

Generating Fuzzy Term Sets for Software Project Attributes using Fuzzy C-Means and Real Coded Genetic Algorithms

Ali Idri¹, Azeddine Zahi² and Alain Abran³

¹ Department of Software Engineering
ENSIAS, Mohamed V University
Rabat, Morocco, E-mail : { idri, elkoutbi}@ensias.ma

² Department of Computer Science FST,
Sidi Mohamed Ben Abdellah University, Fez, Morocco
E-mail : azeddine.zahi@fst-usmba.ac.ma

³ École de Technologie Supérieure
1180 Notre-Dame Ouest,
Montreal, Canada H3C 1K3
E-mail : aabran@ele.etsmtl.ca

ABSTRACT

This paper investigates with the fuzzy representation of software project attributes. The aim is to generate fuzzy sets and their membership functions from numerical data of software project attributes. The proposed fuzzy sets generation process consists in two main steps: First, we use the well-known Fuzzy C-Means algorithm (FCM) and the Xie-Beni validity criterion to decide on the number of fuzzy sets. Second, we use a Real Coded Genetic Algorithm (RCGA) to build membership functions for these fuzzy sets. Membership functions can be trapezoidal, triangular or Gaussian. This study uses the software attributes given in the COCOMO'81 dataset.

Keywords : *Software project attributes, Fuzzy clustering, Real Coded Genetic Algorithms.*

1. INTRODUCTION

Software project attributes are used by estimation models in software engineering to predict some important attributes of future entities such as software development effort, software reliability and programmers productivity. For example, software cost estimation models use as inputs some software project attributes, also called cost drivers, such as software size, software reliability, and experience of the personnel involved in the software project in order to estimate the required software development effort (Boehm, 1981) (Boehm, 1995) (Burgess, 2001) (Idri, 2002) (Shepperd, 1997) (Vicinanza, 1990) (Wittig, 1997).

In general, many software project attributes are measured either on Nominal or Ordinal scale type composed of linguistic values such as, *low, very low, complex, and important*. For example in the COCOMO II software cost estimation model (Boehm, 1995) 17 among 23 cost drivers are measured on an Ordinal scale composed of six linguistic values: *very low, low, nominal, high, very high,*

and *extra-high*. As a consequence, when dealing with linguistic values handling imprecision, uncertainty and partial truth is unavoidable. However, the software engineering community often uses numbers or classical intervals to represent these linguistic values. Furthermore, such transformations and representations do not mimic the way in which humans interpret linguistic values and consequently cannot deal with imprecision and uncertainty. To overcome this limitation, we have suggested the use of fuzzy sets rather than classical interval (or numbers) to represent linguistic values (Idri, 2000) (Idri, 2001) (Idri, 2002). The main motivation of fuzzy sets theory, founded by Zadeh in 1965, is apparently the desire to build a formal quantitative framework that captures the vagueness of humans knowledge since it is expressed via natural language.

Fuzzy set theory (Zadeh, 1965) suggests through the fuzzy set concept a more suitable representation of linguistic values. Indeed, a fuzzy set, by contrast to a classical set, is associated with a membership function, which maps the elements of a domain W in a real interval $[0,1]$. Thus, a fuzzy set representation captures the vagueness of one linguistic value by the use of gradual rather than abrupt-step membership function. In this paper, we investigate the fuzzy representation of linguistic values measuring software project attributes.

Fuzzy representation of linguistic values has been successfully used in many other fields such as control, image processing, and pattern recognition. An overview of techniques to generate fuzzy sets and their membership functions is presented in (Medasani, 1998). They may be grouped into two major categories: (1) empirical techniques which construct membership functions from expert knowledge (Idri, 2000) (Sicilia, 2005), and (2) automatic techniques, which construct membership functions from historical data using clustering techniques (Liao, 2001) (Chen, 2005) (Guillaume, 2004). In an earlier work (Idri, 2000), we have empirically built fuzzy sets of twelve COCOMO'81 cost drivers based on their descriptions given

in (Boehm, 1981). These fuzzy sets are associated with trapezoidal membership functions. The aim of this work is to generate fuzzy sets and their membership functions using the fuzzy C-Means clustering technique and a Real Coded genetic algorithm.

