| | |
|---|---|
| **Titre:** Title: | Automatisation du processus de construction des structures de données floues |
| **Auteur:** Author: | Sofiane Achiche |
| **Date:** | 2005 |
| **Type:** | Mémoire ou thèse / Dissertation or Thesis |
| **Référence:** Citation: | Achiche, S. (2005). Automatisation du processus de construction des structures de données floues [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/8199/ |

UNIVERSITÉ DE MONTRÉAL

# AUTOMATISATION DU PROCESSUS DE CONSTRUCTION DES STRUCTURES DE DONNÉES FLOUES

SOFIANE ACHICHE
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR (Ph.D.)
(GÉNIE MÉCANIQUE)
FÉVRIER 2005

*Your file Votre référence*
*ISBN: 978-0-494-48880-5*
*Our file Notre référence*
*ISBN: 978-0-494-48880-5*

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée:

# AUTOMATISATION DU PROCESSUS DE CONSTRUCTION DES STRUCTURES DE DONNÉES FLOUES

présentée par: ACHICHE Sofiane

en vue de l'obtention du diplôme de: Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de:

M. PARIS Jean, Ph.D., président

M. BARON Luc, Ph.D., membre et directeur de recherche

M. BALAZINSKI Marek, Ph.D., membre et codirecteur de recherche

M. BENAOUDIA Mokhtar, Ph.D., membre et codirecteur de recherche

M. AUBIN Carl-Éric, Ph.D., membre

M. KAJL Stanislaw, Ph.D., membre

à mes parents, à Kamila et Yda.

# REMERCIEMENTS

Je voudrais avant tout remercier mon directeur de recherche Luc Baron, ainsi que mon codirecteur de recherche Marek Balazinski, pour m'avoir accordé leur confiance totale. Leurs qualités scientifiques et humaines m'ont été d'une précieuse aide.

Je remercie Mokhtar Benaoudia du Centre de Recherche Industrielle du Québec (CRIQ), pour ses conseils et sa grande patience face à mes innombrables questions.

Je remercie également, les distingués membres du jury qui ont accepté de lire et évaluer ma thèse.

Ma reconnaissance va au Centre d'Excellence pour la Valorisation des Copeaux du CRIQ pour son soutien financier.

Merci à tous mes amis, d'ici et d'ailleurs, d'être toujours là pour moi.

Enfin, un grand merci à toute ma famille ; ma mère, mon père, mes soeurs et frères pour leur soutien continuel et pour leur amour inconditionnel.

# RÉSUMÉ

Cette thèse est consacrée à l'automatisation, par le biais d'algorithmes génétiques, du processus de construction de structures de données floues. Le nom plus générique de "base de connaissances floues" est plus souvent utilisé. Il est à noter qu'une structure de données floues est une base de connaissances floues sans son moteur d'inférences. L'objectif principal de ce travail est de démontrer que par l'utilisation d'algorithmes génétiques, il est possible de générer automatiquement une structure de données floues sans avoir besoin d'un expert humain.

Les algorithmes génétiques, développés dans le cadre de cette thèse, suivent deux paradigmes de codage différents, soit : un codage binaire traditionnel et un codage hybride combinant les nombres réels et les entiers (binaires). Les opérations d'évolution sont adaptées à chacun des deux algorithmes. Le codage hybride comprend deux parties distinctes dans son mécanisme de reproduction, soit : un croisement adapté aux nombres réels pour la base de faits et un croisement simple pour la base de règles (partie binaire du codage hybride). Les deux algorithmes d'apprentissage sont testés sur des ensembles de données synthétiques représentant des surfaces 3D. Une étude comparative de leurs comportements respectifs est entreprise et ce en tenant compte de critères de performance différents et variés tels que : la précision, la simplicité et le temps d'apprentissage des bases de connais-

sances floues. Il en est ressorti la supériorité du codage hybride quant à la tâche d'apprentissage des structures de données floues et ce sur tous les critères considérés. Cette partie a donné lieu à une première publication dans une revue scientifique. L'un des problèmes importants rencontrés dans l'apprentissage automatique par algorithmes métaheuristiques (dont font partie les algorithmes génétiques) est la convergence prématurée des solutions. Afin de remédier à cette problématique, des techniques pour augmenter la diversité au sein de la population de solutions sont proposées. Ces techniques utilisent des stratégies de reproduction multiples (utilisant différents mécanismes de croisement) associées à une stratégie de famille nombreuse. Ces approches ont démontré leur supériorité par rapport aux stratégies traditionnelles de reproduction (i.e. application d'un seul mécanisme de croisement). Aussi, une étude sur l'amélioration de l'équilibre entre l'exploitation et l'exploration au cours de l'évolution des solutions est proposée. Celle-ci a permis de prouver l'existence de stades d'évolution dans un algorithme génétique et de mettre en évidence l'influence des niveaux d'exploitation et d'exploration sur ses performances. L'exploration suivie de l'exploitation relaxée et, finalement, de l'exploitation est l'ordre d'évolution préconisé. Une application de l'apprentissage automatique (hybride) à un problème de suivi d'usure d'outils a donné des résultats concluants. Cette partie a donné lieu à deux autres publications. Une Application de l'algorithme génétique hybride, avec méthode évolutive de reproduction, au problème de prédiction de la qualité de la pâte thermomécanique (domaine

des pâtes et papiers) a été entreprise. La qualité de la pâte thermomécanique est définie par sa blancheur ISO. L'apprentissage des structures de données floues se fait en utilisant plusieurs combinaisons de variables d'entrées fournies par le Chip Management System ($CMS^{©}$), appareil permettant de caractériser la qualité des copeaux de bois en aval du procédé de fabrication des pâtes thermomécaniques, par le biais du traitement d'image (caméra RGB) et d'un capteur à infrarouges proches évaluant l'humidité surfacique des dits copeaux. Cette approche est innovatrice du fait de la prédiction de la qualité de la pâte à partir de la qualité des copeaux en utilisant des mesures synthétiques (non prises en laboratoire), ce qui permet une prédiction/contrôle de la qualité de la pâte en ligne. Cette partie a donné lieu à une quatrième et cinquième publication. Enfin, une discussion sur l'ensemble de la recherche menée dans cette thèse est présentée.

# ABSTRACT

This thesis presents the automatic generation of fuzzy data structures. A fuzzy data structure is a fuzzy knowledge base without its inference engine. The generic name of "fuzzy knowledge bases" will be used throughout this thesis. The optimization tool used for the automatic learning is a genetically based algorithm. The first objective of this research is to prove the feasibility of the automatic generation of fuzzy knowledge bases (without the need of a human expert).

The genetic algorithms developed in this thesis follow two coding paradigms: a traditional binary coding and a new real/binary-like coding (hybrid coding). The evolution operators are adapted to each algorithm. In the hybrid coding the reproduction mechanism is made of two distinct parts: a specialized crossovers suited for the factual base (real coded part) and a traditional single point crossover used for the rule base (binary-like coded part). A comparative study on the learning performances of both algorithms, using synthetic data obtained from theoretical 3D surfaces, is done taking into account several performance criteria such as: the precision, the simplicity and the learning time of the genetically-generated fuzzy knowledge bases. From this comparative study, the hybrid coding emerged as the most efficient for most of the performance criteria, which prove the advantage of adapting a genetic algorithm to the optimization problem under study. This part

resulted into the first publication presented in this thesis. One of the most tedious problems encountered in automatic learning using meta-heuristic algorithms (genetic algorithms being a part of the meta-heuristic algorithms family) is the premature convergence. In order to overcome these problems (only the hybrid approach is considered) several methods to improve the diversity within the population of solutions are developed. These methods use multiple reproduction strategies (using different crossover mechanisms) along with a crowded family strategy. These approaches showed their superiority when compared with the conventional reproduction strategies (using a single crossover mechanism through the entire evolution). Furthermore, a study on enhancing the performance of the genetic learning by improving the balance between exploration and exploitation within the individuals is done. This study showed the existence of evolution stages in genetic algorithms and also the influence of the exploitation/exploration levels on genetic learning performance. Exploration at the early stages of the evolution, followed by relaxed exploitation during the evolution stage and exploitation in the last stages is the order that improves the learning performance. Genetic learning on experimental data obtained from a tool wear monitoring application gave very satisfactory results. This part resulted into two more publications. An application of the evolutionary algorithms to the thermomechanical pulp and paper process (TMP) was performed, where the quality of the pulp is defined by the ISO brightness. The learning of the fuzzy knowledge bases is performed using input variables obtained

from a Chip Management System ($CMS^{©}$). The $CMS^{©}$ characterizes the quality of wood chips upfront of the TMP process using sensors such as: an RGB camera and near-infrared sensor. This approach allows an online prediction of the pulp quality, since no laboratory measurements are needed for the prediction. This part resulted into a fourth and fifth publication. Finally, a general discussion followed by a set of recommendations and conclusions close this thesis.

# TABLE DES MATIÈRES

# LISTE DES FIGURES

# LISTE DES NOTATIONS ET DES SYMBOLES

AG : Algorithme génétique.

AGB : Algorithme génétique codé en binaire.

AGCR : Algorithme génétique codé en nombres réels.

AGCRB : Algorithme génétique codé aux nombres réels et binaires.

AOM : Average of maximums.

BF : Base de faits.

BGA : Algorithme génétique codé en binaire BINARY GENETIC ALGORITHM.

$BLX - \alpha$ : Croisement aléatoire BLENDED CROSSOVER $\alpha$.

BR : Base de règles.

COG : Centre de gravité CENTER OF GRAVITY.

CMS : Chip Managment System.

CRI : Règle de composition d'inférence COMPOSITIONAL RULE OF INFERENCE.

CRIQ : Centre de Recherche Industrielle du Québec.

EEB : Equilibre de l'exploitation et de l'exploration EXPLORATION/EXPLOITATION BALANCE STRATEGY.

EFS : Ensemble de faits spécifiques.

ELX : Croisment linéaire EXTENDED LINE CROSSOVER.

FC : Convergence rapide (Fast Convergence).

FDSS : Système flou d'aide à la décision FUZZY DECISION SUPPORT SYSTEM.

FKB : Base de connaissances floue FUZZY KNOWLEDGE BASE.

$G$ : Génotype.

GA : Algorithme génétique GENETIC ALGORITHM.

H : Teinte de la couleur HUE.

IA : Intelligence artificielle

$K$ : Nombre maximum de règles floues.

KB : Base de connaissances KNOWLEDGE BASE.

L : Luminance de la couleur.

LC : Combinaison linéaire de trois croisement.

LD : Manque de diversité LACK OF DIVERSITY.

LT : Temps d'apprentissage LEARNING TIME.

MCOG : Centre de gravité modifé MODIFIED CENTER OF GRAVITY.

MRF : Méthode de raisonnement flou LEARNING TIME.

MI : Moteur d'inférences.

MISO : Entrées multiples sortie simple MULTIPLE INPUT SINGLE OUTPUT.

MIMO : Entrées multiples sorties multiples MULTIPLE INPUT MULTIPLE OUTPUT.

MOM : Moyenne des maximums MEAN OF MAXIMA.

$N$ : Nombre de prémisses ou nombre d'entrées.

NAX : Croisement non-uniform NON-UNIFORM ARITHMETICAL CROSSOVER.

$P$ : Taille de la population.

PTM : Pâte thermomécanique.

$R$ : Agrégation des règles floues.

RBLGA : Algorithme génétique hybride REAL/BINARY LIKE CODED GENETIC ALGORITHM.

RBCGA : Algorithme génétique hybride REAL/BINARY LIKE CODED GENETIC ALGORITHM.

RG : Génotype aux nombres réels.

RGB : Red Green Blue.

RMS : Racine de la moyenne carrée ROOT-MEAN-SQUARE.

RP : Atteindre un plateau REACHING A PLATEAU.

S : Saturation de la couleur.

SA : Application simultanée de trois croisements.

SAD : Système d'aide à la décision.

SADF : Système d'aide à la décision flou.

*also* : Connection de phase SENTENCE CONNECTIVE.

$b$ : Nombre de bits.

$e$ : Bit activé = 1 / désactivé = 0.

$g$ : Génotype d'un individu.

$n$ : Nombre de sous-ensembles flous.

$p$ : Phénotype d'un individu.

$\Delta_{RMS}$ : Erreur rms.

$\phi$ : Indice de performance.

$\omega$ : Pondération.

$\wedge$ : Opérateur minimum.

$\vee$ : Opérateur maximum.

$\circ$ : Opérateur d'inférence composée.

$*$ : Opérateur produit.

$\sum$ : Opérateur somme.

$*_t(\cdot)$ : Opérateur norme-t de $(\cdot)$.

# LISTE DES TABLEAUX

# LISTE DES ANNEXES

# INTRODUCTION

De nos jours, les problèmes technologiques sont devenus très complexes (un grand nombre de variables influentes). La gestion de l'information et des connaissances doit donc se faire de façon globale, de là le besoin d'outils d'aide à la décision. L'intelligence artificielle peut servir à développer ce type de systèmes et ainsi pallier à cette globalisation de l'information. Les systèmes d'aide à la décision (SAD) utilisant l'intelligence artificielle sont une application particulière des systèmes à base de connaissances. Ils sont capables de raisonnements logiques sur différentes entités et les représentent par des symboles. Ces systèmes servent essentiellement à la reproduction du raisonnement d'experts humains dans un domaine donné.

Une des principales caractéristiques du raisonnement humain est la possibilité de se baser sur des données imprécises ou incomplètes. Ce type de raisonnement est impossible en utilisant la logique au sens de Boole, mais tout à fait faisable grâce à la logique floue. La logique au sens propre du mot (Booléenne) est une conception des mécanismes de la pensée qui ne devrait jamais être floue. Néanmoins, beaucoup de concepts réels ne se basent pas sur un choix radical entre une proposition et sa négation (blanc ou noir, positif ou négatif...), mais sur toute une nuance de propositions qui pourraient exprimer des états intermédiaires. En logique floue, le résultat d'une opération s'exprime comme une probabilité plutôt que comme une

certitude et peut, outre les valeurs vrai et faux, être probablement vrai, peut-être vrai, peut-être faux ou probablement faux, ce qui s'approche de la façon dont les humains raisonnent et distinguent entre le possible, le probable et le vraisemblable. La logique booléenne est associée à la théorie booléenne des ensembles ; par contre, la logique floue est associée de la même manière à la théorie des sous-ensembles flous [69].

## 0.1 Notions de base sur la logique floue

Afin de mieux comprendre les sections qui suivent, nous allons définir brièvement les concepts de base de la théorie des sous-ensembles flous. Il est à noter que, dans le texte, les termes sous-ensembles flous et ensembles flous expriment la même chose.

### 0.1.1 Définition d'un sous-ensemble flou

Un ensemble classique possède des éléments qui satisfont un ensemble de propriétés précises. Plus formellement, un sous-ensemble $A$ d'un ensemble de références $X$ peut être décrit à partir de sa fonction caractéristique $\chi_A : X \rightarrow \{0, 1\}$ de la manière suivante :

$$\chi_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{sinon} \end{cases}, \tag{1}$$

Par exemple, le sous-ensemble $A$ des gens dont la taille varie entre 1m65 et 1m80 a pour fonction caractéristique :

$$\chi_A(x) = \begin{cases} 1 & \text{si } 1\text{m}65 \leq x \leq 1\text{m}80 \\ 0 & \text{sinon} \end{cases}, \tag{2}$$

Considérons un ensemble $B$ des tailles proches des 1m75, la propriété "proche" n'est pas précise car $B$ ne peut être caractérisé par une fonction caractéristique qui scinderait en deux les tailles : celles qui avoisinent les 1m75 et celles qui ne les avoisinent pas. On est alors amené à introduire une généralisation de cette fonction caractéristique en une fonction d'appartenance afin de considérer les tailles qui ne sont pas trop éloignées de 1m75 sans être vraiment proche de 1m75.

Ainsi, une fonction d'appartenance permet de mettre en évidence les nuances d'appartenance pour les éléments de l'ensemble de référence $X$ et permet de définir un sous-ensemble flou de $X$. De là découle la définition suivante :

Un sous-ensemble flou $F$ de $X$ est défini par une fonction d'appartenance $\mu_F$ qui associe à tout élément $x$ de $X$ une valeur réelle $\mu_f(x)$ dans l'intervalle $[0,\ 1]$. Ainsi définie, toute fonction à valeurs dans l'intervalle $[0,\ 1]$ est un sous-ensemble flou. Néanmoins, toute fonction de ce type ne peut être interprétée conceptuellement comme étant un sous-ensemble flou que lorsqu'elle coïncide avec une description sémantique plausible [17].

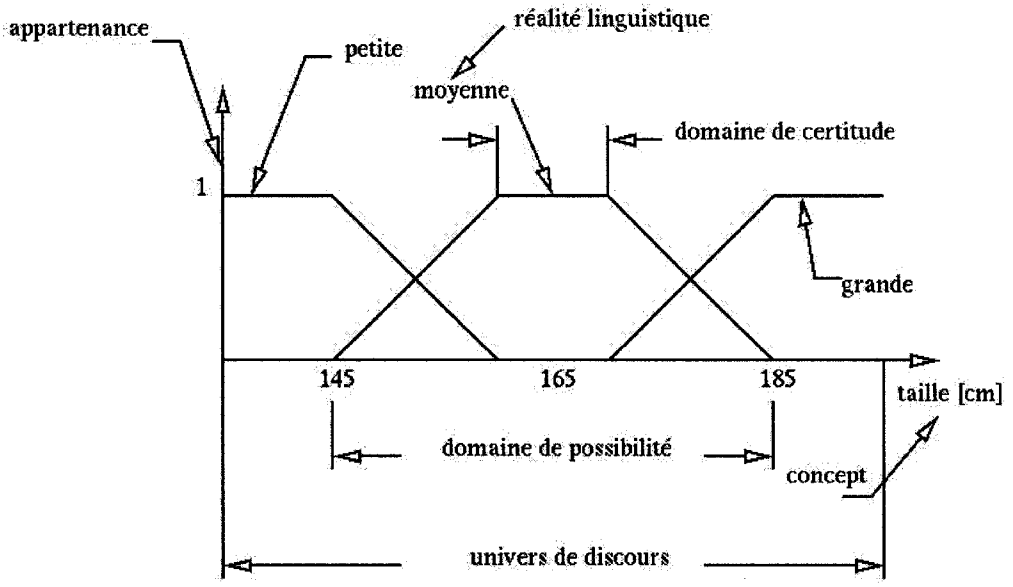


FIG. 1 Termes et concepts en logique floue

La figure 1 montre des sous-ensembles flous (de forme trapézoïdale) ainsi que les termes et concepts les plus usuels.

**Concept :** domaine d'appartenance des différents faits. Par exemple, la taille.

**Réalité linguistique :** expression linguistique d'un fait. Par exemple, pour le concept de la taille, on a des réalités linguistiques : petite, moyenne et grande.

**Évaluation :** évaluation faite par l'observation et le jugement d'un cas particulier. Par exemple, on regarde une personne et on évalue sa taille à environ 1m70.

**Domaine de discours :** champ de définition d'un concept. Dans notre exemple, la taille est définie pour une personne d'âge adulte, de sexe masculin, vivant en Amérique du nord.

**Possibilité d'appartenance :** appelé aussi probabilité d'appartenance, c'est le niveau [0, 1] de l'adhésion à un concept ou à l'évaluation dans le domaine de discours.

**Fonction d'appartenance :** un sous-ensemble flou $A$ d'un ensemble $X$, appelé référentiel ou univers de discours.

### 0.1.2 Raisonnement approximatif

Le raisonnement approximatif c'est tout mécanisme capable d'utiliser et de prendre en compte des connaissances imprécises, floues ou incertaines afin de produire de nouvelles connaissances tel que le raisonnement humain est capable de le faire. La logique floue se base sur la définition de la proposition floue.

**Proposition floue élémentaire :** elle est définie à partir d'une variable linguistique. Par exemple "Toto est petit" ($S$ est $A$). Il s'agit alors de définir le degré de vérité de cette proposition et elle est donnée par la fonction d'appartenance $\mu_A$ de $A$, qui peut être exprimée par :

$$\mu_A : X_S \rightarrow [0,\ 1], \tag{3}$$

cela représente le degré avec lequel chaque valeur de $X_S$ est susceptible de confirmer la proposition.

**Proposition floue générale :** une proposition floue générale est une composition de plusieurs propositions floues élémentaires. Par exemple, "Toto est petit" et "Toto est jeune".

### 0.1.3 Règles floues

Une règle floue est une implication entre deux propositions, un lien qui les unit. Si l'on considère les deux propositions déjà citées, nous pourrions former une règle floue exprimant "Si Toto est jeune" alors "Son salaire est bas", la deuxième partie représentant la conclusion et la première partie la prémisse.

La valeur de vérité qui doit être associée à la règle floue est donnée par l'agrégation des valeurs de vérité de la prémisse et de la conclusion par la fonction associée à l'implication.

### 0.1.4 Inférence floue

Le problème de représentation en logique floue consiste à passer d'une règle floue, qui est un objet linguistique à une relation floue, qui est un objet mathématique.

Zadeh a étendu la notion du *modus ponens* de la logique classique au contexte flou soit le *modus ponens généralisé* :

- modus ponens généralisé : on connaît les deux prémisses.

1. si $x$ est $A'$ avec la fonction d'appartenance $\mu_{A'}$,

   et

2. si $x$ est $A_{\mu_A}$ alors $y$ est $B_{\mu_B}$.

– On peut déduire la conséquence $y$ est $B'$ avec la fonction d'appartenance $\mu_{B'}$.

$A$, $B$, $A'$ et $B'$ sont des sous-ensembles flous.

La fonction d'appartenance de $B'$ est calculée comme une combinaison de $\mu_A$ et $\mu_{A\Rightarrow B}$ :

$$\mu_{B'}(y) = sup\ T_{x\in X}[\mu_{A'}, \mu_{A\Rightarrow B}(x,y)], \tag{4}$$

pour une $t-norm\ T$ appelée l'opérateur *modus ponens généralisé.* Les deux principaux opérateurs *modus ponens généralisés* sont :

1. Opérateur Mamdami (minimum) : $\mu_{B'}(y) = max_x[\mu_{A'}\ min\ \mu_{A\Rightarrow B}(x,y)]$.
2. Opérateur Larsen (produit) : $\mu_{B'}(y) = max_x[\mu_{A'} \bullet \mu_{A\Rightarrow B}(x,y)]$,

Dans cette thèse c'est l'opérateur Larsen qui est utilisé.

## 0.2 Problématique et motivation de la recherche

Les SAD basés sur la logique floue (SADF) sont souvent utilisés en ingénierie et ils ont prouvé leur efficacité dans des domaines très variés. Cette efficacité est due au fait que les entrées et sorties de ces systèmes sont souvent des entités réelles (ou transformées en telles) assemblées par des fonctions non linéaires, ce qui correspond aux problèmes rencontrés dans la réalité. Les SADF sont souvent très bien adaptés

aux problèmes où il y a une multitude d'informations à gérer, un environnement peu précis ou de l'information vague.

Les domaines de prédilection des SADF sont :

- systèmes à comportement non linéaires ;
- systèmes avec perturbations non prévisibles ;
- systèmes dépendant de l'expertise d'un opérateur humain.

Les SADF représentent une alternative pour les applications où les stratégies classiques de contrôle ou de prise de décision sont souvent peu ou pas efficaces. Dans la plupart des cas, les SADF s'appliquent dans les deux cas suivants :

- le besoin de l'expertise d'un opérateur humain pour construire une base de connaissances ;
- présence d'une grande non linéarité. Dans lequel cas il est souvent difficile (jusqu'à l'impossibilité) de développer un modèle mathématique exprimant le comportement du problème en question.

Les SADF traitent l'information de façon ordonnée comme indiqué à la figure 2. Les données collectées (évaluations) passent par :

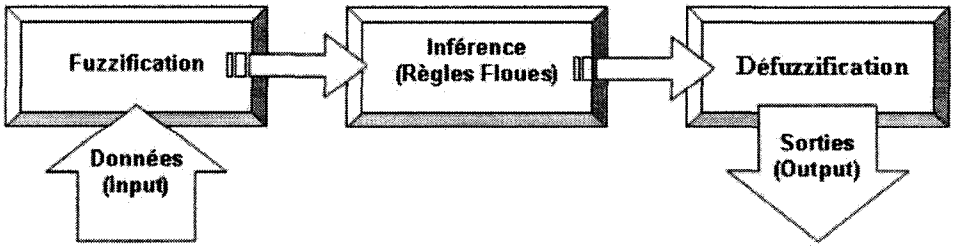


FIG. 2 Processus de traitement de l'information dans un SADF

- la fuzzification : catégorisation de l'information en sous-ensembles flous ;
- l'inférence (mise en marche des règles floues) : interconnexions entre les différents sous-ensembles flous des différentes prémisses, pointant vers les différentes conclusions ;
- la déffuzification : obtention de valeurs numériques.

Ces trois étapes définissent donc le fonctionnement d'un SADF. La modélisation par logique floue d'un quelconque procédé passe par la construction de bases de connaissances qui permettront le fonctionnement de SADF.

Pour construire une base de connaissances, il faut passer par le processus suivant :

- définir les prémisses : il y a autant de prémisses que de variables d'entrée sélectionnées ;
- définir les conclusions : il y a autant de conclusions que de variables de sorties ;
- définir le nombre de sous-ensembles flous : sur chacune des prémisses et conclusions, il faut définir le nombre adéquat de sous-ensembles flous susceptible de bien représenter le procédé ;
- définir le type de sous-ensembles flous à utiliser : triangulaires, trapézoïdaux, sigmoïdes, etc. (en se basant sur les connaissances disponibles) ;
- définir la répartition des sous-ensembles flous : leurs positions sur les prémisses et sur les conclusions ;
- définir les règles floues : qui représentent la relation entre les différents sous-

ensembles flous sur les prémisses et sur les conclusions ;

- choix de la méthode de raisonnement flou : une MRF est une procédure basée sur un système d'inférence, qui donne des conclusions à partir d'un canevas fait par la base de connaissances ;
- choix de la méthode de déffuzzification : transformation des données (réponses) floues en valeurs exactes.

La mise en oeuvre de ces bases de connaissances est une tâche très ardue car elle doit être faite manuellement par un expert en la matière. Cette génération manuelle de base de connaissances pose les principales difficultés suivantes :

- construction manuelle de bases de connaissances très coûteuse en terme de temps ;
- une grande affluence de données et de paramètres complique considérablement la modélisation des connaissances et oblige l'expert à connaître et la logique floue et le domaine de son application.

Il est donc essentiel de développer et d'optimiser des techniques pour automatiser la représentation des connaissances. Le design de ce type de systèmes induit l'implémentation d'une méthode de raisonnement flou, supportée par des algorithmes d'apprentissage et de recherche de solutions. Dans ce travail, nous nous concentrerons sur les SADs qui utilisent les bases de règles floues comme outil de représentation des connaissances acquises (SADF).

