

La Logique Floue Appliquée aux Modèles d'Estimation d'Effort de Développement de Logiciels

Cas du Modèle COCOMO'81

Ali IDRI & Alain ABRAN

Résumé

Au début des années'90, grâce aux travaux de Zadeh et autres, la logique floue connaîtra un succès considérable au niveau du développement de ses fondements théoriques ainsi qu'au niveau de ses applications aux différents domaines (robotique, médecine, traitement des images, etc.). Or comme la plupart des modèles d'estimation ont été publiés dans la décade précédente, plusieurs des concepts utilisés actuellement dans ces modèles d'estimation sont encore basés sur la logique classique. Notre travail consiste à 'fuzzifier' ces concepts.

Actuellement, dans les modèles d'estimation d'effort, plusieurs facteurs tels l'expérience des programmeurs, la fiabilité d'un logiciel et la complexité d'un module sont mesurés sur une échelle ordinale composée de qualifications (valeurs linguistiques en logique floue) telles que 'très bas', 'bas' et 'moyen'. Ces valeurs linguistiques sont représentées, dans ces modèles, par des intervalles classiques. Cette représentation pose de sérieux problèmes dans le processus d'estimation et affecte la précision des estimations d'un modèle. Pour remédier à ceci, nous avons jugé utile d'appliquer quelques concepts de la logique floue aux modèles d'estimation du coût. Notre première application concerne le modèle COCOMO'81 de B. W. Boehm. Il s'agit d'évaluer le COCOMO'81 en représentant ses valeurs linguistiques par des ensembles flous plutôt que par des intervalles classiques. Les résultats obtenus indiquent que la logique floue permet au modèle COCOMO'81 de tolérer les imprécisions et de diminuer ainsi sa sensibilité à des petites perturbations au niveau de ses entrées.

Abstract

When most software cost models were published, Fuzzy Logic (FL) was not yet grounded on solid theoretical foundations. This has not been achieved until Zadeh and others did so in the nineties. Thus, it is not surprising that some of the concepts defined or used in these models are somewhat incompatible with the Fuzzy Logic. Most software cost models use many factors (cost drivers) such as size, reliability and complexity to estimate the effort required to develop the software. In most cases, these factors are measured on an ordinal scale with ranking such as 'Very Low' and 'Low' (linguistic values in fuzzy logic). These linguistic values are represented by classical intervals. This representation leads to serious problems in the estimation process and affects the accuracy of the software cost models. In order to avoid this, we apply some concepts of fuzzy logic to these models. Our first application uses the COCOMO' 81 cost model, specifically its intermediate version, where its six linguistic values will be represented by fuzzy sets rather than classical intervals. Our fuzzy intermediate COCOMO'81 model is evaluated next on three datasets deduced from the original COCOMO'81 database and is found significantly better than the classical intermediate COCOMO'81. This work is also applicable to the COCOMOII model.

Mots-clés : La mesure de logiciels, Les modèles d'estimation du coût, COCOMO'81, Logique floue.

INTRODUCTION

Bien que des chiffres précis soient difficiles à établir, il est de plus en plus admis que les coûts de développement de logiciels représentent une large part des coûts informatiques. La croissance de ces coûts ainsi que les difficultés que l'on rencontre pour les prévoir et les contrôler, car ils sont basés sur des impondérables tels que la productivité du personnel ou la taille du logiciel, constituent encore aujourd'hui une préoccupation des chercheurs et des responsables informatiques. C'est pour rationaliser, donc, le processus d'estimation du coût que plusieurs modèles ont été proposés. Ces modèles peuvent être classifiés en deux catégories :

- les modèles algorithmiques, et
- les modèles non-algorithmiques.

Les modèles algorithmiques sont les plus populaires (du moins dans la littérature). Ils sont développés en appliquant des méthodes statistiques (la régression simple ou multiple, l'Analyse en Composantes Principales, l'analyse Bayésienne, etc.) ou des méthodes de l'analyse numérique telle que l'interpolation polynomiale sur des données historiques. Des exemples de tels modèles sont COCOMO'81 [Boehm 81], COCOMOII [Boehm 95], IBM-FSD [Walston 77], PUTNAM-SLIM [Putnam 78], et ceux se basant sur les points de fonction [Albrecht 83] [Matson 94]. Les avantages et les inconvénients de ces modèles sont :

Avantages :

- Ils sont faciles à développer et à utiliser
- Ils sont facilement programmables

Inconvénients :

- Ils exigent une forme à la fonction de prédiction, souvent de la forme $Effort = A \times size^B$
- Ils doivent être calibrés dans un nouveau contexte
- Ils n'offrent pas une bonne explication à leur comportement

Les modèles non-algorithmiques ont été développés pour remédier aux inconvénients mentionnés ci-dessus. Ils sont basés sur des approches de l'intelligence artificielles telles que les réseaux de neurones, le raisonnement par analogie, et les arbres de régression. Les avantages et les inconvénients de ces modèles sont :

Avantages :

- Ils peuvent modéliser des relations complexes entre les différents facteurs affectant le coût.
- Dans le cas des réseaux de neurones, ils incluent automatiquement des algorithmes d'apprentissage.
- Leur comportement est facilement compréhensible sauf peut-être le cas des réseaux de neurones. Ces derniers sont souvent considérés comme étant 'black boxes' ; des études récentes ont montré qu'on peut transformer un système se basant sur les réseaux de neurones en un système équivalent se basant sur des règles floues [Benitez 97]

Inconvénients

- Ils ne sont pas faciles à développer
- On ne peut pas les utiliser sans les programmer
- Ils impliquent des calculs très intensifs pour générer l'estimation

