Comparison and links between two 2-tuple linguistic models for decision making

Isis Truck*

CHArt-EA4004 Laboratory
University Paris 8
2 rue de la Liberté
93526 Saint-Denis (FRANCE)

Abstract

This paper deals with linguistic models that may prove useful for representing the information during decision making.

Data extraction is a complex problem, especially when dealing with information coming from human beings (linguistic assertions, preferences, feelings, etc.) and several models have emerged to overcome difficulties in expressing the data. Among those models, we are interested in two of them: the 2-tuple semantic model and the 2-tuple symbolic model.

In this paper we stress on a comparison between both models and we prove that links can be made between them. An interesting result is obtained: the 2-tuple semantic model can generate a partitioning that is identical to the one that would be generated thanks to the symbolic 2-tuple model. This permits to compare the models and mix the two when there is a need to use one model or another, depending on the applications, and then to reach a consensus. In closing, an example of the use is given to demonstrate the

*Phone: +33 149 407 325. Fax: +33 149 407 348
Email address: isis.truck@univ-paris8.fr (Isis Truck)

value of the method.

Keywords: 2-tuple linguistic models, decision making, comparison

PACS: 89.20.Ff, 02.50.Le, 07.05.Mh

2010 MSC: 03E70, 03E72, 62C86, 90B50

---

1. Introduction

Decision making implies to choose from several alternatives according to a prior analysis that often involves information coming from human beings and uncertainty (Howard, 1968; Holloway, 1979). Besides, being able to catch the meaning conveyed by linguistic information is essential before making an analysis. One of the most tricky issue is to deal with human thoughts, assertions, preferences, feelings, etc. because of the intrinsic nature of natural language. Since decades, a lot of works have been done to take into account the vagueness and the imprecision of linguistic information using the fuzzy set theory (Zadeh, 1965; Bellman and Zadeh, 1970). Among those works, several computational models (Zadeh, 1975) have been proposed to manage uncertainty, especially when it is not of probabilistic nature (Martínez and Herrera, 2012). In particular, a new model has appeared in 2000 called the *2-tuple fuzzy linguistic representation model* (Herrera and Martínez, 2000) which gave birth to a kind of "2-tuple family" composed of several other similar models: the 2-tuple fuzzy linguistic model able to deal with *unbalanced linguistic term sets* (Herrera et al., 2008), the *2-tuple proportional model* (Wang and Hao, 2006, 2007), the *2-tuple semantic model* (Abchir and Truck, 2013) and the *generalized symbolic modifiers*, seen as the foundation for *2-tuple symbolic model* (Truck and Akdag, 2006). Research related to the

2-tuple linguistic model is abundant and worth considering in depth (Dong et al., 2009; Pei et al., 2010; Wei, 2010; Wang, 2011; Xu and Wang, 2011; Li and Dong, 2014; Dursun and Karsak, 2014).

Besides, it is important to consider all these improvements and more generally all those models to keep variety and flexibility in the design and implementation of decision making processes. To do so, one solution is to try to unify the various models by proposing a "superset", each model being one piece of a bigger picture (*i.e.* the superset). A vectorial approach as a unification of several 2 tuple models have been recently proposed but does not include the 2-tuple semantic model (Truck and Malenfant, 2010). Our proposal in this paper is to reconsider the 2-tuple semantic model in light of the 2-tuple symbolic model and to make a specific link between both models.

The rest of the paper is organized as follows: section 2 describes various 2-tuple models from the literature and their characteristics. The third section aims at showing that the 2-tuple symbolic model can be compared to the 2-tuple semantic model while section 4 considers a case study with two examples to demonstrate the feasibility of the approach. Section 5 addresses decision-making systems in real world application, with particular emphasis on assessments during foreign language acquisition. We finally draw some conclusions and future works in section 6.

2. Some computational models to describe data

2.1. The 2-tuple fuzzy linguistic representation model

We now focus on the first model designed to handle uncertainty in linguistic statements, keeping a simple computational model: the *2-tuple fuzzy*

linguistic representation model (Herrera and Martínez, 2000). This model considers a pair (s_i, α) where s_i is one among $g + 1$ linguistic terms from a linguistic term set S and $\alpha \in [-.5, .5[$ is called *symbolic translation* and catches the uncertainty about a term (each term being represented with a triangular fuzzy set). The underlying idea is that the result of any computation must be expressed in the *domain* (set of departure). So *domain* and *codomain* (target set) are supposed to be the same. For example, if $S = \{s_0 : very_low, s_1 : low, s_2 : medium, s_3 : high, s_4 : very_high\}$ and the computation is an arithmetic mean between $(s_2, 0)$, $(s_3, 0)$, $(s_3, 0)$ and $(s_3, 0)$ then the resulting 2-tuple is $(s_3, -.25)$ and may represent the linguistic statement "almost high", with $s_3 \in S$.

In this model, it is also required that there exist negation *Neg*, max and min operators defined over this set (Herrera et al., 2000):

(i) a negation operator $Neg(s_i) = s_j$ such that $j = g - i$

(ii) a max operator: $\max(s_i, s_j) = s_i$ if $s_i \geq s_j$

(iii) a min operator: $\min(s_i, s_j) = s_i$ if $s_i \leq s_j$

An improvement of this model has been proposed to take into account unbalanced linguistic term sets (Herrera et al., 2008) that are commonly used to describe erratic phenomenas and irregular progression in the patterns such as weather events or behavior under the influence of alcohol. The improvement is a method that uses linguistic hierarchies (Cordón et al., 2001) to build a partition on the set of departure. A linguistic hierarchy LH in a level t is a set of $n(t)$ equidistant terms, each term being represented with a triangular fuzzy set. Linguistic hierarchies are meant to express granularity. Each fuzzy unbalanced linguistic 2-tuple (s_i, α) is composed of two terms $s_j^{n(t)}$

and $s_k^{n(t')}$ from either the same or a different level of hierarchy, *i.e.* t does not necessarily equal t'. Those terms are the upside (left) and the downside (right) of the 2-tuple respectively.

2.2. The 2-tuple semantic model

The 2-tuple semantic model is inspired by the 2-tuple fuzzy linguistic model formalism (Abchir and Truck, 2013) and aims at solving issues where linguistic term sets are *very* unbalanced. Symbolic translations α are used to construct and generate the data set and the obtained 2-tuples are twofold. Indeed, they all are composed of two half 2-tuples: an upside and a downside 2-tuple, except the first and the last 2-tuples of the partition. Specifically, the linguistic input values are represented by a pair (s, v) where s is a linguistic term and v the position (a numerical value) of s on the axis.

