

Impacts des techniques de construction des ensembles flous sur la précision d'un modèle d'estimation des coûts de logiciels par analogie floue

Ali Idri^{*}, Azeddine Zahi^{**}, Mohammed ElKoutbi^{*} and Alain Abran^{***}

^{*}ENSIAS, Université Mohmed V Souissi, Rabat, Maroc
idri@ensias.ma, elkoutbi@ensias.ma

^{**}FST, Université Sidi Mohmed Ben Abdellah, Fès, Maroc
azahi@fst-usmba.ac.ma

^{***}Ecole de Technologie Supérieure, 1180 Notre –Dame Ouest, Montreal, Canada H3C 1K3
alain.abran@etsmtl.ca

Abstract: Fuzzy Analogy est une approche d'estimation des coûts de développement de logiciels qui remédie à la problématique d'utilisation des valeurs linguistiques tout le long d'un processus d'estimation basé sur la technique CBR (*Case-Based reasoning*). En effet, Fuzzy Analogy utilise la logique floue pour représenter et traiter convenablement les valeurs linguistiques. Dans un précédent travail, nous avons validé Fuzzy Analogy sur la base de projets COCOMO'81 en utilisant une représentation floue empirique des différents attributs affectant le coût; les ensembles flous utilisés sont associés à des fonctions trapézoïdales. Cet article, propose une validation de Fuzzy Analogy sur la même base de projets avec une représentation floue générée automatiquement à partir de données historiques. Les ensembles flous et leurs fonctions d'appartenance sont obtenus par l'algorithme FCM (*Fuzzy C-Means*) et un Algorithme Génétique à Codage Réel (AGCR). Les fonctions d'appartenance de ces ensembles flous peuvent avoir différentes formes : trapézoïdale, triangulaire et gaussienne. Une comparaison de la précision des estimations fournies par Fuzzy Analogy, utilisant une technique empirique pour la génération des ensembles flous, avec celle de Fuzzy Analogy, utilisant la technique FCM-AGCR pour la génération des ensembles flous, est aussi présentée dans cet article.

Key words: Estimation des coûts de logiciels, Raisonnement à base de cas, Classification floue, Algorithmes génétiques à codage réel.

INTRODUCTION

L'estimation des coûts de développement de logiciels a fait l'objet de plusieurs investigations ces dernières années. Ces investigations ont abouti au développement et à la validation d'un certain nombre de techniques. L'estimation par analogie est l'une des techniques qui a suscité l'intérêt des chercheurs dans le domaine de l'estimation des coûts de logiciels. C'est une forme de raisonnement à base de cas (*Case Based Reasoning* –CBR) qui s'adapte bien au problème de la prédiction des coûts de développement de logiciels [Vicinanza, 1990] [Shepperd 1997] [Idri, 2002a]. La technique CBR en estimation des coûts de logiciels est basée sur l'affirmation suivante : « *similar software projects have been similar costs* », et se compose de trois étapes : 1) Identification des projets logiciels par un nombre d'attributs significatifs et indépendants ; 2) Evaluation de la similarité entre le nouveau projet

logiciel et tous les projets logiciels historiques ; 3) Utilisation des coûts réels des projets similaires au nouveau projet pour en déduire une estimation de son coût. Cependant, l'utilisation de la technique CBR en estimation des coûts ne permet pas de traiter convenablement le cas des projets logiciels dont certains attributs sont mesurés par des valeurs linguistiques, souvent floues et imprécises, telles que *bas, élevé, très élevé, etc.* Dans cet optique, une version d'estimation par analogie, nommée *Fuzzy Analogy*, a été proposée [Idri, 2002a]. Fuzzy Analogy incorpore la logique floue pour traiter les imprécisions et les incertitudes engendrées par l'utilisation des valeurs linguistiques tout le long du processus CBR. Ainsi, au niveau de la représentation des valeurs linguistiques, Fuzzy Analogy utilise les ensembles flous plutôt que des intervalles classiques ou des nombres. Au niveau de l'évaluation de la similarité entre les projets logiciels, Fuzzy Analogy utilise des mesures basées sur les quantificateurs

linguistiques RIM (*Regular Increasing Monotone*) [Idri, 2001].