Jusqu'à présent, ces deux classes de modèles d'estimation (algorithmiques ou non-algorithmiques) utilisent la logique classique : une proposition ne peut être que 'vraie' ou 'fausse'. L'utilisation de cette logique impose des restrictions majeures sur ces deux types de modèles. Tout d'abord, ils ne tolèrent pas les imprécisions dans le processus d'estimation ; de plus ils ne sont pas adaptés aux cas où la plupart des variables affectant le coût sont qualitatives. Pour remédier à ces deux limites, nous utiliserons la Logique Floue (FL) au lieu de la logique classique pour représenter les différents facteurs (variables linguistique dans la logique floue) affectant le coût de développement. Notre recherche s'intéressera donc à l'application de la logique floue aux deux types des modèles d'estimation du coût. Dans cet article, nous traitons le cas des modèles algorithmiques et plus particulièrement le modèle COCOMO'81

Depuis son apparition, le modèle COCOMO'81 ne cesse de faire l'objet de plusieurs travaux de calibration, d'amélioration, et de reformulation [Amrane 93] [Boehm 89] [Boehm 95] [Idri 97a] [Idri 97b] [Marwane 91] [Miyazaki 85]. Le présent travail est une contribution à cette réflexion par l'application du concept d'un ensemble flou au modèle COCOMO'81 Intermédiaire. Cette application consiste à représenter les valeurs linguistiques du modèle par des ensembles flous plutôt que par des intervalles 'classiques'. Le modèle COCOMO'81 existe en trois versions: simple, intermédiaire et détaillé. Le choix de la version intermédiaire du modèle peut être justifié par:

- C'est la version la plus utilisée.

- La précision d'estimation de la version intermédiaire, mesurée en terme de l'erreur relative, est nettement supérieure à celle de la version simple et se rapproche de celle de la version détaillée (Table 1).
- Les données publiées ne permettent l'évaluation que des versions simple ou intermédiaire [Boehm 81].
- La version simple ne prend pas en considération assez de facteurs (seulement deux)

	Version du COCOMO'81		
	Simple	Intermédiaire	Détaillé
Erreur Relative $RE \leq 20$	25%	68%	70%

Table 1. Comparaison de la précision des 3 versions du COCOMO'81

Cet article est composé de quatre sections. Dans la première section, nous présentons d'une manière succincte les concepts fondamentaux de la logique floue. Dans la deuxième section, nous rappelons le principe du modèle COCOMO'81 intermédiaire. Dans la troisième section, nous expliquons pourquoi et comment la logique floue peut être appliquée au modèle COCOMO'81 intermédiaire. La quatrième section décrit l'évaluation du COCOMO'81 intermédiaire lorsque les facteurs sont mesurés par des ensembles flous plutôt que des par intervalles 'classiques'.

1. LA LOGIQUE FLOUE

La logique floue a été fondée par Zadeh [Zadeh 65]. Depuis son apparition, elle ne cesse de faire l'objet de plusieurs travaux de recherches ayant comme objectifs le perfectionnement et l'application de cette nouvelle logique. Pour Zadeh [Zadeh 94], la logique floue est la théorie des ensembles flous (Fuzzy Set Theory). FST est une extension de la théorie des ensembles dits 'ensembles classiques'. La fonction d'appartenance d'un ensemble classique A est définie par :

$$m_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases}$$

Ceci signifie qu'un élément x est soit dans A ($\mu_A(x)=1$) ou non ($\mu_A(x)=0$). Or dans plusieurs situations, il est parfois ambigu si x appartient ou non à A. Par exemple [Jager 95], considérons l'ensemble A représentant les PCs qui sont 'trop chers' pour une population d'étudiants. Après une enquête menée au sein de cette population, un PC ayant un prix supérieur ou égal à \$2500 sera déclaré 'trop cher', ayant un prix inférieur ou égal à \$1000 n'est pas 'trop cher'. Il existe un nombre important de PCs ayant un prix entre ces deux limites. Dans cet intervalle, on peut utiliser des valeurs, comprises strictement entre 0 et 1, pour classer les prix comme étant partiellement 'trop cher'. C'est la définition d'un ensemble flou : un ensemble avec une fonction d'appartenance à valeurs (???) dans [0,1]. Ce degré d'appartenance indique la valeur de vérité de la proposition 'un PC est trop cher'. Un ensemble flou A est noté par :

$$A = \int_X m_A(x) / x$$

où $\mu_A(x)$ est la fonction d'appartenance à A et X est l'ensemble de toutes les valeurs de x (IR dans la plus part des cas). Si $\mu_A(x)$ est égal à 1 alors il est sûr et certain que x est dans A ; $\mu_A(x)$ est égal à 0 implique que sûr et certain x n'appartient pas à A.

La figure suivante illustre la différence entre la représentation de la valeur 'trop cher' par un ensemble classique et par un ensemble flou.

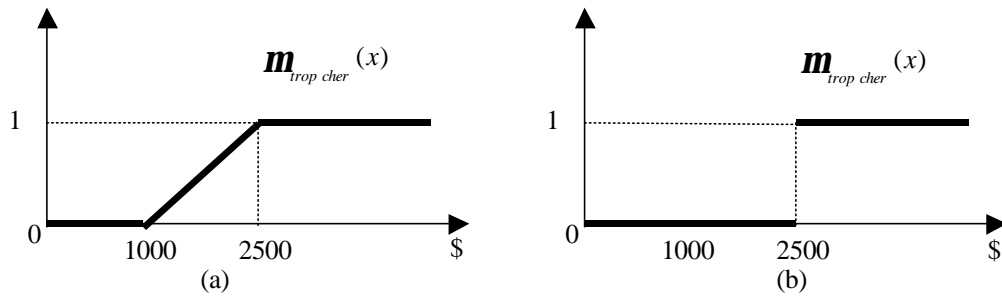


Figure 1. Ensemble flou vs Ensemble classique pour la valeur linguistique 'trop cher'

Parmi les autres axes de la théorie des ensembles flous (FST), on trouve aussi l'arithmétique floue, la théorie des graphes flous, l'analyse des données floues, etc. Dans ce travail, nous s'intéresserons à l'application du concept de l'ensemble flou au modèle COCOMO'81 intermédiaire.