The proposed fuzzy sets generation process consists in two main steps (Figure 1). First, we use the well-known Fuzzy C-Means algorithm (FCM) and the Xie-Beni validity criterion to decide on the number of clusters (fuzzy sets) (Bezdek, 1981) (Xie and Beni, 1991). Second, we use a Real Coded Genetic Algorithm (RCGA) to build membership functions for these fuzzy sets (Herrera, 2003) (Mühlenbein, 1993). Membership functions can be trapezoidal, triangular or Gaussian. Fuzzy C-Means algorithm is a fuzzy clustering method used to generate a known number of clusters. The determination of this number is still an open problem in clustering. Often, an empirical knowledge or a set of evaluation criteria is used to choose the best set of clusters. In this work, we use the fuzzy cluster validity criterion proposed in (Xie, 1991).

This study uses a dataset that contains 252 historical Software projects. This dataset is deduced from the COCOMO'81 dataset (Boehm, 81). Each project is described by 13 attributes: the software size measured in KDSI (Kilo Delivered Source Instructions) and the remaining 12 attributes are measured on a scale composed of six linguistic values: 'very low', 'low', 'nominal', 'high', 'very high' and 'extra high'. These 12 attributes are related to the software development environment such as the experience of the personnel involved in the software project, the method used in the development and the time and storage constraints imposed on the software (Table 1).

Table 1: COCOMO attributes selected for fuzzification

Attributes	Designation
SIZE	Software Size
DATA	Database Size
TIME	Execution Time Constraint
STOR	Main Storage Constraint
VIRTMIN, VIRT MAJ	Virtual Machine Volatility
TURN	Computer Turnaround
ACAP	Analyst Capability
AEXP	Applications Experience
PCAP	Programmer Capability
VEXP	Virtual Machine Experience
LEXP	Programming Language Experience
SCED	Required Development

This paper is organized as follows. Section 2 describes briefly the Fuzzy C-Means algorithm and presents the results of its application to the software project attributes of the COCOMO'81 dataset. Section 3 presents how a Real Coded Genetic Algorithm is used to build membership functions of the fuzzy sets generated by the FCM algorithm. Section 4 presents and discusses the obtained

membership functions when applying RCGA to software project attributes of the COCOMO'81 dataset. A conclusion and an overview of future work conclude this paper.

2. FUZZY C-MEANS ALGORITHM FOR CLUSTERING SOFTWARE PROJECT ATTRIBUTES

2.1 FCM algorithm: An overview

Fuzzy C-means algorithm (FCM) is a fuzzy clustering technique which is different from classical C-means that uses hard partitioning. FCM uses fuzzy partitioning such that a data point can belong to all clusters with different membership grades between 0 and 1. FCM is an iterative algorithm that aims to find cluster centers (centroids) that minimize the following objective function:

$$\text{Min } J_m(U, C) = \sum_{i=1}^{i=c} \sum_{j=1}^{j=c} (u_{ij})^m \|x_j - c_i\|^2 \quad (1)$$

$$\text{Subject to } \sum_{i=1}^c u_{ij} = 1, \forall j = 1, \dots, n \quad (2)$$

where

$X = \{x_1, \dots, x_n\}$ is a data set of points;

c is the desired number of clusters;

m is the control parameter of fuzziness;

$U = (u_{ij})$ is the partition matrix, containing the membership values of all data in all clusters;

$C = (c_i)$ is the set of cluster centers

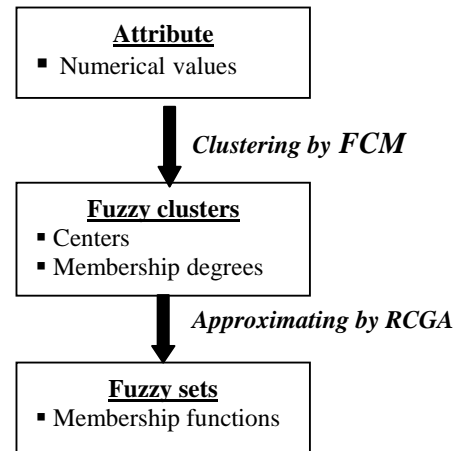


Figure 1. Fuzzy sets generation process

To obtain a fuzzy partition using the FCM algorithm, the membership matrix (U) is randomly initialized according to Equation 2. To reach a minimum of the objective function, there are two conditions. First, the centers are computed according to the Equation 3. Second, the matrix U is calculated according to the Equation 4. By iteratively updating the cluster centers and the membership grades for

each data point, FCM moves the cluster centers to the "right" locations within a data set.

$$c_i = \frac{\sum_{j=1}^n u_{ij}^m x_j}{\sum_{j=1}^n u_{ij}^m} \quad (3)$$

$$u_{ij} = \frac{1}{\sum_{k=1}^c \left(\frac{\|x_j - c_i\|}{\|x_j - c_k\|} \right)^{2/(m-1)}} \quad (4)$$

The outline of the FCM algorithm can be stated as follows (Bezdek, 1981):

- Step 1. Randomly initialize the membership matrix (U) that has constraints in Equation 2.
- Step 2. Calculate centroids(c_i) by using Equation 3.
- Step 3. Compute dissimilarity between centroids and data points using Equation 1. Stop if its improvement over previous iteration is below a threshold.
- Step 4. Compute a new U using Equation 4. Go to Step 2.