Le chapitre 1 de cette thèse est une étude bibliographique mettant en valeur les travaux précédents dans le domaine de la génération automatique de bases de connaissances floues. L'objectif de cette étude est de situer le travail dans le cadre des développements les plus récents, notamment sur les algorithmes utilisés et les aspects de génération automatique de la base de connaissances. Le chapitre 2 présente l'organisation générale de la thèse par articles. Le chapitre 3 présente deux algorithmes génétiques de génération automatique de bases de connaissances, les deux algorithmes sont développés selon deux paradigmes différents (codage binaire et hybride). Le chapitre 4 présente plusieurs stratégies évolutives de reproduction, appliquées à l'algorithme génétique hybride afin d'éviter le problème de convergence prématurée, fait courant dans l'optimisation stochastique, une application au domaine de l'usinage y est présentée. Le chapitre 5 présente l'application principale de l'outil d'apprentissage développé, qui est le domaine des pâtes thermomécaniques et plus spécifiquement la prédiction de la blancheur de la pâte à partir de données captées en ligne. Enfin, une discussion générale sur l'ensemble de la recherche entreprise dans le cadre de ce mémoire, suivie de recommandations et de conclusions mémoire closent la thèse.

# CHAPITRE 1

# REVUE BIBLIOGRAPHIQUE

## 1.1 Introduction

La définition de l'intelligence a fait l'objet des pensées d'illustres philosophes et mathématiciens tels que : Aristote, Platon, Copernic et Galilé. Ils ont tous essayé d'expliquer le principe du raisonnement humain. Néanmoins, la première clé qui a permis de synthétiser l'intelligence est apparue grâce au philosophe anglais Thomas Hobbes qui, dans les années 1650, expliquait le raisonnement humain comme étant une suite d'opérations symboliques qui pouvait être modélisé mathématiquement. De ce concept dérive facilement la conclusion qu'une machine capable de manipuler des concepts mathématiques serait donc capable de raisonnement propre à l'être humain. Le terme "intelligence artificielle (IA)" n'est apparu qu'en 1956 proposé par John McCarthy [8]. Dans les années 1800, le mathématicien anglais George Boole a formulé des lois de pensées qui utilisaient des règles de logique. Boole définissait que toute opération logique ne pouvait prendre fait que dans deux types de réponses : "vrai ou faux", "oui ou non" et "tout ou rien (i.e. 1 ou 0)". En 1938, Claude Shannon a démontré l'intérêt d'utiliser de la logique Booléenne sur des machines (allumer et éteindre un circuit électrique) [57], la première application industrielle a suivi en 1946 à l'Université de Pennsylvanie, ce qui représentait la

première génération d'ordinateurs.

### 1.1.1 Domaines d'application de l'intelligence artificielle

L'utilisation des machines comme des instruments de reproduction de l'intelligence humaine a amené le développement de plusieurs branches de l'IA, soit :

1. traitement du langage naturel (indexation automatique de texte, analyse de style et de grammaire de textes, traduction automatisée, etc.) ;
2. vision artificielle (traitement d'images, capteurs de mouvement, etc.) ;
3. robotique (contrôle de robots, manipulateurs mobiles indépendants, etc.) ;
4. recherche de solution et planification (traitement de données) ;
5. apprentissage (apprentissage par l'exemple, apprentissage par la tâche, etc.) ;
6. systèmes experts ou systèmes à bases de connaissances (traitement de connaissances plutôt que de données numériques).

Dans cette thèse, nous considérons la partie de l'IA traitant des systèmes à bases de connaissances (systèmes experts).

## 1.2 Systèmes à base de connaissances

C'est dans les années 1960 jusqu'au début des années 1970 que cette branche de l'IA a émergé. Les systèmes à base de connaissances regroupent une famille de logiciels ayant en commun un structure caractérisée par une nette séparation entre

la *connaissance*, propre au domaine d'application, et les procédures d'utilisation de cette connaissance, chargées du *contrôle/prédiction* [7]. Un système expert capture l'expertise humaine et la stocke dans la **base de connaissances**. Un programme, appelé **moteur d'inférences**, a pour tâche de construire un lien entre les connaissances pour traiter un sujet particulier soumis par l'utilisateur.

Une base de connaissances regroupe généralement [22] :

- Les connaissances factuelles qui permettent de décrire le domaine d'application, nous noterons cette partie **BF**, i.e., base de faits.
- Les connaissances déductives qui permettent par exemple d'obtenir de nouvelles connaissances. Elles sont généralement représentées sous la forme de règles de production, nous noterons cette partie **BR**, i.e., base de règles.

L'union de la BF et de la BR forme une base de connaissances notée **BC** (voir équation 1.1).

$$BC = BF \cup BR. \tag{1.1}$$

La BC et le moteur d'inférences, noté **MI**, permettent de résoudre des problèmes (réponses) à partir d'observations appelées ensemble de faits spécifiques, noté **EFS**. La figure 1.1 illustre la structure d'un système à base de connaissances.

Un système à BC, utilisant des sous-ensembles flous pour classifier et/ou carac-

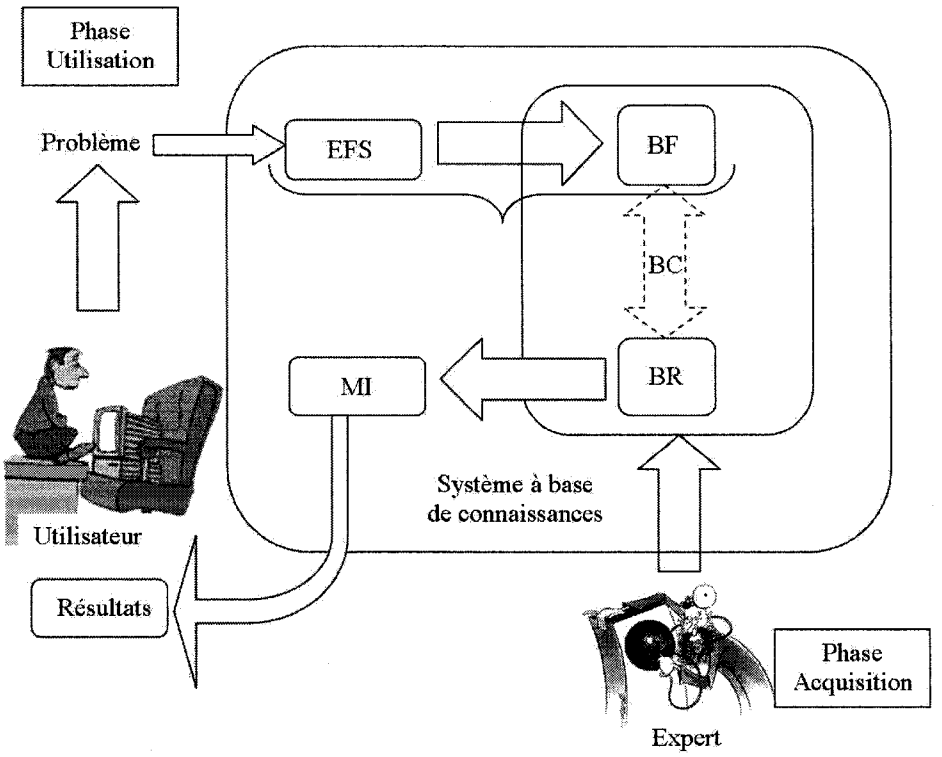


FIG. 1.1 Représentation schématique d'un système à base de connaissances (BC) ; BF - base de faits, BR - base de règles, MI - Moteur d'inférence, EFS - ensemble de faits spécifiques.

tériser l'information pour ses entrées-sorties (définition de la BF), des règles floues pour ses implications (contenu de la BR) et un moteur d'inférences flou (MI basé sur la logique floue) est appelé un "système expert flou", ou bien un "système à base de connaissances floues", ou sinon un "Système d'aide à la décision flou" noté **SADF**. Les SADF ont fait preuve d'efficacité dans plusieurs domaines différents, ce qui la rend une bonne option d'utilisation dans des systèmes d'aide à la décision. En fabrication, on peut citer l'exemple d'une machine à électro-érosion développée par Mitsubishi et dont le contrôle de l'avance est assuré par contrôleur "flou" et

le contrôle de la force de coupe par un "contrôleur neuro-flou" [9] et de choix de condition de coupe [10].

Les applications sont devenues de plus en plus nombreuses comme la prédiction de maintenance préventive [11], allocation de tolérances [28], l'application au contrôle des fours industriels [50], l'aide à la planification de chirurgies de scoliose [45]...etc.

## 1.3 Génération automatique de bases de connaissances floues

La génération de BC pour un SADF présente plusieurs difficultés du fait que la BC dépend essentiellement de l'application étudiée, ce qui rend sa qualité intrinsèquement reliée à son propre contenu. Aussi, il y a le besoin de remplacer l'expert humain qui construit généralement les BC manuellement utilisant le principe de l'essai erreur, à partir de données numériques ou bien sa propre expertise du domaine à modéliser. Une tâche souvent longue, ardue et qui met un biais important sur la BC, dans le sens que deux experts construiraient très probablement des BC différentes à partir des mêmes faits ou des mêmes données. Un expert humain présente d'autres inconvénients dont [8] :

- il est enclin à la fatigue sous une charge mentale et physique importante ;
- il peut oublier des détails cruciaux ;
- il peut être inconsistant dans ses décisions dépendant des jours ;
- il a une mémoire active limitée ;

- il est incapable d'analyser vite un grand volume de données ;
- il est lent à retrouver des détails dans sa mémoire ;
- il est toujours sujet au biais ;
- il peut cacher, mentir ou même mourir.

Ces raisons rendent donc le développement d'un outil de génération automatique de BC pour les SADF une nécessité. Dans cette thèse, nous nous concentrons sur la génération automatique de BC pour les SADF à partir de données numériques. Plusieurs travaux de recherche ont été faits sur l'apprentissage des BC à partir de données numériques. Ils se scindent en trois parties principales :

**1. Génération automatique de la BR seulement**

Les travaux de recherche sur la génération automatique des BC se sont, en grande partie, basées sur l'aspect de la génération automatique de la base de règles floues (BR) et cela en utilisant une répartition et un nombre de sous-ensembles flous prédéfinis (BF prédéfinie). Une méthodologie d'amélioration des règles linguistiques floues à partir de données numériques est présentée dans [20], des méthodes d'apprentissage de règles de classification ont été proposées en [26,63,68] et des apprentissages à partir d'observations expérimentales sont proposés en [32,37,39]. Tous ces travaux de recherche traitent une seule partie du problème d'optimisation qui nous est posé. Aussi, cette approche (i.e. génération de la BR uniquement) rend la qualité de la BR dépendante de celle de la BF pré-établie. Cette dépendance est un grand désavantage, vu que des travaux ont prouvé que la performance d'une BC

est plus sensible au choix de sa BF qu'à la composition même de la BR[18,24,71].

## 2. Génération automatique de la BF

La génération manuelle de la BF (des sous-ensembles flous) est principalement basée sur l'intuition comme proposé par Zadeh [70]. Plusieurs chercheurs se sont penchés sur l'automatisation de la génération de la BF, vu l'importance de celle-ci dans la détermination de la qualité d'une BC. Une méthode basée sur l'inférence par approximation de formes, utilisant les connaissances en géométrie de base, est proposée en [52]. La préférence relative, se basant par exemple sur des sondages, des préférence individuelles, des préférences d'experts et autres sont proposées en [55], d'autres ont fait usage de plusieurs méthodes statistiques pour définir les sous-ensembles flous [27]. Dans les dernières années, un grand nombre de travaux ont été faits sur l'utilisation d'algorithmes d'apprentissage, comme les réseaux de neurones [34] et les algorithmes génétiques [41,43]. Toutes ces approches ont en commun le fait d'utiliser une BR qui leur est inextricablement associée, un désavantage que nous voulons éviter.

## 3. Génération automatique de la BF et de la BR

En général, la méthode la plus simple pour arriver à une bonne BC est une optimisation préliminaire de la BF, une fois la BR créée [18,23,35]. Une approche qui

nuit à une coopération complète entre les différents aspects du problème à optimiser. Néanmoins, tout le domaine de la BC est couvert cette fois-ci. Il est à noter, par contre, que le nombre de sous-ensembles flous dans la BF doit rester inchangé vu que le nombre de règles floues est fixe, ce qui représente une contrainte peu intéressante. D'autres travaux se sont intéressés à la méthode inverse, soit optimiser la BF *a priori* plutôt qu'*a posteriori* [24,48,64], les résultats dégagés restent similaires et gardent les mêmes désavantages.

En 2000 – 2002, une approche pour l'optimisation complète de la BC utilisant des algorithmes génétiques (codés en binaire) [4,12,16] a été proposée. L'approche consiste à lancer l'apprentissage de la BC en lui allouant une complexité maximale, i.e. nombre maximum possible de sous-ensembles flous sur les prémisses et les conclusions,tout en permettant au nombre de règles floues d'être inférieur au maximum obtenu par les combinaisons d'interconnexions entre les sous-ensembles flous. Néanmoins, l'optimisation reste évolutive vers une simplification de la BC, en réduisant le nombre d'informations requises (règles, sous-ensembles flous, etc.) au plus bas niveau possible, sans pour autant trop perdre en précision du modèle flou créé, grâce à des mécanismes spécifiques adaptés au problème en étude. Cette approche minimaliste est celle préconisée dans les travaux présentés dans cette thèse.

### 1.3.1 Algorithmes d'apprentissage de la BC

La génération de BC pour les SADF peut être catégorisée comme un problème d'optimisation qui peut être multi-objectifs et multi-variables. Multi-variables du fait de la présence de plusieurs entités à prendre en considération, telles que : le nombre de sous-ensembles flous, les règles floues, la forme des sous-ensembles flous, le moteur d'inférences, etc. Multi-objectifs, à savoir qu'il peut y avoir plusieurs objectifs à prendre en compte comme la capacité de la BC à coller aux données d'apprentissage (précision), la simplicité de la BR (nombre de règles floues), la simplicité de la BF (nombre de sous-ensembles flous sur les prémisses), etc.

De par le nombre important de paramètres à optimiser dans une base de connaissances et la possibilité de satisfaire à plusieurs objectifs en même temps, nous pouvons classer la tâche de génération automatique de BC dans la catégorie de problèmes d'optimisation de grande taille pouvant être uni-objectifs ou multi-objectifs. Dans la littérature, une attention particulière a été portée sur les problèmes à deux critères tels que le "branch and bound" (branches et zonage) [56,62,65] et la programmation dynamique [19]. Ce type de méthode peut être efficace pour les problèmes de petite taille. Néanmoins, la génération de BC présente les difficultés simultanées des problèmes de complexité non polynomiale complets (NP-Complet) et la multi-objectivité, ce qui rend les procédures exactes très peu efficaces. Les

méthodes heuristiques sont efficaces et nécessaires pour résoudre des problèmes de grande taille et/ou multi-objectifs. Les méthodes heuristiques peuvent être divisées en deux grandes familles :

- algorithmes spécifiques à un seul problème : systèmes experts spécifiques, utilisant les connaissances du domaine [30] ;
- algorithmes généraux indépendants du domaine : appelés les métaheuristiques.

Les métaheuristiques comportent la famille des algorithmes stochastiques de type ***Monte-Carlo***. Cette famille comprend essentiellement le recuit simulé, les réseaux de neurones et les algorithmes génétiques, des algorithmes basés sur la reproduction de phénomènes naturels. Les points communs de ces algorithmes sont [67] :

- se sont des algorithmes approximatifs (heuristiques), ils ne garantissent pas l'obtention d'une solution optimale ;
- ils sont aveugles, il ne savent pas s'ils ont atteint une solution optimale ou proche de l'optimum, il faut donc leur dire de s'arrêter par le biais d'un critère d'arrêt ;
- ils ont des propriétés d'escalades (hill climbing), ce qui leur permet de s'adapter à des changements de direction du problème ;
- ils sont assez généraux pour être facilement implémantés sur plusieurs différents problèmes d'ingénierie ;
- sous certaines conditions, ils peuvent atteindre asymptotiquement une solution optimale.

Il serait vain d'essayer de montrer la supériorité d'un algorithme métaheuristique par rapport à l'autre, il a même été prouvé que la performance moyenne de tous ces types d'algorithmes (métaheuristiques) est très comparable si l'on ne considère pas la connaissance spécifique du domaine et aussi la qualité d'implémentation de l'algorithme [66]. Néanmoins, l'aspect évolutif des algorithmes génétiques (AG), leur capacité d'optimiser plusieurs solutions à la fois (population de solutions), la possibilité d'en personnaliser les mécanismes d'évolution, leur capacité à négocier des contraintes multiples et complexes (parfois même, en partie, contradictoires), ainsi que leur robustesse, les rendent très attrayants pour l'apprentissage automatique des BC pour les SADF.

#### 1.3.1.1 Algorithmes génétiques : choix de codage

Plusieurs travaux ont utilisé des AG pour construire des BC [4, 26, 32, 64]. Chaque AG a été utilisé dans le but de l'apprentissage des différentes parties d'une base de connaissances selon les différentes approches déjà citées, soit :

- apprentissage et optimisation de la BR en utilisant une BF prédéfinie ;
- apprentissage et optimisation de la BF (des sous-ensembles flous en forme, taille et nombre) ;
- apprentissage de la BF et de la BR simultanément.

Les représentations en codage binaire (algorithmes génétiques codés en binaire : AGB) ont dominé la plupart des travaux utilisant les AG, du fait de leur efficacité

[31] et de la simplicité de leur implémentation numérique. Par contre, les bonnes propriétés des AG ne découlent pas de l'utilisation de génotypes binaires—vecteurs binaires—[51] mais de leur aspect évolutif, d'où la redirection des recherches vers une représentation non binaire des AG. L'alternative est "les algorithmes génétiques codés aux nombres réels" (AGCR). Les AGCR ont été plus récemment utilisés dans plusieurs applications d'optimisations numériques [35, 36, 44, 61]. L'avènement et l'utilisation des AGCR sont venus pallier à l'un des principaux défauts des AG codés aux nombres binaires, soit la basse résolution des solutions. De plus, la plupart des problèmes d'optimisation se passent dans des espaces réels (des espaces continus) auxquels les AGCR sont mieux adaptés. Dans cette thèse, les AGCR sont préférés aux AGB, néanmoins les deux approches ont été développées par l'auteur pour des raisons de comparaison et de combinaison.

### 1.3.2 Problème de réduction de la BR

La complexité croissante des problèmes technologiques a comme conséquence la démultiplication des paramètres influents. Modéliser ce genre de problème à grand nombre de variables demande des ressources informatiques considérables. Aussi, il est possible que certaines variables soient redondantes (linéairement dépendantes) et d'autres peuvent perturber le modèle. Un AG d'optimisation (recherche de solutions) sera d'autant plus efficace si le nombre de variables à optimiser est moindre.

Les BC subissent une explosion combinatoire de leur BR (nombre de rèlges floues) lorsque l'on doit traiter des problèmes de grande dimension. Il y a deux façons principales de pallier à ce problème (de façon individuelle ou combinée) :

1. réduire l'espace de recherche des BR de la BC en réduisant le nombre de variables d'entrée du SADF ;
2. minimiser le nombre de règles : certaines règles peuvent être inutiles ou même répétées, d'autres nuisent au système car elles peuvent être en contradiction avec un groupe de règles floues (mauvaise coopération), d'autres règles peuvent être aberrantes dans le sens qu'elles couvrent des domaines physiquement/mathématiquement impossibles.

La réduction du nombre de règles floues se fait soit en combinant des règles ou bien en sélectionnant un sous-ensemble de règles floues, tout en maintenant le niveau de performance de la BC ou bien même l'améliorer (cas de l'élimination de règles non-coopératives). Plusieurs approches pour réduire la taille de la BR ont été étudiées, entre autres, l'utilisation des réseaux de neurones [21], des techniques de *clustering* [33] , des méthodes de transformations orthogonales [59] ainsi que des mesures de similarités [58]. Toutes ces techniques font partie du *Data-mining* (analyse de données), un point qui est actuellement en étude par l'auteur et d'autres collaborateurs, et ce spécialement dans le domaine du clustering [49], mais qui ne sera pas abordé dans cette thèse vu que les avancements sont encore mineurs. Aussi,

des AG ont été utilisés pour la réduction du nombre de règles floues en choisissant un sous-ensemble parmi une base déterminée [38,54]. Parallèlement, des travaux de l'auteur [4,12,16] ont inclus la réduction des règles floues dans les mécanismes de croisement de l'AG, approche utilisée dans cette thèse.

#### 1.3.2.1 Superposition des sous-ensembles flous sur les prémisses d'entrées de la BC

La superposition des sous-ensembles flous sur les prémisses d'entrées est une condition utilisée dans tous les travaux de l'auteur concernant la génération automatique de BC. Une approche qui dans un premier lieu, était destinée à réduire le nombre de paramètres à coder et aussi permettre une représentation complète de l'espace étudié (bonne interpolation). Néanmoins, cette condition fait que les BC obtenues sont plus compactes (i.e. deux intersections, au plus, entre deux sous-ensembles flous), ce qui réduit le nombre de règles floues nécessaires et donc la taille de la BR. Une preuve formelle de l'apport de ce genre de choix, quant à la qualité de la BC, n'a été rapporté que très récemment [42,53].

## 1.4 Conclusion

On peut constater dans cette étude bibliographique le grand intérêt que suscite le domaine de l'intelligence artificielle en général et des systèmes d'aide à la décision basés sur la logique floue en particulier. Plusieurs travaux se sont intéressés à l'as-

pect de génération et d'apprentissage automatiques des bases de connaissances floues. Ces travaux s'appuient sur plusieurs méthodes d'apprentissages, allant des méthodes plus classiques aux métaheuristiques. Toutefois, la plupart des travaux ne s'occupent que d'une seule partie de la génération de BC, soit la BF ou la BR. Ceux proposant des approches duales souffraient du manque de flexibilité quant à la dépendance de la BC par rapport à la BF ou vice-versa. Aussi, les travaux prenant en compte l'optimisation et la complexité de la BC sont très rares, surtout d'un point de vue général et non pas spécifique à une certaine application. La recherche proposée dans cette thèse implique ces deux aspects pour la génération de BC pour les SADF en utilisant des AG. De nouvelles approches dans l'implémentation des AG sont proposées, permettant d'atteindre les objectifs contradictoires de simplicité et de précision de la BC, ainsi que des nouvelles techniques permettant l'amélioration de l'aspect évolutif des AG développés.

# CHAPITRE 2

# ORGANISATION GÉNÉRALE DE LA THÈSE

La thèse présentée ici est sous la forme d'une thèse par articles (trois articles la composent). Les résultats et les discussions des travaux réalisés dans cette étude sont présentés en détail dans les articles présentés aux chapitres 3, 4 et 5 et ont été soumis pour publication dans des revues internationales reconnues dans leur domaine respectif, soit : l'intelligence artificielle pour le premier, la mécanique appliquée et théorique pour le deuxième et finalement l'application de l'intelligence artificielle pour le troisième. Au moment où ce texte est rédigé, les deux premiers articles sont déjà publiés, le dernier est soumis pour fins de révision. Ce chapitre est une synthèse générale (en français) de la thèse afin de ressortir la complémentarité des articles.

La première section de ce chapitre présente les notions générales permettant de comprendre le fonctionnement d'un AG ainsi que des notions de base sur les pâtes thermomécaniques, procédé sur lequel sera appliqué l'apprentissage automatique de bases de connaissances floues pour fins de prédiction et/ou contrôle de la qualité de ladite pâte, une brève synthèse de la recherche est ensuite présentée. Finalement, une synthèse des trois articles est faite.

## 2.1 Généralités sur les algorithmes génétiques

Un AG est une méthode d'optimisation stochastique qui évalue une fonction objectif à un nombre fini de points [31]. Cette méthode est basée sur l'analogie avec le mécanisme de génétique naturel et imite l'approche Darwinienne de la sélection naturelle [15]. En général, un AG est caractérisé par :

1. un codage de chaque solution possible, sous forme d'un ensemble de nombres réels pour les AGCR et sous forme d'une chaîne de bits pour le AGB (ce qui représente le chromosome de l'individu) ;
2. un indice de performance permettant d'évaluer la qualité de chaque solution ;
3. un ensemble initial de solutions, appelé population initiale, généralement construit aléatoirement ou en se basant sur des connaissances *a priori* ;
4. un ensemble d'opérateurs de reproduction, mutation et sélection naturelle, afin de permettre l'évolution de la population de génération en génération.

Les AG utilisent une amélioration itérative des individus à chaque génération pour converger de façon stochastique vers un optimum global. Ceci est fait au moyen de trois opérations : croisement, mutation et sélection naturelle.

### 2.1.1 Croisement

L'évolution de la population à chaque génération est obtenue par la reproduction des meilleurs individus, basée sur leurs habilités à survivre à la sélection naturelle.

La reproduction est généralement faite par croisement du chromosome des parents afin d'obtenir le chromosome des enfants. Le croisement se fait à l'aide d'une fonction de croisement qui produit une certaine combinaison entre les chromosomes des parents, comme suit :

- les parents sont sélectionnés en se basant sur leur indice de performance, les meilleurs sont favorisés ;
- le chromosome des enfants est formé par le biais d'une fonction de croisement établie a priori.

### 2.1.2 Mutation

La mutation est la création aléatoire d'un nouveau chromosome d'un individu lors de la reproduction à partir d'un individu existant. La mutation permet de considérer des solutions complètement différentes et ainsi potentiellement trouver de meilleures solutions.

### 2.1.3 Sélection naturelle

La sélection naturelle est appliquée de façon à conserver les individus les plus prometteurs basée sur leurs indices de performance. Elle se fait par un système de pondération simple des individus d'une même population. Un biais est mis en place pour favoriser la reproduction des meilleurs, sans éliminer la possibilité de faire reproduire les moins bons, dans le but de conserver une certaine diversité dans la

population. Par simplicité, la taille de la population peut être conservée constante. La figure 2.1 montre une vue d'ensemble du fonctionnement d'un algorithme génétique.