2. COCOMO'81 INTERMÉDIAIRE

Le modèle COCOMO'81 intermédiaire est une extension du modèle COCOMO'81 simple. Il en diffère principalement par la considération de 15 facteurs, en plus de la taille et le mode du logiciel, comme étant les principaux facteurs affectant le coût de développement. Ces 15 facteurs justifient la variation entre les coûts de deux logiciels de même taille. Cette variation n'est pas révélée ni expliquée par le COCOMO'81 simple. L'équation d'estimation du modèle COCOMO'81 intermédiaire de l'effort de développement en Homme-Mois d'un projet logiciel est :

$$HM = a \times taille^b \prod_{i=1}^{15} c_{ij} \quad (1)$$

où *taille* désigne la taille d'un logiciel mesurée en Kilo Instructions Source Livrées (KISL),
a, *b* des constantes qui dépendent du mode du projet logiciel (organique, semi-détaché ou intégré),
C_{ij} le multiplicateur d'effort correspondant à la *j*^{ème} catégorie (valeur linguistique) sélectionnée du *i*^{ème} facteur de coût,
et *k_i* le nombre de catégories (valeurs linguistiques) correspondant au *i*^{ème} facteur de coût

Ces 15 facteurs sont regroupés en quatre classes (Table 2). Les 75 multiplicateurs d'effort correspondant sont donnés dans la table 3.

Attributs Produit

RELY (Required Software Reliability) : Fiabilité requise
DATA (Data Base Size) : Taille de la base de données
CPLX (Product Complexity) : Complexité du logiciel

Attributs Matériel

TIME (Execution Time Constraint) : Contrainte du temps d'exécution
STOR (Main storage Constraint) : Contrainte de la taille mémoire
VIRT (Virtual Machine Volatility) : Instabilité de la machine virtuelle
TURN (Computer Turn around Time) : Temps de restitution des travaux
VEXP (Virtual Machine Experience) : Expérience dans la machine virtuelle

Attributs Personnel

ACAP (Analyst Capability) : Compétence des analystes
AEXP (Application Experience) : Expérience du domaine d'application
PCAP (Programmer Capability) : Compétence des programmeurs
LEXP (Programming Language Experience) : Expérience du langage de programmation

Attributs Projet

MODP (Modern Programming Practices) : Pratiques des méthodes modernes de programmation
TOOL (Use of Software Tools) : Utilisation d'outils logiciels
SCED (Required Development Schedule) : Contrainte des échéanciers de développement

Table 2. Les 15 facteurs du modèle COCOMO'81 intermédiaire

L'effet sur le coût de chacun des 15 facteurs de la table (2) est exprimé en terme de ratio de productivité (Figure 1), qui est le rapport entre la productivité (exprimée en ISL par HM) d'un projet exploitant au mieux le facteur en question et la productivité d'un projet n'exploitant pas ce facteur ; tous les autres facteurs sont supposés identiques dans les deux projets.

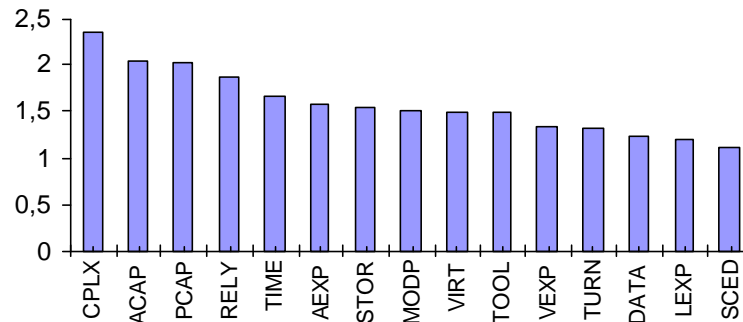


Figure 1. Comparaison des ratios de productivité des 15 facteurs

Attribut	Valeurs linguistiques					
	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.75	0.88	1.00	1.15	1.40	
DATA		0.94	1.00	1.08	1.16	
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
TIME			1.00	1.11	1.30	1/66
STOR			1.00	1.06	1.21	1.56
VIRT		0.87	1.00	1.15	1.30	
TURN		0.87	1.00	1.07	1.15	
ACAP	1.46	1.19	1.00	0.86	0.71	
AEXP	1.29	1.13	1.00	0.91	0.82	
PCAP	1.42	1.17	1.00	0.86	0.70	
VEXP	1.21	1.10	1.00	0.90		
LEXP	1.14	1.07	1.00	0.95		
MODP	1.24	1.10	1.00	0.91	0.82	
TOOL	1.24	1.10	1.00	0.91	0.83	
SCED	1.23	1.08	1.00	1.04	1.10	