2.2 Empirical results

This subsection presents the obtained results when applying the FCM algorithm to the COCOMO'81 software projects attributes. The calculations were made using a software prototype developed with Matlab under a Microsoft Windows PC environment.

For each software project attribute, several experiments were conducted with the FCM algorithm each time using different initial matrix U. The desired number of clusters (c) is varied within the interval [3,6] because all the COCOMO'81 attributes are evaluated on a scale composed of at most six values (Boehm, 1981). The parameter m is fixed to 2 in all experiments. As mentioned earlier, we use the Xie-Beni criterion to decide on the number of clusters to be used in the next section. Table 2 shows the variation of the Xie-Beni index according to the number of clusters for each COCOMO'81 attribute.

For each attribute, we choose the number of clusters that minimizes the value of Xie-Beni criterion. (bold cell in table 2). Figures 2 and 3 show the fuzzy partition generated by the FCM algorithm of the DATA and TIME attributes respectively.

After generating fuzzy sets (clusters) with their partition by FCM, we use a Real Coded Genetic Algorithm (Herrera, 2003) (Mühlenbein, 1993) to build membership functions for these clusters; membership functions can be trapezoidal, triangular or Gaussian.

Table 2: Variation of Xie-Beni index according to the number of clusters for the COCOMO'81 attributes

Attributes	Number of clusters			
	3	4	5	6
SIZE	0.012	0.037	0.055	0.113
DATA	0.012	0.269	0.168	0.120
TIME	0.076	0.060	0.083	0.068
STOR	0.072	0.091	0.066	0.111
VIRTMIN	0.087	0.076	0.083	0.089
VIRTMAJ	0.078	0.109	0.064	0.066
TURN	0.144	0.093	0.070	0.1031
ACAP	0.102	0.068	0.099	0.075
AEXP	0.065	0.184	0.130	0.104
PCAP	0.077	0.086	0.087	0.058
VEXP	0.078	0.068	0.057	0.073
LEXP	0.121	0.091	0.123	0.087
SCED	0.128	0.148	0.135	0.151

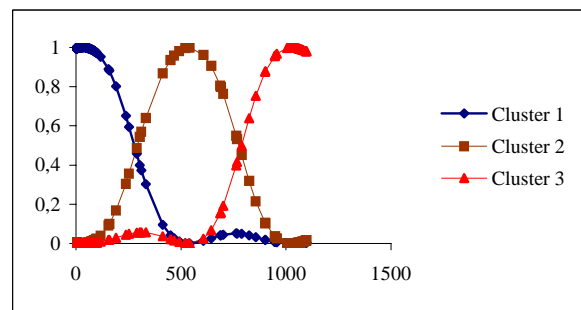


Figure 2: fuzzy partition for DATA attribute

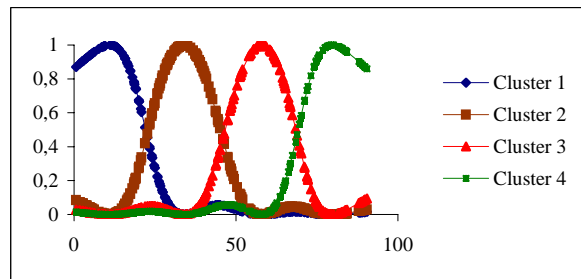


Figure 3: Fuzzy partition for TIME attribute

3. BUILDING MEMBERSHIP FUNCTIONS OF FUZZY SETS USING REAL CODED GENETIC ALGORITHM

3.1 Problem formulation

Let us suppose we know a partition composed by c fuzzy clusters generated when applying the FCM algorithm to a given dataset $X = \{x_1, \dots, x_n\}$. Consider that

