HAL
open science

Extraction de règles linguistiques floues par programmation gènétique

Perfecto Malaquias Quintero Flores, Federico del Razo Lopez, Anne Laurent, Pascal Poncelet

To cite this version:

Perfecto Malaquias Quintero Flores, Federico del Razo Lopez, Anne Laurent, Pascal Poncelet. Extraction de règles linguistiques floues par programmation gènétique. LFA: Logique Floue et ses Applications, Nov 2010, Lannion, France. lirmm-00504072

HAL Id: lirmm-00504072

https://hal-lirmm.ccsd.cnrs.fr/lirmm-00504072

Submitted on 31 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Extraction de Règles Linguistiques Floues par Programmation Génétique

P. M. Quintero Flores[1,2] F. Del Razo Lopez[3] A. Laurent[2] P. Poncelet [2]
[1] I T Apizaco, Mexique
[2] LIRMM - Univ. Montpellier 2 - CNRS, France
[3] I T Toluca, Mexique

{quinterofl,laurent,poncelet}@lirmm.fr
delrazo@ittoluca.edu.mx

Résumé :

Les systèmes génétiques flous permettent de traiter l'imperfection des données du monde réel et de concevoir des systèmes de contrôle, de diagnostic médical, d'aide à la décision, de fouille de données, etc. Un système génétique flou est un système capable d'apprendre à l'aide d'un algorithme évolutionnaire (stratégies évolutives, programmation évolutive, algorithmes génétiques, programmation génétique, évolution différentielle, etc.). Dans cet article nous proposons une méthode d'apprentissage de règles linguistiques floues fondée sur la programmation génétique. Notre approche permet d'extraire des règles de Mamdani, à partir de grandes bases de données d'entrée - sortie de systèmes. La méthode a été testée dans deux contextes, le premier dans le domaine de la psychologie, le deuxième dans le domaine chimique et le traitement des eaux usées.

Mots-clés :

Systèmes flous, systèmes génétiques flous, programmation génétique.

Abstract:

Genetic Fuzzy Systems have proven to be practical tools for handling uncertainty in the real world and the design of control systems, medical diagnosis, decision making, data mining, etc.. A genetic fuzzy system is a fuzzy system with layers of learning with an evolutionary algorithm (evolutionary strategies, evolutionary programming, genetic algorithms, genetic programming, differential evolution, etc.). In this paper we propose a learning method of fuzzy linguistic rules supported by genetic programming. Our approach is to identify the Mamdani type fuzzy rules from large databases of input - output of the subsystems tarjet, this independently of the implementation of such systems. The method was tested in two experiments, the first in the field of psychology, the second in chemical and wastewater treatment.

Keywords:

Fuzzy systems, genetic fuzzy systems, genetic programming.

1 Introduction

Au cours des années 80, les systèmes flous (SF) ont démontré leur énorme potentiel pour résoudre les problèmes de classification, de modélisation, d'aide à la décision, de diagnostic ou encore de contrôle [15], dans un grand nombre d'applications. Dans de nombreux cas, la clé du succès est la capacité des SF d'intégrer l'expertise humaine vague et incertaine [15] et [12]. Dans les années 90, malgré les succès antérieurs des SF, le manque de capacités d'apprentissage a été le verrou qui a suscité de nombreux travaux [12]. Ont alors émergé, aux côtés des réseaux de neurones et arbres de décision flous, les systèmes génétiques flous [6].

Un système génétique flou (SGF) est un système flou ayant une capacité d'apprentissage fondés sur un algorithme évolutionnaire (stratégies évolutives, programmation évolutive, algorithmes génétiques, programmation génétique, évolution différentielle, etc.) [8]. Dans les dix dernières années, les travaux de recherche en SGF ont notamment porté sur les systèmes temps réel [1, 2, 11].

Dans cet article, nous proposons une méthode d'apprentissage de règles linguistiques floues fondée sur la programmation génétique. Notre approche permet d'identifier des règles de Mamdani, à partir de grandes bases de données d'entrée - sortie de systèmes. La méthode a été testée dans deux contextes, le premier dans le domaine de la psychologie et le deuxième dans le domaine chimique et le traitement des eaux usées.