Definition 1. *(Abchir and Truck, 2013) Let $\mathcal{S}$ be an unbalanced ordered linguistic term set and U be the numerical universe where the terms are projected. Each linguistic value is defined by a unique pair $(\mathsf{s}, \mathsf{v}) \in \mathcal{S} \times U$. The numerical distance between s_i and s_{i+1} is denoted by d_i with $d_i = \mathsf{v}_{i+1} - \mathsf{v}_i$.*

Definition 2. *(Abchir and Truck, 2013) Let $S = \{s_0, \ldots, s_p\}$ be an unbalanced linguistic label set and (s_i, α) be a linguistic 2-tuple. To support the unbalance, S is extended to several balanced linguistic label sets, each one denoted $S^{n(t)} = \{s_0^{n(t)}, \ldots, s_{n(t)-1}^{n(t)}\}$ defined in the level t of a linguistic hierarchy LH with $n(t)$ labels. There is a unique way to go from $\mathcal{S}$ (Definition 1) to S according to a partitioning algorithm detailed in (Abchir and Truck, 2013).*

Definition 3. *(Abchir and Truck, 2013) Let $l(t, n(t))$ be a level from a linguistic hierarchy. The* grain *g of $l(t, n(t))$ is defined as the distance between*

two 2-tuples $(s_i^{n(t)}, \alpha)$.

The grain g of a level $l(t, n(t))$ is denoted $g_{l(t,n(t))} = 1/(n(t) - 1)$. For instance, the grain of the second level of the hierarchy[1] is $g_{l(2,5)} = .25$. The grain of level $l(t-1, n(t-1))$ is twice the grain of level $l(t, n(t))$, *i.e.* $g_{l(t-1,n(t-1))} = 2g_{l(t,n(t))}$

The algorithm of the partitioning is not given here but the main principles are recalled. The result of the partitioning is the assignment of one or two label(s) $s_i^{n(t)}$ to each term s_k. The selection of label $s_i^{n(t)}$ depends on both the distance d_k (Definition 1) and the numerical value v_k. The search for the best level of hierarchy is the next step. The best level is the nearest one — since all of them are known in advance — *i.e.*, the level with the grain closest to d_k. Then the right $s_i^{n(t)}$ is selected to match v_k with best possible accuracy. i is chosen such that it minimizes the quantity $\min_i |\Delta^{-1}(s_i^{n(t_k)}, 0) - \mathsf{v}_k|$, with $\Delta^{-1}(s_i, \alpha)$ the function that computes the numerical equivalent value of (s_i, α), *i.e* $\Delta^{-1}(s_i, \alpha) = i + \alpha$ (see (Herrera and Martínez, 2000) for further details about Δ and Δ^{-1} functions).

As linguistic hierarchies are distributed on $[0, 1]$ by default, a scaling is needed in order that they match universe U.

The algorithm returns a set of bridge unbalanced linguistic 2-tuples with a level of granularity that may not be the same for the upside than for the downside (see an example in table 2).

[1] We recall that there is only one way to construct a level of a linguistic hierarchy.

2.3. The 2-tuple symbolic model

The third 2-tuple model comes from the *symbolic* framework proposed by De Glas at the end of the eighties (De Glas, 1989).

Unlike Zadeh's, there are no fuzzy sets in De Glas' framework because he assumes that the universe of discourse is finite and discrete. Zadeh, as for him, considers a continuous or discrete universe of discourse on which data take their values, and a continuous set of values between 0 and 1 that expresses the membership of data to values of the universe of discourse.

According to De Glas, the membership may be partial and is expressed with $x \in_\alpha A$ which means that x belongs to A with degree α or that x satisfies A with degree α. The latter is a boolean assertion. A is a multiset, α (also denotes v_α) is a linguistic term or an adverbial expression such as *very_low*, *low*, *medium*, etc. More precisely according to De Glas, "x is $v_\alpha A$" can be written "x (is v_α)A" or "x is A" is τ_α-true. Each v_α has a truth degree τ_α. To unify the notations, we denote τ_α as τ_i or τ_j.

So under De Glas' framework, there is no second dimension in the data representation anymore, in order to keep only the abscissa axis that contains the degrees associated to the possible values. These degrees belong to a scale that permits to introduce a total order relation, denoted $\leq$.

Several tools have been proposed to allow for a modification of data and tools to allow for an aggregation of data (Truck and Akdag, 2006, 2009). Data are linguistic ones and seen as symbols. Several *Generalized Symbolic Modifiers* (GSMs) have been proposed and are defined through a totally ordered set of M truth degrees $\mathcal{L}_M = \{\tau_0, \ldots, \tau_i, \ldots, \tau_{M-1}\}$ ($\tau_i \leq \tau_j \Leftrightarrow i \leq j$). Four basic operators have been defined: $\vee$ (max), $\wedge$ (min), $\neg$ (symbolic

negation, with $\neg\tau_j = \tau_{M-j-1}$) and the Łukasiewicz implication $\rightarrow_L$: $\tau_i \rightarrow_L \tau_j = \min(\tau_{M-1}, \tau_{M-1-(i-j)})$ (Truck and Akdag, 2006).

Truth degree $\tau_{i'}$ is computed according to a GSM m with a radius ρ, denoted m_ρ. Actually m_ρ modifies the pair $(\tau_i, \mathcal{L}_M)$ into another pair $(\tau_{i'}, \mathcal{L}_{M'})$.

Definition 4. *(Truck and Akdag, 2006)*

$$\begin{array}{rcl} m_\rho \colon \mathcal{L}_M & \rightarrow & \mathcal{L}_{M'} \\ \tau_i & \mapsto & \tau_{i'} \end{array}$$

The position of degree in the scale is denoted $p(\tau_i) = i$. A proportion is associated with each linguistic degree: $\mathrm{Prop}(\tau_i) = \frac{p(\tau_i)}{M-1}$.

Three distinct GSM families have been defined: weakening, reinforcing (see Table 1) and central ones (see Definition 5 for an example of such a GSM, called DC for ***d****ilation* and ***c****entral*).

Definition 5. *(Truck and Akdag, 2006)*

$$\mathrm{DC}(\rho) = \begin{cases} \tau_{i'} = \tau_{i\rho} \\ \mathcal{L}_{M'} = \mathcal{L}_{M\rho-\rho+1} \end{cases}$$

3. Using the 2-tuple symbolic model to build a meaningful data partition

3.1. Preliminaries

The GSMs used in the 2-tuple symbolic model are tools which modify pairs (*value*, *scale*) greatly or slightly and which possibly change the scale. The scale is either eroded or dilated, or possibly unchanged. Thus, parallels can be drawn between those GSMs and the change of linguistic hierarchy level

MODE NATURE	Weakening	Reinforcing
Erosion	$\tau_{i'} = \tau_{\max(0,i-\rho)}$ $\mathcal{L}_{M'} = \mathcal{L}_{\max(1,M-\rho)}$ EW(ρ)	$\tau_{i'} = \tau_i$ $\mathcal{L}_{M'} = \mathcal{L}_{\max(i+1,M-\rho)}$ ER(ρ) $\tau_{i'} = \tau_{\min(i+\rho,M-\rho-1)}$ $\mathcal{L}_{M'} = \mathcal{L}_{\max(1,M-\rho)}$ ER$'$(ρ)
Dilation	$\tau_{i'} = \tau_i$ $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ DW(ρ) $\tau_{i'} = \tau_{\max(0,i-\rho)}$ $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ DW$'$(ρ)	$\tau_{i'} = \tau_{i+\rho}$ $\mathcal{L}_{M'} = \mathcal{L}_{M+\rho}$ DR(ρ)
Conservation	$\tau_{i'} = \tau_{\max(0,i-\rho)}$ $\mathcal{L}_{M'} = \mathcal{L}_M$ CW(ρ)	$\tau_{i'} = \tau_{\min(i+\rho,M-1)}$ $\mathcal{L}_{M'} = \mathcal{L}_M$ CR(ρ)

Table 1: Summary of weakening and reinforcing GSMs. (Truck and Akdag, 2006)

through grain g (from Definition 3) underlying the partitioning algorithm that achieves the semantic 2-tuples.