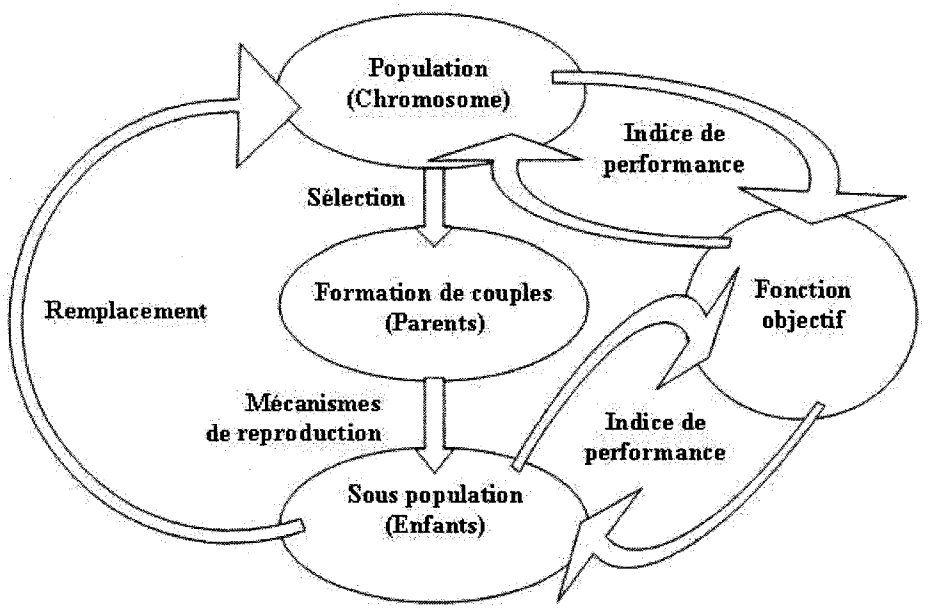


FIG. 2.1 Vue d'ensemble du fonctionnement de l'AG

## 2.2 Généralités sur le procédé de pâtes thermomécanique

Le traitement des papiers modernes se déroule en deux stades :

1. Dissociation des agglomérats cellulosiques liés par la lignine, de façon à faire apparaître les fibres à l'état individuel, dans les usines à pâte.
2. Accroître la surface des fibres libérées, puis assurer leur rapprochement intime, de manière à former une feuille, dans les usines à papier.

Il existe trois sortes principales de pâtes : les pâtes mécaniques, les pâtes mi-chimiques et les pâtes chimiques. Dans cette section nous considérons les pâtes mécaniques en général et les thermomécaniques en particulier.

### 2.2.1 Pâte mécanique

La classification simplifiée des pâtes mécaniques est comme suit :

- Pâtes mécaniques de meules (à partir de rondins) ;
- Pâtes mécaniques de copeaux ;
- Pâtes thermomécaniques ;

La pâte thermomécanique est obtenue avec des désintégrateurs à disques, traitant à température élevée (plus de 100°C) les copeaux de bois. Elle intervient à 100% dans le papier journal (sans adjonction de pâte chimique) lorsqu'elle est thermomécanique. Elle entre aussi dans les cartons de toutes espèces, sous forme écrue ou blanchie. La pâte blanchie est une pâte écrue qui a subit un blanchiment.

#### 2.2.1.1 Blanchiment de la pâte thermomécanique

Il s'agit de traitements chimiques en une seule phase (quelquefois deux) utilisant des réactifs très spécifiques qui permettent d'améliorer très sensiblement la blancheur des pâtes.

Les réactifs utilisés sont essentiellement le peroxyde d'hydrogène ($H_2O_2$) et l'hydrosulfite de sodium ($Na_2S_2O_4$).

## 2.3 Recherche proposée

La recherche proposée ici concerne la génération automatique des BC pour les SADF. L'outil d'optimisation utilisé est un AG au codage hybride. Il utilise un double paradigme de codage, binaire et réel. Aussi, les mécanismes de croisement proposés sont des mécanismes spécifiques adaptés à la tâche d'optimisation. Des techniques d'amélioration du comportement de l'AG sont aussi proposées. Le dernier volet de cette thèse concerne la construction de modèles prédictifs de la qualité de la pâte thermomécanique. En résumé, les objectifs de cette thèse sont :

**1. Objectif 1**

Développer un module de génération automatique de BC pour les SADF.

**2. Objectif 2**

Explorer les effets de nouvelles techniques de reproduction sur les performances de l'AG face au problème de convergence prématurée.

**3. Objectif 3**

De par une étroite collaboration avec le Centre de Recherche Industrielle du Québec (CRIQ), développer des modèles prédictifs de la qualité des pâtes thermomécaniques. Ces modèles seront en forme de BC, desquelles il sera possible de développer des systèmes de contrôle prédictif.

## 2.4 Algorithmes génétiques hybride et binaire pour la génération automatique de bases de connaissances (article 1)

Un algorithme génétique codé en binaire pour la génération automatique de BC pour les SADF a été proposé par l'auteur en 2001 [16]. Ces travaux ont fait objet de plusieurs publications [4,5,12–14,16]. L'AG codé en binaire, certes performant, posait néanmoins certains problèmes tels que :

1. basse résolution des solutions : résolution limitée par le nombre de bits alloués aux paramètre à optimiser ;
2. espace de recherche des solutions appartenant à l'ensemble des réels, d'où le besoin de transformer les chaînes de bit d'entiers à des réels (de génotype à phénotype) ;
3. présence de convergence prématurée, amplifiée lorsque le problème à modéliser est complexe.

Ces raisons nous ont poussés à développer un nouvel outil de génération automatique de BC, en se basant sur le paradigme de codage aux nombres réels. Le nouveau codage proposé combine les propriétés du codage binaire avec celui du codage aux nombres réels. Cette partie traite donc de ce nouvel outil et de ses performances par rapport à un codage binaire.

### 2.4.1 Présentation

Le chapitre 3 est consacré à la présentation d'un outil de génération automatique de BC, utilisant deux AG différents, à partir de données numériques. Le SADF utilisé comme outil de validation des BC est le "FDSS Fuzzy-Flou" [10].

Toutes les BC développées dans ce chapitre (et dans cette thèse) sont à plusieurs entrées et une sortie, ce qui les classe dans le type MISO (*MULTIPLE INPUT SINGLE OUTPUT*). Les paramètres gérés par les deux algorithmes sont :

- le nombre et la répartition des sous-ensembles flous sur les prémisses d'entrées ;
- le nombre et la répartition des sous-ensembles flous sur la conclusion ;
- les règles floues permettant d'interconnecter les différents sous-ensembles flous.

Le MI utilisé dans les deux AG est celui proposé par Larsen (Larsen t-norm), la méthode de deffuzification choisie est le centre de gravité. Afin de réduire le nombre de paramètre à optimiser et du fait même augmenter la vitesse de convergence, nous n'avons considéré que des sous-ensembles flous de forme triangulaire sur les prémisses (ils se superposent) et des triangles isocèles, à bases égales, sur la conclusion, ce qui n'alloue aucun biais vers une sortie en particulier (autre que le choix de la règle floue).

Le nombre de sous-ensembles flous sur les prémisses, ajouté à celui sur les conclusions, plus le nombre de règles floues actives dans le modèle, définit le niveau de complexité de la BC développée.

Le premier AG présenté est un AG codé en binaire (AGB), le second est un AG combinant les propriétés du codage binaire et du codage réel, soit un AG codé aux nombres réels et binaires (AGCRB). Les deux AG servent à l'apprentissage de BC à partir de données numériques, ils permettent la réalisation du paradigme contradictoire qui réside dans le fait de générer des BC précises (reproduction des données numériques avec une erreur minimale) et simple (faible niveau de complexité).

L'AGB utilise un croisement simple pour la BF et la BR ; aussi, la BR est générée avec des règles inactives. L'AGCRB utilise le mécanisme de croisement *Blended Crossover* $\alpha$ pour la BF et un croisement simple pour sa BR, ce choix est motivé par le fait que la BR ne contient que des nombres entiers, ce qui rend son génotype similaire à une chaîne de bits (nombres entiers).

Les performances de l'AGB et de l'AGCRB sont évaluées par les points suivants :

- niveau de simplicité ;
- temps d'apprentissage ;
- influence de la taille de la population sur leur performance.

Les deux AG utilisent le même niveau de complexité au départ de l'apprentissage. Les données d'apprentissage sont obtenues à partir de surfaces en trois dimensions, possédant des degrés de complexité variés. Les surfaces sont inspirées des séries de tests proposés par *De Jong* [25].

La structure du chapitre 3 peut être résumée comme suit :

- une introduction contenant une revue bibliographique ;
- une présentation du fonctionnement d'un SADF ;
- une présentation du fonctionnement de l'AGB et de l'AGCRB ;
- une présentation du paradigme d'apprentissage de BC ;
- les essais de validation des deux AG sur des surfaces synthétiques de différentes complexités ;
- une comparaison et une discussion des résultats des deux AG quant à leurs niveaux de performances ;
- une interprétation des résultats obtenus et une conclusion générale.

Cet article a été soumis, accepté et publié dans la revue "Engineering Applications of Artificial Intelligence".

Achiche, S., Baron, L. et Balazinski, M., "Real/Binary-Like Coded Versus Binary Coded Genetic Algorithms to Automatically Generate Fuzzy Knowledge Bases : A Comparative Study", *Engineering Applications of Artificial Intelligence*, Vol. 17,

No. 4, pp. 313-325, 2004.

## 2.5 Stratégies multicombinatoires pour éviter la convergence prématurée dans les algorithmes génétiques (article 2)

L'AGCRB développé a prouvé son efficacité ; néanmoins, comme c'est le cas pour presque tous les algorithmes métaheuristiques, une fois que l'évolution atteint un certain niveau de maturité, les individus composant la population de solutions tendent à trop se ressembler (mêmes performances, structures trop semblables, etc.), ce qui freine ou met un terme à l'évolution et dans des rares cas, seule le mutation permet d'explorer de nouvelles solutions. Ce phénomène est appelé la convergence prématurée. C'est cette problématique qui a motivé le travail présenté dans le chapitre 4.

### 2.5.1 Présentation

Le chapitre 4 est consacré à la présentation de techniques permettant d'éviter ou de réduire l'impact de la convergence prématurée sur l'AGCRB dans sa tâche de génération automatique de BC à partir de données numériques. Dans une famille (humaine) nombreuse, il est rare que les enfants aient tous la même spécialisation, il est rare aussi des les voir tous réussir (dans différents domaines sociaux) aux mêmes niveaux pourtant, d'un point de vue puerement génétique, ils partagent tous les gènes de mêmes parents ; aussi, ces mêmes parents peuvent donner naissance à

des enfants très différents en taille, teint, etc. Ce sont ces deux idées qui sont exploitées et transférées au monde des AG. L'hypothèse se basait sur l'apport d'une approche de famille nombreuse au sein même de l'AG, ainsi que l'apport d'une variation des mécanismes de croisement (imitant ainsi les variations des gènes hérités par des enfants différents de mêmes parents) améliorerait le comportement du AGCRB quant au problème de la convergence prématurée. Pour ce faire, les changements sont apportés aux mécanismes de croisement de la BF où, selon la stratégie d'évolution préconisée, trois différents mécanismes sont utilisés, soit :

1. *blended crossover* $\alpha$ ;
2. *non-uniform arithmetical crossover*;
3. *extended line crossover.*

Le mécanisme *blended crossover* $\alpha$ permet la variation de son appartenance à un état d'exploration ou d'exploitation de l'information génétique par le biais de sa variable d'équilibre $\alpha$, les autres sont stables (exploitation pour le deuxième et plus d'exploration pour le troisième).

Les différentes approches de stratégies évolutives proposées dans cet article sont :

- application unique : un mécanisme de croisement à la fois ;
- combinaison linéaire : une combinaison linéaire des trois mécanismes est utilisée ;

- application simultanée : les trois mécanismes sont utilisés simultanément dans l'évolution ;
- distribution de l'exploitation et de l'exploration : un changement dans le niveau d'exploration/exploitation est fait dépendant de l'état d'avancement de l'évolution (début, milieu ou fin), deux différents agencements sont testés.

Les résultats obtenus sont comparés et discutés. La meilleure des stratégies est choisie pour application sur des données expérimentales et elle est comparée à une stratégie de croisement unique.

La structure du chapitre 4 peut être résumée comme suit :

- une introduction contenant une revue bibliographique ;
- une présentation sommaire du SADF ;
- une présentation du fonctionnement de l'AGCRB ;
- une revue des différents mécanismes de croisement utilisés ;
- une présentation des différentes stratégies d'évolution ;
- les essais de validation des différentes stratégies proposées sur des surfaces synthétiques 3D ;
- une discussion et une interprétation des performances des différentes stratégies ;
- application de la meilleure stratégie d'évolution sur des données expérimentales (données de tournage) ;
- une conclusion générale.

Cet article a été soumis, accepté et publié dans la revue "Journal of Theoretical and Applied Mechanics".

Sofiane Achiche, Marek Balazinski et Luc Baron, "Multi-Combinative Strategy to Avoid Premature Convergence in Genetically-Generated Fuzzy Knowledge Bases", *Journal of Theoretical and Applied Mechanics*, Vol. 42, No. 3, pp. 417-444, 2004.

## 2.6 Prédiction en ligne de la blancheur ISO de la pâte thermomécanique (article 3)

La qualité du papier est assujettie à un ensemble de facteurs directement liés à l'état des copeaux de bois. Cet état est déterminé par des caractéristiques propres et intrinsèques telles que l'essence de bois, la teneur en écorce, la carie, les noeuds, la siccité, la granulométrie ou encore la densité. D'autres facteurs intrinsèques sont susceptibles de perturber certaines propriétés physico-chimiques des copeaux et entraîner la dégradation de la qualité des papiers produits. C'est le cas notamment des conditions d'entreposage des copeaux, de la gestion de piles, des cours à bois ou encore de variations climatiques qui induisent des modifications dans les propriétés des copeaux.

Actuellement, il ne semble pas exister de connaissances établies et tangibles concernant les "influences concomitantes" des caractéristiques des copeaux de bois sur les

procédés. Aucune approche globale multidimensionnelle n'est mise en oeuvre pour mettre en évidence les relations entre le comportement des copeaux et les procédés de transformation en pâte et papier. Il n'existerait pas non plus de définition universelle et communément acceptable sur la qualité ou la fraîcheur des copeaux. Ce sujet est encore à ses débuts. Les connaissances sur la réalité comportementale des copeaux sont très diffuses et éparpillées.

La logique floue combinée à des AG est entièrement appropriée pour saisir la connaissance de ce domaine à partir de données numériques comme il a été prouvé dans les deux articles précédents. Les BC (BF et BR) sont particulièrement bien adaptées pour tisser les relations de causes à effets qui lient les paramètres des copeaux, des pâtes et des papiers correspondants. Néanmoins, dans cette thèse, nous nous sommes arrêtés au lien entre les copeaux et la pâte.

### 2.6.1 Présentation

Les copeaux sont la matière première principale intervenant dans la fabrication des pâtes et papiers. Malheureusement, aucune usine ne possède de systèmes de mesure permettant d'évaluer en continu ; la qualité de cet intrant qui, pourtant, influence directement la qualité du papier fabriqué. Ainsi, le CRIQ a développé un système de mesure en continu ; des propriétés des copeaux, nommé "Chip Management System $CMS^{©}$". Ce système permet de quantifier la qualité des copeaux

qui entrent dans le procédé de raffinage, il permet, entre autres, d'identifier les contaminants (écorce, carie, noeuds, etc.) et mesurer la dimension, la luminance et l'humidité des copeaux, ainsi que des propriétés de couleurs grâce à une caméra RGB. Ainsi, il devient possible d'anticiper la qualité du papier fabriqué à partir des caractéristiques des copeaux et des paramètres de contrôle du procédé. Il existe deux procédés de défibrage des copeaux et de mise en pâte, soit le procédé de pâtes thermomécaniques (PTM) et le procédé Kraft. Le procédé PTM est celui étudié dans cet article (et de par le fait même dans cette thèse). Dans cet article, l'AG-CRB avec une technique de reproduction évolutive et la logique floue ont la tâche de prédire la qualité de la pâte thermomécanique produit dans les industries de fabrication des PTM.

Les BC développées permettent un maintien de la qualité de la pâte à un niveau raisonnable et cela malgré les fluctuations dans la qualité de la matière première qui entre dans le procédé (en l'occurrence les copeaux), et cela en jouant sur la concentration des éléments de blanchiment (peroxyde ou hydrosulfite). L'approche capitalise sur le système d'inspection par vision numérique $CMS^{©}$ ainsi qu'un banc d'essai expérimental du procédé PTM à l'usine pilote de l'Université du Québec à Trois-Rivières.

À partir d'une lecture de la qualité des copeaux $CMS^{©}$ et des paramètres opéra-

tionnels du procédé, un modèle prédictif de la qualité de la pâte produite est établi et permet d'estimer les écarts par rapport aux valeurs de la qualité désirée. Des correctifs sur les paramètres opérationnels pourront être possibles grâce à la BR de la BC, et ce afin de rétablir la qualité à un niveau acceptable, i.e. une compensation des pertes de propriété de la pâte dues à une dégradation possible de la qualité des copeaux par une action sur les concentrations d'éléments de blanchiment.

La structure du chapitre 4 peut être résumée comme suit :

- une introduction contenant une revue bibliographique ;
- une présentation du $CMS^{©}$ et des variables mesurées ;
- une présentation du plan d'expérience établi et des données récoltées ;
- une explication du choix des paramètres faisant partie de l'apprentissage (entrées/sorties) ;
- une présentation sommaire du SADF ;
- une présentation sommaire du AGCRB et du choix de la méthode de reproduction ;
- une application aux données du procédé PTM (pour l'hydrosulfite et le peroxyde) ;
- une discussion des résultats obtenus ;
- une conclusion générale.

Cet article a été soumis à la revue " Engineering Applications of Artificial Intelli-

gence".

Sofiane Achiche, Luc Baron, Marek Balazinski et Mokhtar Benaoudia.

"Online Prediction of Pulp Brightness using Fuzzy Logic Models", soumis à *Engineering Applications of Artificial Intelligence*, Elsevier Editions, décembre 2004.

# CHAPTER 3

# REAL/BINARY-LIKE CODED VERSUS BINARY CODED GENETIC ALGORITHMS TO AUTOMATICALLY GENERATE FUZZY KNOWLEDGE BASES: A COMPARATIVE STUDY

Sofiane Achiche, Luc Baron et Marek Balazinski.
Département de génie mécanique, École Polytechnique de Montréal
P.O. 6079, station Centre-Ville, Montréeal, Québec, Canada, H3C 3A7
sofiane.achiche@polymtl.ca

## 3.1 Abstract

Nowadays fuzzy logic is increasingly used in decision-aided systems since it offers several advantages over other traditional decision-making techniques. The fuzzy decision support systems (FDSS) can easily deal with incomplete and/or imprecise knowledge applied to either linear or nonlinear problems. This paper presents the implementation of a combination of a **r**eal/**b**inary-**l**ike coded **g**enetic **a**lgorithm (RBLGA) and a **b**inary coded **g**enetic **a**lgorithm (BGA) to automatically generate **f**uzzy **k**nowledge **b**ases (FKB) from a set of numerical data. Both algorithms allow one to fulfill a contradictory paradigm in terms of FKB precision and simplicity (high precision generally translates into a higher level of complexity) considering a

randomly generated population of potential FKBs. The RBLGA is divided into two principal coding methods: 1) a real coded genetic algorithm (RCGA) that maps the fuzzy sets repartition and number (which drives the number of fuzzy rules) into a set of real numbers, and 2) a binary like coded genetic algorithm that deals with the fuzzy rule base relationships (a set of integers). The BGA deals with the entire FKB using a single bit string, which is called a genotype. The RBLGA uses three reproduction mechanisms, a BLX-$\alpha$, a simple crossover and a fuzzy set reducer, while the BGA uses a simple crossover, a fuzzy set displacement mechanism and a rule reducer. Both GAs are tested on theoretical surfaces, a comparison study of the performances is discussed, along with the influences of some evolution criteria.

## 3.2 Introduction

Nowadays fuzzy logic is increasingly used in decision-aided systems since it offers several advantages over other traditional decision-making techniques. The fuzzy decision support systems (FDSS) can easily deal with incomplete and/or imprecise knowledge applied to either linear or nonlinear problems. In most cases decision making systems are used when there is:

- an expert available to manually construct the FKB;
- an important nonlinearity of the modeled process.

Hence, there is a difficulty to build a good enough mathematical model (impossible in some cases) that emulates the behavior of the problem to be solved. The construction of FKBs requires the evaluation of each potential solution (the generated FKBs), which allows to establish the accuracy level when comparing their behaviors (outputs) to the one in the learning data. The manual construction of an FKB requires the evaluation of each proposition made by the expert to measure its performance. If the result is not satisfactory, the expert determines the modifications either to the fuzzy sets repartition, to the fuzzy rule base or to both, in the hope of improving the accuracy. This process is extremely long and tedious since it is completely dependent on the expert's intuition, experiences and his understanding of the problem's behavior. Therefore, a multi-criteria optimization tool is highly desirable. The genetic algorithms (GAs) are powerful stochastic optimization methods [12] and are considered here as an optimization tool for the automatic generation of FKBs. The GA allows one to improve FKB performance in terms of accuracy and simplicity with respect to one or several performance criteria. This can be done automatically, i.e., without the need of expert intuition. This paper presents an overview of the FDSS software Fuzzy-Flou, developed at Ecole Polytechnique (Canada) and the University of Silesia (Poland) [4] which was used for the validation tests. Then, it presents a brief description of the BGA and an extended one of the RBLGA which was used in this work, explaining the specifications of the reproduction and mutation mechanisms of each. Finally validation results are

presented along with a comparative study of their performances inspected through different evolution parameters and performance paradigms.

### 3.3 Fuzzy Decision Support System

In this section we present a rule based approach to decision making using fuzzy logic techniques, based on the compositional rule of inference (CRI). This approach is used to handle uncertain knowledge and was developed in the sixties by L.A. Zadeh [21]. Such knowledge can be collected and delivered by a human expert (e.g. decision-maker, designer, process planner, machine operator, etc.). The CRI may be written in the form:

$$U = (C \times \cdots \times B \times A) \circ R \tag{3.1}$$

where $R$ represents the global relation that aggregates all the fuzzy rules, ($A$, $B$, ..., $C$) represents the inputs (observations) and $U$ represents the output (conclusion). The symbol $\circ$ represents the CRI operator. Three defuzzification methods are usually available, i.e., center of gravity (COG), average of maximums (AOM) and the modified center of gravity (MCOG) [4]. The FKB consists of two components: the linguistic term base (database) and the fuzzy production rule base. The database is divided into two parts: fuzzy premises and fuzzy conclusions. Figure 3.1 shows a screen printout of the premises and a conclusion (on the right), the fuzzy rules

and settings (on the left) of the FDSS Fuzzy-Flou software. FKBs can be divided

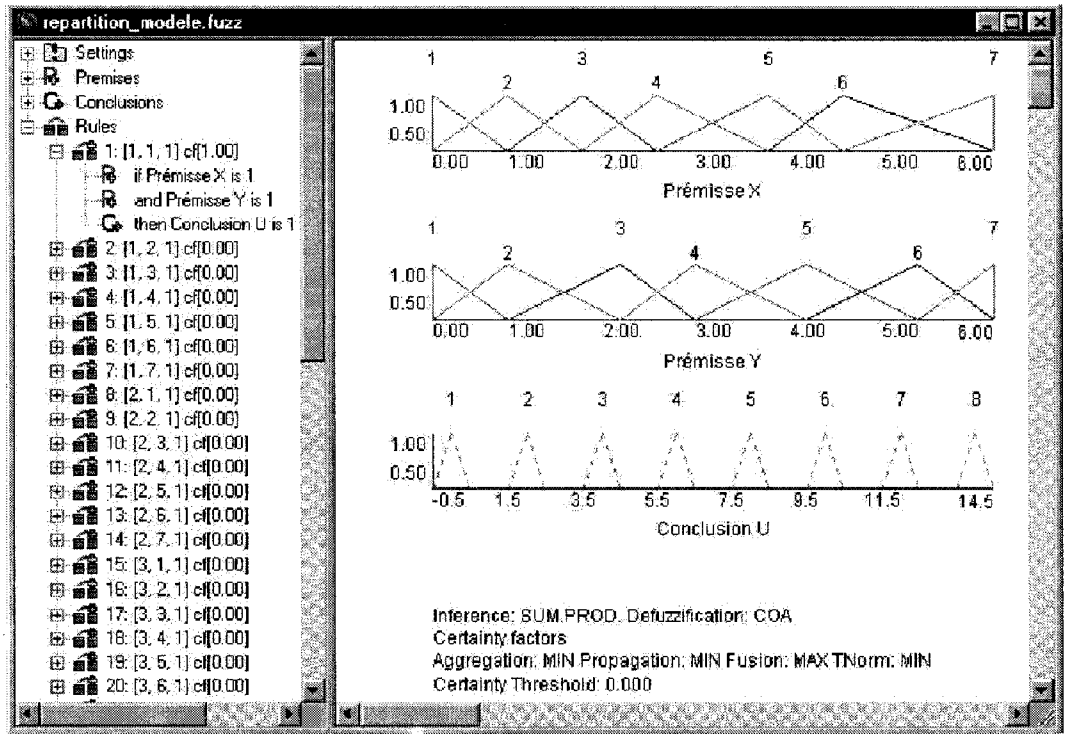


Figure 3.1 Screen shot of the FDSS Fuzzy-Flou

into two main categories:

- multiple inputs single output systems (MISO);
- multiple inputs multiple outputs systems (MIMO).

In this paper only MISOs are considered. The deffuzification mechanism used is the COG. The fuzzy rules are a finite number of heuristic fuzzy rules of the *if then* type.

## 3.4 Automatic Generation of Fuzzy Knowledge Bases using GAs

The automatic generation of fuzzy knowledge bases is performed using a GA. A GA is based on the analogy of the mechanics of natural genetics, and imitates the Darwinian survival-of-the-fittest approach [6]. The GA uses four basic operations: *crossover*, *mutation*, *evaluation* and *natural selection.* Crossover and mutation are used respectively to generate and modify an FKB genotype and the natural selection sorts the different FKBs according to the performance criteria.

Several research used GAs for the automatic generation of FKBs [7,10,13,20]. As shown in Fig.3.2, each individual of a population is a potential FKB. The method uses iterative improvement of individuals at each generation to converge toward multiple optima simultaneously. When the number of unknown parameters increases, GA exhibits only a polynomial increase of the complexity [8,16], while the other optimization techniques show an exponential increase. Figure 3.2 presents the *encoding/decoding scheme* as well as the four basic operations of the developed GA learning software [5].

### 3.4.1 Binary Coded Genetic Algorithm

The binary representation (BGA) of the genotype dominated most of the works using GAs, since the efficiency of the binary coded GAs was proved theoretically [12].