Table 3 : Les 75 multiplicateurs de l'effort du COCOMO'81 intermédiaire

3. POURQUOI ET COMMENT LA LOGIQUE FLOUE PEUT-ÊTRE APPLIQUÉE AU MODÈLE COCOMO'81 INTERMÉDIAIRE ?

Dans le modèle COCOMO'81 intermédiaire, 15 facteurs (Table 3) parmi 17 sont mesurés sur une échelle de type 'ordinal'¹. Cette échelle est composée de six valeurs linguistiques : 'very low', 'low', 'nominal', 'very high', et 'extra-high'. Ces six valeurs sont représentées, dans le modèle, par des intervalles classiques ([Boehm 81], pp. 119). Par exemple, le facteur DATA, expliquant l'influence de la taille de la base de données sur le coût du logiciel, est mesuré par le ratio :

$$\frac{D}{P} = \frac{\text{Database size in bytes or characters}}{\text{Program size in DSI}}$$

Ensuite, une valeur linguistique est attribuée à DATA en fonction du ratio D/P :

¹ Une échelle est de type 'ordinal' si les seules opérations applicables sur cette échelle sont la distinction et l'ordre [Fenton 97]

	Low	Nominal	High	Very High
D/P	D/P < 10	10 ≤ D/P < 100	100 ≤ D/P < 1000	D/P ≥ 1000

Table 4 : Valeurs linguistique du facteur DATA

Par conséquent chaque projet logiciel appartient à une et une seule classe. Par exemple, si D/P est égal à 9,99 alors le DATA du projet logiciel sera déclaré 'low'; si D/P est égal à 10,01 alors il sera 'nominal'. Ceci pose de sérieux problèmes pour les projets ayant des ratios D/P au voisinage des limites. Supposons que nous avons deux projets logiciels P₁ et P₂ tels que :

- Pour tous les facteurs ayant une influence croissante sur le coût (RELY, DATA, CPLX, TIME, STORE, VIRT, TURN), P₁ est juste avant la borne inférieure de l'intervalle représentant la valeur 'high', P₂ est juste après.
- Pour les facteurs ayant une influence décroissante sur le coût (ACAP, AEXP, PCAP, VEXP, LEXP, MODP, TOOL), P₁ est juste après la borne inférieure de l'intervalle représentant la valeur 'nominale', P₂ est juste après.

Dans ce cas, si les deux projets logiciels P₁ et P₂ ont le même effort nominal (Effort_{nom}=A×size^B), 15H-M, alors l'effort ajusté (Équation 1) de P₁ est inchangé, tandis que celui de P₂ est 52HM!

Pour remédier à ce problème, dans le cas du modèle COCOMO'81 intermédiaire, nous proposons de représenter les valeurs linguistiques utilisées dans le modèle par des ensembles flous plutôt que des ensembles classiques. Les avantages de cette représentation sont :

- Elle est plus générale ;
- Elle imite la manière avec laquelle les humains interprètent les valeurs linguistiques ;
- La transition d'une valeur linguistique à la valeur linguistique suivante est progressive plutôt que brusque.

Par exemple, pour le cas du facteur DATA, nous avons défini pour chaque valeur linguistique un ensemble flou avec une fonction d'appartenance μ (Figure suivante).

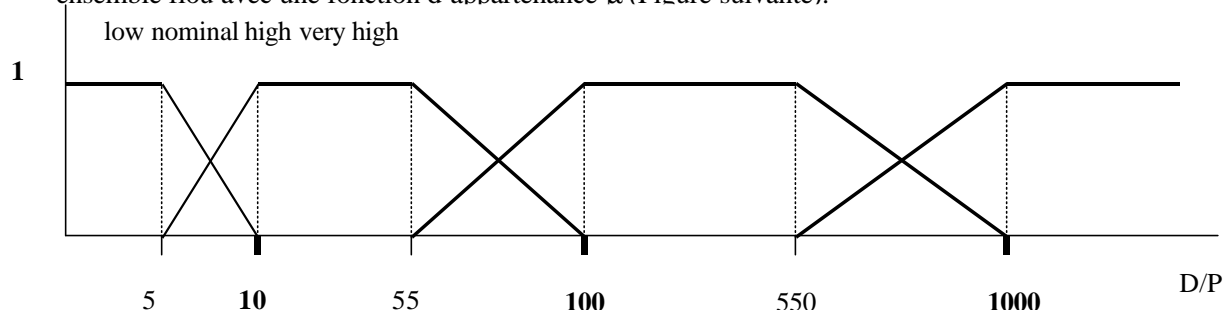


Figure 1. Les fonctions d'appartenance des différents ensembles flous définis pour les quatre valeurs linguistiques du facteur DATA.

Nous avons aussi défini les ensembles flous correspondant aux autres facteurs du modèle COCOMO'81 intermédiaire (voir Annexe 1). Parmi ces 15 facteurs, les quatre attributs RELY, CPLX, MODP, et TOOL n'ont pas été 'fuzzifiés' car leurs descriptions données dans [Boehm 81] sont insuffisantes.

4. EVALUATION

Dans cette section, nous évaluons le modèle COCOMO'81 intermédiaire en utilisant l'équation (1) et les nouveaux multiplicateurs d'effort (F_Cij) obtenus à partir des ensembles flous. Les multiplicateurs d'effort, F_Cij, sont calculés à partir des multiplicateurs d'effort classiques (Table 3) et les fonctions d'appartenance μ associées aux différents ensembles flous définis pour les facteurs du modèle :

$$F_{-}C_{ij} = F(m_{A_i}^V(P) \dots m_{A_j}^V(P), C_{i1} \dots C_{ij})$$

La détermination de la fonction F nécessite une étude très approfondie. Elle peut être linéaire, polynomiale, exponentielle, ou autres. Pour simplifier, nous considérons que F est une fonction linéaire :

$$F - C_{ij} = \sum_{j=1}^{k_i} m_{A_j}^{V_i}(P) \times C_{ij} \quad (2)$$

Où $m_{A_j}^{V_i}$ est la fonction d'appartenance à l'ensemble flou A_j et les A_j s sont les ensembles flous associés au facteur V_i .