Indeed, the 2-tuple semantic model as well as the symbolic 2-tuple model, do not differentiate between the various hierarchies as long as the approximation is adequate or good enough to express the data without loss of information. In particular, using a coarse or a thin grain scale to express a value is considered as strictly equivalent with the symbolic 2-tuples. Only the ratio $\text{Prop}(\tau_i, \mathcal{L}_M)$ counts. For instance, the pair $(\tau_i, \mathcal{L}_M) = (\tau_1, \mathcal{L}_3)$ is strictly equivalent to the pair $(\tau_2, \mathcal{L}_5)$ or the pair $(\tau_4, \mathcal{L}_9)$ because $\text{Prop}(\tau_1, \mathcal{L}_3) = \frac{1}{2} = \text{Prop}(\tau_2, \mathcal{L}_5) = \frac{2}{4} = \text{Prop}(\tau_4, \mathcal{L}_9) = \frac{4}{8}$. On the contrary, they are not really equivalent when considering semantic 2-tuples because the partitioning algorithm always tries to find the coarsest grain scale to decrease computation time.

3.2. A new partitioning algorithm for the 2-tuple symbolic model

We now propose to build a partitioning using GSMs in the 2-tuple symbolic model.

Several values must first be taken into account. Those values are not known in advance and nor is it known how many there are. This means that the partitioning is **temporal**, being calculated **on the fly** and the input values are **not always ordered**.

The algorithm is the following (see Algorithm 1): first, the input values denoted $A, B, C...$ are either zero or positive integers (indeed the value i from τ_i is zero or a positive integer). Those integers are denoted val_A, val_B, val_C, etc.

Thus the input values have to be transformed into positive integers by means of a multiplying factor γ to be computed.

Each value is then transformed into a pair $(\tau_i, \mathcal{L}_M)$ according to the following rule: the M chosen value equals $\max(2, val + 1)$ because $M \geq 2$ by nature, and τ_i equals τ_{val}. The procedure being iterative, let us denote this first M: M_1.

If the next input value cannot be expressed inside the current scale (*i.e.* if $val_B \geq M_1$) then it is necessary to choose another M, called M_2. M_2 equals $val_B + 1$. The pair (B, val_B) can now be defined as being equivalent to $(\tau_{val_B}, \mathcal{L}_{M_2})$.

Besides, as M has changed (M_1 became M_2), it is compulsory to recompute the previous pairs in this new scale of M truth degrees (here we must recompute the correspondence between (A, val_A) and the symbolic 2-tuples). As a result, (A, val_A) who was equal to $(\tau_{val_A}, \mathcal{L}_{M_1})$ is transformed

into $(\tau_{val_A}, \mathcal{L}_{M_2})$ **through a DW GSM**: DW$(M_2 - M_1)$.

It continues the partitioning process for each value, keeping in mind that coefficient γ should always lead to integer values. If not, γ (denoted γ_{old}) must be changed into a new value $\gamma = \gamma_{old} * c$, with c a positive non-zero multiplying factor.

Then all the previous pairs have to be recomputed **using a DC GSM**: DC(c). New pairs $(\tau_{val}, \mathcal{L}_{M_2})$ are thus obtained.

The iterative steps only stop when all the input values have been processed.

Algorithm 1 Partitioning algorithm using GSMs

Require: $A, B, C, ...$ are the input values denoted by pairs (A_0, val_{A_0}), $(A_1, val_{A_1}), ...,$ $(A_i, val_{A_i}), ...$ (the last pair is known only at runtime since the cardinality is unknown ahead of time)

$\gamma \leftarrow 1;\ M \leftarrow 2;\ i \leftarrow j \leftarrow 0$

while there are input values A_i **do**
  if $(val_{A_i} * \gamma) \notin \mathbb{N}$ **then**
    $\gamma_{old} \leftarrow \gamma$
    $\gamma \leftarrow \gamma_{old} * c$ such that $(val_{A_i} * \gamma) \in \mathbb{N}$
    while there are A_j values already processed **do**
      recompute val_{A_j} (*i.e.* $(\tau_\zeta, \mathcal{L}_M)$) using DC($c$): $(\tau_{\zeta_{old}}, \mathcal{L}_{M_{old}}) \xrightarrow{\text{DC}(c)} (\tau_\zeta, \mathcal{L}_M)$
      $j \leftarrow j + 1$
    end while
    $M_{old} \leftarrow M$
    $j \leftarrow 0$
  end if
  $val_{A_i} \leftarrow val_{A_i} * \gamma$
  if $val_{A_i} \geq M$ **then**
    $M_{old} \leftarrow M$
    $M \leftarrow val_{A_i} + 1$
    while there are A_j valued already processed **do**
      recompute val_{A_j} (*i.e.* $(\tau_\zeta, \mathcal{L}_M)$) using DW($M - M_{old}$):
        $(\tau_{\zeta_{old}}, \mathcal{L}_{M_{old}}) \xrightarrow{\text{DW}(M - M_{old})} (\tau_\zeta, \mathcal{L}_M)$
      $j \leftarrow j + 1$
    end while
  end if
  compute val_{A_i}: its associated pair is $(\tau_{val_{A_i}}, \mathcal{L}_M)$
  $j \leftarrow 0$
end while
return the set of associated pairs $(\tau_\zeta, \mathcal{L}_M)$ to each (A_i, val_{A_i})

4. Case study

We now consider two examples. The first one deals with the blood alcohol concentration (BAC in percentage) in the USA, while the second one focuses on studies on foreign language learning.

4.1. First example

In this example, it is focused on five main values: 0% means no alcohol, .05% is the legal limit for drivers under 21, .065% is an intermediate value (illegal for young drivers but legal for the others), .08% is the legal limit for drivers older than 21 and .3% is considered as the BAC level where risk of death is possible (Abchir and Truck, 2013).

The set of pairs (s, v) is thus the following:
$\{(\mathit{NoAlcohol}, .0), (\mathit{YoungLegalLimit}, .05), (\mathit{Intermediate}, .065), (\mathit{LegalLimit}, .08), (\mathit{RiskOfDeath}, .3)\}$.

Table 2 recalls the resulting semantic 2-tuples obtained by the partitioning algorithm from (Abchir and Truck, 2013).