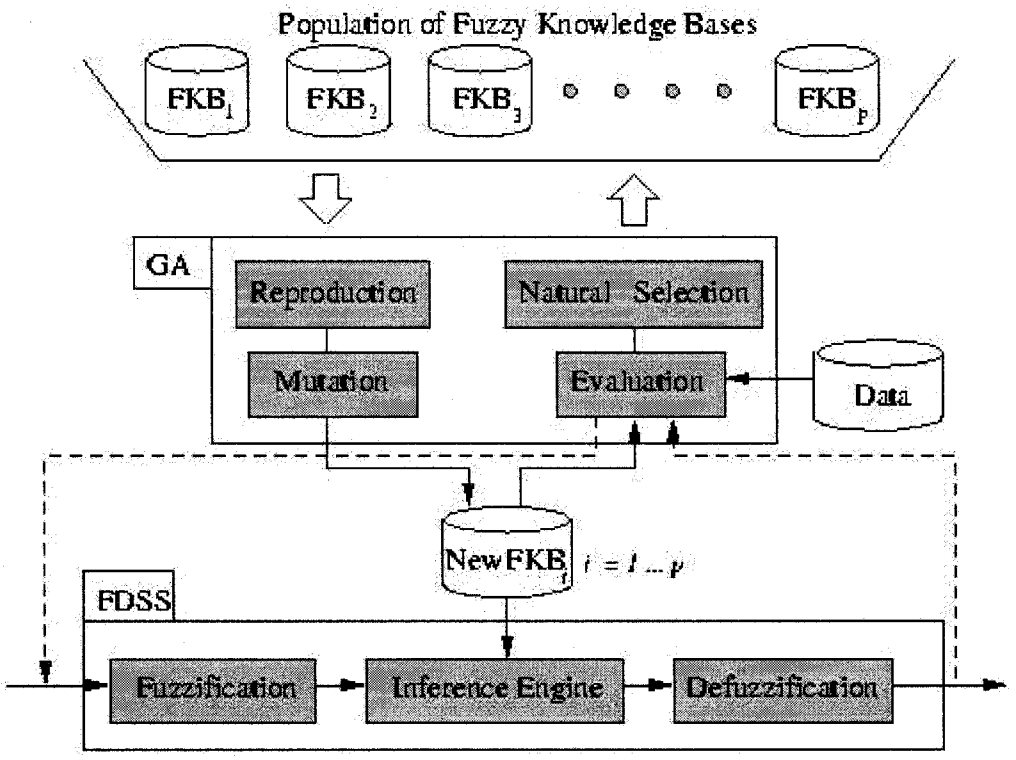


Figure 3.2 The GA learning process of an FDSS Fuzzy-Flou knowledge base

The other obvious reason is the rather simple numerical implementation of BGA. This evolutionary process operates directly on the *genotype*—i.e., the coded physical characteristics into bit string—of individuals rather than on its *phenotype*—i.e., the physical characteristics themselves—.

#### 3.4.1.1 Coding

The *genotype* $G$ of an FKB is the coding of the fuzzy sets and the rule base into a bit-string:

$$G \equiv \{\ G_{sets},\ G_{rules}\ \}, \tag{3.2}$$

where $G_{sets}$ and $G_{rules}$ are respectively the genotypes of the fuzzy sets and the fuzzy rules.

***Input/output premises***

The FDSS Fuzzy-Flou allows the use of trapezoidal membership functions (see Fig. 3.3). For the sake of coding simplicity, we consider only non-symmetrical triangular fuzzy sets on the premises and symmetrical triangular fuzzy sets on the conclusion. Therefore the position of each fuzzy set is given as $m1 = m2 =$ position of the summit, $am$ and $bm$ are set to reach the positions of the previous and next summit, while $hm = 1$ as shown in Fig.3.3. The size of the $G_{sets}$ depends on the number of premises $N$, the number of fuzzy sets $N_i$ on each premise $i$ and the number of bits $b_r$ allocated to specify the resolution on the position. For example, if $b_r = 4$, the genotype of the fuzzy sets of premise $i$ is given as

$$G_{X_i} \equiv \{\ \underbrace{1001}_{summit_1}\ \ \underbrace{1110}_{summit_2}\ \cdots\ \underbrace{0111}_{summit_{K_i}}\ \} \tag{3.3}$$

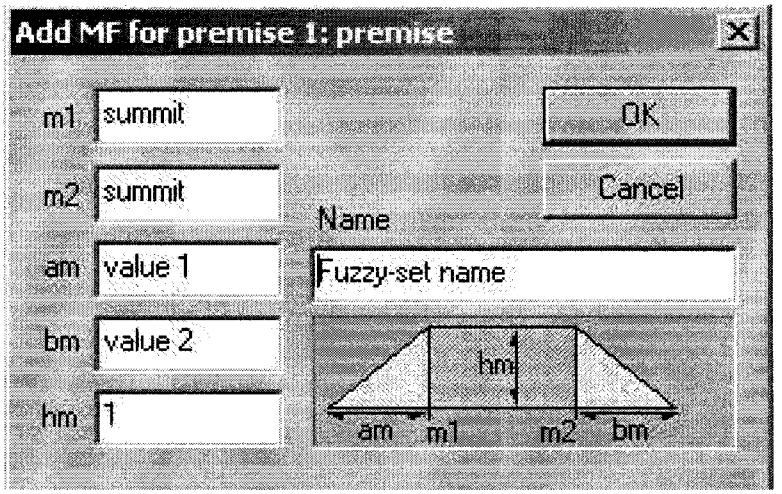


Figure 3.3 Fuzzy set

where $K_i$ is the number of fuzzy sets on premise $X_i$ excluding the two summits located at the extreme values of each premise. The total size of $G_{sets}$ is given as

$$size(G_{sets}) = (\sum_{i=1}^{N} K_i \; b_r) + K_y \; b_r \tag{3.4}$$

where $K_y$ is the number of fuzzy sets on the conclusion. However on the conclusion, the number of fuzzy sets is equal to the number of the coded summits since the limits are also coded.

***Fuzzy rules***

The genotype of the fuzzy rules must contain information about all of the possible combinations connecting one fuzzy set on each premise to a fuzzy set on the conclusion. For $N$ input premises and $K_i$ fuzzy sets on each premise $i$ (including the

limits), the maximum number of fuzzy rules $K$ is computed as:

$$K = (K_1) \times (K_2) \times \cdots \times (K_N) \tag{3.5}$$

The fuzzy rules are coded as an ordered list of combination of the premises, each having an enable/disable bit, denoted $e$—0 for disable; 1 for enable—together with a conclusion fuzzy set number. Each rule is coded into a 4 bit string, i.e.,

$$G_{rules} \equiv \{\underbrace{e101}_{rule_1}\ \underbrace{e111}_{rule_2}\ \cdots\ \underbrace{e011}_{rule_K}\}. \tag{3.6}$$

#### 3.4.1.2 BGA Reproduction Mechanisms

The evolution of the population is achieved by reproduction of the best individuals based on their ability to survive natural selection. This reproduction is performed with a combination of the four following operators:

***1. Simple Crossover***

The main reproduction mechanism is performed by crossing the genotype of the parents, in order to obtain the genotype of two children. The crossover technique used in BGA is a *simple crossover* [19] as shown in Fig.3.4. This part of the reproduction mechanism is governed by an initiating probability $p_1$.

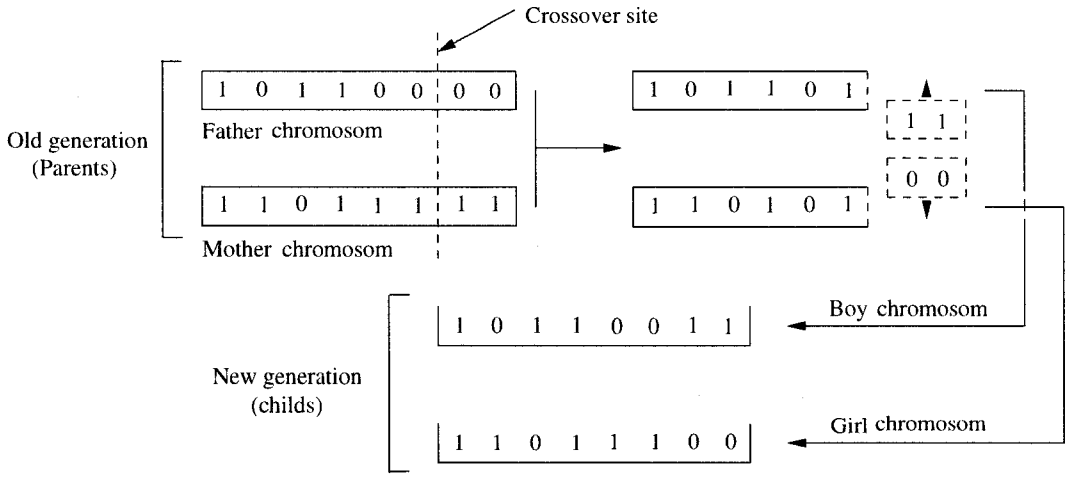


Figure 3.4 Simple crossover

### *2. Fuzzy Sets Displacement*

Displacement of the fuzzy sets is performed (with an initiating probability $p_2$) by randomly selecting a fuzzy set on a premise. The summit of the selected fuzzy set is then moved by one step of resolution toward the left or the right, with an equal probability (see Fig. 3.5). This reproduction operator has the virtue of trying different fuzzy set repartitions, while decreasing the number of fuzzy sets by superimposing two or more of them.

### *3. Fuzzy-Rules Reduction*

The reduction of the number of fuzzy rules is performed with a probability $p_3$ given by

$$p_3 = 1 - p_1 - p_2. \tag{3.7}$$

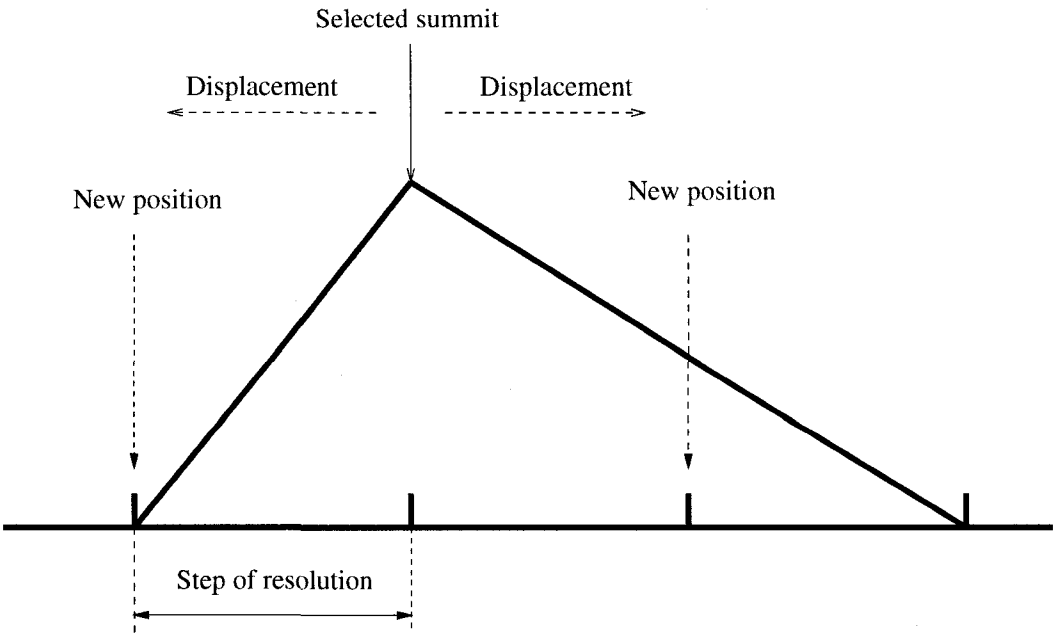


Figure 3.5 Fuzzy-Sets Displacement

One of the $K$ fuzzy rules is randomly selected and deactivated—the bit $e$ is set to disable—as shown in Fig. 3.6. Obviously, this reproduction operator does not always generate a reduction in the number of fuzzy rules (the case when $e$ is already set to 0), but gradually works toward that direction. The bias toward the reduction of the number of rules tends to produce less complex FKBs.

### *4. Mutation*

Mutation is a random inversion of a bit in the genotype of a new member of the population as shown in Fig. 3.7. Mutation makes it possible to try completely different solutions [9]. The probability of mutation $p_4$ should be kept very small in order to give the other reproduction operators precedence for improving the pop-

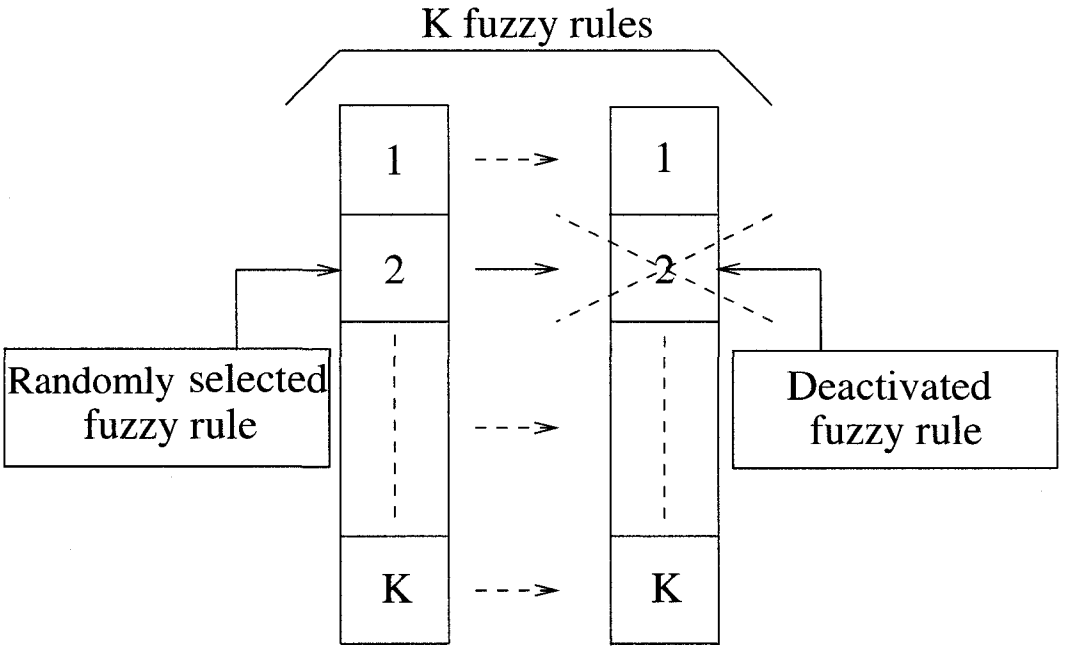


Figure 3.6 Fuzzy-Rules Reduction

ulation.

This way of seeking completely different solutions allows the algorithm to jump out of a local optimum, and potentially fall into more promising regions.

### 3.4.2 Real/Binary like Coded Genetic Algorithm

The most important success of the GAs remains in their evolution paradigm rather than in the way they are coded, hence the occurrence of real coded genetic algorithms (RCGA) in recent years, where they were mostly used in numerical optimization works [14, 18].

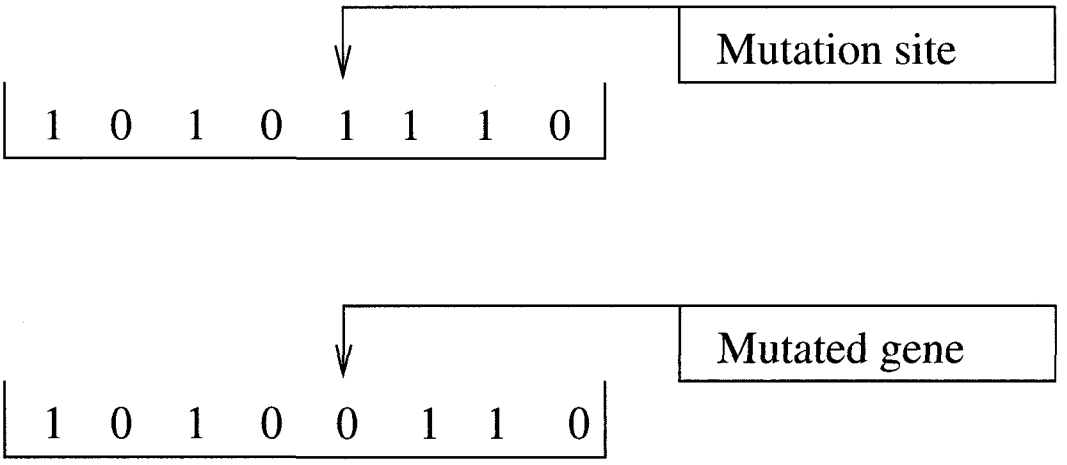


Figure 3.7 Mutation of a genotype

The use of RCGAs overcomes one of BGAs weaknesses, which is the low resolution of the solutions [5]. Moreover, most optimization works are done in a continuous mathematical space (real space). However, the FKBs contain two cooperative but distinct parts:

- the premises and the conclusion, along with the fuzzy sets on them;
- the fuzzy rules base.

They are distinct in such a way that the first one deals with real numbers while the second one uses integer numbers, since the fuzzy rules are simple pointers to the index of a fuzzy set on the conclusion. For this matter, we used a combination between an RCGA and a BGA where the binary part is mapped into a string of integers, the algorithm is called Real/Binary-like Coded Genetic Algorithm (RBLGA).

### 3.4.3 Coding

The *genotype RG* corresponds to several independent sets of reals and a set of integers.

$$RG \equiv \{\ RG_{sets},\ RG_{rules}\ \}, \tag{3.8}$$

where $RG_{sets}$ and $RG_{rules}$ are respectively the genotypes of the fuzzy sets and the fuzzy rules. The *genotype* can be described as follows:

**Input/Output Premises**

There are as many real number sets as there are premises in the problem, and one set for the conclusion. Each set contains a predefined maximum number of real numbers representing the location of the summit of each fuzzy set on each premise and the conclusion. The two summits located at the minimum and maximum limits of each premise and the conclusion are not coded, since they are constant throughout the evolution (similar to the BGA coding).

As in the BGA we consider non-symmetrical-overlapping triangular fuzzy sets for premises and symmetrical triangular fuzzy sets for the conclusion. The genotype of the fuzzy sets of premise $i$ is given as

$$RG_{X_i} \equiv \{\ \underbrace{x_1}_{summit_1}\ \underbrace{x_2}_{summit_2}\ \cdots\ \underbrace{x_i}_{summit_{K_i}}\ \} \tag{3.9}$$

where $K_i$ is the number of fuzzy sets on the premise $i$ (or the conclusion). The limits of the premises are not included in the sets. $RG_{sets}$ is then given as

$$RG_{sets} \equiv \{ \underbrace{RG_{X_1}}_{premise_1}, \underbrace{RG_{X_2}}_{premise_2}, \cdots \underbrace{RG_{X_i}}_{premise_i}, \cdots \underbrace{RG_{X_c}}_{conclusion} \} \tag{3.10}$$

**Fuzzy Rules**

The fuzzy rules are coded as a set of integers representing an ordered list of the combination of the premises. Each integer in the set represents a conclusion fuzzy set summit (see Fig. 3.8). The genotype of the fuzzy rules is given as

$$RG_{rules} \equiv \{ \underbrace{r_1}_{rule_1}, \underbrace{r_2}_{rule_2}, \cdots, \underbrace{r_K}_{rule_K} \}. \tag{3.11}$$

The number of fuzzy rules $K$ is computed using Eq. 3.5.

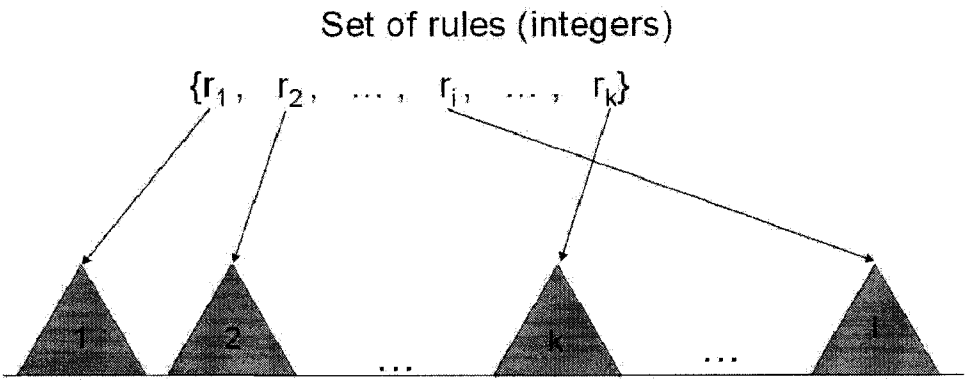


Figure 3.8 Coding of a fuzzy rules set

The initial population of FKBs is composed of $P$ randomly generate FKBs. The *genotype* of each new solution contains all the sets mentioned above. However, as we will explain below, the size of the sets can decrease.

#### 3.4.3.1 RBLGA Reproduction Mechanisms

Reproduction is performed by *crossover* of the parent's *genotype* to obtain the offspring's *genotype* (or two offsprings). The reproduction of the FKBs in the RBLGA is performed through three crossover mechanisms, each one having a certain purpose to achieve, as explained below.

**1. Multi Crossover**

The multi-crossover mechanism is a combination of two crossovers applied on different parts of the *genotype.*

**Premises/Conclusion Crossover**

The mechanism used is called *blending crossover* $\alpha$ (*BLX-*$\alpha$) [11], where $\alpha$ determines the exploitation/exploration level of the offspring obtained from the selected parents. Exploitation indicates the usage of the interval between the values of the two parents; the exploration uses an interval outside of these two limits, hence trying new solutions (see Fig .3.9).

If $A = \{x_1, x_2, \cdots, x_i, \cdots, x_n\}$ and $B = \{y_1, y_2, \cdots, y_i, \cdots, y_n\}$ represents

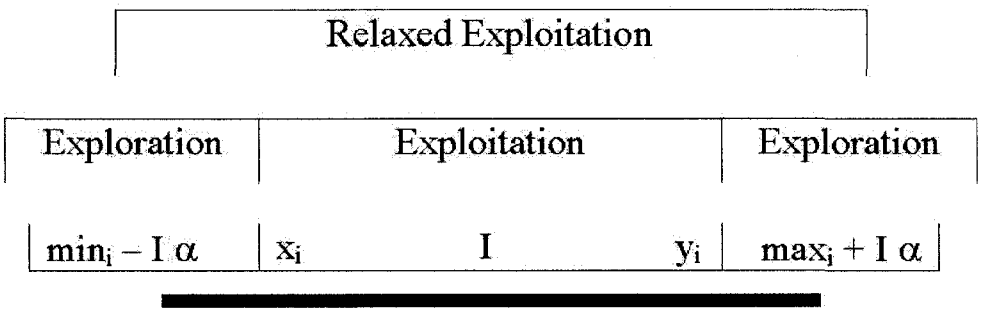


Figure 3.9 Blended crossover $\alpha$ – BLX-$\alpha$

the two selected parents, $C$ the offspring obtained by the crossover of $A$ and $B$ then $C = \{z_1, z_2, \cdots, z_i, \cdots, z_n\}$, where $z_i$ are randomly selected in the interval $[min_i$ -I $\alpha$, $max_i$ + I $\alpha]$ where:

- $max_i$ = maximum $\{x_i, y_i\}$;
- $min_i$ = minimum $\{x_i, y_i\}$;
- I = $\{max_i - min_i\}$.

Figure 3.9 shows the above-described mechanism. The parameter $\alpha$ controls the exploitation/exploration level, knowing that:

- in the exploitation zone, the offspring inherits behaviors close to those of his parents;
- in the exploration zone, the offspring is a result of an exploration, therefore his attributes will be distant from his parent's average.

In order to avoid any bias in either direction (exploitation or exploration) the value of $\alpha$ is set to 0.5, which provides offsprings in the zone named the relaxed exploitation zone (see Fig. 3.9).

**Fuzzy Rules Crossover**

Since the part of the *genotype* representing the fuzzy rule base is composed of integer numbers, the crossover on this part of the *genotype* is done by a *simple crossover*. The use of a BLX crossover is not suitable in this case, because of the integer nature of the values (the rules will tend to be the value zero most of the time). The operation is performed by inverting the end part of the *sets* of the parents at a randomly selected *crossover* site as shown in fig 3.10. These two

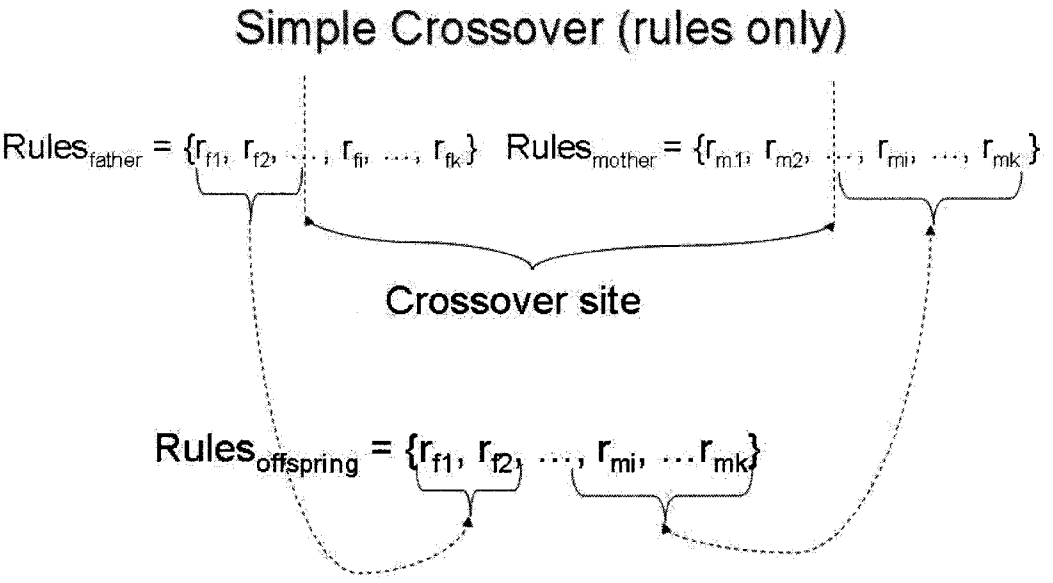


Figure 3.10 BLGA Simple crossover

mechanisms are governed by an initiating probability $pr_1$.

### 2. Fuzzy Set Reducer

This mechanism aims to increase the simplicity level of the FKBs by randomly selecting a fuzzy set on a premise and erasing it together with the corresponding fuzzy rules. This mechanism allows one to obtain different and more simple (less information) solutions (i.e.; FKBs). This mechanism is governed by the initiating probability $pr_2$.