Dans nos travaux antérieurs, Fuzzy Analogy a été validé sur la base de projets logiciels COCOMO'81 qui contient 63 projets [Idri, 2002a] [Idri, 2002b]. Chaque projet logiciel est décrit par un ensemble d'attributs mesurés sur une échelle ordinale composée de six valeurs linguistiques : 'très bas', 'bas', 'moyen', 'élevé', 'très élevé' et 'extra-élevé' [Boehm, 1981]. Les ensembles flous, représentant ces six valeurs linguistiques, sont associés à des fonctions trapézoïdales empiriquement construites à partir des descriptions fournies dans [Boehm, 1981]. Dans ce travail, nous proposons une validation de Fuzzy Analogy sur la base COCOMO'81 en utilisant une représentation floue générée automatiquement à partir de la base de projets historiques de COCOMO'81. Le processus d'obtention des ensembles flous associés aux valeurs linguistiques de chaque attribut est composé de deux étapes :

- La première étape permet de construire une partition floue avec un nombre adéquat de clusters (ensembles flous). Pour ce faire, nous utilisons l'algorithme FCM (*Fuzzy C-Mean*) avec le critère de validité de Xie-Beni pour décider sur le nombre de clusters [Bezdek, 1981] [Xie, 1991].
- La deuxième étape permet de construire les fonctions d'appartenance associées aux ensembles flous (clusters) de la partition générée par l'algorithme FCM. Ces fonctions d'appartenance peuvent avoir une forme trapézoïdale, triangulaire, ou gaussienne. Pour ce faire, nous utilisons un algorithme génétique à codage réel pour trouver les fonctions qui approchent au mieux la partition floue obtenue par FCM [Herrera, 2003] [Mühlenbein, 1993].

Les projets historiques utilisés dans la validation du Fuzzy Analogy sont ceux de la base de données COCOMO'81 [Boehm, 1981]. Chaque projet est décrit par 13 attributs : la taille du logiciel mesurée en KDSI (Kilo Delivered Source Instructions) et 12 attributs mesurés sur une échelle composée de 6 valeurs linguistiques: 'très bas', 'bas', 'moyen', 'élevé', 'très élevé' et 'extra élevé'. Ces 12 attributs représentent l'environnement de développement tels que l'expérience des analystes, la fiabilité logicielle et la volatilité de la machine virtuelle (Tableau 1).

Cet article est composé de 4 sections. Dans la section 1, nous présentons brièvement le processus d'estimation de l'approche Fuzzy Analogy. La section 2 porte sur la construction des ensembles flous, associés aux attributs des projets du COCOMO'81 (Tableau 1), ainsi que leurs fonctions d'appartenance. La section 3 présente la validation de Fuzzy Analogy sur COCOMO'81 en utilisant les ensembles flous obtenus à partir de données historiques (Section 2). Cette validation consiste en l'évaluation de la précision des estimations fournies par Fuzzy Analogy. La section 4 conclut et présente des pistes de recherche futures sur ce thème.

Tableau 1. Les attributs des projets logiciels de COCOMO '81

| Attributs | Désignation |
|----------------------|---------------------------------|
| 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 |

1. ESTIMATION DES COÛTS DE LOGICIELS PAR FUZZY ANALOGY

Fuzzy Analogy est une "fuzzification" de la procédure classique d'estimation des coûts par analogie. Son objectif est de permettre la tolérance des imprécisions tout le long du processus d'estimation par analogie ainsi que la gestion des imprécisions au niveau des coûts estimés. Son processus d'estimation est composé, comme dans le cas de la procédure classique d'estimation par analogie, de trois étapes :

- Dans l'étape d'Identification des projets, un ensemble d'attributs, indépendants et significatifs pour l'estimation des coûts, sont identifiés pour caractériser les projets logiciels. Chaque projet est décrit par un ensemble d'attributs mesurés sur une échelle composée de valeurs linguistiques telles que 'très bas', 'bas', etc. Dans Fuzzy Analogy, chaque valeur linguistique est représentée par un ensemble flou ayant une fonction d'appartenance. Les ensembles flous ainsi que leurs fonctions d'appartenance peuvent être obtenus soit empiriquement à partir de connaissances disponibles du domaine soit automatiquement à partir de données brutes (souvent numériques) en utilisant les techniques de clustering.
- Dans l'étape d'Evaluation de la similarité entre les projets logiciels, Fuzzy Analogy utilise de nouvelles mesures de similarité intégrant les concepts de la logique floue et permettant un traitement convenable des projets logiciels décrit par des valeurs linguistiques [Idri, 2001]. Ces mesures évaluent la similarité globale, $d(P_1, P_2)$, entre deux projets P_1 et P_2 en combinant les similarités individuelles, $d_{v_j}(P_1, P_2)$, mesurées selon les différents attributs, V_j , décrivant P_1 et P_2 . Les similarités individuelles sont évaluées par les formules :

$$d_{V_j}(P_1, P_2) = \begin{cases} \max_k \min(\mu_j(P_1), \mu_j(P_2)) \\ \text{max-min aggregation} \\ \sum_k \mu_j(P_1) \times \mu_j(P_2) \\ \text{sum-product aggregation} \end{cases} \quad (1)$$