L'article est organisé comme suit : la section 2 présente les travaux existants par rapport à des SGF. La section 3 introduit notre approche

d'apprentissage de règles linguistiques floues fondée sur la programmation génétique. Les expérimentations sont rapportées dans la section 4. Enfin la section 5 contient les conclusions de l'article.

2 Apprentissage supervisé avec Programmation Génétique

Dans [17], les auteurs présentent une technique d'extraction de connaissances à partir des bases de données médicales dans le domaine de l'orthopédie, dont l'une contient des données sur les fractures et l'autre sur la scoliose. Les connaissances acquises sont représentées dans deux structures : règles de production et réseaux bayésiens. Les règles contiennent des connaissances concernant les motifs, tandis que les réseaux bayésiens contiennent des relations de causalité entre fractures et la scoliose des enfants. Cette technique utilise un algorithme appelé programmation génétique générique, pour la génération automatique des règles de production. La structure générale et les paramètres de réseaux bayésiens sont générés par programmation évolutionnaire et un algorithme génétique. Un système appelé Lil-GP dans [7] est décrit avec un ensemble de données sur 138 cas de patients présentant des douleurs abdominales.

[19] décrit une méthode d'apprentissage des règles linguistiques floues appelée GEFREX et basée sur un algorithme génétique et un modèle de réseau neuronal artificiel. La méthode a été appliquée avec succès à des problèmes d'approximation, de classification et de prédiction de séries temporelles. Un algorithme d'exploration floue de données est présenté dans [3, 10] pour extraire les règles d'association et les fonctions d'appartenance floues par le biais d'un apprentissage génétique. La technique a été appliquée pour rechercher des motifs fréquents dans les bases de données des recensements de population et pour la prédiction des facteurs de risque de l'athérosclérose.

Dans [4] et [13], les auteurs abordent le problème de la précision et de l'interprétabilité d'un modèle flou et exposent une approche basée sur l'utilisation des algorithmes évolutionnaires multiobjectifs avec application à des problèmes de régression et de prise de décisions.

Une méthode d'apprentissage de règles linguistiques floues compactes et précises basée sur la programmation génétique et une grammaire du contexte-libre est présentée dans [5]. La technique est appliquée aux problèmes de classification de dimension élevée. Dans le cadre de l'extraction de comportements d'usage Web, [21] introduit l'utilisation de l'algorithme de programmation génétique multi-instance appelé G3P-MI pour modéliser les préférences des utilisateurs dans les systèmes d'index de recommandation Web. Le système apprend l'intérêt des utilisateurs par des règles qui ajoutent compréhensibilité et clarté aux modèles découverts et augmentent la qualité des recommandations. [9] présente une étude approfondie des travaux concernant l'application de la programmation génétique dans le cadre de l'apprentissage évolutif des classifieurs.

Une étude concernant l'application des algorithmes évolutionnaires dans la conception des fonctions d'appartenance et des règles floues pour les systèmes de contrôle flou en général et pour un système d'évitement d'obstacles basé sur le comportement pour un robot mobile est présenté dans [14]. Les algorithmes évolutionnaires utilisés sont des stratégies évolutives et des algorithmes génétiques. [20] démontre le potentiel de la PG dans l'identification automatique des bases de règles floues pour le système de contrôle flou intelligent. Ce système gère avec succès un robot mobile le long d'un chemin défini dans un plan, selon ce modèle spatio-temporel développé par Hemami pour une classe de modèles réduits de véhicules basse vitesse (moins de 2 m/s). La structure des règles floues autorisées est décrite par une grammaire contexte-libre restreinte et est arborescente. [16] présente une analyse poussée de l'utilisation des algorithmes évolutionnaires dans l'apprentissage des comportements des

systèmes de contrôle flou.