### 3. Mutation

*Mutation* is the creation of an individual by altering the gene of an existing one. The probability $pr_3$ governs the occurrence of this mechanism. The *mutation* used in the RBLGA is a Random mutation (uniform) [17], applied to one randomly selected individual, as follows:

- an individual $C$ is randomly selected, $C = \{z_1, z_2, \cdots, z_i, \cdots, z_n\}$;
- a mutation site is randomly set in the interval [1,n];
- the selected value "$z_i$" is replaced by $z'_i = \text{random}(a_i, b_i)$, where $a_i$ and $b_i$ are the limits enclosing $z_i$.

## 3.5 Learning Process

The learning process is formulated as an optimization problem applied to the numerical data, using the BGA and the RBLGA in ordre to produce near to optimal FKBs.

An FKB contains the following entities/information:

1. the number of premises (inputs) and the number of conclusions (outputs);
2. the number of fuzzy sets and their distribution on the premises and the conclusions;
3. the fuzzy rules (fuzzy rule base).

Item 1 is a part of the problem's input data and all the features in items 2 and 3 are a part of the learning process. The maximal complexity on each premise (i.e.; maximal number of fuzzy sets) is fixed at the beginning of the optimization and therefore these entities are not a part of the learning process (the maximal complexity can differ from premise to premise). After few executions, maximal complexity can be readjusted to a higher number if required.

The goal of the learning process is to generate FKBs while maximizing the performance criteria in terms of accuracy ($\phi_{rms}$) and simplicity level ($\phi si$). Criteria $\phi_{rms}$ and $\phi_{si}$ are defined in section 3.5.1. The oprimizarion problem can be defined as

follow:

For the BGA:

$$Max \;\; f(\phi_{rms},\ \phi_{si}) \;\text{ with } \quad G: \;\; Binary\ Genotype\ . \tag{3.12}$$

For the RBLGA:

$$Max \;\; \phi_{rms} \text{ with } \quad RG: \;\; Real\ Genotype\ . \tag{3.13}$$

### 3.5.1 Performance Criteria

The performance criteria allow one to compute the ratings of each FKB. Those performance ratings are used by the RBLGA and the BGA in order to perform natural selection. The principal performance criterion is the accuracy level of the FKB (approximation error) in reproducing the outputs of the learning data. The approximation error of the FKB is measured using the root-mean-square error

$$\Delta_{rms} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(GA_{output_i} - data_{output_i})^2} \tag{3.14}$$

where $M$ represents the number of points in the sampled data. The fitness value is evaluated as a percentage of the conclusion length base ($L$), as follows:

$$\phi_{rms} = \frac{L - \Delta_{rms}}{L} \times 100 \,, \tag{3.15}$$

while $(100 - \phi_{rms})$ being the error percentage.

The RBLGA uses the value of $\phi_{rms}$ as the only evaluation criterion to perform the natural selection, however the BGA uses a second performance criterion that rates the simplicity of the FKBs ($\phi_{si}$). The use of this additional criterion is due to the particularities of the reproduction mechanisms (fuzzy set displacement and fuzzy rules reduction) that are less straight forward than the ones used by the RBLGA when it comes to reducing the size of the FKB. The fitness value, denoted $\phi_{si}$, is defined as:

$$\phi_{si} \equiv \frac{K - n_a}{K} \,, \tag{3.16}$$

where $K$ is the maximum number of fuzzy rules (using the initial complexity) and $n_a$ is the number of active rules of the FKB under evaluation. In order to chose between these two contradictory objectives, the BGA uses a weighted sum of the two objective functions, i.e.:

$$\phi_{BGA} = \omega\phi_{rms} + (1 - \omega)\phi_{si}, \tag{3.17}$$

where $\omega$ sets the bias between $\phi_{rms}$ and $\phi_{si}$.

### 3.5.2 Natural selection

Natural selection is performed on the population by keeping the *most* promising individuals based on their fitness value. This is equivalent to using solutions that are closest to the optimum. For convenience, we keep the size of the population constant.

#### 3.5.2.1 Natural selection in the BGA

The first generation starts with $P$ FKBs and additional $Ps$ are generated by reproduction and mutation. These brand new FKBs are then evaluated. *Natural selection* is applied on the $2 \times P$ FKBs by ranking them based on $\phi_{bga}$ and $\phi_{rms}$. We keep the first $P/2$ non identical —to maintain diversity in the population— FKBs of the two lists.

#### 3.5.2.2 Natural selection in the RBLGA

The first generation begins with $P$ FKBs, and the same number of additional FKBs are generated by reproduction and mutation. Moreover, in the RBLGA *natural selection* is applied on the $2 \times P$ FKBs by ordering them according to the principal performance criterion $\phi_{rms}$ and keeping the $P$ first FKBs.

The size $P$ has to be selected depending upon the performances of the computer in

use. A high value of $P$ generally ensures a better diversity in the population, which helps to avoid premature convergence. However it increases the learning time.

## 3.6 Validation Results

The BGA and RBLGA learning performances are now investigated using three examples of known behavior in terms of type $z = f(x, y)$ $3D$ surfaces, where the nodes are the learning set of sampled data. We have used three different surfaces of different complexities to have a better idea on the generality of the results (rather than a specific result to a specific surface). The evolution and selection criteria used for both algorithms are set to the following values:

**BGA's evolution/selection criteria:**

- $p_1 = 85.0\%$;
- $p_2 = 13.5\%$;
- $p_3 = 1.5\%$;
- $p_4 = 5.0\%$;
- $\omega = 1.0$;
- Maximal complexity: 6 fuzzy sets (including the limits) on each premise and 8 on the conclusion (8 being the maximum allowed by the coding resolution).

From this set of parameters we can say that the dominant evolution of the BGA is performed using the *simple crossover* with a probability $p_1$. The emphasis on the *Fuzzy-Rules reduction* reproduction mechanism is set at a low level to let the FKBS evolve toward a lower complexity level naturally rather than in a forced way. With $\omega = 1$ the emphasis of the selection mechanism is put on $\phi_{rms} = \phi_{BGA}$. These values have been chosen with respect to the conclusions driven from the work presented in [3].

**RBLGA's evolution/selection criteria:**

- $pr_1 = 85.0\%$;
- $pr_2 = 15.0\%$;
- $pr_3 = 5.0\%$;
- Maximal complexity: 6 fuzzy sets (including the limits) on each premise and 8 on the conclusion (no limits on the number, since it can match the number of fuzzy rules).

The evolution is mostly governed by the *multi-crossover reproduction* mechanism, however the *fuzzy set reducer* weight is not negligible which will tend to produce more simple FKBs.

**Note1:** In the RBLGA the value of $pr_2$ is equal to the sum of the probabilities $p_2$ and $p_3$ of the BGA. This equality sets a good comparative basis, since the

mechanism driven by these probabilities performs similar changes in the FKBS, i.e., reducing the size of the fuzzy rule base.

**Note2:** The maximal complexity is set in order to fix the maximum size of the genotypes, it can be changed if the early results are not satisfying. However, the number of variables taking part in the learning process can decrease through the generations and moreover, in the same population both the BGA and the RBLGA can deal with individuals of different sizes below the fixed maximum.

### 3.6.1 Theoretical Surfaces

The theoretical surfaces used for the learning process are as follows:

**1. Sinusoid surface**

The theoretical sinusoid surface (fig. 3.11) is defined as

$$z = sin(x\ y) \text{ with } \begin{array}{l} 0 \leq x \leq 1.6 \\ 0 \leq y \leq 1.4 \end{array}, \tag{3.18}$$

**2. Exponential surface**

The theoretical concave surface (fig. 3.12) is defined as

$$z = exp(x^2 + y^2) \text{ with } \begin{array}{l} -1.5 \leq x \leq 1.5 \\ -2 \leq y \leq 2 \end{array}, \tag{3.19}$$

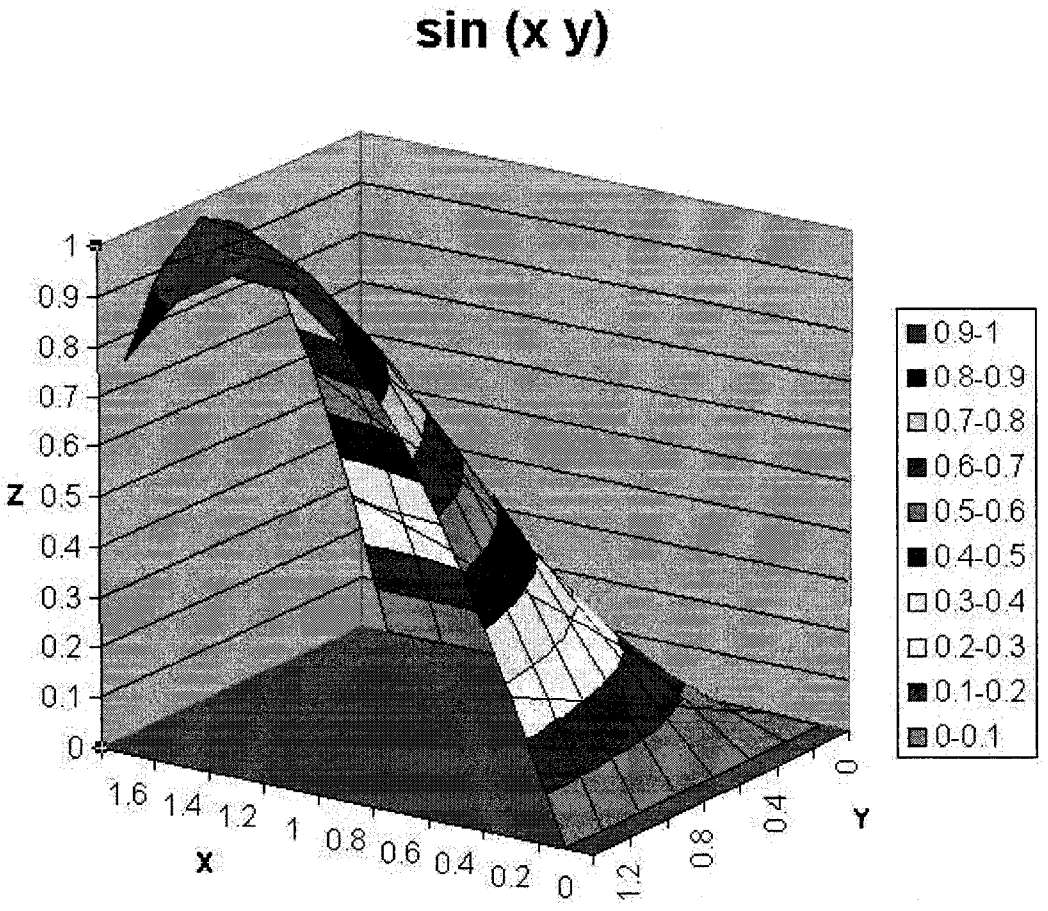


Figure 3.11 Theoretical sinusoid surface

### 3. Hyper-tangent surface

The theoretical Hyper-tangent surface (fig. 3.13) is defined as

$$z = tanh(x\ (x^2 + y^2)) \text{ with } \begin{matrix} -0.2 \leq x \leq 1.4 \\ -0.2 \leq y \leq 1.4 \end{matrix}, \tag{3.20}$$

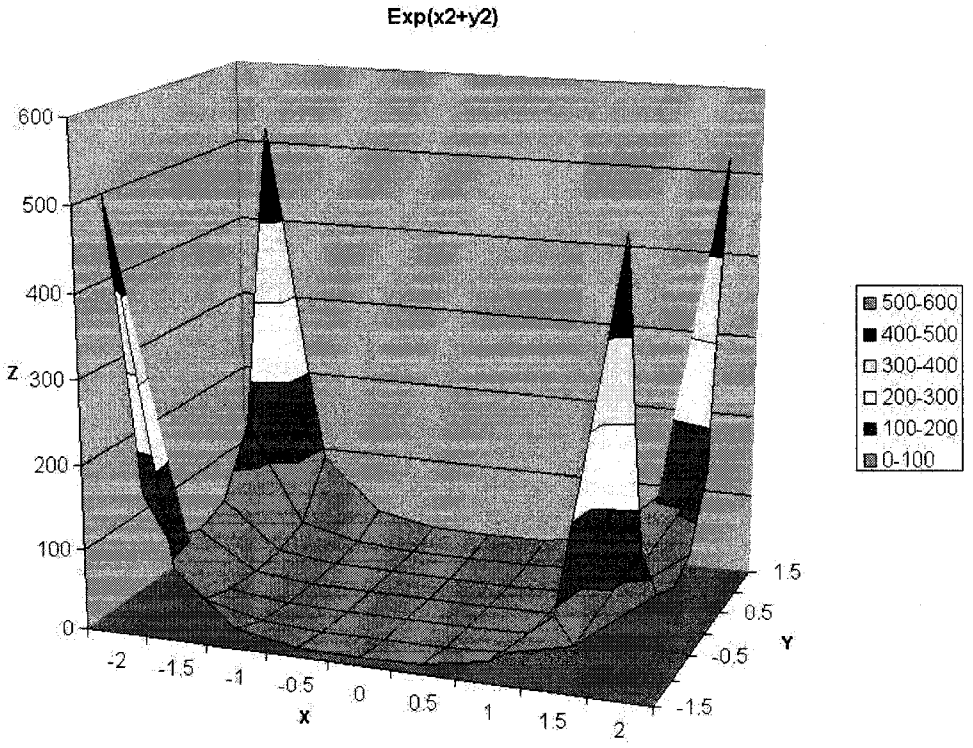


Figure 3.12 Theoretical exponential surface

### 3.6.1.1 FKBs Accuracy level

To measure the accuracy levels (fitness levels) of both the BGA and the RBLGA, in generating FKBs, and for the sake of comparison, several runs have been made on the three surfaces cited above. The population size $P$ was set to 100. Runs were performed for 10, 50, 100, 500, 1000 and 2500 generations. The best individual's fitness level has been taken into account at the last generation. The average value of the three different results obtained for each theoretical surface was computed, as shown in Fig. 3.14 and compiled in Table. 3.1.

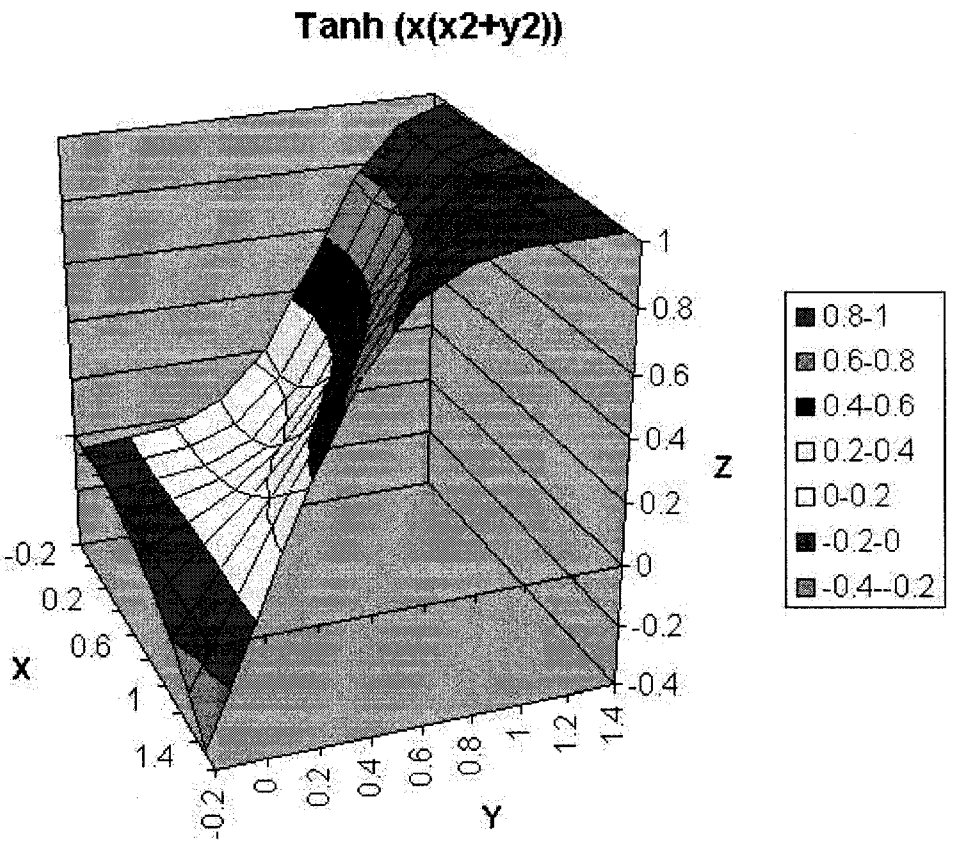


Figure 3.13 Theoretical hyper-tangent surface

Fig. 3.14 shows the superiority of RBLGA when it comes to accuracy for the same number of evaluated solutions (number of generation × population size). For instance, the highest level of accuracy obtained by the BGA after the exploration of 250 000 FKBs (i.e. 91.65%) is below the one achieved by the RBLGA after exploring 5 000 FKBs, meaning that the RBLGA is able to create a more versatile population of FKBs faster than the BGA. The BGA was late in the accuracy race even when 50 000 times more solutions where explored. This leads to the conclusion that when dealing with a complex model to map into an FKB, the RBLGA

Table 3.1 Average $\phi_{rms}$ percentage versus the number of generations

| Population size | Generation Number | Fitness BGA | Fitness RBLGA |
|---|---|---|---|
| 100 | 10 | 68.15% | 79.93% |
| 100 | 50 | 77.73% | 93.38% |
| 100 | 100 | 80.52% | 95.08% |
| 100 | 500 | 86.49% | 95.72% |
| 100 | 1000 | 89.99% | 95.78% |
| 100 | 2500 | 91.65% | 95.94% |

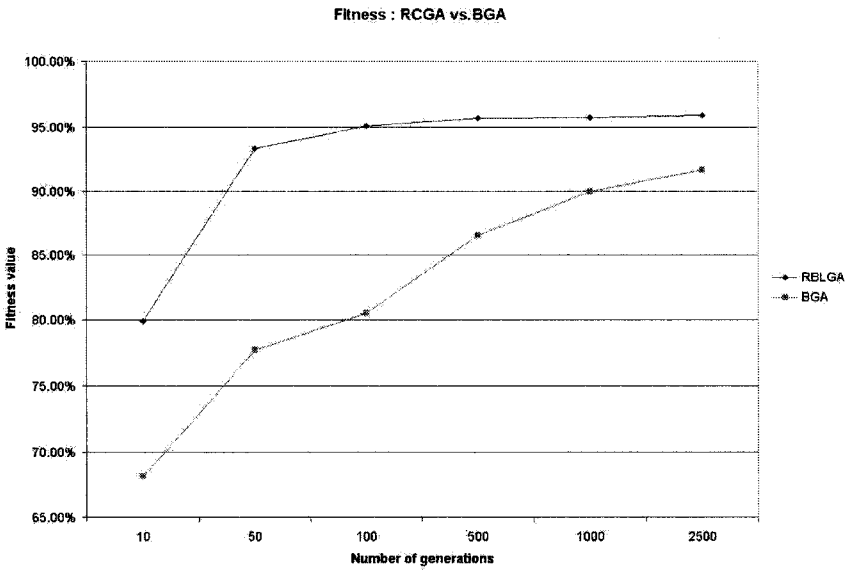


Figure 3.14 Fitness values : RBLGA vs. BGA

will most probably produce a more satisfactory accurate FKB.

Figure 3.14, also shows that the RBLGA tends to reach an accuracy plateau faster than the BGA, which drags the problem of premature convergence, very often a drawback to real coded GAs.

#### 3.6.1.2 The Simplicity level of the FKBs

In this section, the simplicity level of the FKBs is studied. The simplicity level of an FKB is inversely proportional to the number of fuzzy rules representing the fuzzy rule base. The two main advantages of constructing more simple FKBs are:

1. a simple FKB is more flexible toward a manual tuning by a human expert;
2. a simple FKB tends to be a more general model to the existing problem as reported by the authors in [3].

Table 3.2 shows the averages of the rule base size obtained for the three different surfaces.

Figure 3.15 shows that the BGA is more successful in creating simple FKBs,

Table 3.2 Size of the rule base versus the number of generations

| Population size | Generation Number | # of rules BGA | # of rules RBLGA |
|---|---|---|---|
| 100 | 10 | 14.33 | 23.67 |
| 100 | 50 | 12 | 19 |
| 100 | 100 | 12.33 | 19 |
| 100 | 500 | 11 | 19 |
| 100 | 1000 | 11.33 | 19 |
| 100 | 2500 | 12.33 | 19 |

and the RBLGA reached a simplicity plateau with only 19 fuzzy rules. Taking into account that the primary maximum number of fuzzy rules was 36, 19 fuzzy rules still represents approximately a 50% simplification rate. The difference in the simplicity level between the BGA and the RBLGA is quite predictable from of

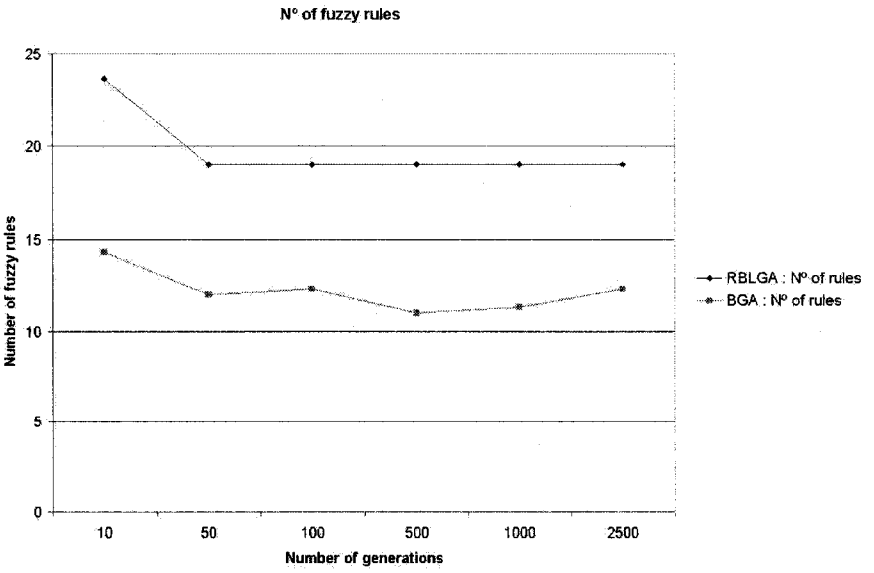


Figure 3.15 Number of fuzzy rules : RBLGA vs. BGA

the differences in the reproduction mechanisms. The BGA simplifies the FKBs in three different ways:

1. the first randomly generated population of individuals is already created with inactive fuzzy rules;

2. the *Fuzzy Sets Displacement* reproduction mechanism can reduce a set of fuzzy rules if a summit is separated by a step of resolution from his neighbor;

3. the *Fuzzy Rules Reduction* reproduction mechanism randomly reduces the rules.

while the only way the RBLGA reduces rules is by reducing the number of fuzzy sets on the premises (by erasing rather than displacing). The lowest number of

fuzzy rules the BGA proposed is around 12—≈ 67%—which is better than the RBLGA.

#### 3.6.1.3 Learning time

We studied the accuracy and the complexity/simplicity level of the FKBs. However, an important aspect of the automatic generation of FKBs is the GAs learning time. In this section, we explore the learning time (LT) of the BGA versus the LT of the RBLGA. The comparison is based on the average LT obtained for the same three surfaces, using the same evolution/selection criteria. Figure 3.16 illustrates

Table 3.3 Learning time RBLGA vs. BGA

| Population size | Generation Number | LT BGA [min] | LT RBLGA [min] |
|---|---|---|---|
| 100 | 10 | 0.02 | 0.04 |
| 100 | 50 | 0.03 | 0.21 |
| 100 | 100 | 0.05 | 0.41 |
| 100 | 500 | 0.20 | 2.18 |
| 100 | 1000 | 0.41 | 4.84 |
| 100 | 2500 | 1.18 | 16.63 |

the results that were compiled in table 3.3. It is quite obvious that the BGA outperforms the RBLGA, since for the same number of explored solutions, the LT for the BGA is much lower. Also the slope of the curve representing the evolution of the RBLGA is higher than the one of the BGA, which means that the LT increases faster for the RBLGA. However, even if the LT of the BGA is very advantageous, the accuracy result is still very low (see Table 3.1).

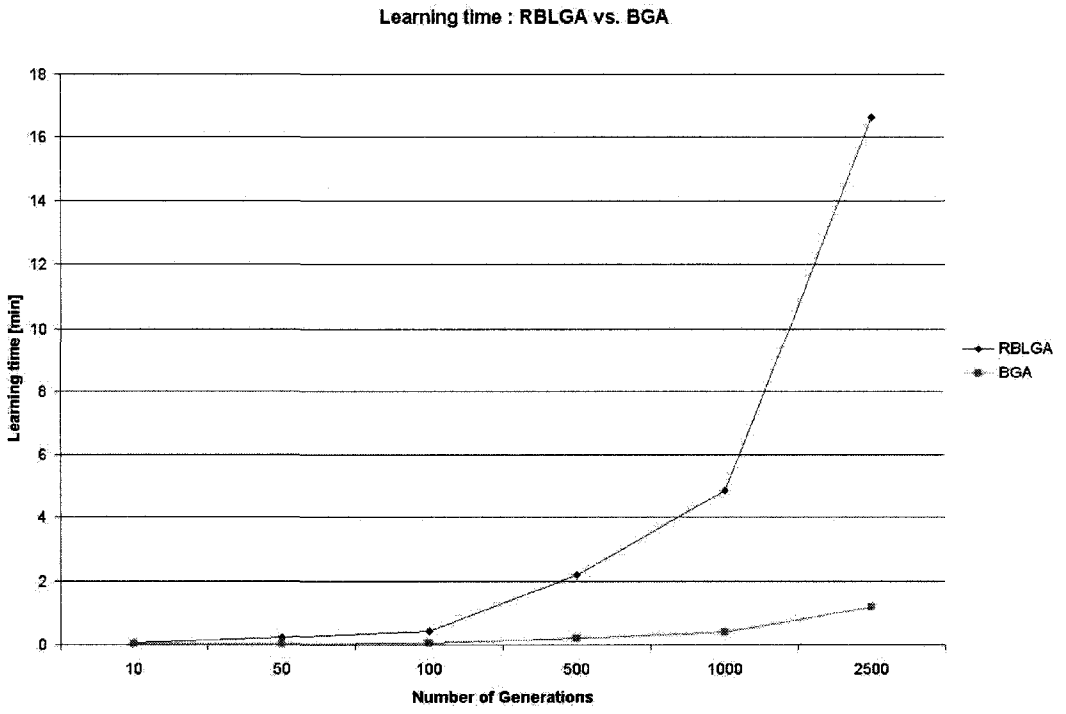


Figure 3.16 Learning time : RBLGA vs. BGA

The first question is: for approximatively the same LT, which GA performs better (i.e.; better fitness)?