- The first input value to be used to construct the partition is $(A, 0)$: it represents in percentage the absence of alcohol in the blood. With the symbolic 2-tuple notation, $(A, 0)$ can be written $(\tau_i, \mathcal{L}_j) = (\tau_0, \mathcal{L}_2)$ or $(\tau_i, \mathcal{L}_j) = (\tau_0, \mathcal{L}_3)$ or $(\tau_i, \mathcal{L}_j) = (\tau_0, \mathcal{L}_4)$, etc.;

- the second input value is $(B, .05)$: it represents in percentage the legal limit of alcohol in the blood for drivers under age 21;

- the third input value is $(C, .06)$: it represents in percentage the early stage to behavioral disinhibition (the value has rounded up from .065

linguistic term	level	semantic 2-tuple
NoAlcohol_R	$l(3,9)$	$(s_0^9, 0)$
YoungLegalLimit_L	$l(3,9)$	$(s_1^9, .0417)$
YoungLegalLimit_R	$l(5,33)$	$(s_5^{33}, .01)$
Intermediate_L	$l(5,33)$	$(s_6^{33}, 0)$
Intermediate_R	$l(4,17)$	$(s_3^{17}, 0)$
LegalLimit_L	$l(4,17)$	$(s_4^{17}, .0167)$
LegalLimit_R	$l(1,3)$	$(s_1^3, -.2333)$
RiskOfDeath_R	$l(1,3)$	$(s_1^3, 0)$

Table 2: The 2-tuple set for the BAC example where L and R are the upside (L for *left*) and downside (R for *right*) of labels respectively. (Abchir and Truck, 2013)

to .06 to shorten the following example, but the algorithm works with any value as long as it is positive);

- the fourth input value is $(D, .08)$: it represents in percentage the legal limit of alcohol in the blood for drivers that are 21 years of age or older;
- the fifth input value is $(E, .3)$: it represents in percentage the limit at which there is a potential for death or serious harm to life.

We now take two different assumptions according to the order in which the input values appear.

4.1.1. First assumption

Let us assume that the input values appear in the following order: B, C, A, E, D. Once the algorithm 1 is complete, we shall obtain:

(i) for $(B, .05)$ value

- $(\gamma * .05) \notin \mathbb{N}$ because $\gamma = 1$. As a result, $\gamma_{old} = 1$, $\gamma = c = 100$ and $val_B = .05 * 100 = 5$
- $val_B \geq 2$ then $M_{old} = 2$ and $M = 5 + 1 = 6$. Scale $\mathcal{L}_M$ equals $\mathcal{L}_6$
- $(B, .05)$ is thus equal to $(\tau_{val_B}, \mathcal{L}_6)$ *i.e.* $(\tau_5, \mathcal{L}_6)$

(ii) for $(C, .06)$ value

- $(\gamma * .06) \in \mathbb{N}$ because $\gamma = 100$
- $val_C = .06 * 100 = 6$
- $val_C \geq 6$ then $M_{old} = 6$ and $M = 6 + 1 = 7$. Scale $\mathcal{L}_M$ equals $\mathcal{L}_7$
- because of new scale $\mathcal{L}_M$, to reflect the changes, B must be recomputed using $\mathrm{DW}(M - M_{old}) = \mathrm{DW}(7-6) = \mathrm{DW}(1)$: $(\tau_5, \mathcal{L}_6) \overset{\mathrm{DW}(1)}{\longrightarrow} (\tau_5, \mathcal{L}_7)$
- As a result, $(C, .6)$ is thus equal to $(\tau_6, \mathcal{L}_7)$

(iii) for $(A, 0)$ value

- $(\gamma * 0) \in \mathbb{N}$ because $\gamma = 100$
- $val_A = 0 * 100 = 0$
- $val_A < 7$ so no recalculation
- $(A, 0)$ is thus equal to $(\tau_0, \mathcal{L}_7)$

(iv) for $(E, .3)$ value

- $(\gamma * .3) \in \mathbb{N}$ because $\gamma = 100$
- $val_E = .3 * 100 = 30$
- $val_E \geq 7$ then $M_{old} = 7$ and $M = 30 + 1 = 31$. Scale $\mathcal{L}_M$ equals $\mathcal{L}_{31}$
- B must be recomputed using $\mathrm{DW}(31-7) = \mathrm{DW}(24)$: $(\tau_5, \mathcal{L}_7) \overset{\mathrm{DW}(24)}{\longrightarrow} (\tau_5, \mathcal{L}_{31})$
- C must be recomputed using $\mathrm{DW}(24)$: $(\tau_6, \mathcal{L}_7) \overset{\mathrm{DW}(24)}{\longrightarrow} (\tau_6, \mathcal{L}_{31})$
- A must be recomputed using $\mathrm{DW}(24)$: $(\tau_0, \mathcal{L}_7) \overset{\mathrm{DW}(24)}{\longrightarrow} (\tau_0, \mathcal{L}_{31})$
- As a result, $(E, .3)$ is thus equal to $(\tau_{30}, \mathcal{L}_{31})$

(v) for $(D, .08)$ value

- $(\gamma * .08) \in \mathbb{N}$ because $\gamma = 100$
- $val_D = .08 * 100 = 8$
- $val_D < 31$ so no recalculation
- $(D, .08)$ is thus equal to $(\tau_8, \mathcal{L}_{31})$

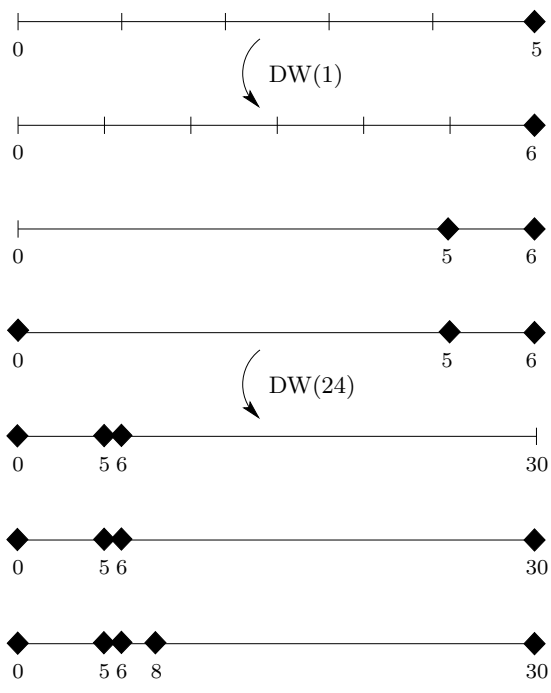
Figure 1 summarizes the execution of the algorithm with those five ordered input values.