Cette évaluation consiste à comparer les estimations du 'fuzzy' COCOMO'81 intermédiaire par rapport aux valeurs réelles du coût. Comme dans le cas de la version COCOMO'81 intermédiaire classique, les cinq quantités suivantes sont utilisées comme indicateurs du degré de précision :

- Pourcentage de projets ayant une Erreur Relative (ER) inférieure à 20:

$$ER(P_i) = \left| \frac{MM_{est}^{P_i} - MM_{act}^{P_i}}{MM_{act}^{P_i}} \right| \times 100$$

où $MM_{est}^{P_i}$ est la valeur estimée de l'effort du projet P_i ,
et $MM_{act}^{P_i}$ est la valeur réelle de l'effort du projet P_i

- Min de ER_i ,
- Max de ER_i ,
- Moyen de ER_i ,
- Ecart-type de ER_i

L'évaluation se fera sur trois bases de données déduites, déduite en utilisant une procédure aléatoire à partir de la base de données du modèle COCOMO'81. En effet, cette dernière contient seulement les multiplicateurs d'effort associés aux 63 projets du modèle ; or pour calculer les $m_{A_j}^{V_i}(P)$ de la formule (2), nous aurons besoin des valeurs réelles des différents facteurs des 63 projets. Ainsi, chacune des trois bases contient 63 projets décrits par des valeurs générées aléatoirement selon les qualifications de leurs facteurs données dans la base initiale du modèle. Par exemple, le facteur DATA du cinquième projet de la base de données du modèle est déclaré 'low' . Par conséquent, selon la table (4), les trois valeurs générées pour les trois bases sont entre 0 et 10. La table suivante montre les résultats de l'évaluation du 'fuzzy' COCOMO'81 intermédiaire comparés à ceux du COCOMO'81 intermédiaire classique :

	'fuzzy'/classique COCOMO'81 intermédiaire			
	Base de données #1	Base de données #2	Base de données #3	COCOMO'81
Pred(20) (%)	62.14 / 68	46.86 / 68	41.27 / 68	68 / 68
Min RE_i (%)	0.11 / 0.02	0.40 / 0.02	0.06 / 0.02	0.02 / 0.02
Max RE_i (%)	88.60 / 83.58	3233.03 / 83.58	88.03 / 83.58	83.58 / 83.58
Moyen RE_i (%)	22.50 / 18.52	78.45 / 18.52	30.80 / 18.52	18.52 / 18.52
Ecart-Type RE_i	19.69 / 16.97	404.40 / 16.97	22.95 / 16.97	16.97 / 16.97

Table 4. Résultats de l'évaluation

Les résultats de l'évaluation du 'fuzzy' COCOMO'81 intermédiaire sont différents pour les quatre bases de données contrairement au cas de la version classique qui donne les mêmes degrés de précision pour les quatre bases. Ceci implique que :

- Le 'fuzzy' COCOMO'81 intermédiaire tolère les imprécisions au niveau de ses entrées et génère en conséquence des sorties (effort) progressivement différentes. Cette progression permet au modèle

d'être moins sensible à des petites variations dans ses entrées. Cette stabilité du modèle est l'un des dix critères cités par Boehm dans [Boehm 81] que n'importe quel modèle d'estimation du coût doit satisfaire.

- La version classique du COCOMO'81 intermédiaire ne tolère pas les imprécisions au niveau de ses entrées et donne soit les mêmes sorties ou des sorties significativement différentes. Par conséquent, elle est moins stable et très sensible aux différentes variations dans ses entrées.

Pour la base de données originale (la colonne 5 de la table 4), les deux modèles ont la même précision. En effet, cette base de données est un cas particulier qu'on peut obtenir si tous les $m_{A_j}^{V_i}(P)$ sont nuls sauf pour un seul A_j

5. CONCLUSION & PERSPECTIVES

Dans ce travail, nous avons utilisé les ensembles flous plutôt que les ensemble classiques pour représenter les différentes valeurs linguistiques du modèle COCOMO'81 intermédiaire. Ainsi, pour chaque facteur du modèle, nous avons défini les ensembles flous correspondant à ses valeurs linguistiques. Ces ensembles flous ont été représentés par des fonctions d'appartenance trapézoïdales. Les résultats de notre évaluation sont affectés par ce choix. Cependant, il existe d'autres représentations possibles d'un ensemble flou qui pourraient être utilisées (représentation Gaussienne, Bell, etc.). Pour choisir la meilleure représentation, il faudrait étudier la signification de chaque valeur linguistique dans l'environnement à partir duquel le modèle COCOMO'81 intermédiaire a été mis au point.

Notre travail s'applique aussi au modèle COCOMO II où 22 facteurs parmi 24 sont mesurés sur une échelle de type 'ordinal'. Toutefois, parce que les données de ce modèle ne sont pas encore publiées, nous avons traité seulement le cas du COCOMO'81.