$U = (u_{ij}), 1 \leq i \leq n; 1 \leq j \leq n$ is the partition matrix containing membership grades of data $X = \{x_1, \dots, x_n\}$ to the c fuzzy clusters. $C = (c_i), 1 \leq i \leq c$, c_i are the cluster centers. The problem consists on building a set of membership functions $(\mu_i), 1 \leq i \leq c$, that are interpolating the known membership values u_{ij} of the partition matrix U ; membership functions can be trapezoid, triangular, or gaussian. Hence, the problem can be formulated as an optimization problem, which consists of finding the membership functions, $(\mu_i), 1 \leq i \leq c$, minimizing the mean square error defined as follows:

$$MSE(\mu_1, \dots, \mu_c) = \frac{1}{n} \sum_{j=1}^{j=n} \left\| (\mu_1(x_j), \dots, \mu_c(x_j)) - (u_{1j}, \dots, u_{cj}) \right\|^2 \quad (5)$$

subject to the following conditions:

$$\sum_{i=1}^{i=c} \mu_i(x_j) = 1, \text{ for all } x_j$$

$$\mu_i(x_j) = u_{ij}, 1 \leq i \leq c; 1 \leq j \leq n$$

According to the shape of membership functions, which are often not differentials, we suggest approaching this problem with a Real Coded Genetic Algorithm.

3.2 Experiment design of a Real Coded Genetic Algorithm to build membership functions

Genetic algorithms (GAs) are stochastic methods based on the principles of genetics and the natural evolution (Goldberg, 1989) (Holland, 1975). They are used in search and optimization problems. The main idea is to evolve over time a finite part of search space, called *population*, using three operators: *selection*, *crossover*, and *mutation* until a termination criterion is reached. Each element in the population is treated as a chromosome, and represents a candidate solution to the problem. Furthermore, a chromosome is associated with a value called *fitness* which reflects its goodness and its adaptability capabilities; it is often calculated from the objective function. When tackling an optimization problem with variables in a continuous domain, GAs are called Real Coded Genetic Algorithms (RCGAs) (Mühlenbein, 1993) (Herrera, 2003). In this case, each chromosome in the search space is coded by a vector of real numbers and specific operators are used. In our case, the use of an RCGA to find membership functions μ_j requires the determination of certain parameters such as the coding scheme, the fitness function and the various genetic operators (selection, crossover and mutation).

Concerning the coding scheme, a chromosome in the population of our RCGA, $m_i, 1 \leq i \leq M$, represents the set of the unknown membership functions $(\mu_j), 1 \leq j \leq c$ associated to the c fuzzy sets generated by the FCM. The shape of the membership functions can be trapezoidal, triangular or gaussian. Thus, each chromosome encodes a set of membership functions in a real vector (m_i^1, \dots, m_i^K) .

The genes m_i^j are obtained from the shape of the membership functions. Furthermore, in order to avoid incoherent situations, such as the peak value of one function being greater than the peak value of the next one, each gene m_i^j (a real value) must be within a fixed interval. These intervals are often determined by experts. Here, we use the cluster centers $C = (c_j), 1 \leq j \leq c$ to decide on these intervals. Taking into account these aspects, we propose three coding schemes of m_i that are associated to the trapezoidal, triangular, and gaussian shapes respectively.

- For the trapezoidal shape, each membership function, $\mu_j, 2 \leq j \leq c-1$, is represented by 4 parameters $(a_1^j, a_2^j, a_3^j, a_4^j)$; the membership functions μ_1 and μ_c are represented by 2 parameters, (a_3^1, a_4^1) and (a_1^c, a_2^c) respectively. In order to obtain a fuzzy partition, i.e. $\sum_{j=1}^{j=c} \mu_j(x) = 1$, the parameters (a_3^j, a_4^j) of each function $\mu_j, 1 \leq j \leq c-1$ must be the same as the parameters (a_1^{j+1}, a_2^{j+1}) of the next function μ_{j+1} . Thus, only 2 of the parameters $(a_3^j, a_4^j, a_1^{j+1}, a_2^{j+1})$ are considered in the coding scheme. Figure 4 shows the structure of an chromosome m_i encoding trapezoidal membership functions. The size of this chromosome is defined by the expression $K = 2c - 2$. The variation intervals associated to a chromosome $m_i = (m_i^1, \dots, m_i^K)$ are defined in Table 3.