Avec μ_j , les fonctions d'appartenance représentant les ensembles flous associés à V_j . Pour l'évaluation de la similarité globale, $d(P_1, P_2)$, les similarités individuelles sont combinées par l'utilisation de quantificateurs linguistiques de type RIM (*Regular Increasing Monotone Quantifier*) tels que, *all*, *most*, *many*, *at most* α , or *there exists* [Yager, 1996]. Le choix d'un quantificateur RIM, noté Q , dépend des caractéristiques et des spécificités de l'environnement étudié. Q indique la proportion des similarités individuelles nécessaires pour une meilleure évaluation de la similarité globale. Ainsi, la similarité globale entre deux projets P_1 et P_2 est définie par l'une des expressions informelles suivantes:

$$d(P_1, P_2) = \begin{cases} \text{all of } (d_{V_j}(P_1, P_2)) \\ \text{most of } (d_{V_j}(P_1, P_2)) \\ \text{many of } (d_{V_j}(P_1, P_2)) \\ \dots \\ \text{there exists of } (d_{V_j}(P_1, P_2)) \end{cases} \quad (2)$$

• Dans l'étape d'Adaptation, Fuzzy Analogy utilise une nouvelle stratégie pour déduire le coût d'un nouveau projet P en utilisant les coûts réels des projets logiciels les plus similaires à P . Cette stratégie est basée sur les similarités globales, $d(P, P_i)$, du nouveau projet avec les projets historiques et la définition adoptée pour la qualification « P_i est étroitement similaire à P ». Dans le cas de Fuzzy Analogy, la qualification « étroitement similaire » est représentée par un ensemble flou associé à la fonction d'appartenance, $\mu_{\text{voisinage de } l}$, où $\mu_{\text{voisinage de } l}(d(P, P_i))$ représente le degré de vérité de cette qualification. Ainsi, la formule d'adaptation est :

$$Effor(P) = \frac{\sum_{i=1}^{i=N} \mu_{\text{voisinage de } l}(d(P, P_i)) \times Effort(P_i)}{\sum_{i=1}^{i=N} \mu_{\text{voisinage de } l}(d(P, P_i))} \quad (3)$$

2. CONSTRUCTION DES ENSEMBLES FLOUS AVEC LEURS FONCTIONS D'APPARTENANCE POUR LES ATTRIBUTS DE COCOMO'81

Fuzzy Analogy requiert une représentation floue de chaque attribut sélectionné pour contribuer à l'estimation finale du coût de développement d'un projet logiciel. La représentation floue de chaque attribut est composée d'un nombre adéquat

d'ensembles flous représentés par des fonctions d'appartenance de forme trapézoïdale, triangulaire ou gaussienne. Dans la littérature, plusieurs techniques ont été proposées pour la génération des ensembles flous avec leurs fonctions d'appartenance [Medasani, 1998]. Ces techniques peuvent être regroupées en deux catégories :

1) les techniques empiriques qui construisent les ensembles flous ainsi que leurs fonctions d'appartenance à partir des connaissances recueillies auprès des experts du domaine [Idri, 2000] [Sicilia, 2005].

2) les techniques automatiques qui construisent les ensembles flous ainsi que leurs fonctions d'appartenance à partir de données historiques (souvent numériques) en utilisant les techniques de classification et de clustering [Chen, 2005] [Liao, 2001] [Guillaume, 2004].

Dans un précédent travail, nous avons proposé des ensembles flous ainsi que leurs fonctions d'appartenances en se basant sur les descriptions des différents attributs du COCOMO'81 fournies dans [Boehm, 1981] [Idri, 2000]. Cet article propose d'utiliser les techniques de clustering ainsi que les algorithmes génétiques pour construire les ensembles flous associés à chaque attribut de COCOMO'81. Ainsi, le processus de génération se compose de deux étapes (figure 1) [Idri, 2006].

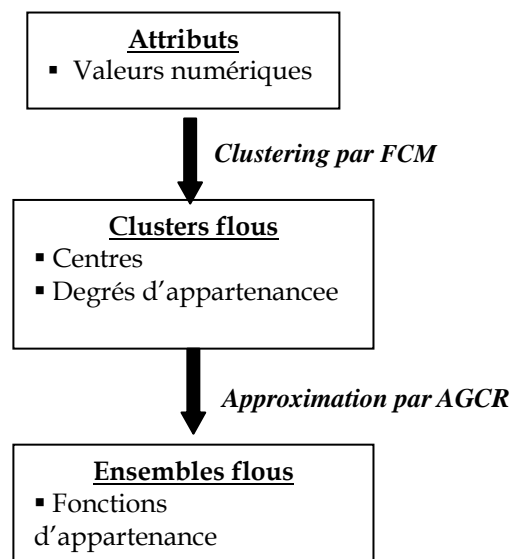


Figure 1. Processus FCM-AGCR de génération des ensembles flous et leurs fonctions d'appartenance