3 PG-RLF : Programmation génétique de règles linguistiques floues

Dans cet article, nous entendons par système flou apprenant un système capable d'apprendre, depuis une (ou plusieurs) source de données, la structure générale de ses règles floues et les paramètres des fonctions d'appartenance (FsA). Nous décrivons ci-dessous notre proposition.

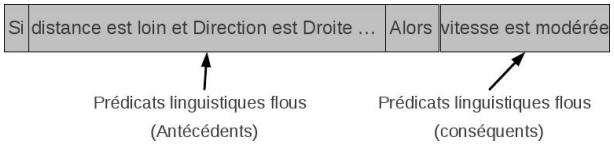

Figure 1 – Structure générale d'une règle floue.

La structure générale d'une règle linguistique floue est décrite figure 1, les paramètres des FsA le sont sur la figure 2. Dans notre approche

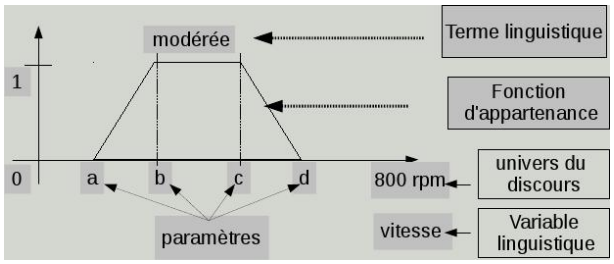

Figure 2 – Paramètres d'une fonction d'appartenance.

d'apprentissage des règles linguistiques floues par la PG, la représentation des individus dans la population se présente comme suit : une règle floue est représentée par une structure d'arbre comme le montre la figure 3. Une base de règles linguistiques floues est représentée elle-aussi par une structure d'arbre dont les sous-arbres sont des règles floues et le nœud racine est l'opérateur logique OU (voir figure 4). La population initiale est formée par deux ou plusieurs bases de règles linguistiques floues ayant

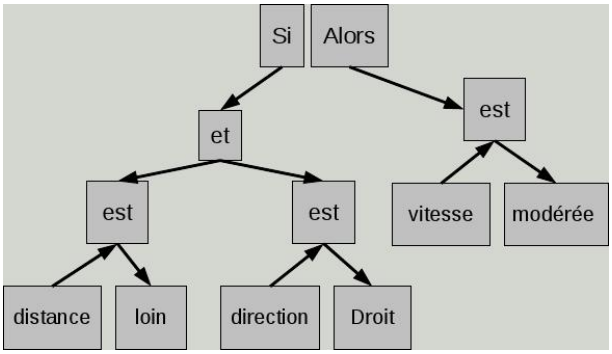

Figure 3 – Arbre d'une règle floue.

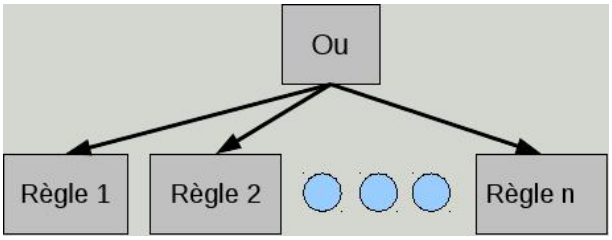

Figure 4 – Arbre d'une base des règles floues.

des aptitudes très faibles. Elles sont donc sujettes à variation et sélection en accord avec l'algorithme de PG résumé figure 6. Ceci permet d'améliorer l'aptitude de chaque base des règles jusqu'à obtenir une base de règles satisfaisant les exigences du problème à résoudre. Dans