Table 3: Variation Intervals of a chromosome m_i associated to trapezoidal membership functions

Gene	Variation interval
m_i^1	$\left[\min(X), c_1 + \frac{c_2 - c_1}{2} \right]$
$m_i^{2l}, 1 \leq l \leq c-1$	$\left[c_l + \frac{c_{l+1} - c_l}{2}, c_{l+1} \right]$
$m_i^{2l-1}, 1 \leq l \leq c-1$	$\left[c_l, c_l + \frac{c_{l+1} - c_l}{2} \right]$
m_i^K	$\left[c_{c-1} + \frac{c_c - c_{c-1}}{2}, \max(X) \right]$

- For the triangular shape, each membership functions $\mu_j, 2 \leq j \leq c-1$ is represented by 3 parameters (a_1^j, a_2^j, a_3^j) ; the membership functions μ_1 and μ_c are represented by 2 parameters, (a_2^1, a_3^1) and (a_1^c, a_2^c) respectively. In order to obtain a fuzzy partition, the parameter a_2^j of each function μ_j is the same as the

parameters a_3^{j-1} and a_1^{j+1} of the adjacent functions μ_{j-1} and μ_{j+1} respectively. Thus, only one of these parameters (the center of a triangular function), is considered in the coding scheme. Figure 5 shows the structure of a chromosome encoding triangular membership functions. The size of this chromosome is given by $K = c$. The variation intervals associated to a chromosome $m_i = (m_i^1, \dots, m_i^K)$ are defined in Table 4.

Table 4: Variation intervals of a chromosome m_i associated to triangular membership functions

Gene	Variation interval
m_i^1	$\left[\min(X), c_1 + \frac{c_2 - c_1}{2} \right]$
$m_i^l, 2 \leq l \leq c-1$	$\left[c_{l-1} + \frac{c_l - c_{l-1}}{2}, c_l + \frac{c_{l+1} - c_l}{2} \right]$
m_i^K	$\left[c_{c-1} + \frac{c_c - c_{c-1}}{2}, \max(X) \right]$

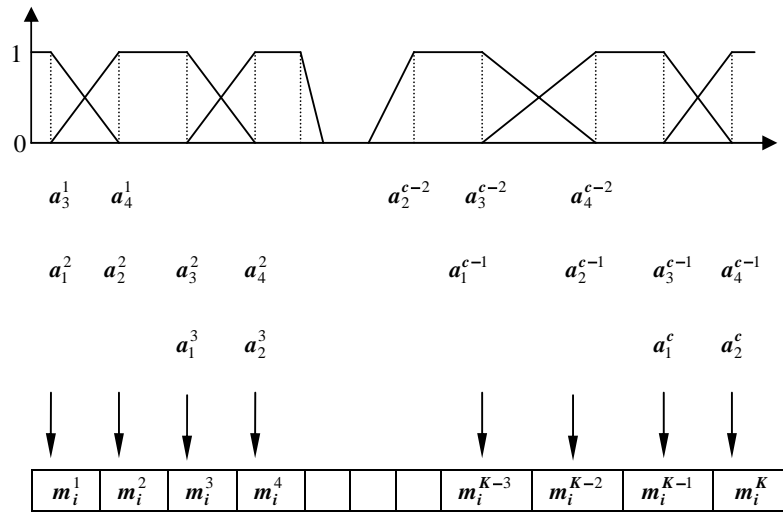


Figure 4: Structure of a chromosome associated to trapezoidal membership functions

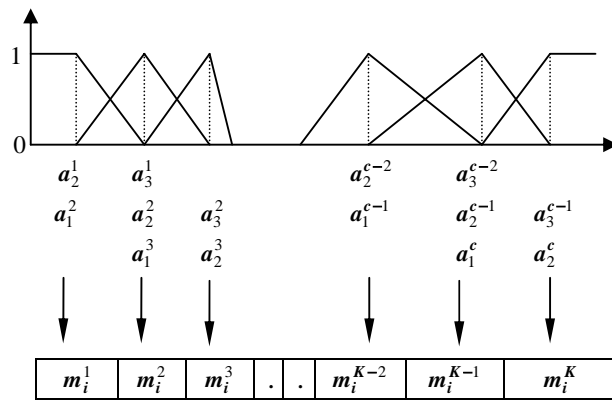


Figure 5: Structure of a chromosome associated to triangular membership functions

▪ For the Gaussian shape, each membership function $\mu_j, 1 \leq j \leq c$ is defined by 2 parameters: the width σ_j and the center c_j . Figure 6 shows the structure of a chromosome encoding gaussian membership functions. The size of this chromosome is given by $K = 2c$. The variation intervals associated to a chromosome $m_i = (m_i^1, \dots, m_i^K)$ are defined in Table 5.