La première étape permet de construire, pour chaque attribut, une partition floue avec un nombre adéquat de clusters (ensembles flous). Pour ce faire, nous utilisons l'algorithme FCM (*Fuzzy C-Mean*) avec le critère de validité de Xie-Beni pour décider sur le nombre de clusters [Bezdek, 1981] [Xie, 1991]. FCM est une méthode de clustering qui permet de construire un nombre spécifique de clusters à partir de données numériques, $X = \{x_1, \dots, x_n\}$, décrivant

l'attribut en question. Le nombre adéquat (c) de clusters (ensembles flous) associés à chaque attribut est déterminé par le critère de validité de Xie et Beni [Xie, 1991]. L'algorithme FCM permet de retrouver itérativement les centres (C_j), $1 \leq j \leq c$ des clusters ainsi que la matrice $U = (u_{ij}), 1 \leq i \leq n, 1 \leq j \leq c$, contenant les degrés d'appartenance, qui minimisent la fonction objective suivante:

$$J_m(U, V) = \sum_{j=1}^{j=c} \sum_{i=1}^{i=n} (u_{ij})^m \|x_i - c_j\|^2 \quad (4)$$

Sous la contrainte :

$$\sum_{j=1}^{j=c} u_{ij} = 1, 1 \leq i \leq n$$

Avec, m un paramètre de contrôle de la fuzzification.

La deuxième étape permet de construire les fonctions d'appartenance associées aux ensembles flous (clusters) générés par l'algorithme FCM. Ces fonctions d'appartenance peuvent avoir une forme trapézoïdale, triangulaire, ou gaussienne. Pour ce faire, nous utilisons un Algorithme Génétique à Codage Réel (AGCR) pour construire les fonctions d'appartenance des ensembles flous associés aux clusters, (C_j), $1 \leq j \leq c$ obtenus par FCM dans l'étape précédente. L'algorithme AGCR, permet de trouver les fonctions (μ_j), $1 \leq j \leq c$ et la matrice U qui interpolent et minimisent la fonction objective suivante :

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

sous les contraintes suivantes : $\sum_{j=1}^c \mu_j(x_i) = 1$, pour

tous les x_i et $\mu_j(x_i) = u_{ij}$, $1 \leq i \leq n$; $1 \leq j \leq c$

L'utilisation d'un algorithme génétique nécessite la spécification d'un certain nombre de paramètres tels que le codage, la fonction d'adaptation et les opérateurs génétiques (sélection, croisement et mutation).

Concernant le codage, un chromosome, $m_i, 1 \leq i \leq M$, de la population, représente les fonctions d'appartenance inconnues (μ_j), $1 \leq j \leq c$, qui sont associées aux (c) ensembles flous (clusters) obtenus par FCM. La forme des fonctions peut être trapézoïdale, triangulaire, ou gaussienne. Ainsi, chaque chromosome, encode les fonctions par un vecteur réel (m_i^1, \dots, m_i^K), dont les composantes sont déterminées à partir de la forme de fonction utilisée. Chaque gène du chromosome doit être maintenu dans un intervalle de variation afin d'obtenir une partition floue. La figure 2 montre un exemple de chromosome qui encode des fonctions trapézoïdales.

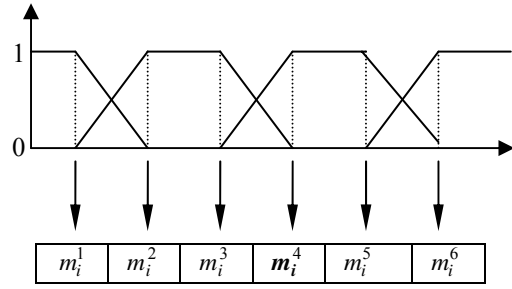


Figure 2. Chromosome associé à une partition floue de 4 fonctions trapézoïdales.

La fitness des chromosomes est évaluée relativement à l'erreur quadratique définie par :

$$MSE(\mu) = \frac{1}{n} \sum_{j=1}^{j=n} \left\| \mu(x_j) - y_j \right\|^2 \quad (6)$$

où $\mu(x_i) = (\mu_1(x_i), \dots, \mu_c(x_i))$ sont les degrés d'appartenance obtenus par les fonctions (μ_j), $1 \leq j \leq c$ et $y_j = (u_{j1}, \dots, u_{jc})$ sont les degrés d'appartenance obtenus par FCM. Plus l'erreur est petite plus le chromosome est robuste.

Enfin, pour les opérateurs génétiques, nous avons utilisé des opérateurs spécifiques aux algorithmes génétiques à codage réel : la sélection par rang [Baker, 1987]; le croisement par recombinaison linéaire et la mutation par le Breeder Genetic Algorithm (BGA) [Mühlenbein, 1993].