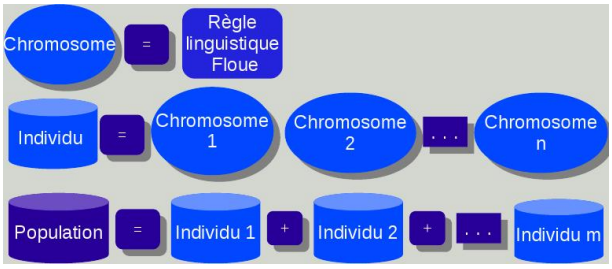

Figure 5 – Terminologie

notre approche un chromosome est donc une règle linguistique floue, un individu est une base de règles linguistiques floues et une population est un ensemble de bases de ces règles (voir figure 5). La sélection est appliquée au niveau des individus et des chromosomes alors que les opérateurs de variation s'appliquent seulement

au niveau des chromosomes. L'évaluation de

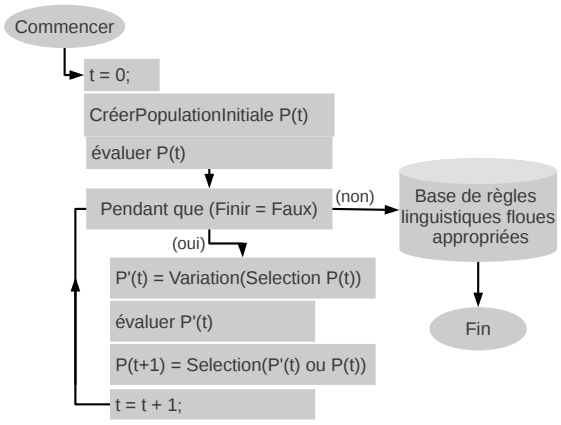

Figure 6 – Évolution des règles floues

chaque base de règles floues avec un vaste ensemble de données de formation est effectuée dans le mécanisme d'inférence floue correspondant (Mamdani, TSK, et Tsukamoto) afin de déterminer l'aptitude de chaque base de règles floues. Les règles floues sont codées avec une grammaire contexte-libre qui est illustrée figure 7.

Symboles Terminaux	=	{Si, Alors, (,), et, est, ou, *, +}
Symboles Non-Terminaux	=	{<RF>, <ANTDTS>, <CONSCT>,
		<PRED_LF>, <VAR_LIN>,.
		<TERM_LF>, < POLYNOME>,.
		<EXP>, <COEF>. }
Productions		<RF>::= Si <ANTDTS> Alors <CONSCT> <ANTDTS> ::= <PRED_LF> [Y <ANTDTS>] <CONSCT>::= <PRED_LF> [Y<ANTDTS>]| <POLYNOME> <PRED_LF> ::= <VAR_LIN> ES <TERM_LIN> <POLYNOME> ::= <VAR_LIN> = <EXP> <EXP> ::= (<COEF>*<VAR_LIN>) {[+ (<COEF> * <VAR_LIN>)]}* + <COEF>

Figure 7 – Grammaire contexte-libre

L'aptitude est calculée pour chaque génération, d'abord au niveau des individus et par la suite au niveau de la population correspondante, moyennant le fait que l'équation (1), représentant l'erreur moyenne minimum admissible pour chaque base de règles floues, doit être inférieure ou égale à 1.

$$f(p) = \frac{1}{n} * \sum_{i=1}^{n} |inf(vs_i) - ve_i| \qquad (1)$$

Où :
- *f(p) :* est l'erreur absolue moyenne d'une base de règles floues de la population p,
- *inf(vs) :* est le vecteur des données produites par le mécanisme d'inférence floue avec les données de formation,
- *ve :* est le vecteur de données de sortie prévue (données enseignants/données attendues),
- *n :* est le nombre de données d'entraînement.

Avec notre approche, il est possible d'obtenir des modèles flous avec un équilibre acceptable entre la précision, l'expressivité et la généralité.

4 Expériences et résultats principaux

Nous mettons en œuvre notre approche en utilisant le langage J++ sur un ordinateur iMac muni du système d'exploitation Mac OS X et un ordinateur PC muni du système d'exploitation Windows XP. Le système dispose d'un bureau de travail comme indiqué figure 8. L'utilisateur dispose d'un gestionnaire de fichiers, d'un gestionnaire de projets et d'une interface graphique qui facilite la mise au point des règles floues et des paramètres des fonctions d'appartenance (triangulaire, trapézoïdale, gaussienne ou en cloche).

