Résumé

Nous considérons le problème d'affectation multicritère qui vise à affecter chaque action d'un ensemble fini A à une des catégories pré-définies. Nous proposons un nouveau concept de taille des catégories qui réfère à "*la proportion de vecteur performance correspondant à une action réaliste affecté à la catégorie*".

Les problèmes d'affectation font référence à une évaluation absolue (l'affectation d'une action ne dépend pas des autres), par opposition aux problèmes de choix et de rangement dans lesquels la nature même du problème conduit à comparer les actions entre elles. Le fait de considérer des contraintes concernant la taille des catégories conduit à définir un problème d'affectation contraint qui fait référence à la fois à l'évaluation absolue et relative.

Après avoir introduit des situations décisionnelles où la notion de taille des catégories se révèle utile en matière de modélisation, ce papier définit formellement le concept de taille des catégories. Nous proposons une utilisation opérationnelle de ce concept qui s'applique aussi lorsque l'ensemble des actions potentielles et/ou les préférences du décideur sont connus de façon imprécise. Nous montrons comment cette notion peut être utilisée dans un processus d'élicitation des préférences. Enfin, pour illustrer l'utilisation de ce concept, nous proposons une procédure qui infère les valeurs des paramètres préférentiels de la méthode UTADIS en prenant en compte des contraintes sur la taille des catégories.

Mots clés: *Aide multicritère à la décision, problématique du tri, taille des catégories*

Abstract

We consider the Multiple Criteria Sorting Problem, that aims at assigning each alternative in a finite set A to one of the predefined categories. We propose a new concept of category size that refers to "*the proportion by which an evaluation vector corresponding to a realistic alternative is assigned to the category*".

Sorting problems usually refer to absolute evaluation (the assignment of an alternative does not depend on the others), as opposed to ranking and choice problems in which the very purpose is to compare alternatives against each other. Considering constraints concerning category size lead to define a Constrained Sorting Problem which refers both to absolute and relative evaluation.

After identifying decision situations in which category size is useful for modelling purposes, this paper defines the concept of category size and proposes a way to compute the size of categories, even when the set of alternatives and/or the preference information is/are imprecisely known. We show how this notion can be used in a preference elicitation process. Finally, in order to illustrate the use of this concept, we propose a procedure to infer the values for preference parameters that accounts for specifications (provided by the DM) about the size of categories, in the context of the UTADIS sorting model.

Keywords: *Multiple Criteria Decision Analysis, Sorting problem, Category size*

Introduction

Modelling a real world decision problem using multiple criteria decision aid involves defining a set of n_{alt} alternatives $A = \{a_1, a_2, \ldots, a_{n_{alt}}\}$ evaluated on n_{crit} criteria $g_1, g_2, ..., g_{n_{crit}}$, each criterion, g_j, being associated to a scale X_j. Several problem statements (or problem formulations) can be considered. Roy [9] distinguishes three basic problem statements: choice, sorting and ranking (see also [1]).

Given a set A of alternatives, choice problems consist in determining a subset $A^* \subset A$, as small as possible, composed of alternatives being judged as the most satisficing. Optimization problems are particular cases of choice problems where A^* is restricted to one alternative. Ranking problems consist in establishing a preference pre-order (either partial or complete) in the set of alternatives A.

Sorting problems consist in formulating the decision problem in terms of a classification, in order to assign each alternative from A to one of the n_{cat} predefined categories $C_1, C_2, \ldots, C_{n_{cat}}$. The assignment of an alternative a to the appropriate category should rely on the intrinsic value of a (and not on the comparison of a to other alternatives from A).

Among these problem statements, a major distinction concerns *relative* versus *absolute* judgment of alternatives. This distinction refers to the way alternatives are considered and to the type of result expected from the analysis. In the first case, alternatives are directly compared one to each other and the results are expressed using the comparative notions of "*better*" and "*worse*". Choice or ranking are typical examples of comparative judgments. The presence (or absence) of an alternative a in the set of best alternatives A^* results from the comparison of a to the other alternatives. Similarly, the position of an alternative in the preference order depends on its comparison to others.

In the *absolute* judgment case, each alternative is considered independently from the others in order to determine its intrinsic value by means of comparisons to norms or references. Sorting problems refer to absolute judgments and consist of assigning each alternative to one of the pre-defined categories. The assignment of an alternative $a \in A$ results from the intrinsic evaluation of a on all criteria (the assignment of a to a specific category does not influence the category to which another alternative should be assigned). Various methods have been developed for such assignment problem (see for instance [7], [3], [6],[10], [8] and [2], see also [12] for a review).

In this paper, we are concerned with sorting problems. More precisely, we are concerned with what we call Constrained Sorting Problems (CSP), related to a new notion regarding the size of the categories, i.e., "*the proportion by which an evaluation vector corresponding to a realistic alternative is assigned to the category*". CSPs arise when the Decision Maker (DM) has an idea about how the alternatives should be distributed among the categories. For instance, imagine a sorting model to assign students to grades A,B,C, and D. The DM is asked the following question "If all of this year's students fall into the same category (it does not matter which), would you be compelled to change the model?". If the answer is yes, then we would have a CSP.

The paper is organized as follows. In the first section, we present four illustrative examples that motivate the usefulness of the notion of category size for decision aiding. We analyze, in section 2, the consequences of using category size in the sorting problem statement and define the CSP, contrasting it to other problem statements. Section 3 formally defines the notion of category size. Two ways to consider category size in preference elicitation processes are considered in section 4. An illustrative example using the UTADIS method is provided in section 5. A final section groups conclusions and further research.

1 Motivating examples

In order to justify the interest of the notion of category size for decision aiding, let us consider several realistic illustrative decision problems in which this notion can play a significant role in the modelling process.

Example A

Consider a corporate distribution company composed of a large number of retails. The head of this company wants to identify the retails that outperform (*i.e.*, the ones that are very profitable and have good results) and those that under-perform (*i.e.*, the ones that seem improperly managed and induce losses). This evaluation is to be grounded on several criteria (*e.g.*, profit, customer complaints, market share, ...). The CEO formulates this problem as a sorting problem in which retails should be assigned to one of the five following categories:

- C_1 Underperforming retail: immediate corrective action required concerning the management (consider replacing the manager),
- C_2 Retail having a bad performance: demand justifying information to the manager in order to clarify the reasons for the bad results,
- C_3 Retail having average performance: no specific action required,
- C_4 Retail having good results: send a congratulation letter to the manager and encourage him/her to pursue his/her effort,
- C_5 Retail that outperform, that have exceptional results stemming from an outstanding management: consider a promotion for the manager.