3. Validation de Fuzzy Analogy

Cette section présente et discute les résultats de la validation de l'approche Fuzzy Analogy sur quatre bases de données déduites à partir de la base originale COCOMO'81. Cette validation consiste en l'évaluation de la précision des estimations fournies par Fuzzy Analogy en utilisant l'indicateur MRE (Magnitude Relative Error) défini par :

$$MRE(P_i) = \left| \frac{Effort_{réel}(P_i) - Effort_{estimé}(P_i)}{Effort_{réel}(P_i)} \right| \quad (7)$$

Pour chaque projet P_i , nous calculons son MRE. En plus, nous utilisons l'indicateur Pred(p) (Prediction Level) défini par :

$$Pred(p) = \frac{k}{N} \quad (8)$$

où k est le nombre de projets ayant un MRE inférieure ou égal à p , N est le nombre total de projets logiciels. Nous fixons p à 0,20 dans cette validation. Ainsi, Pred (0,20) indique le pourcentage de projets logiciels ayant un MRE inférieure ou égal à 0,20.

L'évaluation se fera sur 4 bases de projets déduites à partir de la base originale COCOMO'81 contenant initialement 63 projets [Idri, 2000]. Chaque base contient 63 projets logiciels ; chacun est décrit par 13 attributs (Tableau 1). Une base (DB2) parmi les quatre

sera considérée comme la base de projets historiques ; les autres trois bases contiendront les projets de test.

Fuzzy Analogy sera validée en considérant deux façons différentes pour la construction des ensembles flous ainsi que leurs fonctions d'appartenance :

- 1) Empiriquement en utilisant les descriptions fournies dans [Boehm, 1981] des différents attributs
- 2) Automatiquement en utilisant la procédure FCM-AGCR décrite dans la section précédente.

Pour construire les ensembles flous associés aux attributs COCOMO'81 à l'aide du processus FCM-AGCR (décrit dans la section 2), nous avons utilisé tous les projets des quatre bases (DB1, DB2, DB3, et DB4), à savoir 252 projets logiciels, afin que ces ensembles flous soient représentatifs. Les ensembles flous générés peuvent avoir trois formes de fonctions : trapézoïdale, triangulaire et gaussienne. Le tableau 2 (colonne 'FCM-AGCR') montre le nombre d'ensembles flous générés par le processus FCM-AGCR pour chaque attribut. La figure 3 montre les fonctions d'appartenance associées aux ensembles flous de l'attribut DATA.

Concernant les ensembles flous générés empiriquement en se basant sur les descriptions fournies dans [Boehm, 1981], nous donnons dans le tableau 2, colonne 'Empirique', le nombre d'ensembles flous associés à chaque attribut. La figure 4 montre les fonctions d'appartenance associées aux ensembles flous de l'attribut DATA.

Tableau 2. Nombre d'ensembles flous associés à chaque attribut dans les deux cas : Empirique et FCM-AGCR

| Attributs | #clusters FCM-AGCR | #clusters Empirique |
|------------|--------------------|---------------------|
| Size | 3 | 3 |
| Data | 3 | 4 |
| Virtmineur | 4 | 4 |
| Time | 4 | 4 |
| Virtmajeur | 5 | 4 |
| Stor | 5 | 4 |
| Turn | 5 | 4 |
| Acap | 4 | 5 |
| Aexp | 3 | 5 |
| Pcap | 6 | 5 |
| Vexp | 5 | 4 |
| Lexp | 6 | 4 |
| Sced | 3 | 4 |