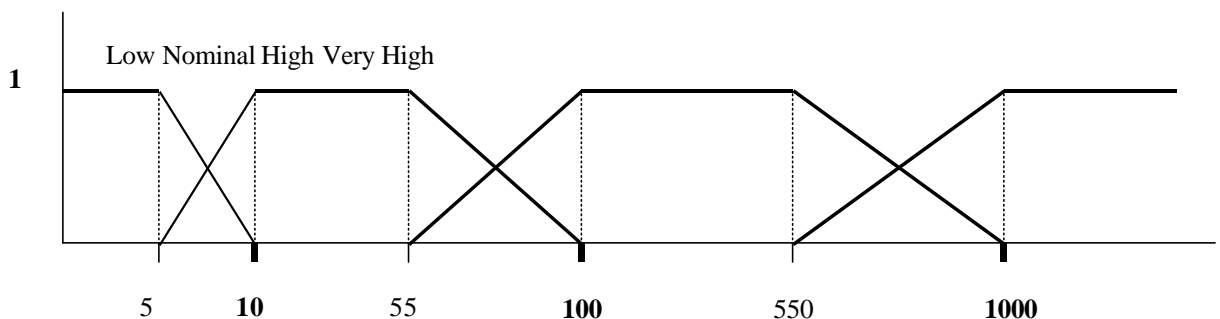
Il existe d'autres définitions utilisées dans le modèle COCOMO'81 qui sont incompatibles avec la logique floue :

- Les trois modes utilisés par le modèle doivent être 'fuzzifiés'.
- La taille d'un logiciel mesurée en KISL peut être représentée par des nombres flous plutôt que des nombres exacts

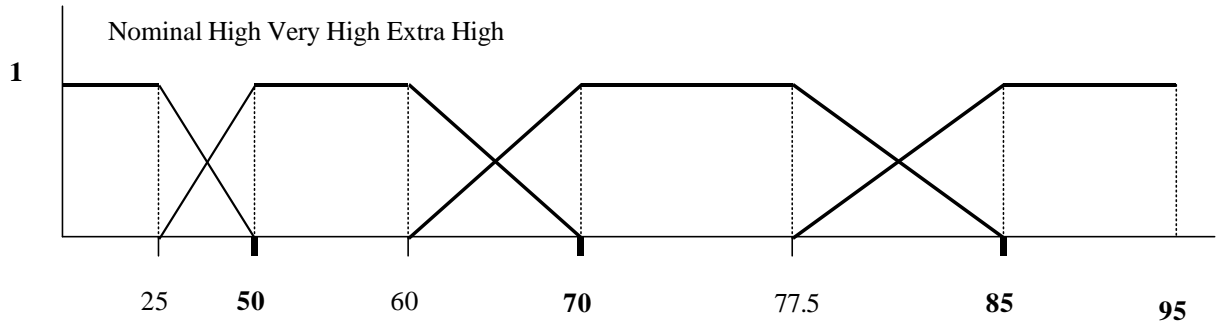
Plus qu'une 'fuzzification' complète du modèle, notre réflexion s'oriente vers les modèles non-algorithmiques en particulier ceux utilisant les réseaux de neurones. Leur principal avantage est qu'ils permettent l'apprentissage : une caractéristique très importante qu'un modèle d'estimation doit satisfaire afin de s'adapter aux éventuels changements dans l'environnement à partir duquel il a été développé. Notre objectif est donc de développer un modèle d'estimation tolérant les imprécisions (logique floue) et permettant l'apprentissage (réseaux de neurones). Un tel modèle supportera le concept du Soft Computing défini par Zadeh [Zadeh 94].

ANNEXE 1 : Les ensembles flous correspondant aux facteurs retenus du modèle COCOMO'81

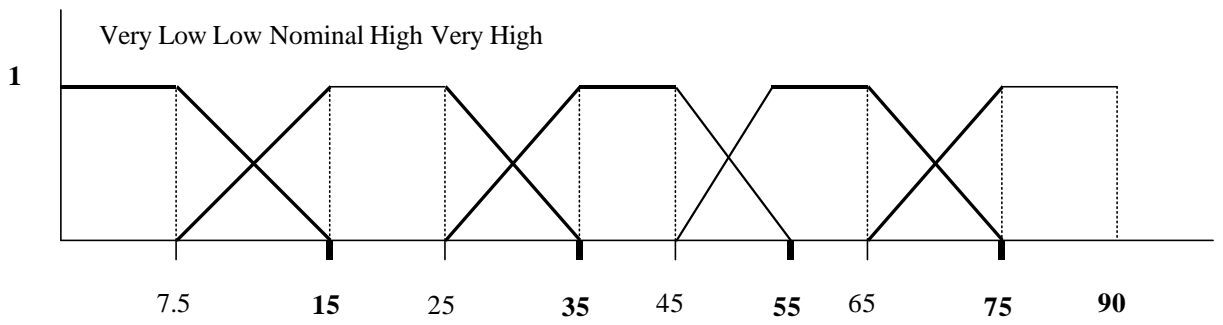
DATA (Data Base Size)



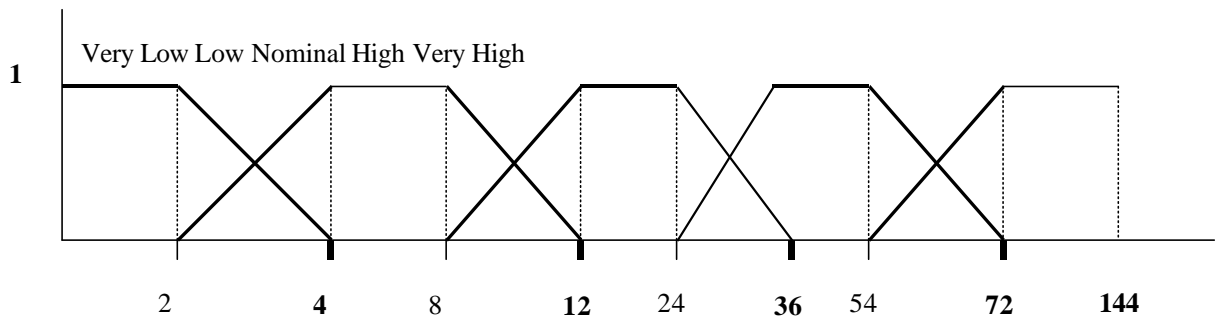
TIME & STOR (Execution Time Constraint, Main Storage Constraint)