4.1.2. Second assumption.

Let us assume that the input values appear in the following order: E, B, A, C, D. Once the algorithm 1 is complete, we shall obtain:

(i) for $(E, .3)$ value

- $(\gamma * .3) \notin \mathbb{N}$ because $\gamma = 1$. As a result, $\gamma_{old} = 1$, $\gamma = c = 10$ and $val_E = .3 * 10 = 3$



step	$(A, 0)$	$(B, .05)$	$(C, .06)$	$(D, .08)$	$(E, .3)$
(i)		$(\tau_5, \mathcal{L}_6)$			
(ii)		$(\tau_5, \mathcal{L}_7)$			
		$(\tau_5, \mathcal{L}_7)$	$(\tau_5, \mathcal{L}_7)$		
(iii)	$(\tau_0, \mathcal{L}_7)$	$(\tau_5, \mathcal{L}_7)$	$(\tau_6, \mathcal{L}_7)$		
(iv)	$(\tau_0, \mathcal{L}_{31})$	$(\tau_5, \mathcal{L}_{31})$	$(\tau_6, \mathcal{L}_{31})$		
	$(\tau_0, \mathcal{L}_{31})$	$(\tau_5, \mathcal{L}_{31})$	$(\tau_6, \mathcal{L}_{31})$		$(\tau_{30}, \mathcal{L}_{31})$
(v)	$(\tau_0, \mathcal{L}_{31})$	$(\tau_5, \mathcal{L}_{31})$	$(\tau_6, \mathcal{L}_{31})$	$(\tau_8, \mathcal{L}_{31})$	$(\tau_{30}, \mathcal{L}_{31})$

Figure 1: Temporal partitioning with GSMs, first assumption.

- $val_E \geq 2$ then $M_{old} = 2$ and $M = 3 + 1 = 4$. Scale $\mathcal{L}_M$ equals $\mathcal{L}_4$
- $(E, .3)$ is thus equal to $(\tau_{val_E}, \mathcal{L}_4)$ *i.e.* $(\tau_3, \mathcal{L}_4)$

(ii) for $(B, .05)$ value

- $(\gamma * .05) \notin \mathbb{N}$ because $\gamma = 10$. As a result, $\gamma_{old} = 10$, $c = 10$, $\gamma = 100$
- E must be recomputed using $\text{DC}(c) = \text{DC}(10)$: $(\tau_3, \mathcal{L}_4) \overset{\text{DC}(10)}{\longrightarrow} (\tau_{30}, \mathcal{L}_{31})$
- $M_{old} = 31$

- $val_B = .05 * 100 = 5$
- $val_B < 31$ so no recalculation
- $(B, .05)$ is thus equal to $(\tau_5, \mathcal{L}_{31})$

(iii) for $(A, 0)$ value

- $(\gamma * 0) \in \mathbb{N}$ because $\gamma = 100$
- $val_A = 0 * 100 = 0$
- $val_A < 31$ so no recalculation
- $(A, 0)$ is thus equal to $(\tau_0, \mathcal{L}_{31})$

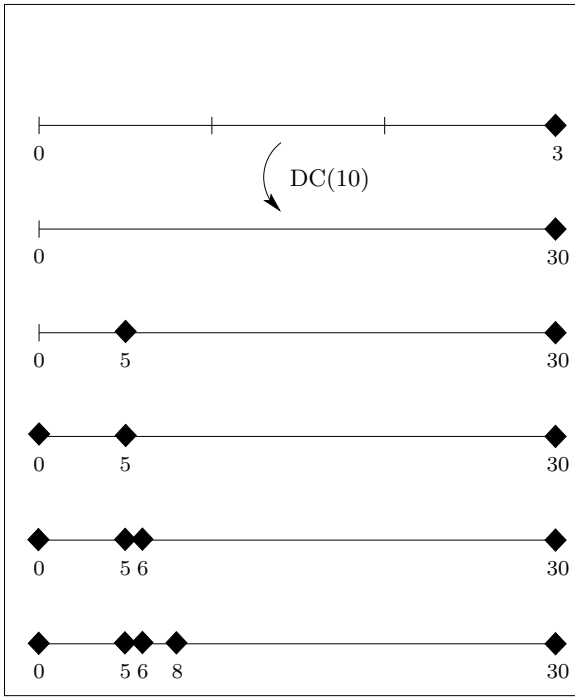
(iv) for $(C, .06)$ value

- $(\gamma * .06) \in \mathbb{N}$ because $\gamma = 100$
- $val_C = .06 * 100 = 6$
- $val_C < 31$ so no recalculation
- $(C, .06)$ is thus equal to $(\tau_6, \mathcal{L}_{31})$

(v) for $(D, .08)$ value

- $(\gamma * .08) \in \mathbb{N}$ because $\gamma = 100$
- $val_D = .08 * 100 = 8$
- $val_D < 31$ so no recalculation
- $(D, .08)$ is thus equal to $(\tau_8, \mathcal{L}_{31})$

Figure 2 summarizes the execution of the algorithm with those five ordered input values.



step	$(A, 0)$	$(B, .05)$	$(C, .06)$	$(D, .08)$	$(E, .3)$
(i)					$(\tau_3, \mathcal{L}_4)$
(ii)					$(\tau_{30}, \mathcal{L}_{31})$
		$(\tau_5, \mathcal{L}_{31})$			$(\tau_{30}, \mathcal{L}_{31})$
(iii)	$(\tau_0, \mathcal{L}_{31})$	$(\tau_5, \mathcal{L}_{31})$			$(\tau_{30}, \mathcal{L}_{31})$
(iv)	$(\tau_0, \mathcal{L}_{31})$	$(\tau_5, \mathcal{L}_{31})$	$(\tau_6, \mathcal{L}_{31})$		$(\tau_{30}, \mathcal{L}_{31})$
(v)	$(\tau_0, \mathcal{L}_{31})$	$(\tau_5, \mathcal{L}_{31})$	$(\tau_6, \mathcal{L}_{31})$	$(\tau_8, \mathcal{L}_{31})$	$(\tau_{30}, \mathcal{L}_{31})$

Figure 2: Temporal partitioning with GSMs, second assumption.

4.2. Second example

The second example deals with assessments and observations made as part of studies on foreign language acquisition.

Let us consider learners answering questions from a teacher who wants to know their level during and at the end of the lesson. The accuracy (and/or relevance) of the responses given by students is the *observation*. For each student, *Accuracy* is the result of the aggregation of several assessments. Indeed, each question is related to one or more concepts and skills such as *morphosyntax, syntax, lexicon*, etc. Thus several questions are grouped to reflect a certain skill.

For each skill, *Accuracy* is expressed in percentage and has the six following values: 0% means that the student only got the wrong answers and did not understand the concept at all, nor developed the skill; 23% means that the student generally got the wrong answers; 50% means that half of the answers were right and the other half were wrong; 77% means that the student understood many principles of the concept, but weaknesses are still observed; 89% means that most of the principles have been understood; 100% means that the student gave only right answers and that the skill is acquired.

After normalization, the set of pairs (s, v) is the following:
$\{(\mathit{VeryBad}, 0), (\mathit{Bad}, .23), (\mathit{Medium}, .5), (\mathit{Good}, .77), (\mathit{VeryGood}, .89), (\mathit{Perfect}, 1)\}$.

Table 3 gives the resulting semantic 2-tuples obtained by the partitioning algorithm from (Abchir and Truck, 2013).