The CEO wants to identify and analyze only the cases corresponding to exceptions, *i.e.*, retails that have either very positive or very negative results. Her idea is that most of the retails fall into the C_3 situation and that only very few of them correspond to C_1 or C_5. She views the distribution of the retails among the five categories as "*gaussian*" (bell-shaped).

Example B

Consider a credit manager in a financial institution who decides whether or not to grant credits to clients. His role is to accept/reject credit files or possibly refer to his superior for difficult or ambiguous cases. His decision is grounded on the various elements documented in the file. This decision problem can be formulated through a multiple criteria trichotomic segmentation (accept/refer to superior/reject).

However, the credit manager does not want to send to many files (no more than 10% in average) to his superior. In such a case, it is natural to conceive a sorting model with three categories in which *the central class is small in size* (when considering a set of "habitual" credit files).

Example C

Each year the director of a department of a firm wants to split the grant budget among her collaborators. She considers four levels of bonuses (A: high, B: medium,C: small,D: null) according to several performance criteria discussed in the beginning of the year with her collaborators (the bonuses packages A, B and C are also stated in the beginning of the year).

The bonus policy of the director is such that very few collaborators get a A-bonus, a little more get a B-bonus, a significant proportion get a C-bonus and a large part of them get no bonus. In this situation, the shape of the category size distribution can be considered as "*increasing*" (or "*decreasing*" according to the way the categories are numbered).

Example D

Each year, the responsible of the University training program faces the same problem when defining the foreign language courses. He wants to split a group of students (approximately 100) into three groups of different levels (beginners, intermediate, advanced). The assignment of a student to a specific class is grounded on his/her skills (oral expression, listening comprehension, grammar, written expression, ...). However, in order to be "fair" to the teachers and students the three classes are intended to be "not too different" in size. Such decision problem can obviously be formulated as a multiple criteria sorting problem. One of the specificities of this problem consists of the "*uniform*" size of the categories representing the three classes.

These four decision problems illustrate prototypical sorting decision situations in which category size somehow intervenes in the modelling process. An analyst designing a decision aiding model for these situations should consider in the modelling process the information concerning the shape of the category size distribution, as suggested by the DM.

The reader will easily imagine various other problems in which this notion is an important aspect of the decision model definition. Each of these examples suggest constraints about the proportion of alternatives assigned to each category. Figure 1 depicts typical category size distributions, which we may call "Gaussian" (Example A), "Dichotomic" (Example B), "Increasing/Decreasing" (Example C), and "Uniform" (Example D).

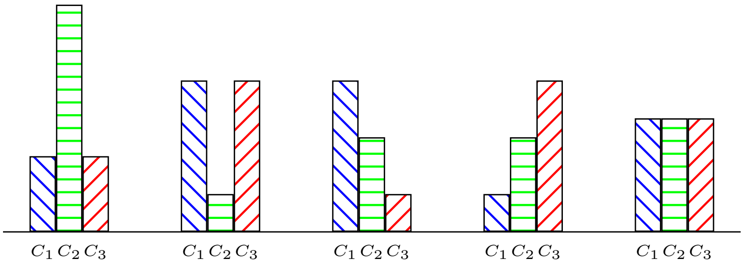

Figure 1: Typical category size distributions

2 Constrained sorting problems

We refer to Constrained Sorting Problem (CSP) in decision situations formulated through a sorting model in which specifications about the size of the categories are introduced. These specifications take the form of implicit or explicit constraints on category size.

In example A for instance, these constraints specify that "the distribution of the retails among the five categories is *gaussian*". The credit manager, in example B, does not want to refer to his superior for more than 10% of the files. In the Example C, the director's view on the "relative proportion" of A, B, C and D-bonuses specifies constraints on the size of corresponding categories. In example D, the constraints express the statement "the three groups should not be *too different* in size".

CSPs differ from "standard" sorting problems. "Standard" sorting problems refer to absolute judgments (no comparison among alternatives is required). However, constraints on the size of categories introduce implicit relative evaluation between alternatives. More precisely, in a "standard" sorting problem the assignment of an alternative a to a specific category only depends on the intrinsic characteristics of a (and the norms defining the categories), while in CSPs its assignment rely not only on its intrinsic characteristics but also on the other alternatives in A.

In example D, a student that has average/low skills might be assigned to the "beginners" class when a large proportion of students have high skills, while some other year he/she could be assigned to the "intermediate" class when the average level of students is lower. In example B, suppose that the credit manager faces an exceptional situation in which a majority of credit files are uncertain and ambiguous. In such a case, a standard sorting model would assign a large proportion of files to the "*central*" class (*i.e.*, refer to superior). The introduction of constraints about the size of this category will "force" the model to assign some alternatives previously assigned to the central class to either of the two remaining ones and will keep in this *refer to superior* category only the most ambiguous cases.

Consequently, it follows that CSPs possess some of the characteristics of the relative problem statements (choice and ranking) namely regarding the dependence of the result with respect to the set of alternatives. In CSPs the assignment of an alternative depends on its intrinsic characteristics, but also on the assignment of the other alternatives. Such problem statement is however distinct from relative problem statements and can be considered as an intermediate situation as it deals both with relative and absolute issues: absolute evaluation is present in the sorting model, and the very idea of sorting, while comparative evaluation stems from the the addition of constraints concerning category size.

It is also interesting to note that a choice problem can be expressed through a CSP, if the categories are ordered. Consider a sorting problem with two categories (C_1: select, C_2: reject). Let us impose that the size of C_1 is equal to n. If we pose $n = 1$, then this problem corresponds to a choice. Moreover, relaxing the constraint concerning the size of C_1 (*i.e.*, increasing the value for n) leads to interesting formulations with respect to the choice problem.

Conversely, some CSPs can be formulated as choice or ranking problems if the categories are ordered. If there exists a size constraint stating that the first category should contain k alternatives, this corresponds to a choice of the best k alternatives from a set. If there exist size constraints stating that k_1 alternatives belong to C_1, k_2 alternatives belong to C_2, etc., this may be accomplished by ranking the alternatives from best to worst and breaking the ranking into n_{cat} segments. However, there exist CPS problems that can not be formulated using a relative formulation (choice or ranking). For instance, a situation like example B can not be solved by ranking the alternatives because we would not know which segment with 10% of the alternatives (in the middle of the ranking) should be selected.