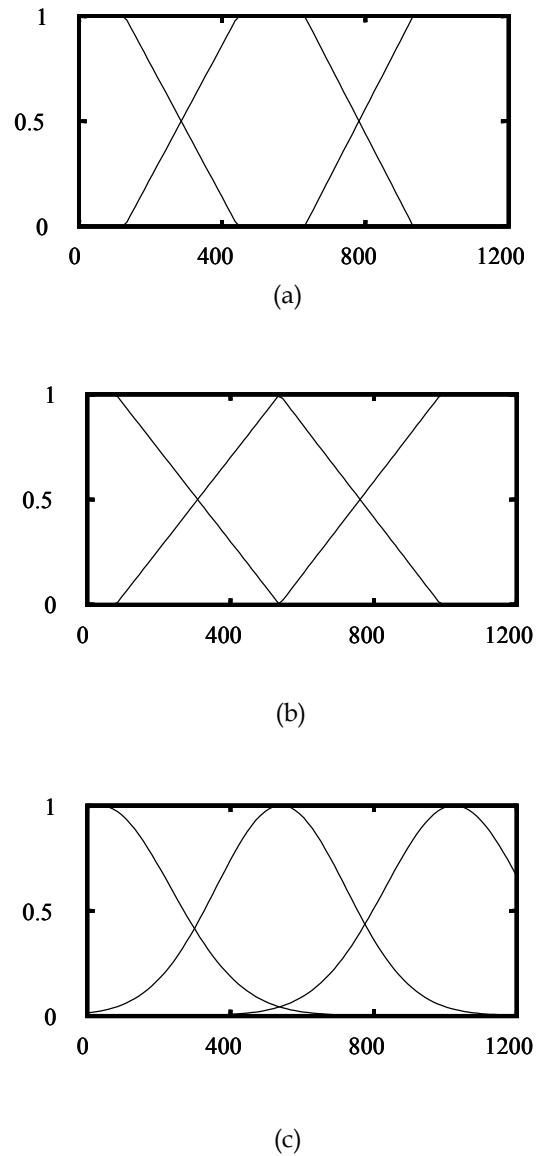


Figure 3. Ensembles flous générés par le processus FCM-AGCR pour l'attribut DATA. (a) Trapezoïdale. (b) triangulaire. (c) Gaussienne.

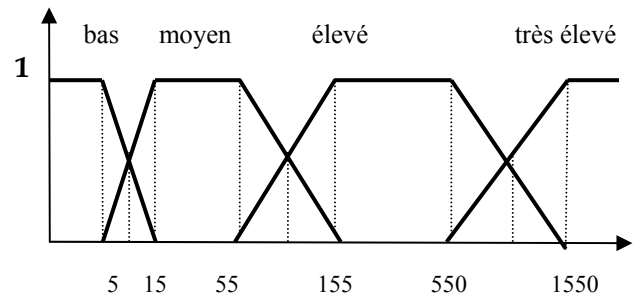


Figure 4. Ensembles flous générés empiriquement pour l'attribut DATA.

Les quantificateurs linguistiques RIM utilisés, Q , dans l'évaluation de la similarité globale entre deux projets logiciels sont définis par l'équation (9) du fait que la base COCOMO'81 ne contient pas les informations nécessaires pour déterminer le quantificateur adéquat. En plus, ce genre de quantificateurs de l'équation (9) peuvent se rapprocher avec une grande précision de plusieurs types de quantificateurs RIM.

$$Q(r) = r^\alpha \quad \alpha > 0 \quad (9)$$

L'analyse des résultats de l'évaluation de la précision des estimations de *Fuzzy Analogy* montre que celle-ci dépend du quantificateur linguistique utilisé pour l'évaluation de la similarité globale (α) et ceci dans les deux cas des processus de génération des ensembles flous (empirique ou FCM-AGCR) et quelque soit la forme des fonctions d'appartenance. (Figures 5 à 7). En général, nous constatons que la précision des estimations, mesurée en terme de $Pred(20)$, est monotone croissante en fonction de α . Cela résulte du fait que nos mesures de similarité sont monotones décroissantes en fonction de α :

- Quand α tend vers zéro, l'évaluation de la similarité globale ne prend en considération que peu d'attributs de ceux utilisés dans la description des projets logiciels; le minimum est de considérer un seul attribut. Ainsi, il est fort probable de trouver, pour deux projets logiciels quelconques de la base de données COCOMO'81, un attribut dont la valeur linguistique associée est la même pour les deux projets. Par conséquent, leur similarité globale serait élevée (au voisinage de 1).

- Quand α tend vers l'infini, l'évaluation de la similarité globale prend en considération la plupart des attributs décrivant les projets logiciels; le maximum doit considérer tous les attributs. Ainsi, il est fort probable de trouver, pour deux projets logiciels quelconques de la base de données COCOMO'81, un attribut dont les valeurs linguistiques associées à ces deux projets sont différentes. Par conséquent, leur similarité globale serait faible (au voisinage de 0).

Quand à la comparaison des précisions des estimations fournies par *Fuzzy Analogy* dépendamment de la procédure utilisée pour générer les ensembles flous des attributs, nous constatons que le cas d'une procédure empirique (courbes DBx-Trapezoidale-Empirique dans les figures 5 à 7) permet de fournir des estimations plus précises que le cas de la procédure FCM-AGCR (courbes DBx-x-FCM-AGCR dans les figures 5 à 7) et ceci pour les trois bases de projets (DB1, DB3 et DB4). Cependant, l'utilisation des méthodes automatiques (Par exemple FCM-AGCR) plutôt que des méthodes empiriques nécessitant l'intervention des experts a deux avantages :1) elles peuvent être appliquées sur des données brutes et ceci indépendamment de la disponibilité de connaissances fournies par des experts; 2) elle offrent la possibilité aux utilisateurs de choisir certains paramètres dans le processus de génération d'une représentation floue tels que, le nombre d'ensembles flous et la forme des fonctions d'appartenance.

