The use of linguistic preference modelling based on 2-tuples and heuristics searches for improving enterprise processes

F. Herrera[1] E. López[2] L. Martínez[3]

Abstract

The processes carried out in an enterprise involve imprecise and vague information. An adequated approach to model this type of information is the use of the fuzzy linguistic approach. Sometimes besides of the modelled in these processes it is neccesary to apply heuristics methods to fix a good solution in their definition space. In this contribution we shall use a new linguistic model representation based on 2-tuple that improves the precision in the processes of computing with words. Combining it with different heuristics as *Genetic algorithms* and *Ant algorithms* it wiil be possible to obtain better solutions than using the classical linguistic models with these heuristics methods.

Keywords: linguistic preference modelling, heuristics.

Introduction

In the enterprise processes there exists a wide range of activities that involve imprecision and vague information in different problems [9, 10]. In this contribution we shall focus our study in those problems that deal with imprecise and vague information modelled by linguistic preferences [11] and need to apply heuristics methods to find a solution in their definition space [12].

Usually the activities carried out in an enterprise present quantitative aspects that are easily assessed by means of numerical values, however in other cases they present qualitative aspects that are difficult to assess by means of precise values. In the latter case, the Fuzzy Linguistic Approach [20] has provided good results. This approach uses linguistic values to model qualitative aspects instead of numerical values. The classical linguistic computational models [4, 5] present an important drawback as it is the lack of precision in the processes of Computing with Words (CW). In this paper, we shall review the 2-tuple linguistic representation model presented in [16, 17] that provides a linguistic computational model that reduces the loss of information in the processes of CW, improving the results obtained by the classical linguistic computational models. We shall use a simple Decision-Making problem, that it is an usual process in the enterprise environment, to show this improvement.

On the other hand, in the enterprise processes we can find complex problems that are difficult to solve with tradicional techniques and for obtaining a solution it is neccessary to use heuristics methods, as genetic algorithms [8] or ant algorithms [6], that are two bioinspired metaheuristics. The integration of these two areas, linguistic preference modelling and metaheuristics, allows us to manage a lot of problems related to the enterprise processes [12, 14, 15].

The Linguistic Preference Modelling

In this section we review briefly the fuzzy linguistic approach [20] and its classical linguistic computational models and afterwards the linguistic preference modelling based on 2-tuples.

Fuzzy Linguistic Approach

Many aspects of different activities in the real world cannot be assessed in a quantitative form, but rather in a qualitative one, i.e., with vague or imprecise knowledge. In that case a better approach may be to use linguistic assessments instead of numerical values. The variables which participate in these problems are assessed by means of linguistic terms [20]. The use of linguistic assessements imply to make computations with them. "Computing with Words" has been applied to different areas. In [21] can be found two volumes consisting of Foundations and Applications providing the current status of theoretical and empirical developments in "Computing with Words".

In this contribution we shall focus in the use of linguistic information for modelling preferences. In order to do that, we have to choose the appropriate linguistic descriptors for the term set and their semantics. One possibility of generating the linguistic term set consists of directly supplying the term set by considering all terms distributed on a scale on which a total order is defined [19]. For example, a set of seven terms S, could be given as follows:

$$S = \{s_0 : N, s_1 : VL, s_2 : L, s_3 : M, s_4 : H, s_5 : VH, s_6 : P\}$$

Usually, in these cases, it is required that the linguistic term set satisfies the following additional characteristics:

1) There is a negation operator: Neg$(s_i) = s_j$ such that $j = g$-i (g+1 is the cardinality).

2) $s_i \leq s_j \iff i \leq j$. Therefore, there exists a minimization and a maximization operator.