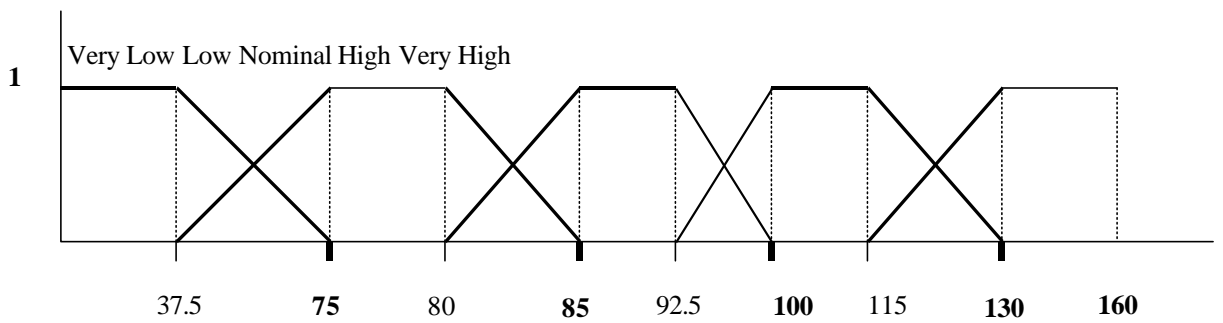
ACAP & PCAP (Analyst Capability, Programmer Capability)



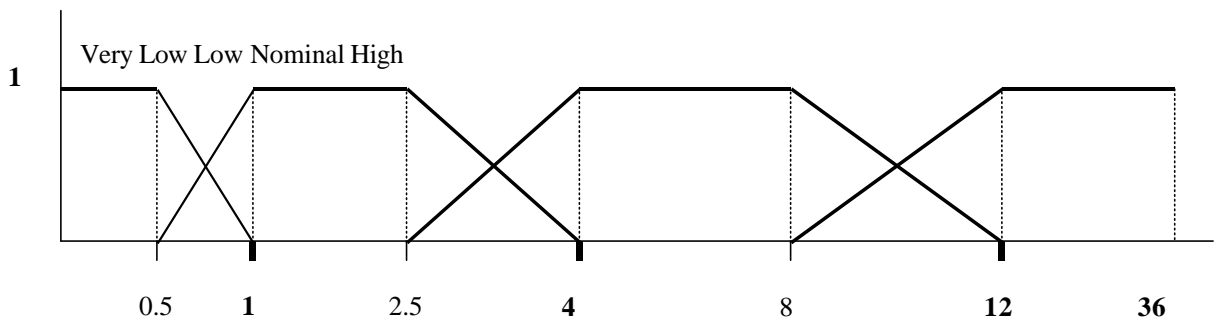
AEXP (Application Experience)



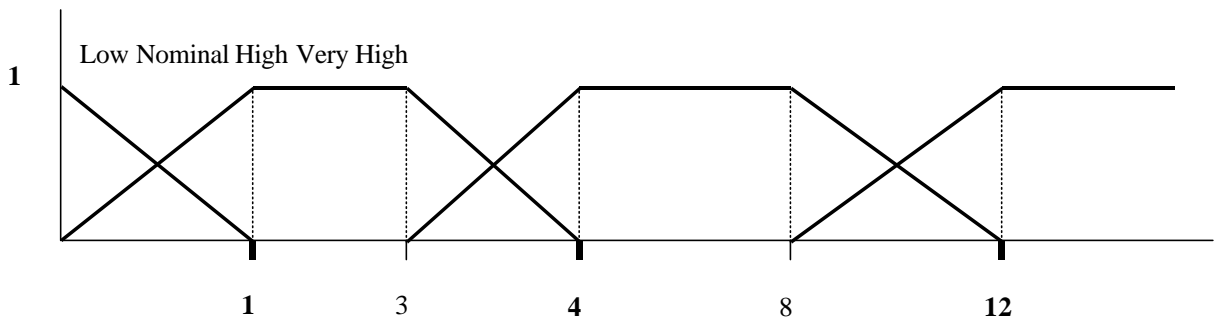
SCED (Required Development Schedule)



VEXP & LEXP (Virtual Machine Experience, Programming Language Experience)

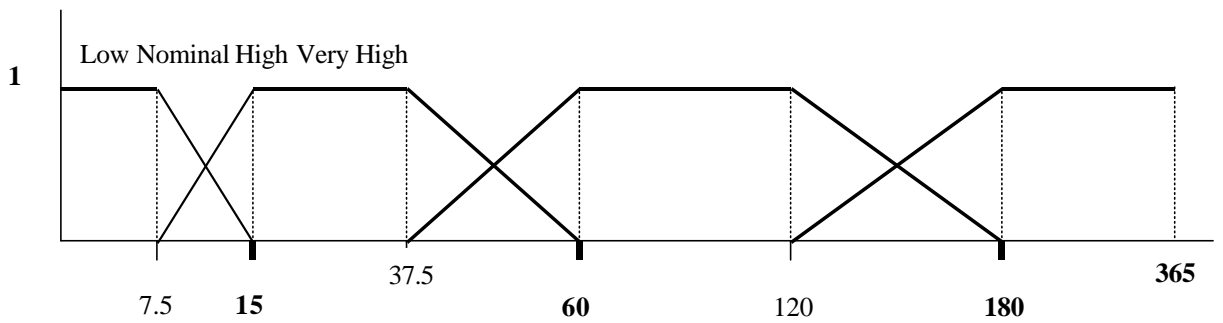


TURN (Computer Turnaround Time)