From Table 3.4, for the same LT ($\approx 0.20\ min$) the RBLGA reaches the fitness

Table 3.4 Fitness BGA vs. RBLGA: same LT

| GA | Population size | Generation Number | LT [min] | Fitness |
|---|---|---|---|---|
| BGA | 100 | **500** | 0.21 | 86.49% |
| RBLGA | 100 | **50** | 0.20 | 93.38% |

level of 93.38%, while the BGA reaches 86.49%, hence, concerning this aspect, the RBLGA outclasses the BGA.

The second question is: what is the LT needed by the BGA to reach an accu-

racy level, near the best ones which are proposed by the RBLGA?

To determine the LT for approximatively the same level of fitness for both AGs, we pushed the evolution of the BGA up-to 10 000 generations. Table 3.5 shows the average of the results obtained for the three surfaces. The BGA reached an accuracy level of 93.41% after 9.62 *min* (and 10 000 generations) while the RBLGA proposed approximatively the same value in around 0.2 *min* (see Table 3.4). From theses different results, we can conclude that the RBLGA is more efficient, relative to the LT.

Table 3.5 LT and Fitness of the BGA after 10 000 generations

| GA | Population size | Generation Number | LT [min] | Fitness |
|---|---|---|---|---|
| BGA | 100 | **10 000** | 9.62 | 93.41 % |

#### 3.6.1.4 Influence of the population size

In this section we explore the influence of the population size on the outcome of the learning process. For the sake of comparison the number of generations is fixed at 100 for both algorithms. Table 3.6, highlights the influence of the population size.

Increasing the population size improves the performances of both algorithms when it comes to accuracy. However, since accuracy and the rule base size are linked, the size of the rule base increases for the RBLGA and remains quite stable for the BGA. The reasons behind the BGA stability are the special reproduction mecha-

Table 3.6 Influence of the population size

| Generation # | Population size | Fitness | # of rules | Learning time [min] |
|---|---|---|---|---|
| BGA | | | | |
| 100 | 25 | 76.89 % | 12.33 | 0.02 |
| 100 | 50 | 80.27 % | 12.00 | 0.03 |
| 100 | 100 | 80.52 % | 12.33 | 0.05 |
| 100 | 200 | 80.60 % | 13.67 | 0.09 |
| RBLGA | | | | |
| 100 | 25 | 93.12 % | 11.00 | 0.09 |
| 100 | 50 | 93.92 % | 20.33 | 0.21 |
| 100 | 100 | 95.08 % | 19.00 | 0.41 |
| 100 | 200 | 96.42 % | 23.33 | 0.89 |

nisms for the fuzzy rules (see 3.6.1.2). The LT increases along with the population size, which is very predictable, since the number of evaluated solutions increases. For both the BGA and the RBLGA, the optimal population size seems to be around 100 individuals, since when increasing from 100 to 200 individuals the accuracy improves by approximately 1.00% for the RBLGA and less than 0.20% for the BGA, while the LT doubles.

#### 3.6.1.5 Comparison and Discussion

The BGA and the RBLGA reacted differently throughout the different tests which were performed in this paper. Table 3.7 summarizes the ability comparison of both GAs, and the check mark $\surd$ gives an edge to one over the other. The RBLGA completely outperformed the BGA regarding fitness level, since the RBLGA was able to produce satisfactory solutions (FKBs) from a restricted set of evaluated

individuals, giving an edge to the RBLGA over the BGA.

For the simplicity level of the FKBs, the BGA gave a lower number of fuzzy rules, the difference wasn't sharp enough from what the RBLGA proposed, the GAs are almost identical on this point giving a slight edge to the BGA.

Considering the LT, for the same number of evaluated solutions, the BGA is faster. However as seen in section 3.6.1.3, in order to obtain a comparative accuracy level the BGA has to run longer and even then the BGA didn't achieve the results obtained by the RBLGA. Both GAs are efficient but an edge is given to the RBLGA since we consider the accuracy of the FKBs as the most important criterion to achieve (as we can see by the weight put on the accuracy in the evolution parameters). From these remarks we can conclude that even if the BGA remains quite

| | Fitness | Simplification | Learning time |
|---|---|---|---|
| BGA | | $\surd$ | |
| RBLGA | $\surd$ | $\surd$ | $\surd$ |

Table 3.7 Performances : RBLGA vs. BGA

efficient, the RBLGA was more convincing in the search for near optimal solutions (FKBs).

## 3.7 Conclusion

From the different tests, we can attest that generally both the binary coded genetic algorithm (BGA) and the real/binary like coded genetic algorithm (RBLGA) are efficient, when dealing with the artificial data. However, in most cases the RBLGA was more satisfactory.

Some principal conclusions can be stated:

- when dealing with a complex process to map into a fuzzy knowledge base, the RBLGA will be more effective than the BGA;
- for a simple process to model, the BGA can be a more appropriate choice since it runs faster along with providing satisfactory results;
- a population size of 100 is a good compromise between the accuracy level and learning time, for both GAs;
- if fast accurate responses are needed the use of the RBLGA is more appropriate, since it reaches high accuracy levels faster than the BGA;
- if a simple fuzzy knowledge base is needed, the BGA will be used, due to the efficiency of the rules reduction mechanisms;
- the evolution parameters can be changed for both GAs, if a new optimization

paradigm has to be set. Better accuracy can be achieved by increasing the initiating probability that governs the crossover mechanisms.

## Acknowledgment

Financial support from the Natural Sciences and Engineering Research Council of Canada under grants RGPIN-203618, RGPIN-105518 ans STPGP-269579 is gratefully acknowledged.

# REFERENCES

[1] Achiche, S., Balazinski, M., Baron, L. and Jemielniak, K., *Tool Wear Monitoring Using Genetically-Generated Fuzzy Knowledge Bases*, Engineering Applications of Artificial Intelligence, Vol. 15 pp. 303–314, Elsevier Science Ltd, 2002.

[2] Balazinski, M., Czogala, E., Jemielnial and K., Leski, J., *Tool Conditions Monitoring Using Artificial Intelligence Methods* Engineering Applications of Artificial Intelligence, Vol. 15 pp. 73–80, Elsevier Science Ltd, 2002.

[3] Balazinski, M., Achiche, S. and Baron, L., *Influence of Optimization and Selection Criteria on Genetically-Generated Fuzzy Knowledge Bases*, International Conference on Advanced manufacturing Technology, Johor Bahru, Malaysia, pp. 159–164, 2000.

[4] Balazinski, M., Bellerose and M., Czogala, E., *Application of Fuzzy Logic Techniques to the Selection of Cutting Parameters in Machining Processes*, International Journal for Fuzzy Sets and Systems, Vol. 61 FSS1157 North-Holland, pp. 307-317, 1993.

[5] Baron, L., Achiche, S. and Balazinski, *Fuzzy Decisions System Knowledge Base Generation Using a Genetic Algorithm*, International Journal of Approximate Reasoning, Vol. 28 pp. 125–148, 2001.

[6] Baron, L., *Genetic Algorithm for Line Extraction*, Technical report EPM/RT-98/06, École Polythechnique de Montréal, 19 pages, 1998.

[7] Cordón, O., Herrera, F. and Villar, P. *Analysis and Guidelines to Obtain a Good Uniform Fuzzy Partition Granularity for Fuzzy-rule Based Systems Using Simulated Annealing*, International Journal of Approximate Reasoning, pp. 187–216, 2000.

[8] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., *A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II*, IEEE Transactions on Evolutionary Computation, V. 6, pp. 182–200, 2000.

[9] Deb, K. and Agrawal, S., *Understanding Interactions among Genetic Algorithm Parameters*, In Foundations of Genetic Algorithms - 5, 1998.

[10] Diederich, J. and Renaud, F., *A Fuzzy Classifier using Genetic Algorithms for Biological Data*, Proceedings of the 8th International Conference of the North American Fuzzy Information Processing Society - NAFIPS - New York, BEE, pp. 680–684, 1999.

[11] Eshelman, L. J. and Schaffer, J. D., *Real- Coded Genetic Algorithms and Interval-Schemata, Foundations of Genetic Algorithms 2*, Morgan Kaufman Publishers, San Mateo, pp. 187–202, 1993.

[12] Goldberg, D. E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 412 pages, 1989.

[13] Surmann, H. and Selenschtschikow, A., *Automatic Generation of Fuzzy Logic Rule Bases: Examples I*, Proceeding of the First International ICSC Conference on Neuro-Fuzzy Technologies, pp. 75–81, 2002.

[14] Herrera, F., Lazano, M. and Verdegay, J.L., *Tunning Fuzzy Logic Controllers by Genetic Algorithms*, International Journal of Approximate Reasoning 12, pp. 299-315, 1995.

[15] Herrera, F. and Lozano, M., *Gradual Distributed Real-Coded Genetic Algorithms*, IEEE Transactions on Evolutionary Computation, Vol. 4, pp. 43–63, 2000.

[16] Lobo F. G., Goldberg D. E.and Pelikan M. *Time Complexity of Genetic Algorithms on Exponentially scaled Problems*, Proceedings of the Genetic and Evolutionary Computation Conference, pp. 151–158, 2000.

[17] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*, New York: Springer 1992.

[18] Ono, I., Kita, H. and Kobayashi, S., *A Real-Coded Genetic Algorithm Using the Unimodal Normal Distribution Crossover*, Natural Computing Series,

Advances in evolutionary computing: theory and applications, pp. 213–237, 2003.

[19] Syswerda, G., *Uniform Crossover in Genetic Algorithms*, Proceedings of the International Conference on Genetic Algorithms, pp. 2–9, 1989.

[20] Valenzuela-Rendon, M., *The Fuzzy Classifier System : A Classifier System for Continuously Varying Variables*, Proceedings of the 4th International Conference on Genetic Algorithms, pp. 346–353, 1991.

[21] Zadeh, L.A., *Outline of New Approach to the Analysis of Complex Systems and Decisions Processes*, IEEE Transactions of Systems, Man and Cybernetics, Vol. 3, pp. 28–44, 1973.

# CHAPTER 4

# MULTI-COMBINATIVE STRATEGY TO AVOID PREMATURE CONVERGENCE IN GENETICALLY-GENERATED FUZZY KNOWLEDGE BASES

Sofiane Achiche, Marek Balazinski et Luc Baron.
Département de génie mécanique, École polytechnique de montréal.
P.O. 6079, station Centre-Ville, Montréal, Québec, Canada, H3C 3A7.
sofiane.achiche@polymtl.ca

## 4.1 Abstract

A growing number of industrial fields are concerned by complex and multi-objective problems. For this kind of problems, optimal decision making is critical. Decision support systems using fuzzy logic are often used to deal with complex and large decision making problems. However the main drawback is the need of an expert to manually construct the knowledge base. The use of genetic algorithms proved to be an effective way to solve this problem. Genetic algorithms model life evolution strategy using the Darwin theory. A main problem in genetic algorithms is the premature convergence and the last enhancements in order to solve this problem include new multi-combinative reproduction techniques. There are two princi-

pal ways to perform multi-combinative reproduction within a genetic algorithm, namely the Multi-parent Recombination, Multiple Crossover on Multiple Parents (MCMP); and the Multiple Crossovers Per Couple (MCPC). Both techniques try to take the most of the genetic information contained in the parents.

This paper explores the possibility to decrease premature convergence in a real/binary like coded genetic algorithm (RBCGA) used in automatic generation of fuzzy knowledge bases (FKBs). The RBCGA uses several crossover mechanisms applied to the same couple of parents. The crossovers are also combined in different ways creating multiple offspring from the same parent genes. The large family concept and the variation of the crossovers should introduce diversity and variation in otherwise prematurely converged populations and hence, keeping the search process active.

*Keywords*: Artificial intelligence, Fuzzy decision support system, Fuzzy knowledge base, learning, premature convergence, genetic algorithm, crossover operators.

## 4.2 Introduction and problem definition

Fuzzy decision support systems use an approximate reasoning that emulates the human thinking process. The thinking is done through fuzzy knowledge bases

(FKB). The behavior of the FKB is easily understandable by a human being, since it is expressed through simple fuzzy sets and a set of linguistic rules. A FKB is generally constructed manually by an expert who translates his own knowledge of the task at hand to propose fuzzy sets—number, shape and repartition— and the appropriate fuzzy rules. The method used by the expert is based on a tedious process of trial and error approach, since fuzzy systems, unlike some other artificial intelligence methods, can't learn from data. This lack implied several researches to automatically generate the rule base of a FKB from the expert knowledge or a numerical set of data [9,10,14]. Other researches, concentrated their efforts on the automatic generation of fuzzy sets [23,27]. Other researches have focused on simultaneous generation of the fuzzy sets and the fuzzy rules [2,6,28]. To perform automatic generation of FKBs, Several optimization algorithms can be used, however with the presence of non derivable functions in the FKBs (inference engine ...etc), the presence of a high number of possible FKBs for each problem encountered and the number of fuzzy rules increasing exponentially with the number of fuzzy sets, genetic algorithms (GAs) appear to be the most promising learning tool. The GAs are powerful stochastic optimization methods presented first in the late 1960's and early 1970's by John Holland [18] and his students. In the mid-1980's these algorithms became more popular and were used in several fields such as machine learning [13]. GAs are considered here as an optimization tool for the automatic generation of FKBs. The GA used in this paper is a real/binary-like

coded genetic algorithm (RBCGA) developed by the authors [1]. The RBCGA allows one to solve a contradictory paradigm in term of FKB precision and simplicity (less fuzzy rules and less fuzzy sets) considering a randomly generated population of potential FKBs. The RBCGA is divided in two principal coding ways. First a real coded genetic algorithm (RCGA) that maps the fuzzy sets repartition and number into a set of real numbers, and second, a binary like coded genetic algorithm deals with the fuzzy rule base relationships (a set of integers is used). Although, GAs have proved to be effective in the automatic generation of FKBs [6], one of their drawbacks is the premature convergence problem (stagnation of the evolution). A GA is said asymptotically convergent if there is no change in the individuals from a generation to the next. At this state, all the individuals look almost alike. GAs can converge even thought the near optimal solution is not yet found, which means that the GA is unable to explore the remaining search space, namely the premature convergence. The premature convergence is generally due to the loss of diversity within the population. This loss can be caused by, the selection pressure, the schemata distribution due to crossover operators, and a poor evolution parameters setting [19,20]. Premature convergence is in some cases considered a complete failure of the GAs [12,13]. To avoid premature convergence, several researches have been performed, such as, dynamic niche sharing to efficiently identify and search multiple niches (peaks) in a multi-modal domain [8], the partition of the population in several subpopulations have been tried, this inducted the distributed genetic al-

gorithms [16]. Multi-combinative techniques have been also newly reported.

There is two principal ways to perform multi-combinative reproduction within a genetic algorithm. The first one being the Multi-parent Recombination, i.e., Multiple Crossover on Multiple Parents (MCMP) [11, 24–26]. The second one is the Multiple Crossovers Per Couple (MCPC). Both techniques try to take most of the genetic information contained in the parents.

This paper explores the possibility to decrease premature convergence in the RBCGA used for the automatic generation of FKBs by the mean of an MCPC strategy. The large family concept and the variation of crossovers should introduce diversity and variation in otherwise prematurely converged populations, and hence, keeping the search process active.

An overview of fuzzy logic based decision system, the FDSS software Fuzzy-Flou, developed at Ecole Polytechnique (Canada) and University of Silesia (Poland) [5] is first presented. This software is used for the validation tests. Then, it presents a description of the RBCGA used in this work, explaining the specificities of the reproduction and mutation mechanisms and their combination to ensure the MCPC strategy. Validation results are presented along with a comparative study of their performances evaluated through different evolution parameters. Finally, an appli-

cation on experimental data is discussed.

## 4.3 Fuzzy Decision Support Systems

A rule-based approach to the decision making using fuzzy logic techniques may consider imprecise vague language as a set of rules linking a finite number of conclusions. The knowledge base of such systems consists of two components: a linguistic terms base and a fuzzy rules base [5]. The former is divided into two parts: the fuzzy premises (or inputs) and the fuzzy conclusions (or outputs). In general, both can contain more than one premise and one conclusion. However, we limit ourself in this paper to systems of $N$ multiple inputs and one single output (MISO). Moreover, for the sake of simplicity, we consider only non-symmetric triangular fuzzy sets on the premises and sharp-symmetric triangular fuzzy sets on the conclusion. The representation of such imprecise knowledge by means of fuzzy linguistic terms makes it possible to carry out quantitative processing in the course of inference that is used for handling uncertain (imprecise) knowledge. This is often called approximate reasoning [29]. Such knowledge can be collected and delivered by a human expert (e.g. decision-maker, designer, process planer, machine operator). This knowledge, expressed by $(k = 1, 2, \cdots, K)$ finite heuristic fuzzy rules of the type MISO, may be written in the form:

$$R^k_{MISO} : \text{if } x_1 \text{ is } X^k_1 \text{ and } x_2 \text{ is } X^k_2 \text{ and } \cdots \text{ and } x_N \text{ is } X^k_N \text{ then } y \text{ is } Y^k, \qquad (4.1)$$

where $\{X_i^k\}_{i=1}^N$ denote values of linguistic variables $\{x_i\}_{i=1}^N$ (conditions) defined in the following universe of discourse $\{\mathbf{X}_i\}_{i=1}^N$; and $Y^k$ stands for the value of the independent linguistic variable $y$ (conclusion) in the universe of discourse $\mathbf{Y}$. The global relation aggregating all rules from $k = 1$ to $K$ is given as

$$R = also_{k=1}^K(R_{MISO}^k). \tag{4.2}$$

where the sentence connective *also* denotes any t- or s-norm (e.g., *min* ($\wedge$) or *max* ($\vee$) operators) or averages. For a given set of fuzzy inputs $\{X_i'\}_1^N$ (or observations), the fuzzy output $Y'$ (or conclusion) may be expressed symbolically as:

$$Y' = (X_1', X_2', \cdots, X_N') \circ R, \tag{4.3}$$

where $\circ$ denotes a compositional rule of inference (CRI), e.g., the *sup*-$\wedge$ or *sup-prod* (*sup*-$*$). Alternatively, the CRI of eq.(4.3) is easily computed as

$$Y' = X_N' \circ \cdots \circ (X_2' \circ (X_1' \circ R)). \tag{4.4}$$

In FDSS Fuzzy-flou, there are four variants of CRI: the sentence connective *also* can be either $\vee$ or *sum* ($\sum$); the compositional operator is the *supremum* (*sup*) of either $\wedge$ or $*$, denoted *sup*$\wedge$ and *sup*$*$; while the sentence connective *and* and the fuzzy relation are always identical to the second part of the latter. For the sake of

brevity, all four variants of CRI—i.e.: $\vee$-*sup*$\wedge$-$\wedge$-$\wedge$; $\vee$-*sup*$*$-$*$-$*$; $\sum$-*sup*$\wedge$-$\wedge$-$\wedge$; and $\sum$-*sup*$*$-$*$-$*$—are expressed as

$$Y' = \left\{ \begin{array}{c} \vee_{k=1}^{K} \\ \\ \sum_{k=1}^{K} \end{array} \right\} \quad \begin{array}{c} sup \\ \{x_i \in X_i\}_{i=1}^{N} \end{array} \quad *_t \left( *_t(X'_N, \cdots, X'_2, X'_1) \ , \ *_t(X_1^k, X_2^k, \cdots, X_N^k, Y^k) \right), \tag{4.5}$$

where $*_t(\cdot)$ denotes the t-norm of $(\cdot)$ defined as either $\wedge$ or $*$. These variants of CRI mechanisms allow us to obtain different conclusions represented as the membership function $Y'$. In FDSS Fuzzy-Flou, there are three defuzzification methods: the center of gravity (COG); the mean of maxima (MOM); and the height method (HM). All the results presented in this paper are obtained with the $\sum$-*sup*$*$-$*$-$*$ CRI and COG as defuzzification. Figure 4.1 shows a screen printout of the premises and conclusion (on the right), the fuzzy rules and settings (on the left) of the FDSS Fuzzy-Flou software.

## 4.4 Real/Binary like Coded Genetic Algorithm

The most important success of the GAs remains in their evolution basis rather than the coding, hence the occurrence of real coded genetic algorithms (RCGA) in recent years, they were, at first, mostly used in numerical optimization works [17]. The use of RCGAs overcomes some of the weaknesses of Binary Coded Genetic Algorithms (BCGAs), such as the low resolution of the solutions [6]. Moreover,

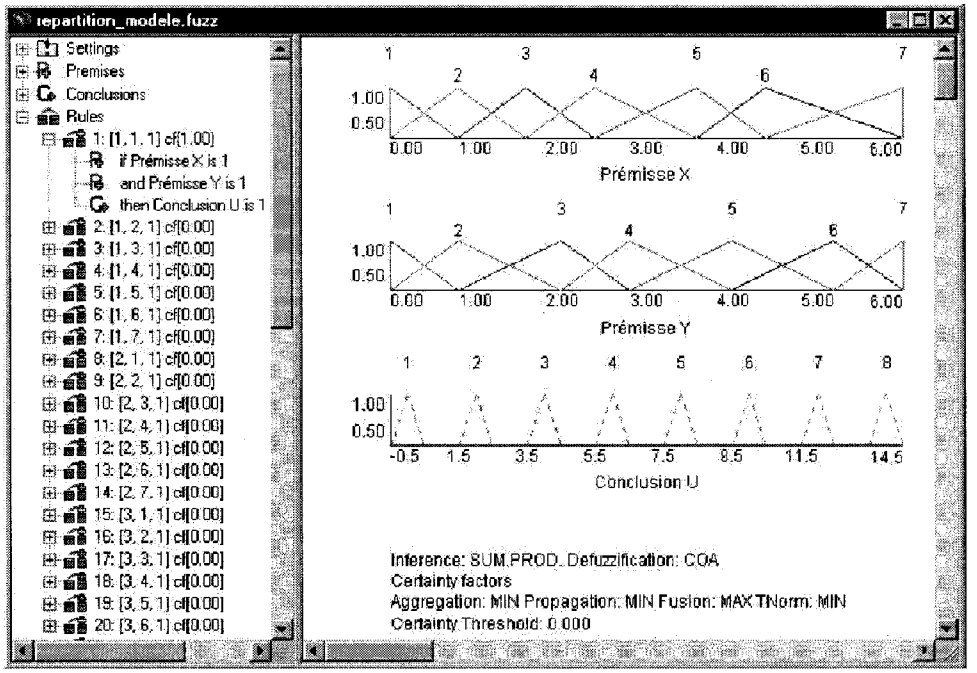


Figure 4.1 Screen shot of the FDSS Fuzzy-Flou

most optimization works are performed in a continuous mathematical space (real space). However, since the FKBs contains two cooperative but distinct parts in terms of:

- premises, conclusions, along with the fuzzy sets on them;
- fuzzy rules base.

They are distinct in a way that the first one deals with real numbers while the second one uses integer numbers, since the fuzzy rules are simple pointers to the index of a fuzzy set on the conclusion. For this matter, we used a combination between an RCGA and a BCGA, where the binary part is mapped into a string of integers, the algorithm is therfore called Real/Binary-like Coded Genetic Algorithm

(RBLGA).

### 4.4.1 Coding

The *genotype* corresponds to several independent sets of reals and a set of integers. Here, the *genotype* can be described as follow:

#### Premises and Conclusion

There is as many real number sets as there is premises in the problem and one set for the conclusion. Each set contains a predefined maximum number of real numbers representing the location of the summit of each fuzzy set on each premise and the conclusion. The two summits located at the minimum and maximum limits of each premise and conclusion are not coded. We consider non-symmetrical-overlapping triangular fuzzy sets for premises and symmetrical triangular fuzzy sets for the conclusion.

#### Fuzzy Rules

The fuzzy rules are coded as a set of integers representing an ordered list of the combination of the premises. Each integer in the set representing a conclusion fuzzy set summit. The initial population of FKBs is composed of $P$ FKBs randomly generated. The *genotype* of each new solution contains all the sets mentioned above, however as we will explain below, the size of the sets can decrease throughout the evolution, but can't increase.

#### 4.4.1.1 Reproduction Mechanisms of the RBCGA

Reproduction is performed by *crossover* of the *genotype* of the parents to obtain the *genotype* of an offspring (or two offsprings). The reproduction of the FKBs in the RBCGA is preformed through three principal crossover mechanisms, each one has its own purpose, as explained below.

**a) Multi Crossover**

The multi-crossover is a combination of crossovers applied on different parts of the *genotype.*

**a.1) Premises/Conclusion Crossover**

For this part of the FKB, three crossover mechanisms are used concurently as explained latter in the paper. For the three reproduction mechanisms, if $A = \{x_1,\ x_2,\ \cdots,\ x_i,\ \cdots,\ x_n\}$ and $B = \{y_1,\ y_2,\ \cdots,\ y_i,\ \cdots,\ y_n\}$ represents the two selected parents, $C$ the offspring obtained by the crossover of $A$ and $B$ then $C = \{z_1,\ z_2,\ \cdots,\ z_i,\ \cdots,\ z_n\}$, where $z_i$ are selected according to the used crossover. The values of the offspring can belong to two main intervals, i.e., the exploitation or the exploration zone. Exploitation means using the interval between the values of the two parents, while the exploration uses an interval outside of these two limits, therefore trying new solutions (see Fig .4.2), knowing that:

- in the exploitation zone, the offspring inherits behaviors close to those of his parent's average, since he was produced between them;

- in the exploration zone, the offspring is a result of an exploration, his attributes are away from his parent's average.

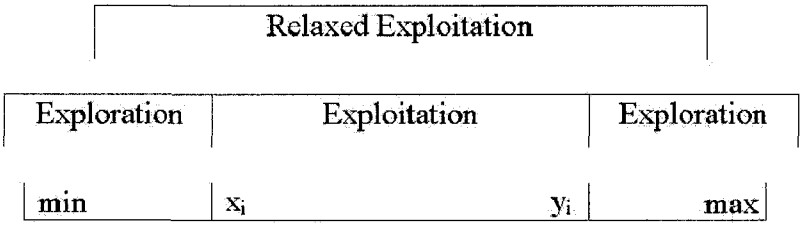


Figure 4.2 Interval action of an offspring gene

### a.1.1) Blended Crossover $\alpha$

The blended crossover $\alpha$ is denoted as $BLX\text{-}\alpha$ [15], where $\alpha$ controls the exploitation/exploration level of the offspring obtained from the selected parents. As shown in Fig. 4.3, the $z_i$ values of the offspring are randomly selected in the interval $[min, max]$ where:

- $max$ = maximum $\{x_i, y_i\}$ - I $\alpha$;
- $min$ = minimum $\{x_i, y_i\}$ + I $\alpha$;
- I = maximum $\{x_i, y_i\}$ - minimum $\{x_i, y_i\}$.