We have informally defined the size of a category as "*the proportion by which an evaluation vector corresponding to a realistic alternative is assigned to the category*". We can distinguish among situations in which the set of alternatives A is completely known or not before building the model:

- *Static CSP*: Suppose that we have a complete description of the set of alternatives A. In this case, constraints on category size may be specified during the model definition. The resulting model explicitly integrates these constraints. For instance in example D, the three groups of students are to be defined considering the actual students of the current year.
- *Anticipatory CSP*: If A is not known beforehand, we face situations with uncertainty concerning the alternatives, where a sorting model is built taking into account constraints about the size of the categories, *given the alternatives that are realistically likely to appear*. In such situations, the model is built (and may be divulged) before the actual alternatives are known, and constraints about category size may possibly be violated when the actual set of alternatives is considered. For instance, in example C, the department director would want to announce the criteria for granting bonuses, and if the employees work exceedingly well she may have to grant more A bonuses than she was expecting.

3 Formal definition of category size

We have referred to the size of category C_k resulting from a sorting model as "*the proportion by which an evaluation vector corresponding to a realistic alternative is assigned to the category* C_k". Let $K = \{1, \ldots, n_{cat}\}$ be the set of category indices. Next we propose some definitions for the size of the k^{th} category, denoted by $\mu(C_k)$. These definitions satisfy the following desirable properties:

$$\begin{cases} \mu(C_k) \geq 0, \ \forall k \in K \\ \sum_{i=1}^{n_{cat}} \mu(C_k) = 1 \\ \mu(\cup_{k\in K'} C_k) = \sum_{k\in K'} \mu(C_k), \ \ \forall K' \subset K \end{cases} \quad (1)$$

Let us consider a specific sorting model that uses a set of preference parameters Ω (such as criteria weights, limits of categories, ...). Let $\mathcal{P}$ denote the domain of possible values for the parameters in Ω. Let $C(a_i, p)$ denote the index of the category to which $a_i \in A$ is assigned when the model parameters take values $p \in \mathcal{P}$. Let each alternative be defined by its evaluations on n_{crit} criteria. For the j^{th} criterion ($j = 1, \ldots, n_{crit}$), the evaluations may take values from a domain X_j.

3.1 Evaluating the size of categories when A and p are known

Given a set of alternatives A completely defined and a vector of parameter values $p \in \mathcal{P}$, it is reasonable to admit that the size of each category equals the proportion of alternatives from A that are assigned to category C_k, *i.e.*,

$$\mu(C_k) = \frac{|\{a_i \in A : C(a_i, p) = k\}|}{|A|} \quad (2)$$

In such situations, there is complete a knowledge about the alternatives, which excludes anticipatory CSPs. Furthermore, the model's parameters have been precisely fixed.

3.2 Evaluating the size of categories when p is known but A is imprecisely known

In this situation we have the knowledge concerning the vector of parameter values $p \in \mathcal{P}$, but the alternatives are not precisely known in advance. This may correspond to the anticipatory CSPs, where the sorting model is being constructed to evaluate alternatives that will appear in the future, such as financial projects, job applicants, research projects, etc.

We will consider that the knowledge of the DMs (e.g. from past evaluations) allows to specify a multivariate probability distribution Ψ on $\cup_{i=1}^{n_{crit}} X_i$ (X_i denotes the scale of criterion g_i). This probability distribution for the evaluations of alternatives should account for possible correlations among criteria. We may therefore consider that a random sample of evaluation vectors following this distribution is a representative set of the actual evaluations of the alternatives that will appear in the future. Following the definition for the category size presented in the previous subsection, we reach a new definition that relates the size of the k^{th} category with the probability of an alternative following the distribution Ψ being assigned to that category:

$$\mu(C_k) = \text{Prob}(C(\tilde{a}, p) = k) \quad (3)$$

where $\tilde{a}$ is a random vector following the distribution Ψ.

3.3 Evaluating the size of categories when p is imprecisely known

In this situation neither the alternatives nor the parameter values are precisely known in advance. The lack of a precise vector of preference-related parameter values may stem from various sources:

- the DMs find it hard to precisely answer some questions regarding their preferences,
- the DMs may not fully understand the role of all the parameters,
- the model may be used in the future and the DMs may not know how their preferences will evolve,
- etc.

We will consider again a multivariate probability distribution Ψ for the evaluations of the alternatives, and introduce a multivariate probability distribution Π (possibly uniform) for the vector of parameters. Although there will seldom exist reasons to consider that some parameter values are more probable than others, using probability distributions for parameter values will permit a general definition of category size, consistent with the one provided in the previous sub-section. Indeed, the latter may be extended to account for the fact that the parameter values are also stochastic:

$$\mu(C_k) = \text{Prob}(C(\tilde{a}, \tilde{p}) = k) \tag{4}$$

where $\tilde{a}$ is a random vector following the distribution Ψ, and $\tilde{p}$ is a random vector following the distribution Π.

3.4 Robustness analysis for the category size

A complementary approach to using the preceding definitions is to find the maximum and minimum size of each category, given a domain for the vector of parameters p. Assuming that the information provided by the DMs allows to define a domain $P \subset \mathcal{P}$, we may compute:

$$\begin{aligned} \mu_{min}(C_k) &= \min(Prob(C(\tilde{a}, p) = k) : p \in P) \\ \mu_{max}(C_k) &= \max(Prob(C(\tilde{a}, p) = k) : p \in P) \end{aligned} \tag{5}$$

This provides the DMs an idea of the interval for the size of each category given the lack of precise knowledge about the parameter values and alternatives. These intervals become narrower as more information about the parameter values is added (*i.e.*, as P becomes smaller). The mean value $\overline{\mu}(C_k)$ for the size of C_k can also provide information.

4 Considering category size in preference elicitation processes

The elicitation of a multiple criteria sorting model amounts at assigning precise values to the preference parameters used by the aggregation model, *i.e.*, to select an appropriate $p^* \in \mathcal{P}$. This work can be done:

- either by a direct questioning procedure with the DM,
- or indirectly through the use of an inference program that induces parameter values that restore holistic judgments (*e.g.*, assignment examples) provided by the DM (see for instance [4] and [5] for such a disaggregation approach).