Concerning the fitness function F, we use the following formula:

$$F(m_i) = \frac{MSE(m_i)}{\sum_{i=1}^M MSE(m_i)} \quad (6)$$

$$MSE(m_i) = \frac{1}{n} \sum_{j=1}^n \|\mu(x_j) - y_j\|^2$$

where

$$\mu(x_j) = (\mu_1(x_j), \dots, \mu_c(x_j)),$$

$$y_j = (u_{j1}, \dots, u_{jc}), \text{ and } M \text{ is the size of the population.}$$

Table 5: Variation intervals of a chromosome m_i associated to Gaussian membership functions

Gene	Variation interval
center m_i^1	$\left[\min(X), c_1 + \frac{c_2 - c_1}{2} \right]$
centers $m_i^{2l-1}, 2 \leq l \leq c-1$	$\left[c_{l-1} + \frac{c_l - c_{l-1}}{2}, c_l + \frac{c_{l+1} - c_l}{2} \right]$
center m_i^{K-1}	$\left[c_{c-1} + \frac{c_c - c_{c-1}}{2}, \max(X) \right]$
Widths $m_i^{2l}, 1 \leq l \leq c$	$[0, L_l], L_l$ is the length of the variation interval of the previous gene: m_i^{2l-1}

For the three genetic operators (selection, crossover and mutation), we use those that are specific to Real Coded

Genetic algorithms:

• **Selection:** The linear ranking is used as a selection operator (Baker, 1987). Fitness values are first sorted into decreasing order. A chromosome is then randomly selected according to its rank in the population with the probability computed as follows.

$$P(m_i) = \frac{1}{M} \left(2 - \eta + 2 * (\eta - 1) * \left(\frac{\text{rank}(m_i) - 1}{M - 1} \right) \right) \quad (7)$$

where M is size of the population, and $\eta \in [0,1]$.

• **Crossover :** The line recombination method is considered as a crossover operator (Mühlenbein, 1993). It performs recombination between real coded chromosomes. Let be P_1 and P_2 the chromosomes to be crossed, and O_1 a chromosome generated by this operator. O_1 is constructed gene by gene, and each gene O_1^l is the result of combining the genes in the parents according to the expression:

$$O_1^l = P_1^l + \alpha (P_2^l - P_1^l) \quad (8)$$

where α a scaling factor chosen uniformly, for all pairs of parents, at random in the interval $[0.25, 1.25]$.

• **Mutation:** As a mutation operator, we consider the Breeder Genetic Algorithm (Mühlenbein, 1993), which performs a mutation of real coded chromosomes by perturbing each gene m_i^l of the chromosome m_i according to the expression:

$$(m_i^l)' = m_i^l \pm \Delta m_i^l \cdot \delta \quad (9)$$

where, Δm_i is the range of the variation interval associated to m_i^l , the sign (-) or (+) is selected with probability 0.5, and δ is a distributed amplitude of the perturbation favoring the worst values.

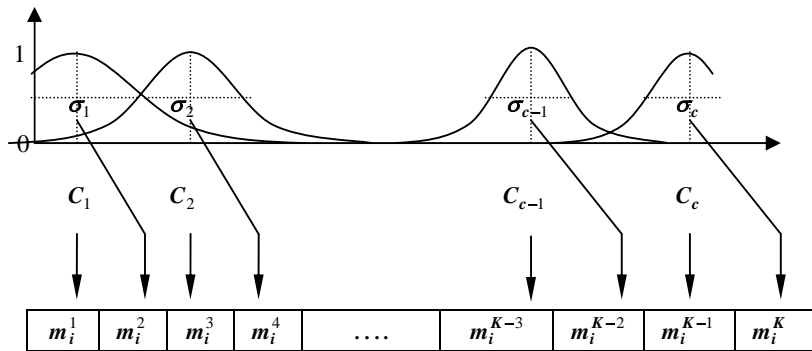


Figure 6: Structure of a chromosome associated to Gaussian membership functions

4. EMPIRICAL RESULTS

This section presents the obtained membership functions when applying the RCGA algorithm to the COCOMO'81 software projects attributes. The calculations were made using a software prototype developed with Matlab under a Microsoft Windows PC environment.

For each software project attribute, we have applied the RCGA algorithm, as it is designed in the previous section, to the fuzzy clusters generated by the FCM algorithm in order to build their membership functions. The RCGA algorithm is applied with populations sized to 300, mutation probability fixed to 0.9, and the number of generation is equal to 200. For each attribute, the number of membership functions is equal to the number of fuzzy clusters generated by the FCM algorithm with the Xie-beni criterion (Section 2). Figure 7 and Figure 8 show three different shapes of membership functions associated to the fuzzy sets of the DATA and TIME attributes respectively.