The semantics of the linguistic terms is given by fuzzy numbers defined in the [0,1] interval. A way to characterize a fuzzy number is to use a representation based on parameters of its membership function [2]. The linguistic assessments given by the users are just approximate ones, some authors consider that linear trapezoidal membership functions are good enough to capture the vagueness of those linguistic assessments. The parametric representation is achieved by the 4-tuple (a, b, d, c), where b and d indicate the interval in which the membership value is 1, with a and c indicating the left and right limits of the definition domain of the trapezoidal membership function [2]. A particular case of this type of representation are the linguistic assessments whose membership functions are triangular, i.e., $b = d$, then we represent this type of membership functions by a 3-tuple (a, b, c). An example may be the following (*Figure 1*):

$$\begin{array}{llll} H = (.5, .67, .83) & VH = (.67, .83, 1) & P = (.83, 1, 1) & \\ N = (0, 0, .17) & VL = (0, .17, .33) & L = (.17, .33, .5) & M = (.33, .5, .67) \end{array}$$

Now, we review the classical linguistic computational models to accomplish CW processes.

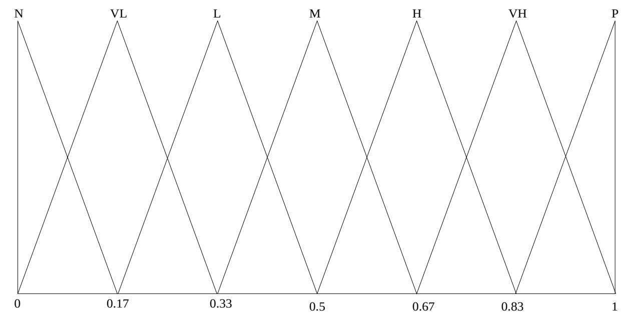

Figure 1: *A Set of Seven Terms with Its Semantic*

Linguistic Computational Model Based on the Extension Principle

The Extension Principle is a basic concept in the fuzzy sets theory [7] which is used to generalize crisp mathematical concepts to fuzzy sets. The use of extended arithmetic based on the Extension Principle [7] increases the vagueness of the results. The results obtained by the fuzzy arithmetic are fuzzy numbers that usually do not match any linguistic term in the initial term set, so a linguistic approximation process is needed to express the result in the original expression domain. In the literature we can find different ways to make linguistic approximations [2, 5]. A linguistic aggregation operation based on the Extension Principle acts according to:

$$S^n \xrightarrow{\tilde{F}} F(\mathcal{R}) \xrightarrow{app_1(\cdot)} S$$

where S^n symbolizes the n cartesian product of S, $\tilde{\mathrm{F}}$ is an aggregation operator based on the Extension Principle, $F(\mathcal{R})$ the set of fuzzy sets over the Real line $\mathcal{R}$, $app_1 : F(\mathcal{R}) \rightarrow S$ is a linguistic approximation function that returns a label from the linguistic term set S whose meaning is the closest to the obtained unlabelled fuzzy number and S is the initial term set.

Symbolic Linguistic Computational Model

A second approach used to operate on linguistic information is the symbolic one [4], that makes direct computations on labels. Usually it uses the ordered structure of the linguistic term sets, $S = \{s_0, \ldots, s_g\}$ where $s_i < s_j$ iff $i < j$, to perform the computations. The intermediate results are numerical values, $\alpha \in [0, g]$, which must be approximated in each step of the process by means of an approximation function $app_2 : [0, g] \rightarrow \{0, .., g\}$ that obtains a numerical value, such that, it indicates the index of the associated linguistic term, $s_{app_2(\alpha)} \in S$. Formally, it can be expressed as:

$$S^n \xrightarrow{C} [0, g] \xrightarrow{app_2(\cdot)} \{0, ..., g\} \longrightarrow S$$

where C is a symbolic linguistic aggregation operator, $app_2(\cdot)$ is an approximation function used to obtain an index $\{0, ..., g\}$ associated to a term in $S = \{s_0, ..., s_g\}$ from a value in $[0, g]$.

The 2-tuple Linguistic Model

This representation model was presented in [16, 17] and it is based on the concept of symbolic translation and use it for representing the linguistic information by means of 2-tuples, (s_i, α), where s is a linguistic term and α is a numerical value representing the symbolic translation.

Let $S = \{s_0, ..., s_g\}$ be a linguistic term set, and $\beta \in [0, g]$ a numerical value in its interval of granularity (e.g.: β is obtained from a symbolic aggregation operation).