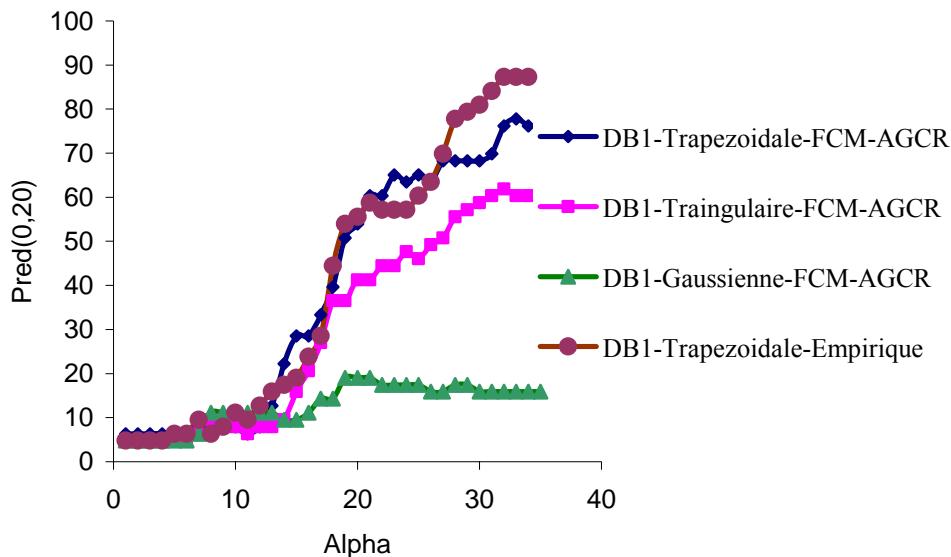


Figure 5. Relation entre la précision de Fuzzy Analogy ($Pred(0,20)$) et le quantificateur (α) utilisé pour la base de données DB1

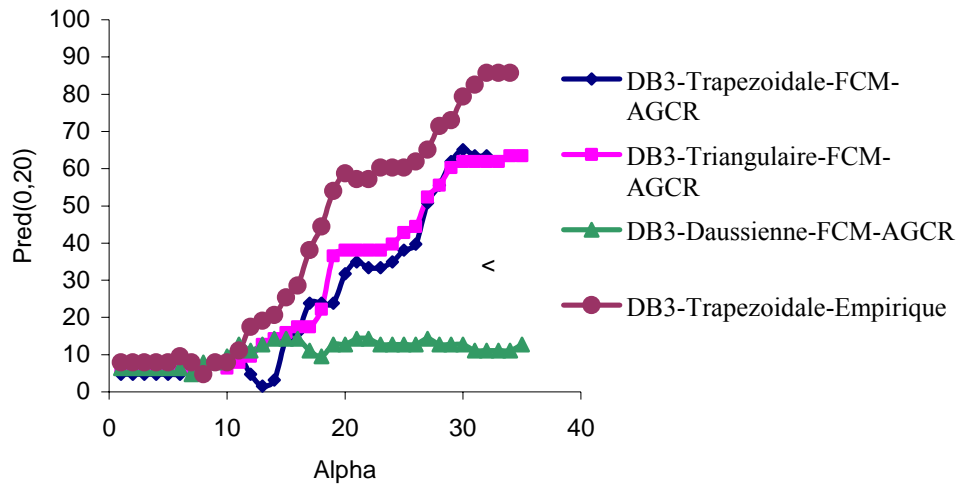


Figure 6. Relation entre la précision de Fuzzy Analogy ($Pred(0,20)$) et le quantificateur (α) utilisé pour la base de données DB3

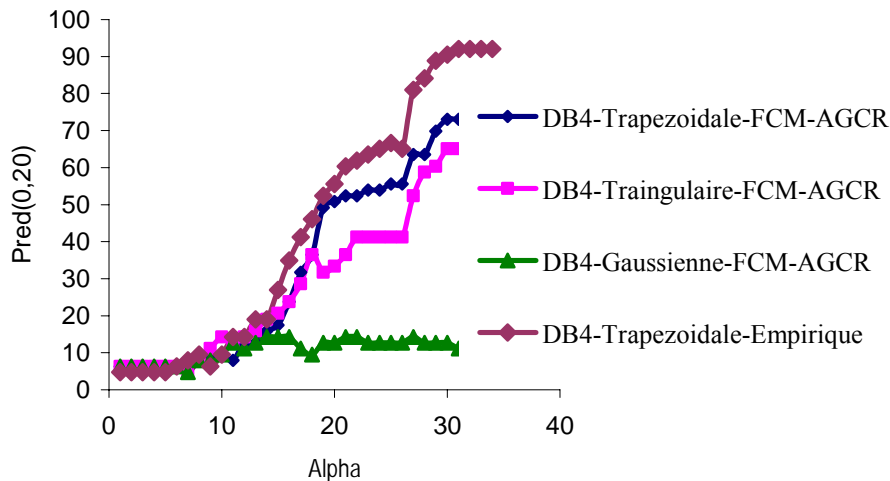


Figure 7. Relation entre la précision de Fuzzy Analogy ($Pred(0,20)$) et le quantificateur (α) utilisé pour la base de données DB4

