_{ij} \wedge w_i)$
symmetric	OWMin, OWMax	$O(n^3 \log  \mathbb{L} )$	Threshold
$k$ -maxitive	Sugeno	$O(n^{3+k} \log  \mathbb{L} )$	Threshold
$k$ -minitive	Sugeno	?	compact MIP

**Table 1:** Summary of our contributions.



## ACKNOWLEDGMENTS

Soufiane Drissi Oudghiri and Mohamed Hachimi's work was supported by the Centre National de la Recherche Scientifique et Technique (CNRST-Morocco). Patrice Perny and Olivier Spanjaard's work was supported by the ANR-14-CE24-0007-01 project "CoCoRiCo-CoDec".

## REFERENCES

- [1] E.M. Bartee, 'Problem solving with ordinal measurement', *Management Science*, **17**(10), 622–633, (1971).
- [2] U. Bossong and D. Schweigert, 'Minimal paths on ordered graphs', Technical Report 24, Report in Wirtschaftsmathematik, Universität Kaiserslautern, (1996).
- [3] S. Bouveret, U. Endriss, and J. Lang, 'Fair division under ordinal preferences: Computing envy-free allocations of indivisible goods', in *European Conference on Artificial Intelligence (ECAI 2010)*, pp. 387–392. IOS Press, (2010).
- [4] S. Brams, P. Edelman, and P. Fishburn, 'Fair division of indivisible items', *Theory and Decision*, **5**(2), 147–180, (2004).
- [5] S. Brams and D. King, 'Efficient fair division – help the worst off or avoid envy?', *Rationality and Society*, **17**(4), 387–421, (2005).
- [6] A. Chateauneuf, M. Grabisch, and A. Rico, 'Modeling Attitudes toward Uncertainty through the Use of Sugeno Integral', *Journal of Mathematical Economics*, **44**(11), 1084–1099, (2008).
- [7] F. Della Croce, V. Th. Paschos, and A. Tsoukias, 'An improved general procedure for lexicographic bottleneck problems', *Op. Res. Letters*, **24**, 187–194, (1999).
- [8] D. Dubois and H. Fargier, 'Lexicographic refinements of sugeno integrals', in *European Conference on Symbolic and Quantitative Approaches to Reasoning and Uncertainty*, pp. 611–622. Springer, (2007).
- [9] D. Dubois and H. Prade, 'Weighted minimum and maximum operations in fuzzy set theory', **39**, 205–210, (1986).
- [10] D. Dubois and H. Prade, 'Possibility theory as a basis for qualitative decision theory', in *IJCAI*, volume 95, pp. 1924–1930, (1995).
- [11] D. Dubois, H. Prade, and A. Rico, 'Residuated variants of Sugeno integrals: Towards new weighting schemes for qualitative aggregation methods.', *Information Sciences*, **329**, 765–781, (2016).
- [12] D. Dubois, H. Prade, and R. Sabbadin, 'Qualitative decision theory with sugeno integrals', in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, pp. 121–128. Morgan Kaufmann Publishers Inc., (1998).
- [13] R. Fortet, 'Applications de l'algèbre de boole en recherche opérationnelle', *Revue Française de Recherche Opérationnelle*, **4**(14), 17–26, (1960).
- [14] P. Gardenfors, 'Assignment problem based on ordinal preferences', *Management Science*, **20**(3), 331–340, (1973).
- [15] J. Gorski and S. Ruzika, 'On k-max-optimization', *Operations Research Letters*, **37**(1), 23–26, (2009).
- [16] M. Grabisch, 'Some lexicographic approaches to the Sugeno integral', in *Proc. Int. Conf. on Information Processing and Management of Uncertainty (IPMU'06)*, pp. 572–579, (2006).
- [17] M. Grabisch and C. Labreuche, 'A decade of applications of the choquet and sugeno integrals in multi-criteria decision aid', *Annals of Operations Research*, **175**, 247–286, (2010).
- [18] M. Grabisch, J.-L. Marichal, R. Mesiar, and E. Pap, *Aggregation Functions (Encyclopedia of Mathematics and its Applications)*, Cambridge University Press, New York, NY, USA, 1st edn., 2009.
- [19] O. Gross, 'The bottleneck assignment problem', Technical Report P-1630, The Rand Corporation, Sta. Monica, CA, (1959).
- [20] P. L. Hammer and S. Rudeanu, *Boolean methods in operations research and related areas*, volume 7, Springer Science & Business Media, 2012.
- [21] A. Junghanns and J. Schaeffer, 'Sokoban: Enhancing general single-agent search methods using domain knowledge', *Artificial Intelligence*, **129**(1), 219–251, (2001).
- [22] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai, 'A practical, decision-theoretic approach to multi-robot mapping and exploration', in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, volume 4, pp. 3232–3238. IEEE, (2003).
- [23] H.W. Kuhn, 'The hungarian method for the assignment problem', *Naval research logistics quarterly*, **2**(1-2), 83–97, (1955).
- [24] V. M. Malhotra, M. P. Kumar, and S. N. Maheshwari, 'An  $O(|V|^3)$  algorithm for finding maximum flows in networks', *Information Processing Letters*, **7**(6), 277–278, (1978).
- [25] J.-L. Marichal, 'Weighted lattice polynomials', *Discrete Mathematics*, **309**(4), 814–820, (2009).
- [26] R. Mesiar, 'Generalizations of k-order additive discrete fuzzy measures', *Fuzzy Sets and Systems*, **102**, 423–428, (1999).
- [27] F.S. Roberts, *Measurement theory, with applications to Decision Making, Utility and the Social Sciences*, Addison-Wesley, Boston, 1979.
- [28] F.S. Roberts, 'Meaningfulness of conclusions from combinatorial optimization', *Discrete Applied Mathematics*, **29**, 221–241, (1990).
- [29] T. Sandholm and S. Suri, 'Side constraints and non-price attributes in markets', *Games and Economic Behavior*, **55**(2), 321–330, (2006).
- [30] David Schmeidler, 'Integral representation without additivity', *Proceedings of the American mathematical society*, **97**(2), 255–261, (1986).
- [31] D. Schweigert, 'Ordered graphs and minimal spanning trees', *Foundations of Computing and Decision Sciences*, **24**(4), 219–229, (1999).
- [32] P.T. Sokkalingam and Y.P. Aneja, 'Lexicographic bottleneck combinatorial problems', *Operations Research Letters*, **23**(1), 27–33, (1998).
- [33] M. Sugeno, *Theory of fuzzy integrals and its applications*, Ph.D. dissertation, Tokyo Institute of Technology, 1974.
- [34] A. Takaci, I. Stajner-Papuga, M. Maric, and D. Drakulic, 'A note on the use of choquet and sugeno integrals in minimal and maximal covering location problems', in *IEEE 12th International Symposium on Intelligent Systems and Informatics, SISY 2014, Subotica, Serbia, September 11-13, 2014*, pp. 159–162, (2014).
- [35] V. Torra and Y. Narukawa, 'The-index and the number of citations: Two fuzzy integrals', *Fuzzy Systems, IEEE Transactions on*, **16**(3), 795–797, (2008).