- The first input value to be used to construct the partition is $(A, 0)$: it represents the value *VeryBad*. $(A, 0)$ can be written $(\tau_i, \mathcal{L}_j) = (\tau_0, \mathcal{L}_2)$ or $(\tau_i, \mathcal{L}_j) = (\tau_0, \mathcal{L}_3)$ or $(\tau_i, \mathcal{L}_j) = (\tau_0, \mathcal{L}_4)$, etc.;
- the second input value is $(B, .23)$: it represents the value *Bad*;
- the third input value is $(C, .5)$: it represents the value *Medium*;
- the fourth input value is $(D, .77)$: it represents the value *Good*;
- the fifth input value is $(E, .89)$: it represents the value *VeryGood*;
- the sixth input value is $(F, 1)$: it represents the value *Excellent*.

As in the previous example, we take two different assumptions according to the order in which the input values appear.

linguistic term	level	semantic 2-tuple
VeryBad_R	$l(3,9)$	$(s_0^9, 0)$
Bad_L	$l(3,9)$	$(s_2^9, -.16)$
Bad_R	$l(2,5)$	$(s_5^{33}, -.08)$
Medium_L	$l(2,5)$	$(s_6^{33}, 0)$
Medium_R	$l(2,5)$	$(s_3^{17}, 0)$
Good_L	$l(2,5)$	$(s_4^{17}, .08)$
Good_R	$l(3,9)$	$(s_1^3, .16)$
VeryGood_L	$l(3,9)$	$(s_4^{17}, .12)$
VeryGood_R	$l(4,17)$	$(s_1^3, .24)$
Perfect_R	$l(4,17)$	$(s_1^3, 0)$

Table 3: The semantic 2-tuple set for the *foreign language learning* example where L and R are the upside and downside of labels respectively.

4.2.1. First assumption

Let us assume that the input values appear in the following order: E, B, D, C, F, A. Once the algorithm 1 is complete, we shall obtain:

(i) for $(E, .89)$ value

- $(\gamma * .89) \notin \mathbb{N}$ because $\gamma = 1$. As a result, $\gamma_{old} = 1$, $\gamma = c = 100$ and $val_E = .89 * 100 = 89$
- $val_E \geq 2$ then $M_{old} = 2$ and $M = 89 + 1 = 90$. Scale $\mathcal{L}_M$ equals $\mathcal{L}_{90}$
- $(E, .89)$ is thus equal to $(\tau_{val_E}, \mathcal{L}_{90})$ *i.e.* $(\tau_{89}, \mathcal{L}_{90})$

(ii) for $(B, .23)$ value

- $(\gamma * .23) \in \mathbb{N}$ because $\gamma = 100$
- $val_B = .23 * 100 = 23$
- $val_B < 90$ so no computations here
- As a result, $(B, .23)$ is thus equal to $(\tau_{23}, \mathcal{L}_{90})$

(iii) for $(D, .77)$ value

- $(\gamma * .77) \in \mathbb{N}$ because $\gamma = 100$
- $val_D = .77 * 100 = 77$
- $val_D < 90$ so no computations here
- As a result, $(D, .77)$ is thus equal to $(\tau_{77}, \mathcal{L}_{90})$

(iv) for $(C, .5)$ value

- $(\gamma * .5) \in \mathbb{N}$ because $\gamma = 100$
- $val_C = .5 * 100 = 50$
- $val_C < 90$ so no computations here
- As a result, $(C, .5)$ is thus equal to $(\tau_{50}, \mathcal{L}_{90})$

(v) for $(F, 1)$ value

- $(\gamma * 1) \in \mathbb{N}$ because $\gamma = 100$
- $val_F = 1 * 100 = 100$
- $val_F \geq 90$ then $M_{old} = 90$ and $M = 100 + 1 = 101$. Scale $\mathcal{L}_M$ equals $\mathcal{L}_{101}$
- E must be recomputed using $\mathrm{DW}(101 - 90) = \mathrm{DW}(11)$: $(\tau_{89}, \mathcal{L}_{90}) \overset{\mathrm{DW}(11)}{\longrightarrow} (\tau_{89}, \mathcal{L}_{101})$

Table 4: Execution of the algorithm for the order E, B, D, C, F, A.

step	$(A, 0)$	$(B, .23)$	$(C, .5)$	$(D, .77)$	$(E, .89)$	$(F, 1)$
(i)					$(\tau_{89}, \mathcal{L}_{90})$	
(ii)		$(\tau_{23}, \mathcal{L}_{90})$			$(\tau_{89}, \mathcal{L}_{90})$	
(iii)		$(\tau_{23}, \mathcal{L}_{90})$		$(\tau_{77}, \mathcal{L}_{90})$	$(\tau_{89}, \mathcal{L}_{90})$	
(iv)		$(\tau_{23}, \mathcal{L}_{90})$	$(\tau_{50}, \mathcal{L}_{90})$	$(\tau_{77}, \mathcal{L}_{90})$	$(\tau_{89}, \mathcal{L}_{90})$	
(v)		$(\tau_{23}, \mathcal{L}_{101})$	$(\tau_{50}, \mathcal{L}_{101})$	$(\tau_{77}, \mathcal{L}_{101})$	$(\tau_{89}, \mathcal{L}_{101})$	$(\tau_{100}, \mathcal{L}_{101})$
(vi)	$(\tau_0, \mathcal{L}_{101})$	$(\tau_{23}, \mathcal{L}_{101})$	$(\tau_{50}, \mathcal{L}_{101})$	$(\tau_{77}, \mathcal{L}_{101})$	$(\tau_{89}, \mathcal{L}_{101})$	$(\tau_{100}, \mathcal{L}_{101})$

- B must be recomputed using DW(11): $(\tau_{23}, \mathcal{L}_{90}) \overset{\text{DW(11)}}{\longrightarrow} (\tau_{23}, \mathcal{L}_{101})$
- D must be recomputed using DW(11): $(\tau_{77}, \mathcal{L}_{90}) \overset{\text{DW(11)}}{\longrightarrow} (\tau_{77}, \mathcal{L}_{101})$
- C must be recomputed using DW(11): $(\tau_{50}, \mathcal{L}_{90}) \overset{\text{DW(11)}}{\longrightarrow} (\tau_{50}, \mathcal{L}_{101})$
- As a result, $(F, 1)$ is thus equal to $(\tau_{100}, \mathcal{L}_{101})$

(vi) for $(A, 0)$ value

- $(\gamma * 0) \in \mathbb{N}$
- $val_A = 0 * 100 = 0$
- $val_A < 101$ so no computations here
- As a result, $(A, 0)$ is thus equal to $(\tau_0, \mathcal{L}_{101})$

As in the first example, table 4 summarizes the execution of the algorithm with those six ordered input values.