In this section we discuss how the concept of category size may be exploited in such preference elicitation processes. As input, we consider a domain $P_0 \subset \mathcal{P}$ for possible parameter values and a set $\bar{A}$ of alternatives to assign to categories. When the alternatives are known in advance, then we consider $\bar{A} = A$. Otherwise (e.g., in anticipatory CSPs), we can consider $\bar{A}$ is a sample generated from distribution Ψ, or a sample from historical data.

The concept of category size may be used to support an elicitation process by trial and error, where the DM chooses a combination of values for the parameters $p \in \mathcal{P}$ and observes the computed category sizes corresponding to it through pictures similar to those in Figure 1. If what the DM sees does not correspond to his/her intuition of what the distribution of category sizes should look like, then he/she may change the parameter values, by trial and error, until a satisfactory distribution is found. However, unless the DM is using a very simple assignment model (e.g. one that depends on few parameters and such that the effects of changing each parameter are easy to predict), then a trial and error process may become cumbersome.

We therefore consider the case where the sorting model is inferred. In such cases, the DM does not have to be an expert in the sorting method, and may simply provide assignment examples, *i.e.*, alternatives for which the DM defines a specific assignment. In addition, the DM will provide some constraints related to his/her intuitive view on the "*size of each category*", and may provide additional constraints on parameter values.

Constraints on category sizes can be expressed in various manners:

- exact values, e.g., "there should be 5 alternatives in C_1";
- bounds, e.g., "there should be at most 5 alternatives in C_1";
- intervals, e.g., "there should between 5 and 10 alternatives in C_1";
- comparisons, e.g., "there should more alternatives category C_1 than C_2".

This information can be translated as constraints on p (defining $P \subset \mathcal{P}$) in a mathematical program, because a set of alternatives $\bar{A}$ has already been fixed as a reference. The details of such a mathematical program will vary from a sorting method to another (the next section provides an example for a UTADIS-like method).

As assignment examples and constraints on category size are expressed through constraints in the inference mathematical program, these two types of constraints might be conflicting. This implies that the inference program should specify how such potential conflicts among these constraints are to be solved (see section 5 for further discussion in the UTADIS framework).

In anticipatory CSPs, where the actual set A will seldom be equal to the forecasted sample $\bar{A}$, the inferred model that satisfies all category size constraints when $\bar{A}$ is considered, may no longer satisfy some of them when A is considered instead. For this reason, when a DM places a constraint like "there should be 5 alternatives in the top category", he/she should expect that *around* 5 alternatives, and not *exactly* 5, will appear in the top category when using the model to sort A. The DM may even determine beforehand (e.g., using Monte-Carlo simulation) the probability of violating each constraint, given distribution Ψ and an inferred model $p^* \in P$.

This characteristic of anticipatory CSPs should not be seen as a drawback. Indeed, it may be essential. For instance, in Example A (Section 1) the CEO may provide size constraints stating that less than 10% of the retails should fall into categories C_1 or C_5, whereas more than 50% should be sorted into C_3. This reflects the CEO's view of normality, and a model might be built using past data as a reference set $\bar{A}$. However, should for some reason the performance of the actual retails A become atypical (e.g. 30% of them fall into C_1), then the CEO would want to be alerted for that. Surely, she would not want a model that would automatically adjust its parameters to keep the underperformers under a 10% size barrier.

5 An illustrative example using a UTADIS-like method

In order to illustrate the use of category size concept, we propose a procedure to infer the preference parameters values that account both for assignment examples and specifications about the size of categories, in the context of the UTADIS sorting model ([3], [11]).

5.1 Brief reminder on the UTADIS method

UTADIS is a multiple criteria sorting method that assigns alternatives $a_i \in A = \{a_1, a_2, \dots, a_{n_{alt}}\}$ to one of the predefined ordered categories $C_1, C_2, \dots, C_k, \dots, C_{n_{cat}}$ (C_1 being the worst category) on the basis of a set of criteria $\{g_1, g_2, \dots, g_j, \dots, g_{n_{crit}}\}$. In what follows, we will assume, without any loss of generality, that preferences increase with the value on each criterion. Let $F = \{1, \dots, n_{crit}\}$ denote the set of criteria indices, $I = \{1, \dots, n_{alt}\}$ denote the set of alternatives indices, and by $K = \{1, \dots, n_{cat}\}$ the set of categories indices. The UTADIS sorting method is an additive utility model of the form:

$$u(a_i) = \sum_{j \in F} u_j(g_j(a_i)) \ \in \ [0, 1], \quad \forall\, i \in I, \text{ where} \tag{6}$$

- $g_j(a_i)$ is the evaluation of alternative a_i on criterion g_j, $\forall i \in I$, $\forall j \in F$,
- u_j is a piecewise linear utility function for criterion g_j, $\forall j \in F$, each u_j ranges on the set $[0, w_j]$ (we pose without loss of generality $\sum_{j \in F} w_j = 1$).

UTADIS assigns each alternative $a_i \in A$ by comparing $u(a_i)$ to a set of category limits, $b_1, \dots, b_k, \dots, b_{n_{cat}-1}$ such that $b_k > b_{k-1}$, $k = 2, ..., n_{cat} - 1$. These limits are used to assign alternatives in the following way (Figure 2 illustrates the assignment process described above):

$$\begin{cases} a_i \in C_1 & \Leftrightarrow \quad 0 \le u(a_i) < b_1 \\ & \quad \vdots \\ a_i \in C_k & \Leftrightarrow \quad b_{k-1} \le u(a_i) < b_k \\ & \quad \vdots \\ a_i \in C_{n_{cat}} & \Leftrightarrow \quad b_{n_{cat}-1} \le u(a_i) \le 1 \end{cases} \tag{7}$$