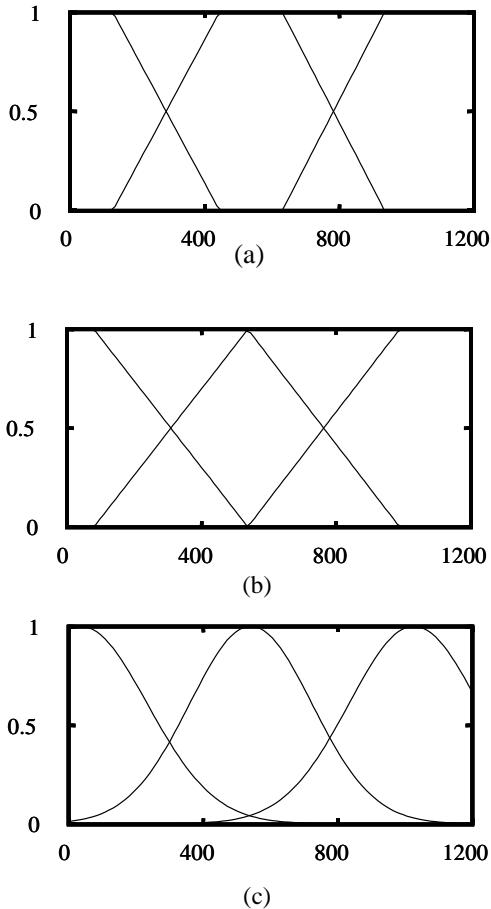


Figure 7: Membership functions associated to the fuzzy sets of the DATA attribute. (a) Trapezoidal. (b) Triangular. (c) Gaussian.

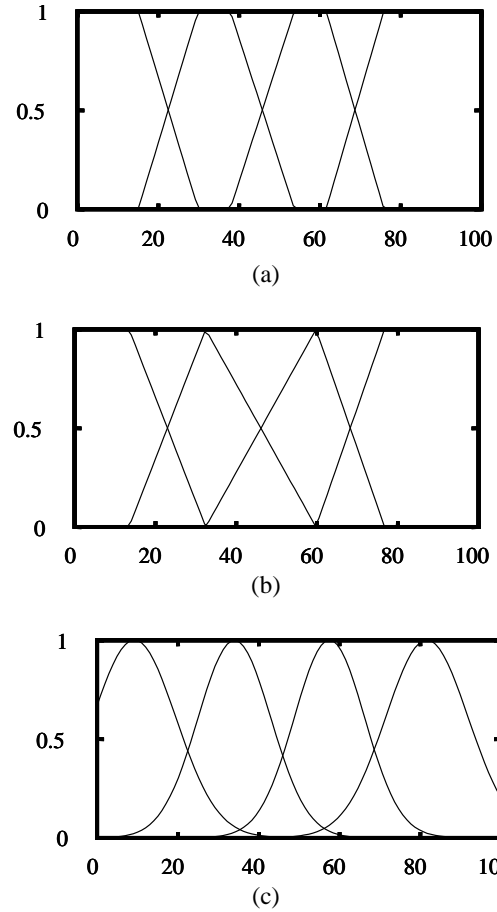


Figure 8: Membership functions associated to the fuzzy sets of the TIME attribute. (a) Trapezoidal. (b) Triangular. (c) Gaussian.

5. CONCLUSION AND FUTURE WORK

In this paper, we have proposed and validated the use of the FCM algorithm and a Real Coded Genetic Algorithm to generate fuzzy sets and their membership functions for software project attributes. The proposed fuzzy sets generation process consists in two main steps. First, we have used the well-known Fuzzy C-Means algorithm (FCM) and the Xie-Beni validity criterion to decide on the number of clusters (fuzzy sets). Second, we have used a Real Coded Genetic Algorithm (RCGA) to build membership functions for these fuzzy sets. Membership functions can be trapezoidal, triangular or Gaussian. This study has used the 13 attributes of the COCOMO'81 dataset.

The obtained fuzzy sets and their membership functions of the 13 attributes of the COCOMO'81 dataset will be used for software cost estimation. Indeed, in some earlier works, we have developed a set of software cost estimation models based on an empirical-construction of fuzzy sets (Idri et al., 2000) (Idri et al., 2002). Hence, we are looking currently at the investigation of the fuzzy sets obtained in this work to compare the accuracy of cost estimation models when using FCM and RCGA rather than empirical knowledge for building fuzzy sets.