Table 5: Execution of the algorithm for the order B,A,E,D,C,F.

step	$(A, 0)$	$(B, .23)$	$(C, .5)$	$(D, .77)$	$(E, .89)$	$(F, 1)$
(i)		$(\tau_{23}, \mathcal{L}_{24})$				
(ii)	$(\tau_0, \mathcal{L}_{24})$	$(\tau_{23}, \mathcal{L}_{24})$				
(iii)	$(\tau_0, \mathcal{L}_{90})$	$(\tau_{23}, \mathcal{L}_{90})$			$(\tau_{89}, \mathcal{L}_{90})$	
(iv)	$(\tau_0, \mathcal{L}_{90})$	$(\tau_{23}, \mathcal{L}_{90})$		$(\tau_{77}, \mathcal{L}_{90})$	$(\tau_{89}, \mathcal{L}_{90})$	
(v)	$(\tau_0, \mathcal{L}_{90})$	$(\tau_{23}, \mathcal{L}_{90})$	$(\tau_{50}, \mathcal{L}_{90})$	$(\tau_{77}, \mathcal{L}_{90})$	$(\tau_{89}, \mathcal{L}_{90})$	
(vi)	$(\tau_0, \mathcal{L}_{101})$	$(\tau_{23}, \mathcal{L}_{101})$	$(\tau_{50}, \mathcal{L}_{101})$	$(\tau_{77}, \mathcal{L}_{101})$	$(\tau_{89}, \mathcal{L}_{101})$	$(\tau_{100}, \mathcal{L}_{101})$

4.2.2. Second assumption

Let us assume that the input values appear in the following order: B, A, E, D, C, F. Running the algorithm 1, we obtain the execution summarized in table 5.

4.3. Results

For each example, when we compare the results obtained with the semantic 2-tuples in table 2 (resp. table 3) and those obtained with this new algorithm (see figures 1 and 2, resp. tables 4 and 5), it is possible to draw the following parallels: the pairs[2] $(s_j^{n(t)}, \alpha_j)$, without any distinction of upside (L) or downside (R) semantic 2-tuples, are linked with $(\tau_i, \mathcal{L}_M)$ values according to the two following assertions:

- (s_j, α_j) is equivalent to τ_i and

[2]The notation α_j is used to distinguish from the various α values obtained in each 2-tuple.

- $n(t)$ is equivalent to $\mathcal{L}_M$.

Moreover, figures 1 and 2 (resp. tables 4 and 5) prove that the resulting partition is the same **regardless the order** that the input values occur (see step (v) — resp. step (vi) — in both executions).

This algorithm we propose in this paper offers a **temporal** and **unique** partitioning (the resulting partition will always be the same with the same input value set) in exhibiting a final scale that *resembles* the universe obtained with the semantic 2-tuples. Of course, the resemblance stops where the fuzzy sets attached to the 2-tuples begin.

These results are interesting if we want to use various 2-tuple representation models (by necessity, by desire or being forced to do it): it is thus possible to unify those models or to go from one model to another more easily. In decision-making systems, this can be very useful.

5. Decision systems using both linguistic 2-tuple models

5.1. Decision-making systems

According to Zimmermann, prescriptive decision theory is defined as follows:

Definition 6. *(Zimmermann, 1987) Given the set of feasible actions, A, the set of relevant states, S, the set of resulting events, E, and a (rational) utility function, u — which orders the space of events with respect to their desirability — the optimal decision under certainty is the choice of that action which leads to the event with the highest utility, such a decision can be described properly by the quadruple $\{A, S, E, u\}$.*

In a fuzzy environment, the decision is the intersection of fuzzy constraints and fuzzy objective functions. The fuzzy constraints and the feasible actions (or alternatives) are often expressed through linguistic values because they come from human beings (Bellman and Zadeh, 1970; Kacprzyk and Yager, 1990; Chen and Mon, 1994; Chang and Chen, 1994; Yager et al., 1994; Kumar and Bauer, 2009).

Among the decision-making systems are the *assessment decision-making systems* where the results of assessment can inform decision-making. In 2003, McMillan suggested that measurement specialists need to adapt technical testing principles to be more relevant to the nature of classroom assessment decision making (McMillan, 2003). The aim of such decision systems is to increase the performance of both the candidates and the teacher who understands better the potential difficulties and can adapt his course. The assessments made are designed to identify capability gaps such as skills, processes, etc. Authors such as Shinn focus on curriculum-based measurement — a way to monitor the candidates, providing current, week-by-week information on the progress people are making — to assess people (especially special children) (Shinn, 1989).

5.2. Real-world application

Let us return to the studies on foreign language acquisition as an example.

In this real-world application (the project is called VILLA, see for instance (Rast et al., 2011)), a useful but tricky thing is the large amount of production and performance data for learners to analyze. At first sight, identify, interpret and analyze those data is not trivial. As noted earlier, the accuracy of the response given by the learners is in fact the result of a weighted aggregation of



Figure 3: An example of a confidence scale used in our study.

assessments related to several questions dealing with a same skill. Another observation is the response time for each question. It is also the result of weighted aggregations. Both *Accuracy* and *ResponseTime* yield pairs (s, v) easily expressible in semantic 2-tuples.

But other observations can be features coming from subjective assessments, judgments, feelings, etc. given by teachers. For instance, a feature applies to the attitude or the behavior of the student, as perceived by the teacher. For each set of questions, a linguistic scale (of the symbolic 2-tuple type) is used to assess the confidence of the learner when answering questions. Figure 3 shows such a scale.

Thus, there is some variation in the nature of those features (aggregations of observations), as described above. To make a decision, it is compulsory to take into account all the features in expressing them in a same model. In this case, the semantic 2-tuple model is the most appropriate because the confidence scale (symbolic 2-tuples) can be transformed into semantic 2-tuples through algorithm 1.

The next step needs to cluster learners in order to characterize them and understand them better (the way they acquire a new language, etc.). This goes beyond this study and will be the subject of upcoming papers with

particular emphasis on clustering in such domains.

6. Conclusions

The novelty in this paper is a formal outcome, which links two linguistic 2-tuple models. Indeed we have proposed a thought about how to compare or unify two different 2-tuple representation models: the 2-tuple semantic model and the 2-tuple symbolic model. The first model is useful to catch the semantics of a linguistic statement, especially when it deals with erratic phenomenas whose patterns are subject to irregular progression, such as legislation on blood alcohol levels where linguistic terms involved are unsymmetrically distributed on the axis.

The second one is based on the symbolic framework and does not consider any fuzzy set. This model is based on modifiers which change the ratio of the linguistic value expressed.

The result obtained in this paper is an algorithm that constructs a partition (with tools of the second model) similar to the one that would be obtained with the first model, regardless the order of appearance of the input values. The only difference between both partitions is the fact that the symbolic 2-tuples don't make use of fuzzy sets. So the algorithm suggests a temporal and unique partitioning useful especially when the number of input values is not known.

Thus the conclusions that can be drawn are the following: parallels between both models have been established. One model or the other can be used to express the data, as soon as the underlying fuzzy sets are not needed in future computations.