Bien que Fuzzy Analogy fournisse des estimations très précises ($Pred(20) \geq 85$) dans le cas d'une génération empirique des ensembles flous, sa précision dans le cas d'une génération des ensembles flous avec la procédure FCM-AGCR est acceptable. En effet, l'indicateur $Pred(20)$ est supérieur à 70 dans le cas d'ensembles flous trapézoïdaux (70 étant le seuil d'acceptation reconnu dans le littérature des modèles d'estimation des coûts de logiciels). Dans le cas d'ensembles flous triangulaires, le $Pred(20)$ de Fuzzy Analogy est aux alentours de 65 ; ce qui est un peu moins du seuil d'acceptation. Cependant, le cas d'ensembles flous gaussiens ne permet pas de générer des estimations précises ($Pred(20)$ aux alentours de 20) du fait que la procédure FCM-AGCR ne garantit pas, contrairement aux cas des formes trapézoïdale ou triangulaire, que les ensembles flous générés

formeront une partition floue (la somme des degrés d'appartenance d'un élément à tous les ensembles flous est égal à 1).

4. CONCLUSION

Dans cet article, nous avons évalué la précision d'un modèle d'estimation des coûts de logiciels baptisé Fuzzy Analogy. Cette évaluation est réalisée sur quatre bases de projets logiciels contenant chacune 63 projets décrits par 13 attributs. Le processus d'estimation de Fuzzy Analogy utilise une représentation floue des attributs décrivant les projets logiciels. Les caractéristiques de cette représentation floue (nombre d'ensembles flous, forme des fonctions d'appartenance, etc.) influencent la précision de Fuzzy Analogy. Ainsi, dans un premier temps, nous avons empiriquement construit cette représentation floue en

se basant sur les descriptions des 13 attributs fournies dans [Boehm, 1981]. Les fonctions d'appartenance générées sont trapézoïdales. Dans un second temps, nous avons utilisé la technique FCM et un algorithme génétique à codage réel (AGCR) pour construire les ensembles flous ainsi que leurs fonctions d'appartenance. Les fonctions d'appartenance générées peuvent être trapézoïdales, triangulaire ou gaussiennes. L'évaluation de la précision des estimations fournies par Fuzzy Analogy a montré que sa précision dans le cas d'une génération empirique de la représentation floue est meilleure que celle dans le cas d'une génération à l'aide de la procédure FCM-AGCR. Ceci peut s'expliquer par le fait que la génération empirique, basée essentiellement sur les connaissances des experts, représente souvent convenablement les situations étudiées. Cependant, elle n'est toujours possible du fait que les connaissances empiriques ne sont pas souvent disponibles. Ainsi, le recours à des techniques automatiques telles que la procédure FCM-AGCR s'avère obligatoire. En plus, son utilisation pour la génération de la représentation floue dans Fuzzy Analogy a permis d'obtenir des estimations acceptables surtout dans le cas d'une représentation floue trapézoïdale.

5. REFERENCES

- [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.
- [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.
- [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. Loazano, A.m. Sanchez. "A taxonomy for the crossover operator for real-coded genetic algorithms: An experiment study." *International Journal of Intelligent. Systems* 2003, 18(3), pp: 309-338
- [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, 2002a] A. Idri, A. Abran, and T. M. Khoshgoftaar. "Estimating Software Project Effort by Analogy Based on Linguistic values", *8th International Symposium on Software Metrics*, June 2002, Ottawa, Canada, pp.21-30. IEEE Computer Society.
- [Idri, 2002b] 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.
- [Idri, 2006]. Idri, A., Zahi, A., Abran, A. , "Generating Fuzzy Term Sets for Software Project Attributes using Fuzzy C-Means and Real Coded Genetic Algorithms", *ICT4M*, Malaysia, November 21-23, 2006, pp. 120-127
- [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.
- [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.
- [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.
- [Yager, 1996] R.R. Yager, "Quantifier Guided Aggregation using OWA Operators", *International Journal of Intelligent Systems*, 11, 1996, pp.49-73
- [Zadeh, 1965] L. A. Zadeh. " Fuzzy sets", *Information and Control*, vol. 8, pp. 338-352.