Definition 1. *The symbolic translation is a numerical value assessed in $[-.5, .5)$ that supports the "difference of information" between a counting of information β assessed in the interval of granularity ,$[0, g]$, of the term set S and the closest value in $\{0, ..., g\}$ that indicates the index of the closest linguistic term in S.*

From this concept we shall develop a linguistic representation model which represents the linguistic information by means of 2-tuples (r_i, α_i), $r_i \in S$ and $\alpha_i \in [-.5, .5)$. r_i represents the linguistic label center of the information and α_i is the value of the Symbolic Translation.

This representation model defines a set of functions to facilitate computational processes over 2-tuples.

Definition 2. *Let $s_i \in S$ be a linguistic term, then its equivalent 2-tuple representation is obtained by means of the function θ as:*

$$\begin{array}{l} \theta : S \longrightarrow (S \ \times \ [-0.5, 0.5)) \\ \theta(s_i) = (s_i, 0) \end{array}$$

Definition 3. *Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and $\beta \in [0, g]$ a value supporting the result of a symbolic aggregation operation, then the 2-tuple that expresses the equivalent information to β is obtained with the following function:*

$$\Delta : [0, g] \longrightarrow S \times [-0.5, 0.5)$$

$$\Delta(\beta) = (s_i, \alpha), \ \ with \left\{ \begin{array}{cc} s_i & i = round(\beta) \\ \alpha = \beta - i & \alpha \in [-.5, .5) \end{array} \right.$$

where $round(\cdot)$ is the usual round *operation, s_i has the closest index label to "β" and "α" is the value of the symbolic translation.*

Proposition 1. *Let $S = \{s_0, ..., s_g\}$ be a linguistic term set and (s_i, α) be a 2-tuple. There is always a Δ^{-1} function, such that, from a 2-tuple it returns its equivalent numerical value $\beta \in [0, g] \subset \mathcal{R}$.*

Proof.

It is trivial, we consider the following function:

$$\begin{array}{l} \Delta^{-1} : S \times [-.5, .5) \longrightarrow [0, g] \\ \Delta^{-1}(s_i, \alpha) = i + \alpha = \beta \end{array}$$

Here we review the computational technique based on 2-tuples presented in [16, 17]:

1. Comparison of 2-tuples

The comparison of linguistic information represented by 2-tuples is carried out according to an ordinary lexicographic order.

Let (s_k, α_1) and (s_l, α_2) be two 2-tuples, with each one representing a counting of information:

- if $k < l$ then (s_k, α_1) is smaller than (s_l, α_2)
- if $k = l$ then
  1. if $\alpha_1 = \alpha_2$ then (s_k, α_1), (s_l, α_2) represents the same information
  2. if $\alpha_1 < \alpha_2$ then (s_k, α_1) is smaller than (s_l, α_2)
  3. if $\alpha_1 > \alpha_2$ then (s_k, α_1) is bigger than (s_l, α_2)

2. Aggregation of 2-tuples

The aggregation of information consists of obtaining a value that summarizes a set of values, therefore, the result of the aggregation of a set of 2-tuples must be a 2-tuple. In [16] we can find some 2-tuple aggregation operators, that are based on classical aggregation operators.

3. Negation operator of a 2-tuple

The negation operator over 2-tuples is defined as:

$$Neg((s_i, \alpha)) = \Delta(g - (\Delta^{-1}(s_i, \alpha)))$$

where $g + 1$ is the cardinality of S, $S = \{s_0, ..., s_g\}$.

Linguistic Decision-Making problem

In this section we shall solve a linguistic Multi-Expert Decision problem using the above linguistic computational models:

Description.

A distribution company needs to renew its computing system, so it contracts a consulting company to carry out a survey of the different possibilities existing on the market, to decide which is the best option for its needs. The alternatives are the following:

x_1	x_2	x_3	x_4
UNIX	WINDOWS-NT	AS/400	VMS

The consulting company has a group of four consultancy departments

p_1	p_2	p_3	p_4
Cost analysis	System analysis	Risk analysis	Techonology analysis

Each department provides a performance vector expressing its preferences for each alternative. The preferences are assessed in the term set S *(see figure 1)*.

		alternatives			
	L_{ij}	x_1	x_2	x_3	x_4
experts	p_1	VL	M	M	L
	p_2	M	L	VL	H
	p_3	H	VL	M	M
	p_4	H	H	L	L

Decision process.