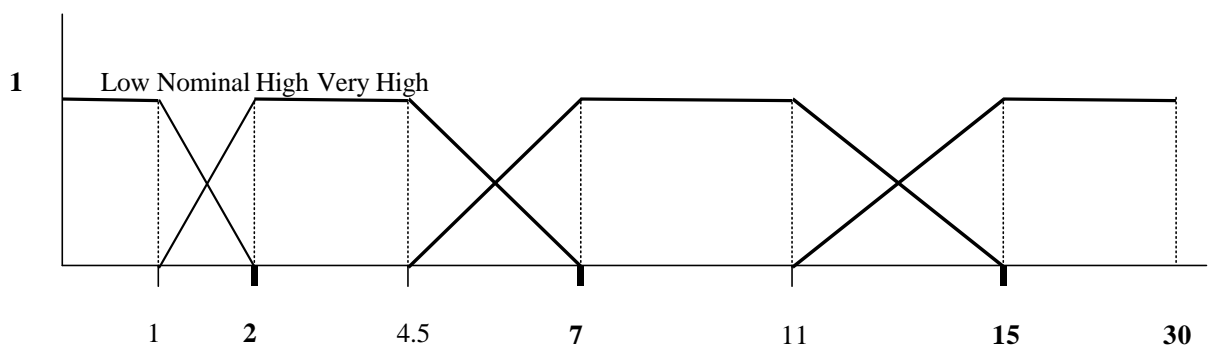


VIRT (Virtual Machine Volatility)

Major



Minor



BIBLIOGRAPHIE

- [**Albreach 83**] Albrecht, A.J., Gaffney, J.E., 'Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation', *IEEE Transactions on Software Engineering*, Vol. SE-9, no 6, November, 1983, pp. 639-647
- [**Amrane 93**] Amrane, B., Slimani, Y., 'ALCOMO Model: A statistical reformulation of the COCOMO'81 model', *Information science and Technology*, April, 1993.
- [**Benitez 97**] Benitez J. M., Castro J. L., Requena I., 'Are artificial Neural Networks Black Boxes?', *IEEE Transactions On Neural Networks*, Vol. 8, no 5, September, 1997, pp1156-1164
- [**Boehm 81**] Boehm, B.W., *Software Engineering Economics*, Prentice-Hall, 1981.
- [**Boehm 89**] Boehm, B.W., Royce, W.W. 'Le COCOMO'81 Ada', *Génie logiciel & Systèmes experts*, 1989.
- [**Boehm 95**] Boehm, B.W., et al., 'Cost Models for Future Software Life Cycle Processes: COCOMO'81 2.0', *Annals of Software Engineering on Software Process and Product Measurement*, Amsterdam, 1995.
- [**Fenton 97**] Fenton, N., Pfleeger, S.L., *Software Metrics: A Rigorous and Practical Approach*, International Computer Thomson Press, 1997.
- [**Gulezian 91**] Gulezian, R., 'Reformulating and Calibrating COCOMO'81', *Journal of Systems Software*, Vol 16, 1991, pp.235-242
- [**Idri 97a**] Idri A., 'Une étude préalable pour la mise au point d'un modèle nationale d'estimation du coût de développement de logiciels : le projet MOCOMO', thèse de de 3^{ème} cycle, ISBN 1530, Juillet, Faculté de sciences, 1997.
- [**Idri 97b**] Idri, A., Griech, B., El Iraki, A., 'Towards an Adaptation of the COCOMO'81 Cost Model to the Software Measurement Theory', In *Proceeding ESEC/FSE*, September, Zurich, 1997.
- [**Jager 95**] Jager, R., 'Fuzzy Logic in Control', Ph.D. Thesis, Technic University Delft, ISBN 90-90008318-9, Dutsh, 1995.
- [**Jones 86**] Jones, C., *Programming Productivity*, McGraw-Hill, New York, 1986.
- [**Marwane 91**] Marwane, R., Mili, A., 'Building Tailor-made Software Cost Model: Intermediate TUCOMO', *Inf. Soft. Techn.*, Vol. 33, no 3, April, 1991, pp.232-238
- [**Matson 94**] Matson, J.E., Barrett, B.E., Mellichamp, J.M., 'Software Development Cost Estimation Using Function Points', *IEEE Transactions on Software Engineering*, Vol. 20, no 4, April, 1994, pp. 275-287
- [**Miyazaki 85**] Miyazaki, Y., Mori, K., 'COCOMO'81 evaluation and tailoring', In *Proceeding Eighth International Conference Software Engineering*, London, UK, August, 1985, pp.292-299
- [**Putnam 78**] Putnam, L.H., 'A General Empirical Solution to the Macro Software Sizing and Estimation Problem', *IEEE Transactions on Software Engineering*, Vol. SE-4, no 4, July, 1978.
- [**Watson 77**] Walston, C.E., Felix, A.P., 'A Method of Programming Measurement and Estimation', *IBM Systems Journal*, Vol 16, no. 1, 1977.
- [**Zadeh 65**] Zadeh, L.A., 'Fuzzy Set', *Information and Control*, Vol. 8, 1965, pp. 338-353.
- [**Zadeh 94**] Zadeh, L. A., 'Fuzzy Logic, Neural Networks, and Soft Computing', *Communications of the ACM*, Vol. 37, no 3, March, 1994, pp.77-84.