La méthode a été testée dans trois contextes, le premier dans le domaine de la psychologie et le deuxième dans le domaine chimique et traitement des eaux usées. Dans les paragraphes suivants, nous décrivons les deux premières applications et les résultats obtenus.

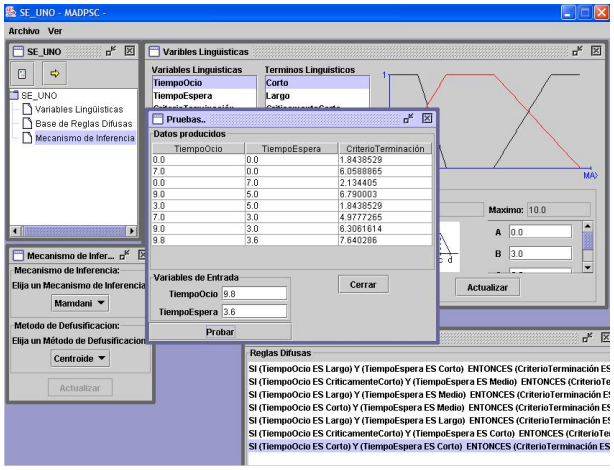

Figure 8 – Bureau de travail du système

4.1 Application d'un système génétique flou en psychologie de l'enfant

Dans le contexte de la psychologie de l'enfant, l'étude du développement des compétences est importante pour les spécialistes. Nous avons travaillé avec l'un d'entre eux en mettant au point diverses stratégies et techniques, parmi lesquelles les jeux de compétences ci-dessous :

- *Malle :* permet de mesurer la capacité visuelle,
- *Labyrinthes :* permet de mesurer l'attention, la concentration et la prise de décision,
- *Puzzles :* permet de mesurer la perception visuelle et l'intégration de l'espace,
- *Analogies :* permet de mesurer la généralisation, l'abstraction, l'analyse et la formation des concepts verbaux.
- *Séries :* permet de mesurer l'attention, l'abstraction, les capacités d'analyse et d'intégration.

Le problème est de faire une telle étude des compétences rapidement et efficacement dans de nombreux groupes d'enfants et de faire une analyse intelligente des données obtenues lors de l'exécution des jeux pour les enfants.
Notre solution consiste en un système multimédia avec des jeux reliés à un système flou pour l'analyse de données intelligente et d'identification des compétences des enfants. Pour la conception de la base de règles floues et les fonctions d'appartenance du système flou, nous avons appliqué notre approche évolutive à l'apprentissage des règles linguistiques floues. Pour chaque jeu de compétence, nous identifions deux variables linguistiques *d'entrée* et une variable de *sortie* définis comme est indiqué dans le tableau 1. Les fonctions d'appartenance sont des fonctions triangulaires et trapézoïdales.

Tableau 1 – Définition des variables linguistiques *d'entrée-sortie*

Nom Variable	*Sorte*	*Ensembles flous*
Hits	Entrée	Peu Suffisamment Tout
Temps	Entrée	Court Normal Long
Aptitude	Sortie	Insuffisante Bonne Excellente

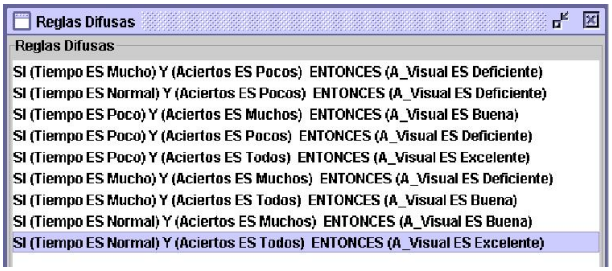

Figure 9 – Base Finale de règles floues SIC.

Le système de programmation génétique a été formé et testé avec deux bases de données de 50 enfants pour cinq jeux de compétences pour un total de 250 dossiers. Le système génère une base de 9 règles linguistiques floues (voir figure 9), après 10 lancers avec 12 générations par lancer et une probabilité de 50 % de mutation. La performance obtenue dans le système flou est comparée à celle d'un expert humain.