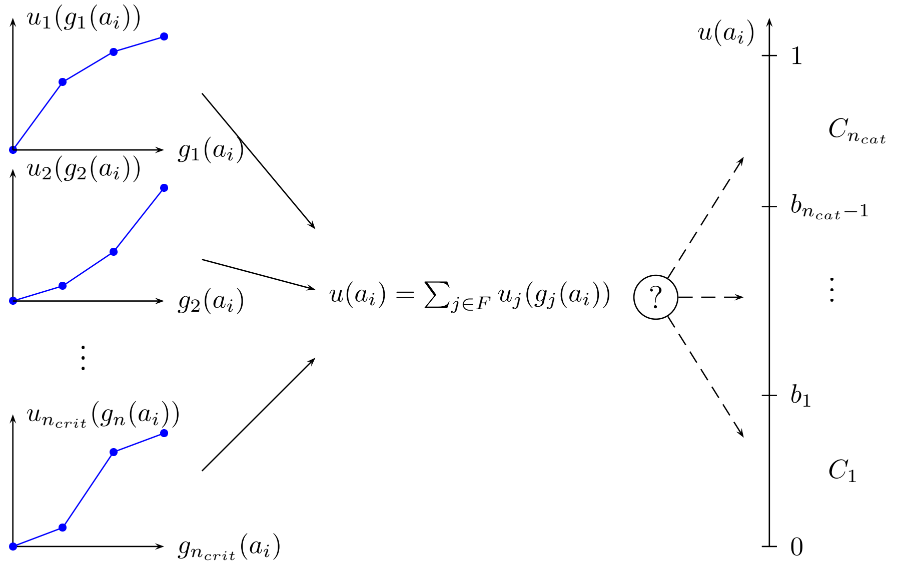

Figure 2: UTADIS sorting scheme

Let g_j^m (g_j^M, respectively) be the minimum (maximum, respectively) evaluation on criterion g_j, $\forall j \in F$. The interval $[g_j^m, g_j^M]$ is divided into L_j equal subintervals: $[g_j^0, g_j^1[$, $\ldots$, $[g_j^l, g_j^{l+1}[$, $\ldots$, $[g_j^{L_j-1}, g_j^{L_j}]$ ($g_j^0 = g_j^m$ and $g_j^{L_j} = g_j^M$), where g_j^l is computed as follows:

$$g_j^l = g_j^m + \frac{l}{L_j}(g_j^M - g_j^m), \quad l = 0, \ldots, L_j \;\text{ and }\; j \in F \tag{8}$$

Each piecewise linear function u_j is defined by the utilities of breakpoints $u_j(g_j^0) \leq u_j(g_j^1) \leq \ldots \leq u_j(g_j^{L_j})$ (we recall that $u_j(g_j^0) = 0$ and $u_j(g_j^{L_j}) = w_j$). If $g_j(a_i) \in [g_j^l, g_j^{l+1}[$, then the partial utility is obtained by linear interpolation: $u_j(g_j(a_i)) = u_j(g_j^l) + \frac{g_j(a_i) - g_j^l}{g_j^{l+1} - g_j^l}(u_j(g_j^{l+1}) - u_j(g_j^l))$. Hence, the parameters of the UTADIS sorting model are the following:

- The utility of each breakpoint g_j^l, that is, $u_j(g_j^l)$, for $j \in F$ and $l = 1, \ldots, L_j$.
- The category limits, b_k, for $k = 1, \ldots, n_{cat} - 1$.

Let $A^* \subset A$ denote a subset of alternatives that the DM intuitively assigns to a specific category (A^* contains the assignment examples). UTADIS aims at inferring the parameters values that best match the assignment examples. Suppose the DM stated that alternative $a_i \in A^\star$ should be assigned to the category C_k ($a_i \to C_k$). This statements generates constraints on the parameters values: $b_{k-1} \leq u(a_i) < b_k$. In order to integrate these constraints in a mathematical program, two slack variables $\delta^-(a_i)$ and $\delta^+(a_i)$ are introduced as follows (ϵ is an arbitrarily small positive constant):

$$\begin{cases} u(a_i) - b_k - \delta^-(a_i) \leq \epsilon \\ u(a_i) - b_{k-1} + \delta^+(a_i) \geq 0 \end{cases} \tag{9}$$

The linear program (10)-(19) infers the parameter values that best restore a set of assignment examples:

$$\begin{aligned}
\min\ z &= \textstyle\sum_{a_i \in C_1} \delta^-(a_i) + \ldots + \sum_{a_i \in C_k} (\delta^-(a_i) + \delta^+(a_i)) + \ldots + \sum_{a_i \in C_{n_{cat}}} \delta^+(a_i) && (10)\\
s.t: & \textstyle\sum_{j \in F} u_j(g_j(a_i)) - b_1 - \delta^-(a_i) \leq \epsilon, \quad \forall a_i \in C_1 && (11)\\
& \textstyle\sum_{j \in F} u_j(g_j(a_i)) - b_k - \delta^-(a_i) \leq \epsilon, \quad \forall a_i \in C_k, \quad k = 2, \ldots, n_{cat} - 1 && (12)\\
& \textstyle\sum_{j \in F} u_j(g_j(a_i)) - b_{k-1} + \delta^+(a_i) \geq 0, \quad \forall a_i \in C_k, \quad k = 2, \ldots, n_{cat} - 1 && (13)\\
& \textstyle\sum_{j \in F} u_j(g_j(a_i)) - b_{n_{cat}-1} + \delta^+(a_i) \geq 0, \quad \forall a_i \in C_{n_{cat}} && (14)\\
& u_j(g_j^{l+1}) - u_j(g_j^l) \geq 0 \ \ \forall j \in F, \ l = 1, \ldots, L_j - 1 && (15)\\
& u_j(g_j^0) = 0, \ \ \forall \ j \in F && (16)\\
& \textstyle\sum_{j \in F} u_j(g_j^{L_j}) = 1 && (17)\\
& b_k - b_{k-1} \geq \epsilon, \quad k = 2, \ldots, n_{cat} - 1 && (18)\\
& \delta^-(a_i), \delta^+(a_i) \geq 0, \quad \forall a_i \in A^\star && (19)
\end{aligned}$$

5.2 Considering category size constraints in UTADIS

Let us consider the data of a real-world application concerning a credit granting application [10]. This application deals with 100 alternatives (Appendix A lists the set of alternatives) evaluated on 7 criteria (all criteria are decreasing) to be assigned to 3 ordered categories: C_1: to reject, C_2: to analyze, C_3: to accept.

The knowledge of the credit manager leads to define interval of variation for the parameters values as follows:

- the shape of the functions u_j are imprecisely known (see Appendix B),
- the criteria weights are such that $w_j \in [0.1, 0.2]$, $\forall j \in F$, (note: $w_j = u_j(g_j^{L_j})$) and
- the categories profiles or limits are such that $b_1 \in [0.5, 0.6]$ and $b_2 \in [0.65, 0.7]$.

Moreover, the credit manager usually sends about 10% of the files to his/her superior for further analysis, which means that, C_2 should contain "approximately" 10% of the files. Hence, our purpose is to built a model taking into account the constraint about the size of category C_2. A simulation study (1000 random simulations for the parameters values considering the imprecise knowledge concerning these parameters) leads to the results given in Table 1.