The process used to solve the problem it is composed by the following steps:

1. To obtain a *collective performance value* over each alternative.
2. A selection process over the collective performance vector is applied.

Using: **(1)** the Linguistic computational model based on the Extension Principle [5], **(2)** the symbolic one [4] and **(3)** the 2-tuple one [16, 17]. We obtain the following solution sets (for a depth description of these computations see [16]):

	C*ollective Values*				**S***olution*
	x_1	x_2	x_3	x_4	
(1)	M	M	L	M	$\{x_1, x_2, x_4\}$
(2)	M	M	L	M	$\{x_1, x_2, x_4\}$
(3)	(M,0)	(M,-.5)	(L,-.25)	(M,-.25)	$\{x_1\}$

Table I. Results using the three methods

An important difference appears in the *"Solution Set"* column in *Table I.* The result obtained by the 2-tuple model is more precise than the sets obtained by the other ones. This is due to:

- Collective values obtained by both classical methods (E.P. and Symbolic) are discrete, then when different alternatives have the same linguistic value as the collective degree we cannot discern which is better than the others.
- In the method based on the 2-tuple representation, the collective values are managed as continuous ones, therefore if several alternatives have the same linguistic term but a different value for the symbolic translation, we can choose the best of those alternatives.

Therefore the linguistic 2-tuple model allows us to obtain better solutions because is more precise than the other ones.

Heuristics Methods

The heuristics methods are used in those complex problems where it is necessary to apply a search algorithm to reach a solution. These methods carry out an exhaustive search in the definition space of the problem to obtain a solution. There exist different heuristics methods, here we shall focus in two well known bioinspired metaheuristics:

Genetic Algorithms [8] are general purpose search algorithms which use principles inspired by natural genetic populations to evolve solutions to problems. Genetic algorithms play a significant role as search technique for handling complex spaces in many fields, in particular in management problems (see [1, 18]).

Ant algorithms [6] are multi-agent systems in which the behaviour of each single agent, called artificial ant, is inspired by the behaviour of real ants. Ants cooperate using an indirect form of communication by means of a pheromone trail that they deposit on the edges of a graph that represent the solution space. Ants can smell pheromone and, when choosing their way, they tend to choose, in probability, paths marked by strong pheromone concentrations. Ants can find shortest paths between good sources and their nest. The artificial ant algorithms use a probabilistic model that describes this phenomenon, using probabilities for representing the pheromone in edges. Ant algorithms are one of the most successful examples of swarm intelligent systems [3]. They have been applied to a great number of problems, a description of this applications can be found in Chapter 2 of [3].

Integration of Linguistic Preference Modelling and Metaheuristics

The integration of these two areas, linguistic preference modelling and metaheuristics, allows us to manage a lot of problems related to the enterprise processes as follows:

- Using linguistic preferences we can represent the expert knowledge in form of alternative assesments. Examples of this, are assignment-selection problem [12] suppliers selection [13], promotion mix management [15].
- The linguistic computation model based on 2-tuples allow us to obtain a linguistic assesment for a solution. An example of a computation linguistic process for a personnel management problem can be found in [14]. The use of the computational model based on 2-tuples on this problem is clear, because it is a generalization of the symbolic one used in [14].
- The linguistic assesments of the solution will act as the fitness function associated to a chromosome in the reproduction phase of a genetic algorihtms. When we use ant algorithm, the linguistic assesments are used for modifying probability value associated to the probabilistic pheromone trail used in the search.
- The heuristic method is used for searching the solution.

Concluding Remarks and Future Works

In this contribution we have reviewed the 2-tuple linguistic model and shown that this model improves the classical ones. Therefore, in those problems in which are combined the linguistic modelling and the heuristics is possible to obtain better results using the 2-tuple linguistic preference modelling.

In this way, as future work, we are working in the use of Ant algorithms for solving a personnel management problem evaluated via the 2-tuple linguistic model.

References

[1] J. Biethahn and V. Nissen, Evolutionary Algorithms in Management Applications, Springer Verlag, 1995.