Future works will focus on global unification of 2-tuple models (including linguistic models such as the fuzzy linguistic 2-tuple model or the proportional fuzzy 2-tuple model). Indeed this contribution should permit to understand the 2-tuple semantic model better, being viewed and perceived through the prism of the 2-tuple symbolic model. And *vice versa*. It should permit to carry on establishing a general model that would comprise all the 2-tuple representation models.

References

Abchir, M., Truck, I., 2013. Towards an Extension of the 2-tuple Linguistic Model to Deal With Unbalanced Linguistic Term sets. Kybernetika 49 (1), 164–180.

Bellman, R. E., Zadeh, L. A., 1970. Decision-making in a fuzzy environment. Management science 17 (4), 141–164.

Chang, P.-L., Chen, Y.-C., 1994. A fuzzy multi-criteria decision making method for technology transfer strategy selection in biotechnology. Fuzzy Sets and Systems 63 (2), 131–139.

Chen, C.-H., Mon, D.-L., 1994. Evaluating Weapon System by Analytical Hierarchy Process Based on Fuzzy Scales. Fuzzy Sets and Systems 63 (1), 1–10.

Cordón, O., Herrera, F., Zwir, I., 2001. Linguistic modeling by hierarchical systems of linguistic rules. IEEE Transactions on Fuzzy Systems 10 (1), 2–20.

De Glas, M., 1989. Knowledge representation in a fuzzy setting. Report 89/48, Université Paris 6.

Dong, Y., Xu, Y., Yu, S., 2009. Computing the numerical scale of the linguistic term set for the 2-tuple fuzzy linguistic representation model. Fuzzy Systems, IEEE Transactions on 17 (6), 1366–1378.

Dursun, M., Karsak, E., 2014. An Integrated Approach Based on 2-Tuple Fuzzy Representation and QFD for Supplier Selection. In: Kim, H. K., Ao, S.-I., Amouzegar, M. A., Rieger, B. B. (Eds.), IAENG Transactions on Engineering Technologies. Vol. 247 of Lecture Notes in Electrical Engineering. Springer Netherlands, pp. 621–634.

Herrera, F., Herrera-Viedma, E., Martínez, L., 2000. A fusion approach for managing multi-granularity linguistic term sets in decision making. Fuzzy Sets and Systems 114, 43–58.

Herrera, F., Herrera-Viedma, E., Martínez, L., 2008. A Fuzzy Linguistic Methodology to Deal With Unbalanced Linguistic Term Sets. IEEE Transactions on Fuzzy Systems 16 (2), 354–370.

Herrera, F., Martínez, L., 2000. A 2-tuple fuzzy linguistic representation model for computing with words. IEEE Transactions on Fuzzy Systems 8 (6), 746–752.

Holloway, C. A., 1979. Decision making under uncertainty: models and choices. Prentice-Hall Englewood Cliffs, NJ.

Howard, R. A., 1968. The foundations of decision analysis. Systems Science and Cybernetics, IEEE Transactions on 4 (3), 211–219.

Kacprzyk, J., Yager, R., 1990. Using Fuzzy Logic with Linguistic Quantifiers in Multiobjective Decision Making and Optimization: A Step Towards More Human-Consistent Models. In: Slowinski, R., Teghem, J. (Eds.), Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Vol. 6 of Theory and Decision Library. Springer Netherlands, pp. 331–350.

Kumar, P., Bauer, P., 2009. Progressive design methodology for complex engineering systems based on multiobjective genetic algorithms and linguistic decision making. Soft Computing 13 (7), 649–679.

Li, C., Dong, Y., 2014. Multi-attribute group decision making based on proportional 2-tuple linguistic model. In: Xu, J., Fry, J. A., Lev, B., Hajiyev, A. (Eds.), Proceedings of the Seventh International Conference on Management Science and Engineering Management. Vol. 241 of Lecture Notes in Electrical Engineering. Springer Berlin Heidelberg, pp. 215–225.

Martínez, L., Herrera, F., 2012. An Overview on the 2-tuple Linguistic Model for Computing with Words in Decision Making: Extensions, Applications and Challenges. Inf. Sci. 207, 1–18.

McMillan, J. H., 2003. Understanding and Improving Teachers' Classroom Assessment Decision Making: Implications for Theory and Practice. Educational Measurement: Issues and Practice 22 (4), 34–43.

Pei, Z., Ruan, D., Liu, J., Xu, Y., 2010. The 2-tuple fuzzy linguistic representation model. In: Linguistic Values Based Intelligent Information Process-

ing: Theory, Methods, and Applications. Vol. 1 of Atlantis Computational Intelligence Systems. Atlantis Press, pp. 33–78.

Rast, R., Dimroth, C., Watorek, M., 2011. Language teaching and acquisition: What can we learn from ab initio learners? Discours, acquisition et didactique des langues, les termes d'un dialogue Collection Universités, série Sciences du Langage, 71–85.

Shinn, M. R., 1989. Curriculum-based Measurement: Assessing Special Children. The Guilford school practitioner series. Guilford Press.

Truck, I., Akdag, H., 2006. Manipulation of qualitative degrees to handle uncertainty: Formal models and applications. International Journal of Knowledge and Information Systems 9 (4), 385–411.

Truck, I., Akdag, H., 2009. A tool for aggregation with words. International Journal of Information Sciences, Special Issue: Linguistic Decision Making: Tools and Applications 179 (14), 2317–2324.

Truck, I., Malenfant, J., 2010. Towards A Unification Of Some Linguistic Representation Models: A Vectorial Approach. In: The 9th International FLINS Conference on Computational Intelligence in Decision and Control. pp. 610–615.

Wang, J., Hao, J., 2006. A new version of 2-tuple fuzzy linguistic representation model for computing with words. IEEE Transactions on Fuzzy Systems 14 (3), 435–445.

Wang, J., Hao, J., 2007. An approach to computing with words based on

canonical characteristic values of linguistic labels. IEEE Trans. Fuzzy Systems 15 (4), 593–604.

Wang, X., 2011. Model for tourism management with 2-tuple linguistic information. AISS: Advances in Information Sciences and Service Sciences 3 (4), 34–39.

Wei, G.-W., 2010. A method for multiple attribute group decision making based on the ET-WG and ET-OWG operators with 2-tuple linguistic information. Expert Systems with Applications 37 (12), 7895–7900.

Xu, Y., Wang, H., 2011. Approaches based on 2-tuple linguistic power aggregation operators for multiple attribute group decision making under linguistic environment. Applied Soft Computing 11 (5), 3988–3997.

Yager, R. R., Goldstein, L. S., Mendels, E., 1994. FUZMAR: An Approach to Aggregating Market Research Data Based on Fuzzy Reasoning. Fuzzy Sets and Systems 68 (1), 1–11.

Zadeh, L. A., 1965. Fuzzy sets. Information Control 8, 338–353.

Zadeh, L. A., 1975. The Concept of a Linguistic Variable and Its Applications to Approximate Reasoning. Information Sciences, Part I, II, III 8,8,9, 199–249, 301–357, 43–80.

Zimmermann, H. J., 1987. Fuzzy Sets, Decision Making, and Expert Systems. International Series in Management Science Operations Research. Springer.