	$\mu_{min}(C_k)$ (%)	$\mu_{max}(C_k)$ (%)	$\overline{\mu}(C_k)$ (%)
C_1	12	73	37.5
C_2	2	64	34.3
C_3	16	62	28.2

Table 1: Results from the simulation process

Considering the imprecision of the data, the size of C_2 is contained in $[2\%, 64\%]$ (see Table 1). However, the credit manager wants to send approximately 10% of the files to his/her superior. As assignment examples, the credit manager has identified some files (a_{42}, a_{53}, a_{61}), for which he/she should refer to his/her superior, as well as some files to be rejected (a_{22}, a_{27}, a_{41}) and some to be accepted (a_{80}, a_{90})

In order cope with the manager's requirements, we define a mathematical program to infer (in the domain of acceptable values of the parameters) a model as compatible as possible with this constraint on the size of C_2. Let us define the decision variables y_{ik} such that:

$$y_{ik} = \begin{cases} 1, & \text{if the alternative } a_i \rightarrow C_k \\ 0, & \text{otherwise.} \end{cases} \quad \forall i : a_i \in A, \; k = 1,2,3 \tag{20}$$

These y_{ik} variables can be defined in a mathematical program by the constraints (21) and (22) where M is a large positive constant and ε a small positive constant.

$$\sum_{j=1}^{n_{crit}} u_j(g_j(a_i)) - b_k + My_{ik} \leq M - \varepsilon, \tag{21}$$

$$-\sum_{j=1}^{n_{crit}} u_j(g_j(a_i)) + b_{k-1} + My_{ik} \leq M, \tag{22}$$

Hence the expression $\sum_{i=1}^{n_{alt}} y_{ik}$ denotes the number of alternatives assigned to category C_k and can be used to constraints on the size of C_k. In our case, the statement 'C_2 *should contain approximately 10 % of the files*" can be formulated by the two following constraints:

$$\begin{cases} \sum_{i=1}^{n_{alt}} y_{i2} \geq 10 - \sigma \\ \sum_{i=1}^{n_{alt}} y_{i2} \leq 10 + \sigma \end{cases} \tag{23}$$

where σ is a variable to be minimized. Moreover, the assignment examples can be easily integrated by setting the values of the corresponding y_{ik} variables ($a_i \rightarrow C_k \Leftrightarrow y_{ik} = 1$[1]). In the mathematical program given hereafter, the y_{ik} binary variables are defined by the constraints (24)-(**??**).

$$\text{Min } \sigma$$

$$\text{s.t.} \quad \sum_{j=1}^{7} u_j(g_j(a_i)) - b_k + My_{ik} \leq M - \varepsilon, \quad i = 1..100, \; k = 1,2 \tag{24}$$

$$-\sum_{j=1}^{7} u_j(g_j(a_i)) + b_{k-1} + My_{ik} \leq M, \quad i = 1..100, \; k = 1,2 \tag{25}$$

$$\sum_{i=1}^{100} y_{i2} \geq 10 - \sigma \tag{26}$$

$$\sum_{i=1}^{100} y_{i2} \leq 10 + \sigma \tag{27}$$

$$w_j \in [w_j^m, w_j^M], \; j = 1..7, \quad \sum_{j=1}^{7} w_j = 1 \tag{28}$$

$$b_2 \geq b_1 + \varepsilon, \tag{29}$$

$$b_k \in [b_k^m, b_k^M], \; k = 1,2 \tag{30}$$

$$y_{i1} + y_{i2} + y_{i3} = 1, \; i = 1..100 \tag{31}$$

$$y_{ik} \in \{0,1\}, \; i = 1..100, \; k = 1,2, \; \sigma \geq 0 \tag{32}$$

After solving this mathematical programming model we have the following results:

[1]Note that such model considers assignment examples as constraints. It is also possible to introduce slack variables as in the standard UTADIS model, using an objective function that defines how violations of assignment examples compensate violation of category size constraints.

- There exists a combination of parameter values that satisfies the credit manager request (assignment examples and $\mu(C_2) = 10\%$): $w = (0.192, 0.100, 0.200, 0.103, 0.200, 0.100, 0.105)$, $b_1 = 0.6$ and $b_2 = 0.65$

- The 10 files to analyze are $\{a_{42}, a_{52}, a_{53}, a_{54}, a_{56}, a_{57}, a_{59}, a_{61}, a_{62}, a_{70}\}$,

- The size of each category are: $\mu(C_1) = 64\%$, $\mu(C_2) = 10\%$ and $\mu(C_3) = 26\%$.

On the basis of this first result, the credit manager may want to refine the sorting model. He/She may for instance state that the file a_{52} (currently assigned to C_2 by the model) is to be rejected, *i.e.*, $a_{52} \rightarrow C_1$. This can be done by assigning the value 1 to $y_{52,1}$. He/She can impose that criterion g_6 is at least as important as g_7 ($w_6 \geq w_7$). Solving the mathematical program with these additional constraints would lead to a new solution and such interactive process continues as long as the credit manager is not satisfied with the resulting sorting model.

Conclusion

Sorting problems consist of formulating the decision problem in terms of a classification so as to assign each alternative from A to one of the predefined categories $C_1, C_2, \ldots, C_{n_{cat}}$. The assignment of an alternative a to the appropriate category should rely on the intrinsic value of a (and not on the comparison of a to other alternatives from A). On the contrary, the very nature of ranking and choice problems is to compare alternatives one to another to determine a preference order or the subset of the best one(s). Hence ranking and choice refer to relative evaluation while sorting refers to absolute evaluation.

In this paper we have motivated the use of the notion of category size in sorting problems and given a formal definition to this notion. We have shown that considering constraints on category size leads to define a new type of problem, that we call *constrained sorting problem* (CSP) that has both an absolute and relative evaluation aspect.

We have shown how to make the category size concept operational even in decision situations where the set of alternatives and/or DM's preferences are imprecise. Finally, we have given an illustration using a UTADIS like sorting method.

We deem the notion of category size and CSP open a new research avenue that ought to be pursued. On the one hand, future research may work on the design of elicitation procedures that allow DMs to specify contraints on category size, thus integrating the notion of category size in the various existing sorting methods. On the other hand, new multicriteria sorting methods might be devised to deal specifically with CSPs.