In order to avoid any bias in either direction (exploitation or exploration) the value of $\alpha$ is set to 0.5, which provides offspring in the relaxed exploitation zone.


Figure 4.3 Blended crossover $\alpha$ – BLX-$\alpha$

### a.1.2) Non-uniform Arithmetical Crossover

The non-uniform Arithmetical crossover (non-uniform arithmetical crossover) [21] is denoted as *NAX*. The $z_i$ values of the offspring are computed as follow:

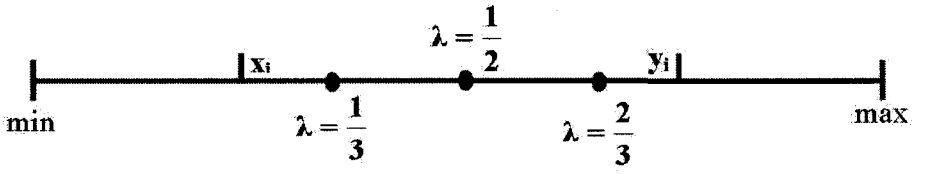


Figure 4.4 Arithmetical crossover *NAX*

- $z_i = \lambda\, x_i + (1-\lambda) y_i$ or $z_i = \lambda\, y_i + (1-\lambda) x_i$.

The $z_i$ values are set to one of these two values at an equal probability (50% each). The value of $\lambda$ is randomely selected at each generation in the interval $[0, 1]$, making the non-uniform part of the mechanism. The *min*, *max* values are the limits, as shown in Fig. 4.4.

### a.1.3) Extended Line Crossover

The extended line crossover [22] is denoted as *ELX*. The $z_i$ values of the offspring are computed as follow:

- $z_i = \beta\, x_i + \beta(y_i - x_i)$.


Figure 4.5 Extended Line Crossover $ELX$

The value of $\beta$ is randomly chosen within the interval $[-0.25, 1.25]$. Again, the $min$, $max$ values are the limits as shown in Fig. 4.5.

**a.2) Fuzzy Rules Crossover**

Since the part of the *genotype* representing the fuzzy rule base is composed of integer numbers, the crossover on this part of the *genotype* is done by a *simple crossover.* The use of a BLX/NAX/$ELX$ crossovers is not suitable in this case, because of the integer nature of the values. The operation is performed by inverting the end part of the *sets* of the parents at a randomly selected *crossover* site as shown in fig 4.6. These two mechanisms are governed by an initiating probability $pr_1$.

**b) Fuzzy Set Reducer**

This mechanism is set to increase the simplicity of the FKBs by selecting on each premise a summit and erasing it from the respective sets representing the premises. The mechanism can be described as follows:

- the set given by $premise_i = \{summit_1, \cdots, summit_i, \cdots, summit_n\}$ is selected;

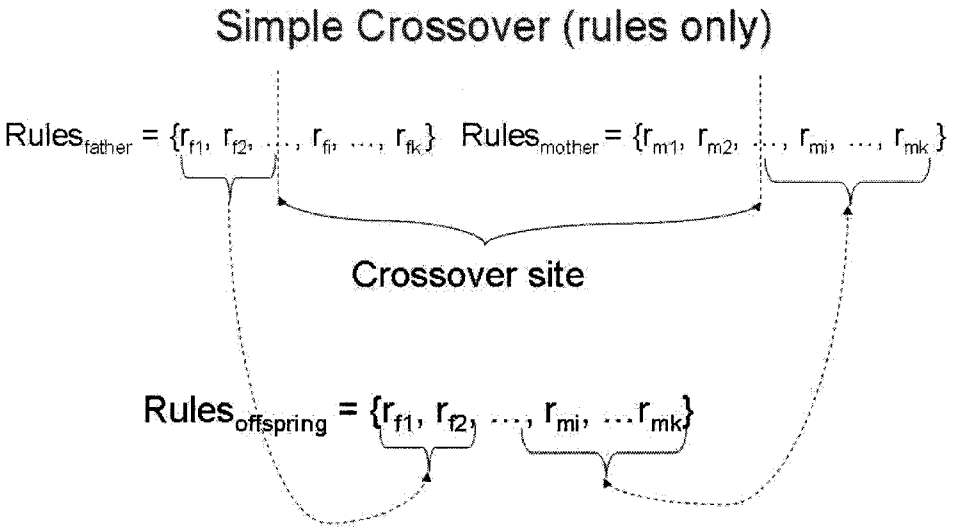


Figure 4.6 Simple crossover of the RBLGA

- one of the summits is randomly selected and erased;
- the operation is repeated for each premise.

Decreasing the number of fuzzy sets reduces the maximum number of fuzzy rules. This mechanism allows one to obtain different and more simple solutions (FKBs). This mechanism is governed by an initiating probability $pr_2$.

### c) Mutation

*Mutation* is the creation of a new individual by altering the gene of an existing one. The probability $pr_3$ governs the occurrence of this mechanism. This paper uses a uniform mutation [21] applied to one randomly selected individual, as follows:

- an individual $C$ is randomly selected, $C = \{z_1, z_2, \cdots, z_i, \cdots, z_n\}$;
- a mutation site "$i$" is randomly chosen in the interval [1,n];

- $z_i$ is changed to $z'_i = \text{random}(a_i, b_i)$, where $a_i$ and $b_i$ are the limits enclosing $z_i$.

**d) Selection Mechanism**

The selection mechanism used in this paper (to select the parents) is performed as follows:

- a real value $"R"$, is randomly generated in the interval [0 1];
- $R$ is multiplied by $S$, $S$ being the fitness values sum of the individuals of the population. The value $RS$ is obtained;
- beginning by the best individual of the population, the fitness values are summed till the result is higher than $RS$. The last added individual is considered as a potential parent;
- the same mechanism is used to select the second parent.

## 4.5 Performance Criteria

The performance criteria allows one to compute the rating of each FKB. This performance rating is used by the RBCGA in order to perform the natural selection. Here, the performance criteria is the accuracy level of a FKB (approximation error) in reproducing the outputs of the learning data. The approximation error is

measured using the root-mean-square error:

$$\Delta_{rms} = \sqrt{\sum_{i=1}^{N} \frac{(RBCGA_{output} - data_{output})^2}{N}} \tag{4.6}$$

where $N$ represents the number of learning data. The fitness value is evaluated as a percentage of the output length ($L = z_{max} - z_{min}$) of the conclusion, i.e., $\phi_{rms} = \frac{L-\Delta_{rms}}{L} \times 100$.

### 4.5.1 Natural selection (elitist approach)

Natural selection is performed on the population by keeping the *most* promising individuals based on their fitness value (elitist approach). This is equivalent to using solutions that are closest to the optimum. For convenience, we keep the size of the population constant.

The first generation begins with $P$ FKBs, and the same number of additional FKBs are generated by reproduction and mutation. Moreover, in the RBLGA *natural selection* is applied on the $2 \times P$ FKBs by ordering them according to the performance criterion $\phi_{rms}$ and keeping the $P$ first FKBs. The size $P$ has to be selected depending upon the performances of the computer in use. A high value of $P$ generally ensures a better diversity in the population, which helps to avoid premature convergence. However it increases the learning time.

## 4.6 Evolutionary Strategy

To apply the MCPC strategy to the BCGA, the crossover mechanisms applied to the genotype are combined in different ways to create multiple different offspring from the same parent's genes. A critical point in overcoming premature convergence is the identification of why it occurs and when it does occur, along with a good balance between exploitation and exploration throughout the evolution. It is widely established that a loss of population diversity leads to premature convergence. The MCPC strategy is used on the reproduction mechanisms. Although we can use recombination strategy on all the parts of the FKB, the quality of the FKB is more sensitive to the quality of the repartition and the number of fuzzy sets [7], and hence, the evolutionary strategy is only applied to the premises. The *Fuzzy Set Reducer* mechanism is not used in the MCPC strategy.

The MCPC strategy will be applied as follow:

- **Single application**: each crossover mechanism is applied independently (BLX-$\alpha$, NAX and $ELX$), the results are used as a comparison basis;
- **Linear combination**: a linear combination of the three mechanisms is used;
- **Simultaneous application**: all three mechanisms are used simultaneously in the evolution;

- **Exploration balance mechanism**: a switch of the reproduction mechanism is set based on some criteria, this mechanism being divided into two main parts.

### 4.6.1 Test Functions

The learning performances of the RBCGA are investigated using three examples of known behavior in term of 3D surfaces of the type $z = f(x, y)$, where the nodes are the learning set of sampled data. We have used three different surfaces of different complexities to have a better idea on the generality of the results. The evolution and selection criteria are set to the following values:

- $pr_1 = 100.0\%$;
- $pr_2 = 0.0\%$;
- $pr_3 = 5.0\%$;
- Maximal complexity: 4 fuzzy sets (including the limits) on each premise and 8 on the conclusion (no limits on the number, it can match the number of fuzzy rules). These numbers were chosen based on performance tests applied on the three surfaces.

The evolution is completely governed by the *multi-crossover reproduction* mechanism, the *fuzzy set reducer* is disabled to allow a more comparable FKBs, otherwise

the number of fuzzy rules can vary excessively. However, the number of fuzzy sets can decrease when two summits are overlapping, hence, the reducing of the fuzzy rule base. The theoretical surfaces are the following:

**1. Sinusoid surface**

The sinusoid surface is defined as

$$z = sin(x\ y) \text{ with } \begin{array}{c} 0 \leq x \leq 1.6 \\ 0 \leq y \leq 1.4 \end{array}, \tag{4.7}$$

**2. Spherical surface**

The spherical surface is defined as

$$z = x^2 + y^2 \text{ with } \begin{array}{c} -2.0 \leq x \leq 2 \\ -2 \leq y \leq 2 \end{array}, \tag{4.8}$$

**3. Hyper-tangent surface**

The hyper-tangent surface is defined as

$$z = thanh(x\ (x^2 + y^2)) \text{ with } \begin{array}{c} -0.2 \leq x \leq 1.4 \\ -0.2 \leq y \leq 1.4 \end{array}, \tag{4.9}$$

In order to measure the fitness (accuracy levels) of the RBCGA in generating FKBs, and for the sake of comparison, several runs have been made on the three

surfaces. The population size $P$ was set to 20 individuals. Runs were performed three times for 10, 50, 100, 500, 1 000, 2 500 and 5 000 generations. The fitness of the best individual, the fitness of the worst individual and the average fitness of all the individuals are taken into account at the last generation for each surface. The average value of the the three different results obtained for each theoretical surface was computed. The runs are performed using the different versions of the RBCGAs (different crossover mechanisms).

#### 4.6.1.1 Test Functions : Single applications

The multi-crossover of the RBCGA uses the $BLX - 0.5$, the $NAX$ and $ELX$, independently. At the end of the evolution the averages of the results obtained for the three surfaces (each test being performed three times) are computed.

Tables 4.1 to 4.7 show the fitness for the best individual, the worst individual in the population, the average fitness of the last population along with the size of the fuzzy rules base corresponding to the crossover mechanism applied.

**1. Blended Crossover BLX-0.5**

As shown in Table 4.1, the highest value obtained is 91.63% after 5 000 generations. However after only 500 generations the fitness is already up to 91.35%. The worst fitness get closer to the best individual with the improvement of the populations, same goes for the mean value, which shows a lack of diversity in the population.

A drawback very difficult to overcome since it is a direct consequence of the elitist philosophy.

Figure 4.7 shows the fitness evolution of the best individual, the fitness reaches a

Table 4.1 Average $\phi_{rms}$ obtained by $BLX - 0.5$

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 82.76% | 77.88% | 79.64% | 16 |
| 50 | 90.68% | 90.48% | 90.54% | 12.67 |
| 100 | 91.01% | 91.01% | 91.01% | 12.67 |
| 500 | 91.35% | 91.17% | 91.18% | 12.67 |
| 1000 | 91.35% | 91.18% | 91.21% | 12.67 |
| 2500 | 91.60% | 91.36% | 91.48% | 12.67 |
| 5000 | 91.63% | 91.54% | 91.59% | 12.67 |

plateau around 92.00%. The number of rules is decreased to 13

**2. Non-uniform Arithmetical Crossover**

In Table 4.2, the highest value obtained is 92.00% after 5 000 generations. The fitness gained around 1.00% from 500 generations to 5 000, which is not negligible but still quite low. The worst fitness is very close to the best one, same for the

Table 4.2 Average $\phi_{rms}$ obtained by $NAX$

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 82.85% | 79.63% | 80.68% | 16.00 |
| 50 | 89.17% | 89.16% | 89.16% | 16.00 |
| 100 | 89.45% | 89.45% | 89.45% | 16.00 |
| 500 | 91.09% | 91.04% | 91.05% | 16.00 |
| 1000 | 91.45% | 91.45% | 91.45% | 16.00 |
| 2500 | 91.83% | 91.72% | 91.73% | 16.00 |
| 5000 | 92.00% | 91.98% | 91.98% | 16.00 |

mean value which can be interpreted as a presence of too many alike individuals

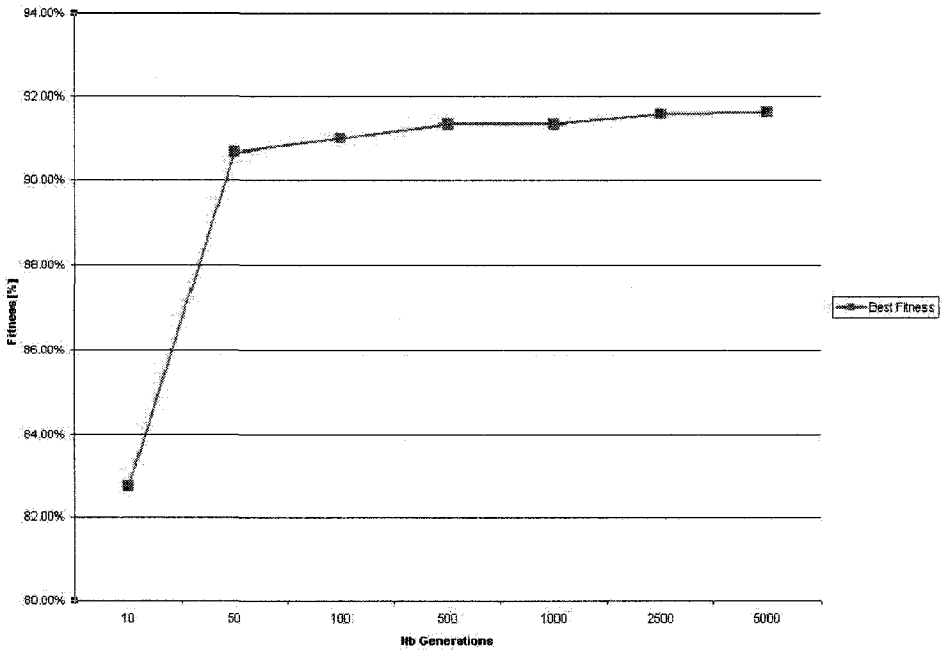


Figure 4.7 Accuracy level for the best individuals using $BLX - 0.5$

in the population. This is reached after 50 generations only. The number of rules stays at 16, which means that no simplification occurred during the evolution. Figure 4.8 shows the fitness evolution of the best individual, the fitness reaches again a plateau around 92.00%. The evolution is slower than for the $BLX - 0.5$, even if the best values are still very comparable (91.63% vs 92.00%).

### 3. Extended-Line Crossover

From Table 4.3, the highest fitness value is 92.78% reached after 5 000 generations. After 100 generations the best fitness reached is 91.10%. The worst and mean fitness values are very close to the best individual from the $50th$ generation.

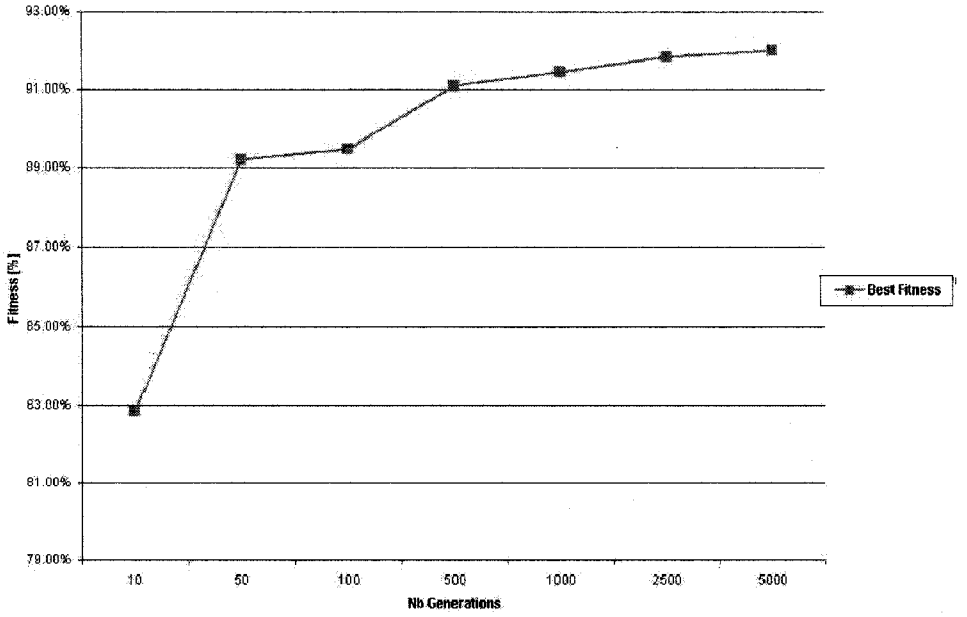


Figure 4.8 Accuracy level for the best individual using $NAX$ strategy

The $ELX$ reached the best fitness compared to $BLX - 0.5$ and $NAX$. From

Table 4.3 Average $\phi_{rms}$ obtained by $ELX$

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 82.63% | 79.39% | 80.37% | 16.00 |
| 50 | 89.53% | 89.46% | 89.49% | 13.67 |
| 100 | 91.10% | 91.10% | 91.10% | 13.67 |
| 500 | 91.56% | 91.56% | 91.56% | 13.67 |
| 1000 | 91.92% | 91.92% | 91.92% | 13.67 |
| 2500 | 92.55% | 91.55% | 91.55% | 13.67 |
| 5000 | 92.78% | 91.77% | 91.77% | 13.67 |

Fig. 4.9, we can see that the fitness evolution of the best individual is better distributed through the generations, however the slow down at the end is still present, where a plateau seems to be reached.

Using the single crossover applications, we have noticed three main negative points:

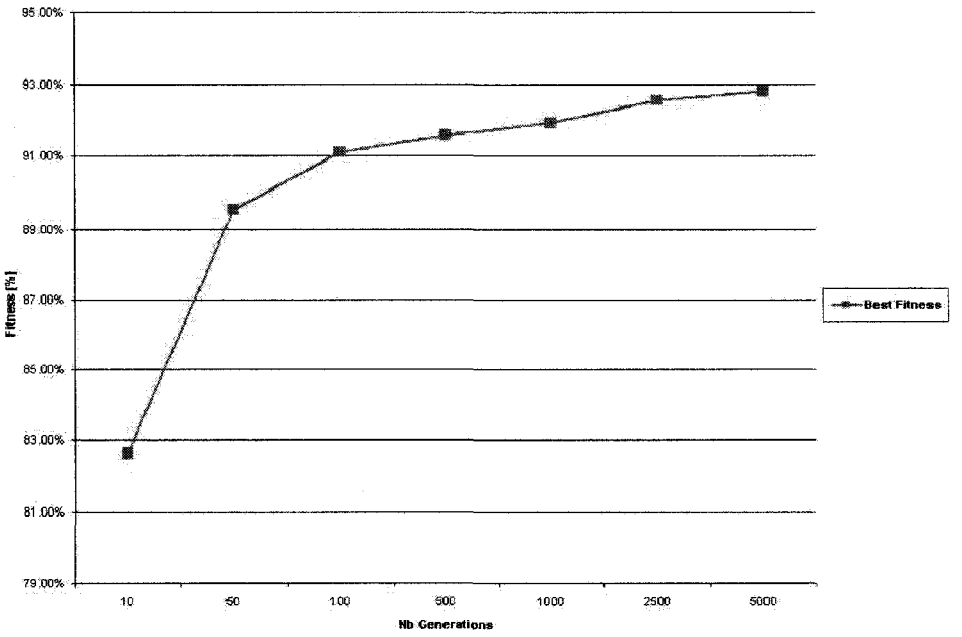


Figure 4.9 Fitness the best individual using $ELX$

1. Fast convergence (premature) : after 50 generations the evolutions slowed down noticeably;

2. Lack of diversity : the worst fitness value gets too close to the best fitness value too fast;

3. Reaching a plateau : a plateau is reached around the fitness value of 92.00%, a plateau being a period in the learning process in which minimum or no progress takes place.

To try to overcome the three drawbacks above, we applied the strategies presented in section 4.6.

#### 4.6.1.2 Test Functions : Linear Combination

The linear Combination is denoted as $LC$. The crossover mechanism is a linear combination of the $BLX-0.5$, the $NAX$ and the $ELX$. From an existing population $P$ and new population of offspring is created, namely $P'$. However, one third of the offspring are generated using $BLX-0.5$, an other one third is generated using $NAX$ and the last third is generated using $ELX$.

The assumption is that the use of different mechanisms should create more diversity and hence decrease the premature convergence and its consequences. Moreover, it is noteworthy to say that in the $LC$, the three mechanisms don't work in competition, but rather in a symbiotic way, since all the offspring created are part of the evolution. The results obtained are presented in Table 4.4.

We can see that the $\approx$ 92.00% plateau is exeeded. The $LC$ reached a fitness value

Table 4.4 Average $\phi_{rms}$ obtained by $LC$

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 81.79% | 78.92% | 79.87% | 14.67 |
| 50 | 89.55% | 89.47% | 89.49% | 13.33 |
| 100 | 89.97% | 89.47% | 89.56% | 13.33 |
| 500 | 91.61% | 91.61% | 91.61% | 13.33 |
| 1000 | 91.82% | 91.77% | 91.77% | 13.33 |
| 2500 | 92.54% | 91.42% | 91.43% | 13.33 |
| 5000 | 93.16% | 93.05% | 93.07% | 13.33 |

of 93.16% after 5 000 generations. From Fig. 4.10, we can see that the speed of convergence decreased (the curve is less abrupt). The plateau, noticed in the single

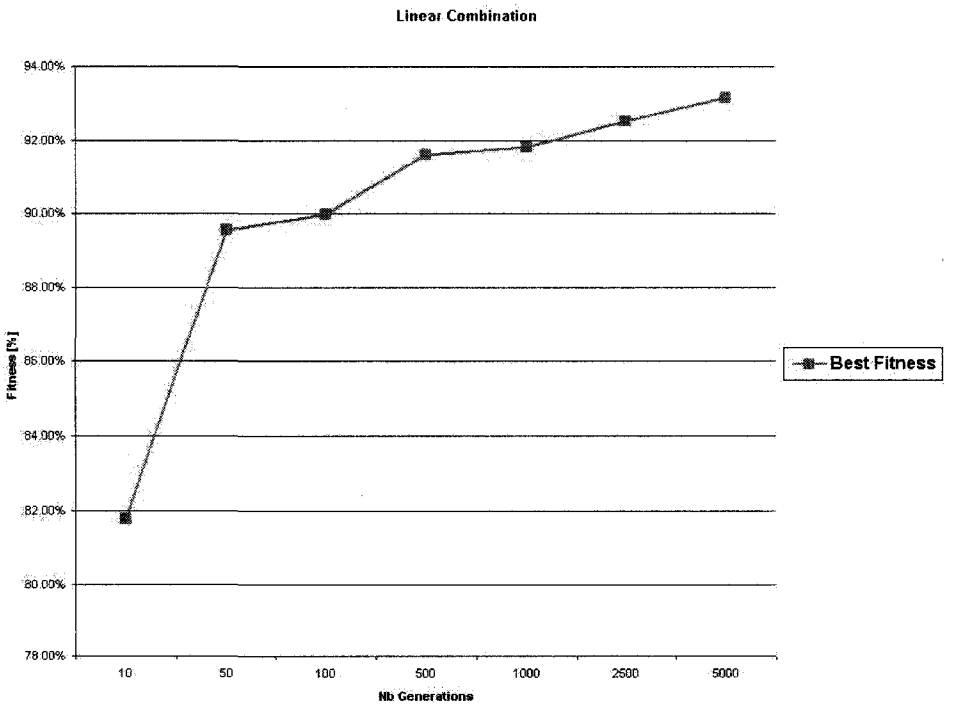


Figure 4.10 Accuracy level for the best individual using $LC$ strategy

mechanisms, is less acute here. In the last stages of the evolution (1 000 generations and more) the improvement is continuing for the best individuals (see Fig. 4.10). The size of the rule base decreased also, since it went from 16 to 13 fuzzy rules, which is equivalent to the best result obtained by the single applications.

#### 4.6.1.3 Test Functions : Simultaneous Application Strategy

The Simultaneous Application Strategy is denoted as $SA$. The crossover mechanism is a simultaneous application of the $BLX - 0.5$, the $NAX$ and the $ELX$. From an existing population $P$ a new population of offspring is created, namely $P'$. However, for each pair of selected parents, 6 children are generated, two of them

with $BLX-0.5$, two with the $NAX$ and the last two with the $ELX$. The different mechanisms shall create diversity and hence decrease the premature convergence and its consequences. However, in the $SA$ strategy the crossover mechanisms are used competitively, in a sense that, only the best offsprings are selected to be a part of the evolution. Table 4.5 shows that the $\approx$ 92.00% plateau is also exeeded.

Table 4.5 Average $\phi_{rms}$ obtained by $SA$

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 83.13% | 78.50% | 79.74% | 13.67 |
| 50 | 89.75% | 89.40% | 89.51% | 13.67 |
| 100 | 90.84% | 90.77% | 90.79% | 13.67 |
| 500 | 92.93% | 92.92% | 92.92% | 13.67 |
| 1000 | 93.23% | 93.23% | 93.23% | 13.67 |
| 2500 | 93.46% | 93.36% | 93.40% | 13.67 |
| 5000 | 93.61% | 93.40% | 93.43% | 13.67 |

The $SA$ reached 93.61% after 5 000 generations. From Fig. 4.11, we can see that the speed of convergence decreased if compared to the single evolution strategies (Fig. 4.7, 4.8 and 4.9). The curve is less abrupt from $50^{th}$ generations. The plateau noticed in the single mechanisms disappeared. After the 1 $000^{th}$ generation the improvement is slowing down but still active (see Fig. 4.11). The $SA$ strategy gave very similar results to the ones obtained by the $LC$ strategy (with a small improvement in the fitness), and the curves are also very alike. The size of the rule base decreased, since it went from a maximum of 16 to around 13 fuzzy rules, which is again equivalent to the results obtained by the best single application ($\approx$ 13 fuzzy rules) and the $LC$ strategy ($\approx$ 13 fuzzy rules).