[2] P.P. Bonissone and K.S. Decker, Selecting Uncertainty Calculi and Granularity: An Experiment in Trading-off Precision and Complexity, in: L.H. Kanal and J.F. Lemmer, Eds., *Uncertainty in Artificial Intelligence* (North-Holland, 1986) 217-247.

[3] E. Bonabeau, M. Dorigo, G. Theraulaz. From Natural to Artificial Swarm Intelligence. Oxford University Press, 1999.

[4] M. Delgado, J.L. Verdegay and M.A. Vila, On Aggregation Operations of Linguistic Labels, *Int. J. of Intelligent Systems* **8** (1993) 351-370.

[5] R. Degani and G. Bortolan, The problem of Linguistic Approximation in Clinical Decision Making, *Int. Journal of Approximate Reasoning* **2** (1988) 143-162

[6] M. Dorigo and G. Di Caro, Ant Algorithms for Discrete Optimization, *Artificial Life* **5** (1999) 137-172.

[7] D. Dubois and H. Prade, Fuzzy Sets and Systems: Theory and Applications, *Academic*, New York, 1980.

[8] D.E. Goldberg, Genetic Algorithms in Serach, Optimisation and Machine Learning, Addison Wesley, 1989.

[9] J. Gil-Aluja, The Interactive Management of Human Resources in Uncertainty. Kluwer Academic, 1997.

[10] J. Gil Aluja, Investment in Uncertainty. Kluwer Academic, 1999.

[11] F. Herrera, E. Herrera-Viedma, E. López, On the Linguistic Approach in Multi-Person Decision Making. Proc. Int. Conf. on Intelligent Technologies in Human-Related Sciences (ITHURS'96), Leon, 1996, Vol. II, 205-213.

[12] F.Herrera, E. López, C. Mendaña, M.A. Rodríguez, Solving an assignment-selection problem with verbal information and using genetic algorithms, *European Journal of Operational Research*, **119** (1999) 326-337.

[13] F.Herrera, E. López, C. Mendaña, M.A. Rodríguez, A Linguistic Decision Model for Suppliers Selection in International Purchasing. In: J. Kacprzyk, L. Zadeh (Eds.),*Computing with Words in Information/Intelligent Systems 2. Applications*, Physica Verlag, 1999, 500-524.

[14] F.Herrera, E. López, C. Mendaña, M.A. Rodríguez, A Linguistic Decision Model for Personnel Management Solved with a Linguistic Biobjective Genetic Algorithm, *Fuzzy Sets and Systems* 2000. In press.

[15] F.Herrera, E. López, C. Mendaña, M.A. Rodríguez, A Linguistic Decision Model for Protion Mix Management Solved with Genetic Algorithms. *Fuzzy Sets and Systems* 2000. In press.

[16] F. Herrera and L. Martínez, A 2-tuple Fuzzy Linguistic Representation Model for Computing with Words, Dept. Computer Sciences and A.I., Granada University, Technical Report#DECSAI-990102, (1999)

[17] F. Herrera and L. Martínez, A 2-tuple Fuzzy Linguistic Representation Model based on a Symbolic Translation, *In Proc. EUROFUSE-SIC'99 Conference*, Budapest, 1999, 25-28.

[18] E. Lopez, C. Mendaña, M.A. Rodriguez. The Election of a Portfolio Through a Fuzzy Genetic Algorithm: The Pofugena Model. In C. Zopounidis (Ed.), *Operational Tools in the Management of Finantial Risks*, Kluwer Academic, 1998.

[19] R.R. Yager, An Approach to Ordinal Decision Making, *Int. J. of Approximate Reasoning* **12** (1995) 237-261.

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

[21] Zadeh, L. and Kacprzyk, J. (Eds.), *Computing with Words in Information/Intelligent Systems Vol. I, Vol II*, Physica-Verlag, Heidelberg, 1999.

1 Department of Computer Science and A.I.. University of Granada. herrera@decsai.ugr.es

2 Department of Economy and Business Management. University of León. ddeelg@dde.unileon.es

3 Department of Computer Science. University of Jaén. martin@ujaen.es