Cette comparaison est illustrée figure 10, où la ligne bleue du graphique correspond à l'avis de l'expert et la ligne rouge correspond aux opinions du système flou, avec une erreur moyenne de tous les cas de $0,29$ décrite par la ligne verte.

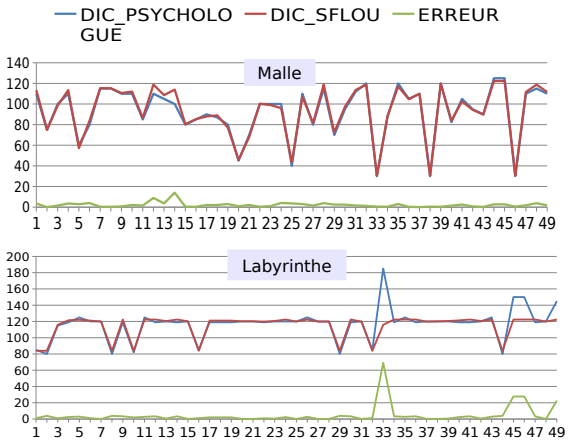

Figure 10 – Performance du système flou.

4.2 Application d'un système génétique flou au contrôle

Cette expérience aborde le problème du contrôle du dosage de floculant et coagulant dans un processus physico-chimique pour le traitement des eaux d'égout.

Tableau 2 – Variables linguistiques *d'entrée-sortie*

Nom Variable	*Sorte*	*Ensembles flous*
Conductivité	Entrée	Basse Moyenne Haute
Total des solides en suspension	Entrée (SST)	Faible Normal Fort
Quantité de coagulant et floculant	Sortie (CCF)	Basse Moyenne Haute

Puisqu'il s'agit d'un système non linéaire, il est difficile de déterminer la quantité optimale à doser pour ces réactifs naturels. Notre solution consiste à modéliser un système de contrôle flou (SCF). Pour la conception de la base de règles floues et l'apprentissage des fonctions d'appartenance du système flou, nous avons appliqué notre approche évolutive. Deux variables linguistiques *d'entrée* et une variable de *sortie* ont été définies comme indiqué tableau 2, et les ensembles flous ont été définis par des fonctions d'appartenance triangulaires et trapézoïdales.

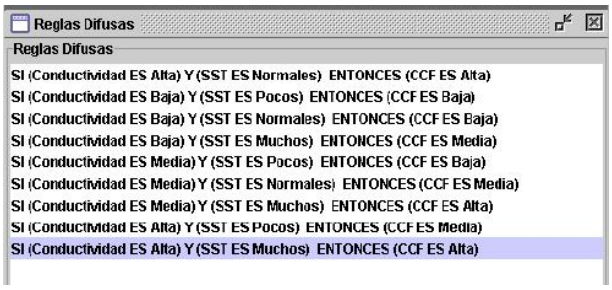

Figure 11 – Base Finale de règles floues SCF.

Le système de programmation génétique a été formé et testé avec deux bases de données de 300 cas différents. Chacune des bases est formée après 12 lancers avec 10 générations par lancer et une probabilité de 50 % de mutation, pour un total de 120 bases de règles floues.

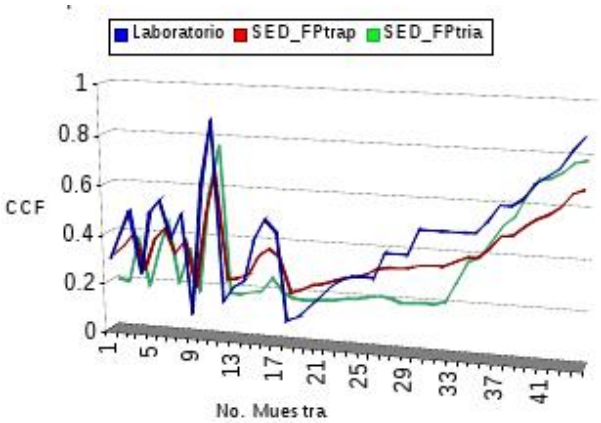

Figure 12 – Performance du SCF.