Acknowledgements:
This work has benefited from the luso-french grant n^{o}500B4 (ICCTI/Ambassade de France au Portugal). The third author was also supported by the grant SFRH/BDP/6800/2001

References

[1] C. Bana e Costa. Les problématiques de l'aide à la décision : vers l'enrichissement de la trilogie choix-tri-rangement. *RAIRO/Operations Research*, 30(2):191–216, 1996.

[2] N. Belacel. Multicriteria assignment method PROAFTN: methodology and medical application. *European Journal of Operational Research*, 125(1):175–183, August 2000.

[3] J.M. Devaud, G. Groussaud, and E. Jacquet-Lagrèze. UTADIS: Une méthode de construction de fonctions d'utilité additives rendant compte de jugements globaux. *European working group on MCDA, Bochum , Germany*, 1980.

[4] E. Jacquet-Lagrèze and Y. Siskos. Assessing a set of additive utility functions for multicriteria decision-making, the UTA method. *European Journal of Operational Research*, 10(2):151–164, 1982.

[5] E. Jacquet-Lagrèze and Y. Siskos. Preference disaggregation: 20 years of MCDA experience. *European Journal of Operational Research*, 130(2):233–245, April 2001.

[6] R. Massaglia and A. Ostanello. N-tomic: A support system for multicriteria segmentation problems. In P. Korhonen, A. Lewandowski, and J. Walenius, editors, *Multiple Criteria Decision Support*, volume 356 of *Lecture Notes in Economics and Mathematical Systems*, pages 167–174. IIASA, 1991. Proceedings of the International Workshop, Helsinki.

[7] J. Moscarola and B. Roy. Procédure automatique d'examen de dossiers fondée sur une segmentation trichotomique en présence de critères multiple. *RAIRO/Operations Research*, 11(2):145–173, May 1977.

[8] P. Perny. Multicriteria filtering methods based on concordance and non-discordance principles. *Annals of Operations Research*, 80:137–165, 1998.

[9] B. Roy. *Méthodologie multicritère d'aide à la décision.* Economica, Paris, 1985.

[10] W. Yu. *Aide multicritère à la décision dans le cadre de la problématique du tri : concepts, méthodes et applications.* PhD thesis, Université Paris-Dauphine, 1992.

[11] C. Zopounidis and M. Doumpos. PREFDIS: a multicriteria decision support system for sorting decision problems. *Computers & Operations Research*, 27(7-8):779–797, June 2000.

[12] C. Zopounidis and M. Doumpos. Multicriteria classification and sorting methods: A literature review. *European Journal of Operational Research*, 138(2):229–246, April 2002.

Appendices

Appendix A: List of alternatives used in the example

Name	g_1	g_2	g_3	g_4	g_5	g_6	g_7
a_1	14.47	14.02	18.02	7.24	22.16	3.02	58.65
a_2	14.64	14.02	18.02	5.46	22.16	3.02	39.44
a_3	14.48	11.01	18.02	17.36	22.16	20.32	39.44
a_4	14.65	11.01	14.11	6.43	22.16	8.91	39.44
a_5	14.43	11.01	18.02	29.25	22.16	31.73	39.44
a_6	14.15	11.01	18.02	19.13	49.87	14.83	58.65
a_7	15.04	14.02	10.47	18.32	12.92	8.91	39.44
a_8	14.71	14.02	18.02	6.43	12.92	3.02	39.44
a_9	14.54	14.02	18.02	4.65	12.92	3.02	58.65
a_{10}	14.59	11.01	18.02	6.43	12.92	8.91	39.44
a_{12}	14.42	11.01	18.02	4.65	12.92	8.91	58.64
a_{11}	14.72	11.01	14.11	18.32	12.92	8.97	39.44
a_{13}	14.65	14.02	18.02	6.43	22.16	14.83	39.44
a_{14}	14.52	11.01	18.02	6.43	22.16	3.02	39.44
a_{15}	14.40	7.99	18.02	6.43	22.16	8.91	39.44
a_{16}	14.23	7.99	18.02	4.65	22.16	8.91	58.65
a_{17}	14.42	11.01	18.02	16.55	12.92	3.02	58.65
a_{18}	13.66	7.99	31.49	81.60	40.64	30.70	61.59
a_{19}	13.02	15.74	18.02	7.24	79.21	42.63	79.32
a_{20}	13.39	4.98	26.96	81.60	69.01	18.77	61.59
a_{21}	13.60	4.02	22.44	4.65	31.40	31.73	58.65
a_{22}	13.75	11.01	22.44	81.13	69.01	7.88	21.71
a_{23}	13.66	11.01	14.11	70.55	69.01	18.77	42.39
a_{24}	13.80	14.02	14.11	60.43	59.11	8.91	20.24
a_{25}	13.04	7.99	22.44	40.19	40.64	6.96	100.0
a_{26}	12.97	7.99	18.02	29.66	31.40	18.77	100.0
a_{27}	13.51	14.02	18.02	29.25	22.16	8.91	39.44
a_{28}	13.26	11.01	14.11	7.24	22.16	7.88	79.32
a_{29}	13.39	11.01	18.02	17.36	22.16	8.91	39.44
a_{30}	13.71	14.02	14.11	6.43	12.92	3.02	39.44
a_{31}	13.24	11.01	14.11	4.65	12.92	6.96	100.0
a_{32}	13.54	14.02	14.11	4.65	12.92	3.02	58.65
a_{33}	13.19	11.01	18.02	7.24	12.92	7.88	79.32
a_{34}	13.59	11.01	14.11	6.43	12.92	8.91	39.44
a_{35}	13.46	11.01	18.02	5.46	12.92	8.91	39.44
a_{36}	13.42	11.01	14.11	4.65	12.92	8.91	58.65
a_{37}	13.52	11.01	14.11	6.43	22.16	3.02	39.44
a_{38}	13.39	11.01	18.02	5.46	22.16	3.02	39.44