Bibliography

- (Baker, 1987) J. E. Baker, "Reducing bias and inefficiency in the selection algorithm", Lawrence Erlbaum Associates, Publishers, ProcICGA 2, pp. 14-21, 1987.
- (Bezdek, 1981) J. Bezdek. "Pattern Recognition with Fuzzy Objective Function Algorithms", Plenum, New York, 1981.
- (Boehm, 1981) B.W. Boehm. "Software Engineering Economics". Prentice-Hall, 1981.
- (Boehm, 1995) B.W. Boehm et al. "Cost Models for Future Software Life Cycle Processes: COCOMO 2.0." Annals of Software Engineering on Software Process and Product Measurement, Amsterdam, 1995.
- (Burgess, 2001) C. J. Burgess and M. Lefley. "Can Genetic Programming Improve Software Effort Estimation?" Information and Software Technology, vol. 43, 2001, pp. 863-873.
- (Chen 2005) S.M. Chen., F. M. Tsai. " A new Method to Construct Membership Functions and Generate Rules from training Instances ", Information and Management Sciences, vol. 16, no 2, pp. 47-72.
- (Goldberg, 1989) D. E. Goldberg, "Genetic Algorithm in Search, Optimization and Learning", Addison-Wesley, 1989
- (Guillaume, 2004) S. Guillaume and B. Charnomordic. "Generating an interpretable Family of Fuzzy Partitions From Data." IEEE Transactions on Fuzzy Systems, vol. 12, no 3, June 2004, pp.324-335.
- (Herrera, 2003) F. Herrera, M. Lozano, A.m. Sanchez. " A taxonomy for the crossover operator for real- coded genetic algorithms: An experiment study." Int. J. Intell. Syst. 2003, 18(3), pp: 309-338
- (Holland, 1975) J. H. Holland, "Adaptation in Natural and Artificial Systems", University of Michigan Press, 1975.
- (Idri, 2000) A. Idri., L. Kjiri, and A. Abran. "COCOMO Cost Model Using Fuzzy Logic." Proceedings of the 7th International Conference on Fuzzy set Theory and Technology. February 2000, Atlantic City, NJ, USA, pp. 219-223.
- (Idri, 2001) A. Idri, and A. Abran, "A Fuzzy Logic Based Measures For Software Project Similarity: Validation and Possible Improvements", 7th International Symposium on Software Metrics, *IEEE computer society*, 4-6 April, England, 2001. pp. 85-96
- (Idri, 2002) A. Idri, T. M. Khoshgoftaar and A. Abran. "Investigating Soft Computing in Case-based Reasoning for Software Cost Estimation." International Journal of Engineering Intelligent Systems, vol. 10, no. 3, 2002, pp. 147-157.
- (Liao, 2002) T.W. Liao., A. K. Celmins., R. J. Hammell. "A fuzzy c-means variant for the generation of fuzzy term sets." Fuzzy sets and Systems, 135, 2003, pp. 241-257.
- (Medasani, 1998) S. Medasani., J. Kim., R. Krishnapuram. "An overview of membership function generation techniques for pattern recognition." Internat. J. Approx. Reas., 19, 1998, pp.391-417.
- (Mühlenbein, 1993) H. Mühlenbein and D. Schlierkamp-Voosen, "Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization", *Evolutionary Computation*, Vol. 1, No. 1, pp.25-49, 1993.
- (Sicilia 2005) Sicilia, M.A., Cuadrado, J.J., Crespo, J. and García-Barriocanal. " Software cost estimation with fuzzy inputs modelling and aggregation cost drivers", *Kybernetika*, 41, 2005, pp. 249-264.
- (Shepperd, 1997) M. Shepperd and C. Schofield. "Estimating Software Project Effort Using Analogies." Transactions on Software Engineering, vol. 23, no. 12, 1997, pp. 736-747.
- (Vicinanza, 1990) S. Vicinanza, and M.J. Prietolla, "Case-Based Reasoning in Software Effort Estimation." Proceedings of the 11th Int. Conf. on Information Systems, 1990.
- (Wittig, 1997) G. Wittig and G. Finnie. "Estimating Software Development Effort with Connectionist Models." Information and Software Technology, vol. 39, 1997, pp. 469-476.
- (Xie, 1991) X. L. Xie., G. Beni, "A validity measure for fuzzy clustering." IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 13, no 8, 1991, pp.841-847.
- (Zadeh, 1965) L. A. Zadeh. " Fuzzy sets" , Information and Control, vol. 8, pp. 338-352.