Cela génère une base de 9 règles linguistiques floues (voir figure 11) avec l'erreur minimum absolu entre 0 et 0,2 avec fonctions d'appartenance trapézoïdales comme indiqué dans les

graphiques des figures 12 et 13 avec une surface de contrôle qui est illustrée par la figure14.

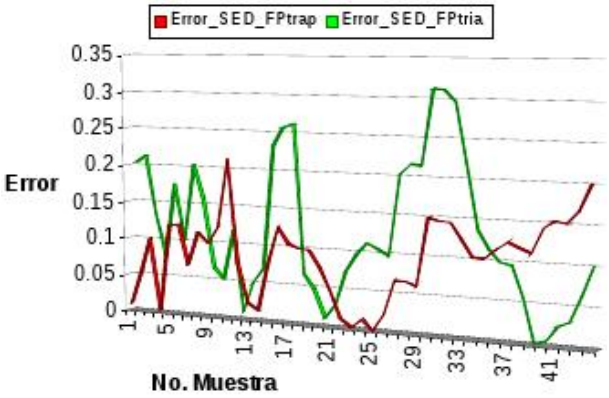

Figure 13 – Erreur du SCF.

Le nombre de lignes bleues représente la quantité dispensée par l'expert et la série de lignes rouges et vertes correspondent aux quantités déterminées par le système génétique flou avec des fonctions d'appartenance trapézoïdales et triangulaires.

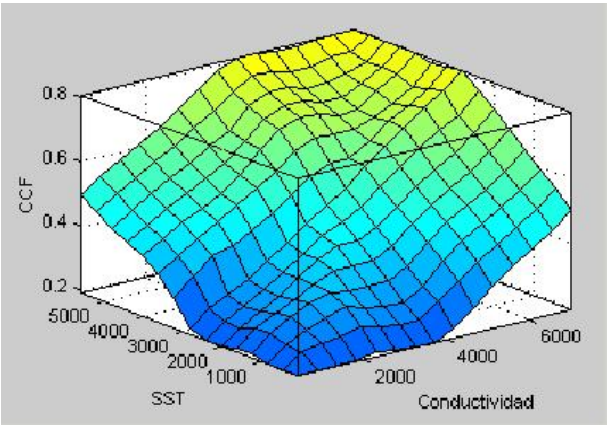

Figure 14 – Surface de contrôle du SCF.

5 Conclusions et travaux futurs

Dans cet article nous présentons une approche pour l'intégration de la programmation génétique et d'un système flou pour l'apprentissage automatique des règles linguistiques floues. Nous expliquons aussi son application dans la modélisation de deux systèmes flous : l'un dans le domaine de la psychologie de l'enfant et l'autre dans la zone de contrôle d'un processus physico-chimique. La technique proposée pour l'apprentissage permet de modéliser des systèmes flous avec un niveau d'expressivité très acceptable, mais doit améliorer sa précision principalement dans le domaine des systèmes de contrôle temps réel.

Références

[1] P. Angelov, D. Filev and N. Kasabov (Eds.). *Evolving Intelligent Systems : Methodology and Applications*. John Willey and Sons, IEEE Press Series on Computational Intelligence, Feb. 2010.

[2] P. Angelov. Evolving Fuzzy Systems. *In Encyclopedia on Complexity and System Science (Bob Meyers Editor-in-Chief)*, 10398 pp., article 194, Springer, June 2009.

[3] J. Alcalá-Fdez, R. Alcalá, M.J. Gacto, F. Herrera. Learning the Membership Function Contexts for Mining Fuzzy Association Rules by Using Genetic Algorithms. *In Fuzzy Sets and Systems*, 160 :7 905-921, 2009.

[4] R. Alcalá, P. Ducange, F. Herrera, B. Lazzerini, F. Marcelloni. A Multi-Objective Evolutionary Approach to Concurrently Learn Rule and Data Bases of Linguistic Fuzzy Rule-Based Systems. *IEEE Transactions on Fuzzy Systems*, 17(5) : 1106-1122, 2009.