following next page

following of the preceding page

Name	g_1	g_2	g_3	g_4	g_5	g_6	g_7
a_{39}	13.92	7.99	18.02	7.24	22.16	6.96	100.0
a_{40}	13.40	7.99	14.11	6.43	22.16	8.91	39.44
a_{42}	13.79	11.01	10.47	18.32	12.92	8.91	39.44
a_{41}	13.04	7.99	22.44	7.24	31.40	14.83	58.65
a_{43}	13.42	11.01	14.11	16.55	12.92	3.02	58.65
a_{44}	13.46	11.01	18.02	6.43	12.92	3.02	39.44
a_{45}	12.97	7.99	18.02	7.24	40.64	14.83	58.65
a_{46}	13.40	11.01	18.02	6.43	22.16	14.83	39.44
a_{47}	13.15	4.98	18.02	6.43	22.16	8.91	39.44
a_{48}	13.24	4.98	14.11	6.43	22.16	20.32	39.44
a_{49}	13.48	1.05	18.02	6.45	31.40	18.77	100.0
a_{50}	12.74	11.01	22.44	91.72	69.01	7.88	21.71
a_{51}	12.07	7.99	18.02	29.25	31.40	8.91	39.44
a_{52}	12.46	11.01	14.11	5.46	12.92	8.91	39.44
a_{53}	12.46	11.01	14.11	6.43	12.92	3.02	39.44
a_{54}	12.07	7.99	14.11	4.65	12.92	7.88	79.32
a_{55}	12.14	7.99	18.02	5.46	22.16	3.02	39.44
a_{56}	10.94	7.99	14.11	7.24	12.92	7.88	79.32
a_{57}	11.08	7.99	18.02	17.36	12.92	8.91	39.44
a_{58}	10.41	1.96	18.02	19.13	22.16	18.77	100.0
a_{59}	11.01	7.99	18.02	17.36	22.16	3.02	39.44
a_{60}	11.04	7.99	14.11	4.65	12.92	3.02	58.65
a_{61}	10.91	7.99	18.02	7.24	12.92	3.02	58.65
a_{62}	11.07	7.99	18.02	5.46	12.92	3.02	39.44
a_{63}	11.40	11.01	14.11	6.43	3.76	3.02	39.44
a_{64}	9.90	4.98	18.02	71.48	59.11	7.88	60.12
a_{65}	9.93	7.99	18.02	61.36	49.87	3.93	79.32
a_{66}	10.25	7.99	14.11	15.58	12.92	8.91	20.24
a_{67}	10.08	7.99	14.11	5.46	12.92	3.02	39.44
a_{68}	9.91	7.99	14.11	7.24	12.92	3.02	58.65
a_{69}	10.65	11.01	10.47	3.69	3.76	8.91	20.24
a_{70}	9.60	4.98	14.11	7.24	12.92	6.96	100.0
a_{71}	9.29	1.05	14.11	7.24	22.16	18.77	100.0
a_{72}	8.94	4.98	18.02	81.60	59.11	7.88	40.92
a_{73}	8.65	1.96	18.02	71.48	59.11	7.88	60.12
a_{74}	8.65	4.98	18.02	61.36	40.64	6.96	100.0
a_{75}	8.46	1.96	18.02	61.36	49.87	6.96	100.0
a_{76}	8.76	4.98	14.11	17.36	22.16	3.02	39.44
a_{77}	9.40	7.99	10.47	3.69	3.76	8.91	20.24
a_{78}	8.66	4.98	14.11	7.24	12.92	3.02	58.65
a_{79}	8.57	4.99	14.11	7.24	12.92	3.93	79.32
a_{80}	8.71	1.96	14.11	5.46	12.92	8.91	39.44

following next page

following of the preceding page

Name	g_1	g_2	g_3	g_4	g_5	g_6	g_7
a_{81}	7.58	1.96	18.02	71.48	49.87	7.88	60.12
a_{82}	7.58	4.98	18.02	61.36	31.40	6.96	100.0
a_{83}	7.31	1.96	14.11	19.13	12.92	7.88	79.32
a_{84}	8.26	7.99	10.47	3.69	3.76	3.02	20.24
a_{85}	8.45	7.99	6.74	3.69	3.76	8.91	20.24
a_{86}	6.33	1.96	18.02	61.36	31.40	6.96	100.0
a_{87}	6.46	4.98	18.02	50.78	12.92	6.96	100.0
a_{88}	6.65	4.98	14.11	5.46	3.76	14.83	39.44
a_{89}	5.28	1.05	18.02	60.43	40.64	3.02	20.24
a_{90}	4.06	1.05	18.02	39.78	12.92	3.02	39.44
a_{91}	4.64	1.96	6.74	5.46	3.76	3.02	39.44
a_{92}	5.11	4.98	3.02	3.69	3.76	3.02	20.24
a_{93}	3.68	1.96	6.74	15.58	3.76	3.02	20.24
a_{94}	3.86	1.96	3.02	3.69	3.76	3.02	20.24
a_{95}	2.56	1.96	6.74	15.58	3.76	14.83	20.24
a_{96}	2.81	1.96	3.02	3.69	5.41	3.02	20.24
a_{97}	1.04	1.05	14.11	49.90	3.76	3.02	20.24
a_{98}	1.48	1.05	3.02	15.58	3.76	3.02	20.24
a_{99}	1.68	1.96	3.02	15.58	5.41	3.02	20.24
a_{100}	0.60	1.05	0.71	3.69	5.41	3.02	20.24

Appendix B: Constraints on the utility functions used in the example

	g_1	g_2	g_3	g_4	g_5	g_6	g_7
u_j^0	0	0	0	0	0	0	0
u_j^1	$[0.25w_1, 0.6w_1]$	$[0, 0.18w_2]$	$[0, 0.16w_3]$	$[0, 0.2w_4]$	$[0, 0.125w_5]$	$[0, 0.175w_6]$	$[0, 0.3w_7]$
u_j^2	$[0.5w_1, 0.65w_1]$	$[0.22w_2, 0.42w_2]$	$[0.14w_3, 0.34w_3]$	$[0.2w_4, 0.4w_4]$	$[0.1w_5, 0.3w_5]$	$[0.18w_6, 0.45w_6]$	$[0.1w_7, 0.5w_7]$
u_j^3	$[0.65w_1, 0.8w_1]$	$[0.62w_2, 0.82w_2]$	$[0.44w_3, 0.64w_3]$	$[0.6w_4, 0.8w_4]$	$[0.7w_5, 0.9w_5]$	$[0.425w_6, 0.7w_6]$	$[0.45w_7, 0.8w_7]$
u_j^4	$[0.75w_1, 0.99w_1]$	$[0.8w_2, w_2]$	$[0.86w_3, 0.99w_3]$	$[0.8w_4, w_4]$	$[0.875w_5, w_5]$	$[0.7w_6, 0.85w_6]$	$[0.75w_7, 0.9w_7]$
u_j^5	w_1	w_2	w_3	w_4	w_5	w_6	w_7

Table 3: Shape of the partial utility functions