[5] F. J. Berlanga, A.J. Rivera, M.J. del Jesus, F. Herrera. GP-COACH : Genetic Programming based learning of COmpact and ACcurate fuzzy rule based classification systems for high dimensional problems. *Information Sciences*, 18(8) : 1183-1200, 2010.

[6] P. P. Bonissone, Y. - O. Chen, K. Goebel, and P. S. Khedkar. Hybrid Soft Computing Systems : Industrial and Commercial Applications. *Proceedings of the IEEE*, 87(9) : 1641-1667, 1999.

[7] B. C. Ceila, H. Lopes, A. Freitas. Genetic Programming for Knowledge Discovery in Chest-Pain Diagnosis. *IEEE in Engineering and Biology* : 38-44, 2000.

[8] O. Cordón, F. Gomide, F. Herrera, F. Hoffmann, L. Magdalena. Ten Years of Genetic Fuzzy Systems : Current Framework and New Trends. *Fuzzy Sets and Systems*, 141(1) : 5-31, 2004.

[9] P. Espejo, S. Ventura, F. Herrera. A Survey on the Application of Genetic Programming to Classification. *IEEE Transactions on Systems, Man, and Cybernetics–Part C : Applications and Reviews*, 40(2) : 121-144, 2010.

[10] A. Fernandez, M.J. del Jesus, F. Herrera. On the 2-Tuples Based Genetic Tuning Performance for Fuzzy Rule Based Classification Systems in Imbalanced Data-Sets. *Information Sciences*, 180(8) : 1268-1290, 2010.

[11] D. Floreano, L. Keller. Evolution of Adaptive Behaviour in Robots by Means of Darwinian Selection. *PLoS Biology*, 8(1) : 1-8, January 2010.

[12] T. Furuhashi. Fusion of Fuzzy/Neuro/Evolutionary Computing for Knowledge Acquisition. *Proceedings of the IEEE*, 89(9) : 1266-1274, September 2001.

[13] M.J. Gacto, R. Alcalá, F. Herrera. Adaptation and Application of Multi-Objective Evolutionary Algorithms for Rule Reduction and Parameter Tuning of Fuzzy Rule-Based Systems. *Soft Computing Springer*, 13(5) : 419-436, 2009.

[14] F. Hoffmann. Evolutionary Algorithms for Fuzzy Control System Design. *Proceedings of the IEEE*, 89(9) : 1318-1333, September 2001.

[15] J.-S. R. Jang, C.-T. Sun and E. Mizutani. *Neuro-Fuzzy and Soft Computing : A Computational Approach to Learning and Machine Intelligence*.Matlab Curriculum Series, Prentice Hall, 1997.

[16] J. Luengo, F. Herrera. Domains of Competence of Fuzzy Rule Based Classification Systems with Data Complexity measures : A case of study using a Fuzzy Hybrid Genetic Based Machine Learning Method. *Fuzzy Sets and Systems*, 161 : 3-19, 2010.

[17] M. Leung, W. Lam, K. Sak, P. Shun, J. Cheng. Discovering Knowledge from Medical Databases Using Evolutionary Algorithms. *IEEE Engineering in Medicine and Biology*,pp 45-55, 2000.

[18] M. Mucientes, J. Alcalá-Fdez, R. Alcalá, J. Casillas. A case study for learning behaviors in mobile robotics by evolutionary fuzzy systems. *Expert Systems With Applications*, 37 : 1471-1493, 2010.

[19] M. Russo. Genetic Fuzzy Learning. *IEEE Transactions on Evolutionary Computation*, 4(2) : 259-273, 2000.

[20] E. Tunstel and M. Jamshidi. On Genetic Programming of fuzzy rule-based systems for intelligent control. *Intelligent Automation and Soft Computing*, 2(3) : 271-284, 1996.

[21] A. Zafra, C. Romero, S. Ventura, E. Herrera-Viedma. Multi-Instance Genetic Programming For Web Index Recommendation. *Expert Systems and Applications*, 36 : 11470-11479, 2009.