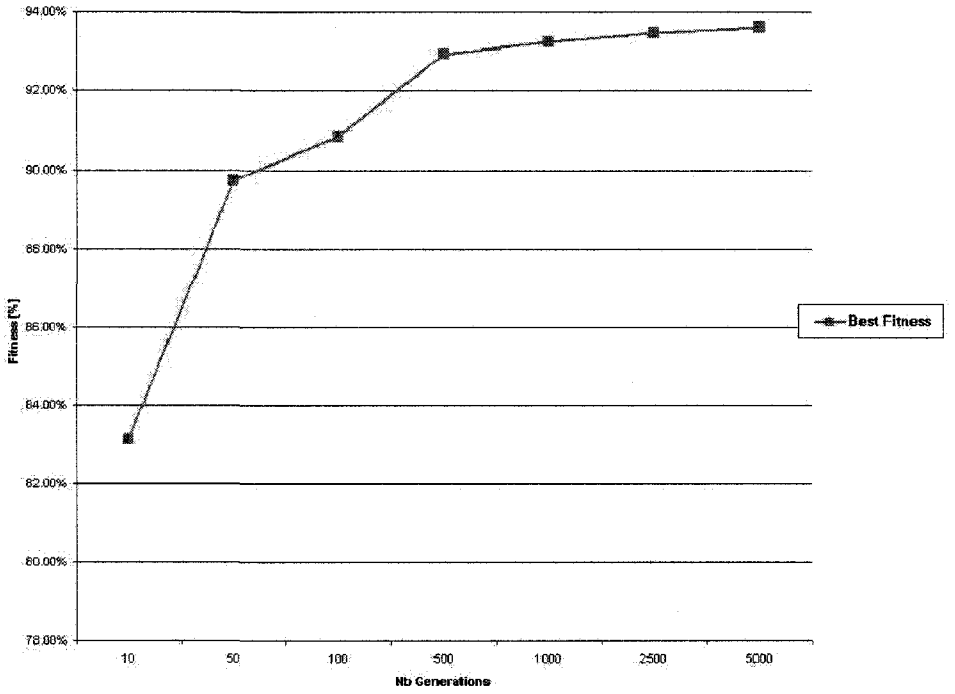


Figure 4.11 Accuracy level for the best individual using $SA$ strategy

#### 4.6.1.4 Test Functions : Exploration/Exploitation Balance Strategy

The exploration/exploitation balance strategy is denoted as EEB. As mentioned earlier in the paper, the two main reasons that cause premature convergence are the loss of diversity in the population and the bad balance between exploration and exploitation throughout the evolution process. In this section, we study the influence of the exploration/exploitation balance on the premature convergence aiming to find which combination is the best as follows:

1. Exploiting the individuals at the early stages of the evolution, applying re-

laxed exploitation through the main part of the evolution and then exploring the individuals at the late stages;

2. Exploring at the early stages of the evolution, applying relaxed exploitation through the main part of the evolution and then exploiting at the late stages of the evolution.

The first combination is called $EEB1$, while the second is called $EEB2$. Both mechanisms use the $BLX-\alpha$, since it is quite easy to control the exploration/exploitation balance through the variation of $\alpha$. The different values given to $\alpha$ influence the exploration, exploitation or relaxed exploitation levels of the crossover mechanism. The three values are set as follow:

- Exploration: $\alpha = 1.00$ for total exploration;
- Relaxed Exploitation: $\alpha = 0.50$;
- Exploitation: $\alpha = 0.1$ for close to maximal exploitation—$\alpha \neq 0.00$, in order to make a difference between the BLX-$\alpha$ and the uniform mutation—.

The question we should ask ourselves is: At which stage of the evolution process these three mechanisms should occur?. The stage of the evolution is defined by the generation number. However, what can be considered an early stage of the evolution? The 10 first generations? How if the maximal number generations is set

to 10? For this matter we assumed the following, considering a maximal number of generations $N$:

- The first quarter (1/4) of $N$ is considered being the early stages;
- The last quarter of the $N$ is considered being the last stages;
- The remaining part (from the end of the first quarter to the beginning of the last one, limits non included) is considered the evolution stage.

In the next sections we present the results obtained for both $EBB_1$ and $EBB_2$ strategies.

**1. Exploitation / Relaxed Exploitation / Exploration** $EEB1$

For $EBB_1$, $\alpha$ is set to 0.10 at the early stages, then changes to 0.50 for the relaxed exploitation stage and finally switches to 1.00 for the last stages of evolution. The results obtained by using $EBB_1$ are reported in Table 4.6. The first noticeable

Table 4.6 Average $\phi_{rms}$ obtained by EEB1

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 84.19% | 78.80% | 80.81% | 16 |
| 50 | 88.71% | 87.49% | 88.03% | 16 |
| 100 | 88.84% | 88.47% | 88.58% | 16 |
| 500 | 90.72% | 90.71% | 90.72% | 16 |
| 1000 | 91.61% | 91.28% | 91.52% | 16 |
| 2500 | 90.97% | 90.90% | 90.93% | 14.67 |
| 5000 | 90.42% | 90.42% | 90.42% | 14.67 |

change is that the best fitness reached isn't for the 5 000 generations test but for the 1 000 generations one. It is the lowest fitness obtained until now (for the best

individual) by the different strategies. From Fig. 4.12, we can clearly see that unlike

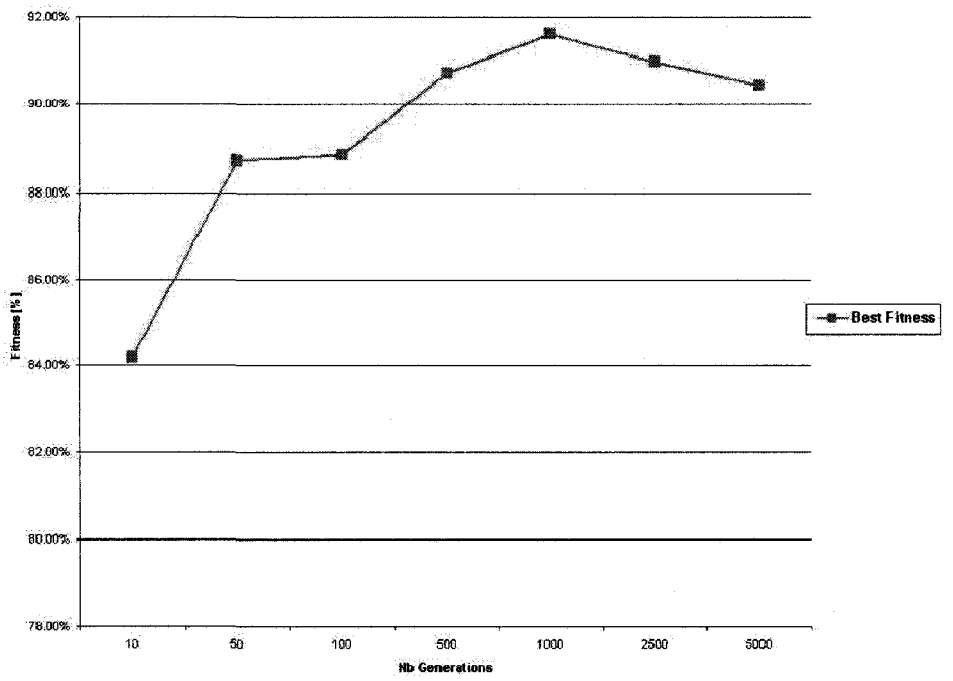


Figure 4.12 Accuracy level for the best individual using EEB1 strategy

for the other tests the fitness level is not proportional to the number of generations. This is due to the fact that the number of exploited and/or explored individuals is different from a test to another. For example a run using 1 000 generations: during 250 generations the new breed is obtained using mostly exploitation, while a run of 5 000 generations, the exploitation is performed during 1 250 generations. A good balance has to be found to obtain the best results with this mechanism, which is not a simple task to do. However, even thought the fitness level is not as high as the other tests, it remains that the evolution of the convergence speed is

quite interesting, since we don't have a plateau as shown in Fig. 4.12.

The number of fuzzy rules is around 15 (see Table 4.6) which is higher than the one from the previous tests, i.e., the FKBs obtained are less simple.

**2. Exploration / Relaxed Exploitation / Exploitation $EEB2$**

For $EBB_2$, $\alpha$ is set to 1.00 at the early stages, then changes to 0.50 for the relaxed exploitation stage and finally switches to 0.10 for the last stages of evolution. The results obtained by using $EBB_2$ are reported in Table 4.7.

The best fitness level reached is for the 5 000 generations test with 93.30%. From

Table 4.7 Average $\phi_{rms}$ obtained by EEB2

| Generation # | Best individual | Worst individual | Mean Fitness | # rules |
|---|---|---|---|---|
| 10 | 81.92% | 79.06% | 80.40% | 12.33 |
| 50 | 91.59% | 91.57% | 91.58% | 9.00 |
| 100 | 92.89% | 92.85% | 92.85% | 9.67 |
| 500 | 93.29% | 93.20% | 92.24% | 9.67 |
| 1000 | 93.29% | 93.23% | 93.24% | 9.67 |
| 2500 | 93.29% | 93.23% | 93.25% | 9.67 |
| 5000 | 93.30% | 93.25% | 93.26% | 9.67 |

the 500 generations test the evolution is stagnating. However, the fitness level achieved exceeded the plateau of the single applications ($\approx$ 92.00%). Figure 4.13 illustrates the fitness evolution of the best individuals, the fitness reaches a plateau around 93.30%. The evolution seems as fast as for the single applications, even if the best values are still higher ($\approx$ 93.00% vs $\approx$ 92.00%). In the $EBB_2$, the

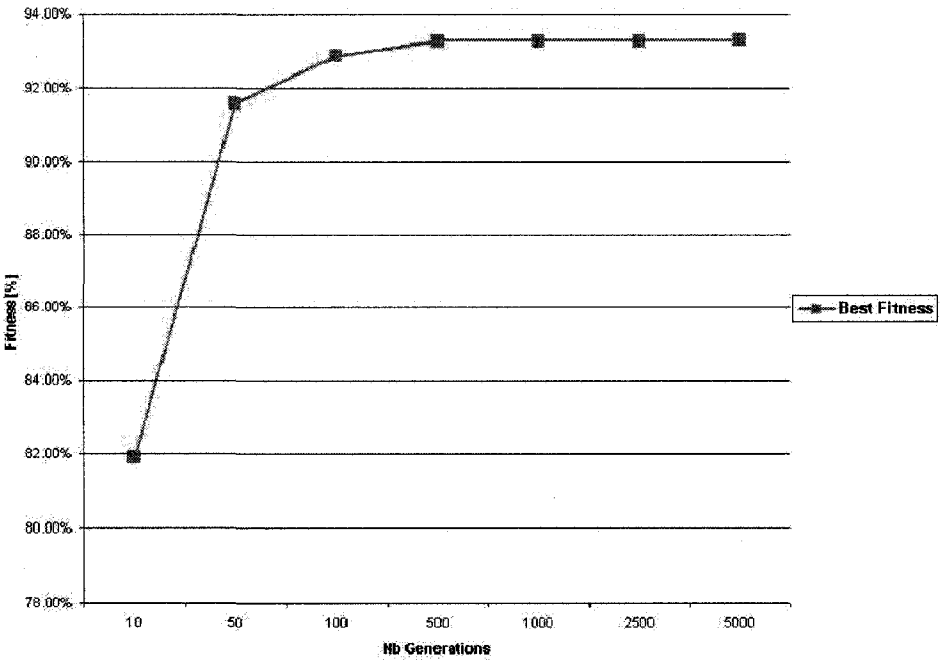


Figure 4.13 Accuracy level for the best individual using EEB2 strategy

problem of the balance between the exploration/exploitation levels and the number of generations didn't occur. The $EBB_2$ seems to be a more natural way to explore solutions, since at the early stages it is better to explore the search space, before the diversity starts to drop. The number of fuzzy rules is around 10 (see Table 4.7) which is the lowest of any other strategies.

### 4.6.2 Evolutionary Strategy: Discussion and Conclusions

The main goal of the MCPC strategy is to improve (or overcome) the behavior of the RBCGA against premature convergence while generating fuzzy knowledge bases

using a small population size. Using three different single crossover applications, we have noticed three negative points:

1. Fast convergence (FC);
2. Lack of diversity (LD);
3. Reaching a plateau (RP).

For the sake of comparison we assume that the FC is overcame if the best fitness value isn't reached before 1 000 generations (improvement of 0.50% or more), the LD is overcame if the $\approx$ 92.00% fitness level is exeeded and finally the RP is overcame if the shape of the curve shows no plateau.

The different combinations of multi-combinative evolution strategy achieved various results for the tests made in this paper. Table 4.8 summarizes the ability of the different strategies, the check mark $\surd$ means that the strategy succeeded in overcoming a negative aspect or achieving a positive one. From table 4.8 it

Table 4.8 Performances : Single Applications vs. MCPC Strategies

| Strategy | Overcoming FC | Simplification | Improving LD | Overcoming RP |
|---|---|---|---|---|
| $BLX-0.5$ | | $\surd$ | | |
| $NAX$ | $\surd$ | | | |
| $ELX$ | $\surd$ | $\surd$ | | |
| $LC$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ |
| $SA$ | $\surd$ | $\surd$ | $\surd$ | $\surd$ |
| $EEB1$ | | | | $\surd$ |
| $EEB2$ | | $\surd\surd$ | $\surd$ | $\surd$ |

is obvious that the only two strategies achieving all the constraints are the $LC$ and $SA$ ones. Moreover, the $EEB2$, even with the fast convergence, remains very efficient because of the high simplicity level of the FKBs (less fuzzy rules), hence the two $\surd$ marks. The $EBB_1$ failed almost in every single criteria, which means that exploitation during the early stages of an evolution is not a good way to perform the optimization process. The best way is to explore the search space while the breed is young and then exploiting at the end for a fine tuning. Among the strategies proposed, the $LC$, $SA$ and $EBB_2$ are the best. The $SA$ and $LC$ strategies achieved similar results. The $SA$ achieved a slightly better fitness level at the 5 000 generations test. The learning time being a very important point in computer learning, the $LC$ surpasses the $SA$ at this point because it produces three times less children, since for a couple of parents $SA$ produces six children, while the $LC$ generates only two. If a fast convergence (time wise) is needed combined with a high simplification rate (less fuzzy rules), the $EBB_2$ strategy must be used. The $EBB_2$ can be very suitable for factory floor applications.

The MCPC strategy overcomes some of the problems faced by the single applications, apart from the $EBB_1$. In the next section, we apply the multi-combinative strategy on a set of experimental data and we compare it with the single applications.

## 4.7 Application to Experimental Data

In this section, we use a multi-combinative and a single application learning processes on a set of experimental data used to predict tool wear ($VB$) based on a measure of the feed ($feed$), the cutting force ($F_c$), the feed force ($F_f$), the depth of cut ($ap$) during turning operations, and the time of turning ($t$) [2]. In order to

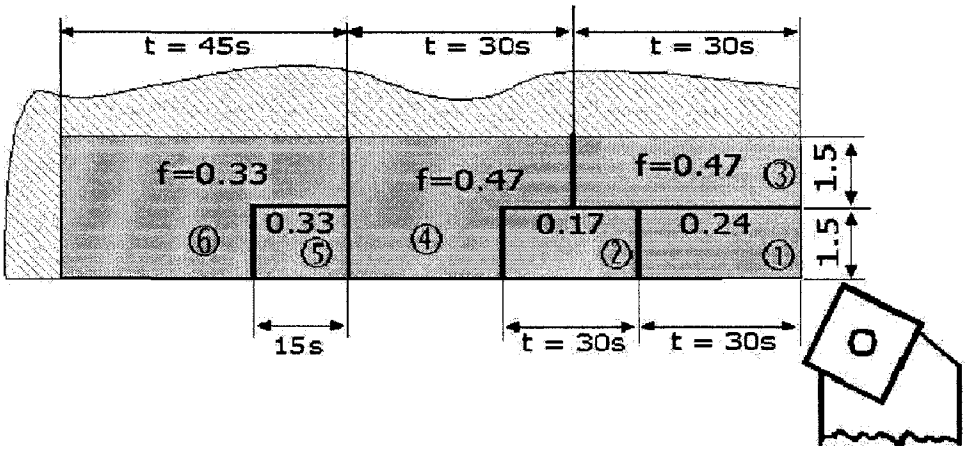


Figure 4.14 Sets of cutting conditions

simulate factory floor conditions, a typical part is machined on a conventional lathe under six different cutting conditions [3], such as shown in Fig. 4.14. The RBCGA is used to automatically generate the FKB, from the set of experimental data collected during this experiment. The maximum complexity is set to 5 fuzzy sets on each of the 5 input premises and 10 fuzzy sets on the conclusion. Therefore, the maximum number of fuzzy rules is given by $K = 5 \times 5 \ldots \times 5 = 5^5 = 3125$.

### 4.7.1 Selection of the strategies and the evolution parameters

Since we are dealing with a factory floor modeling problem, we need a strategy that respects the following:

- achieving a reasonable fitness level in a reasonable time (around 5 minutes);
- a simple FKB is preferable, in case it has to be fine tuned manually for further applications.

From this constraints, we can easily pick the $EBB_2$ application. Because of the simplicity level of its FKBs, the learning will be processed faster by the RBCGA. As for the single application, the $BLX-0.5$ is the one giving the less fuzzy rules, hence it runs faster while achieving comparable results to the other single applications. The evolution parameters of the RBCGA are:

- Multi crossover application % : $pr_1 = 85.00\%$;
- Fuzzy set reducer application % : $pr_2 = 15.00\%$;
- Uniform mutation % : $pr_3 = 05.00\%$
- Population size is fixed at 100;
- Number of generations is fixed at 350.

The number of generations and the population size has been chosen with respect to the convergence time constraint.

### 4.7.2 Application to Experimental Data : Results

As shown in table 4.9, the FKB obtained by $BLX-0.5$ is 94.43% accurate which is a fairly good fitness level. The size of the fuzzy rules base is 32 (reduced from 3125). The highest simplicity level is achieved, since each input premise contains the minimum number of fuzzy sets (2) that are obtained by linking the limits. The

Table 4.9 $\phi_{rms}$ obtained by EEB2 and BLX-0.5 for the experimental data

| Crossover mechanism | Best fitness level | # of fuzzy rules |
|---|---|---|
| BLX-0.5 | 94.43% | 32 |
| $EBB_2$ | 96.11% | 72 |

$EBB_2$ outperformed the $BLX-0.5$ accuracy wise, since it reached 96.13% fitness level. However, the number of fuzzy rules is higher than the one obtained from the $BLX-0.5$, i.e., 72 fuzzy rules. Nevertheless, 72 fuzzy rules remains a very good simplification from the maximal size of 3125. The multi-combinative strategy is very successful in modeling the tool wear monitoring problem. The FKB obtained is accurate but still sufficiently simple for further human tuning.

## 4.8 Conclusion

The mutli-combinative strategies used in this paper helped to overcome some aspects of the premature convergence encountered in the automatic generation of fuzzy knowledge bases using genetic algorithms. The different ways to create new breed from the same pair of parents genes is a good way to improve diversity. How-

ever at the very late stages of the evolution, the presence of too many look alike individuals is still a problem. We can also conclude that exploration/exploitation balance influences the results. The rule to use is as follows: exploration at the early stages of the evolution, relaxing in the middle and exploitation at the end, achieves better results than using exploitation, exploration or relaxed exploitation only. The multi-combinative strategy also proved to be efficient when applied to experimental data. In order to improve the performances of our RBCGA, a new crossover mechanism that increases the number of fuzzy sets, when the minimum of two fuzzy sets is reached on each premise can be added, which is a way to add more diversity in the population.

## Acknowledgment

Financial support from the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618, RGPIN-105518 and STPGP-269579 is gratefully acknowledged.

# REFERENCES

[1] Achiche, S., Balazinski, M. and Baron, L., *Real/Binary-Like Coded Genetic Algorithm to Automatically Generate Fuzzy Knowledge Bases*, IEEE Fourth International Conference on Control and Automation, pp. 799–803, 2003.

[2] Achiche, S., Balazinski, M., Baron, L. and Jemielniak, K., *Tool Wear Monitoring Using Genetically-Generated Fuzzy Knowledge Bases*, Engineering Applications of Artificial Intelligence, 15, pp. 303-314, 2002.

[3] Balazinski, M., Czogala, E., Jemielniak, K. and Leski, J., *Tool Condition Monitoring using Artificial Intelligence Methods*, Engineering Applications of Artificial Intelligence, pp. 75–80, 2002.

[4] Balazinski, M., Achiche, S. and Baron, L., *Influences of Optimization and Selection Criteria on Genetically-Generated Fuzzy Knowledge Bases*, International Conference on Advanced manufacturing Technology, pp. 159–164, 2000.

[5] Balazinski, M., Bellerose, M. and Czogala,E., *Application of Fuzzy Logic Techniques to the Selection of Cutting Parameters in Machining Processes*, International Journal for Fuzzy Sets and Systems, Vol. 61, pp. 307–317, 1993.

[6] Baron, L., Achiche, S. and Balazinski, *Fuzzy Decisions System Knowledge Base Generation Using a Genetic Algorithm*, International Journal of Approximate Reasoning, Vol. 28, pp. 125–148, 2001.

[7] Bonissone, P.P., Khedkar, P.S. et Chen, Y.T., *Gentic Algorithms for Automated Tunning of Fuzzy Controllers, a transportation application*, Fifth International Conference on Fuzzy Systems (FUZZ-IEEE'96), pp. 674–680, 1996.

[8] Brad L. Miller, *Genetic Algorithms with Dynamic Niche Sharing for Multimodal Function Optimization*, International Conference on Evolutionary Computation, 1996.

[9] Casillas, J., Cordón, O. and Herrera, F., *A Methodology to Improve ad-hoc Data-driven Linguistic Rule Learning Methods by Inducing Cooperation among Rules*, Technical Report, #DECSAI–000101, University of Granada, Spain, 2000.

[10] Castellano, G., Attolico, G. and Distante, A., *Automatic Generation of Fuzzy Rules for Reactive Robot Controllers*, Robitics and Autonomous Systems, Vol. 22, pp. 133–149, 1997.

[11] Eiben, A.E. and Bäck, T.,*An Empirical Investigation of Multi-Parent Recombination Operators in Evolution Strategies.*, Evolutionary Computation, 5(3), pp. 347–365, 1997.

[12] Eshelman, L. J. and Schaffer, J.D., *Preventing Premature Convergence by Preventing Incest*, Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 115-122, 1991.

[13] Goldberg, D. E.,*Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 412 pages, 1989.

[14] Hagras, H., Callaghan, V., Colley, M. and Carr-West, M., *A Fuzzy-Genetic Based Embedded-Agent Approach to Learning & Control in Agricultural Autonomous Vehicles*, Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan, pp. 1005–1010, 1999.

[15] Herrera, F. and Lozano, M., *Gradual Distributed Real-Coded Genetic Algorithms*, IEEE Transactions on Evolutionary Computation, Vol. 4, pp. 43–63, 2000.

[16] Herrera, F. and Lazano, M.,*Gradual Distributed Real-Coded Genetic Algorithms*, Technical Report, #DECSAI–97–01–03, Universty of Granada, Spain, 1997.

[17] Herrera, F., Lazano, M. et Verdegay, J.L., *Tunning Fuzzy Logic Controllers by Genetic Algorithms*, International Journal of Approximate Reasoning 12, pp. 299-315, 1995.

[18] Holland, J. H., *Genetic Algorithms and the Optimal Allocation of Trials*, SIAM Journal on Computing, 2(2), pp. 88–105, 1973.

[19] Li, TH., Lucasius, C.B. and Kateman, G., *Optimization of Calibration Data with the Dynamic Genetic Algorithm*, Analytica Chimica Acta, pp. 123–134, 1992.

[20] Mahfoud, S. W., *Niching Methods for Genetic Algorithms* Ph.D. Thesis, University of Illinois at Urbana-Champaign, 1995.

[21] Michalewicz, Z., *Genetic Algorithms + Data Structure = Evolution Programs*, New York: Springer, 1992.

[22] MÃ$\frac{1}{4}$hlenbein, H. and Schlierkamp-Voosen, D., *Predictive Models for the Breeder Genetic Algorithm: I. Continuous Parameter Optimization*, Evolutionary Computation, 1 (1), pp. 25-49, 1993.

[23] Nomura, H., Hayashi, I. and Wakami, N., *A Self-tuning Method of Fuzzy Reasoning by Genetic Algorithm*, Proceedings International fuzzy Systems and Intelligent Control Conference, pp. 236–245, 1992.

[24] Ono, I. and Kobayashi, S., *A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distributed Crossover*, Proceedings of the 7th International. Conference on Genetic Algorithms, pp. 246–253, 1997.

[25] Tsutsui, S.,*Multi-parent Recombination in Genetic Algorithms with Search Space Boundary Extension by Mirroring*, Proceedings of the Fifth International Conference on Parallel Problem Solving from Nature, pp. 428–437, 1998.

[26] Tsutsui, S. and Ghosh, A., *A Study on the Effect of Multi-parent Recombination in Real Coded Genetic Algorithms*, Proceedings of the 1998 IEEE International Conference on Evolutionary Computation, pp. 119–133, 1998.

[27] Valesco, J.R., Lopez, S. and Magdalena, L., *Genetic Fuzzy Clustering for the Definition of Fuzzy Sets*, Proceedings Sixth IEEE International Conference On Fuzzy Systems, FUZZ IEEE, Vol. III, pp.1665–1670, 1997.

[28] Xiong, N. and Litz, L., *Reduction of Fuzzy Control Rules by Means of Premise Learning - Method and Case Study*, Fuzzy Sets and Systems, Vol. 132, pp. 217–231, 2002.

[29] Zadeh, L.A., *Outline of New Approach to the Analysis of Complex Systems and Decisions Processes*, IEEE Transactions of Systems, Man and Cybernetics, Vol. 3, pp. 28–44, 1973.

# CHAPTER 5

# ONLINE PREDICTION OF PULP BRIGHTNESS USING FUZZY LOGIC MODELS

Sofiane Achiche*, Luc Baron*, Marek Balazinski* et Mokhtar Benaoudia**.

* Département de génie mécanique, École polytechnique de montréal.
P.O. 6079, station Centre-Ville, Montréal, Québec, Canada, H3C 3A7.

Département de génie mécanique.
**Centre de Recherche Industrielle du Québec.
333, rue Franquet Sainte-Foy, Québec, Canada G1P 4C7
sofiane.achiche@polymtl.ca

## 5.1 Abstract

The quality of the thermomechanical pulp is influenced by a large number of variables. To control the pulp and paper process (TMP), the process maker has to choose manually the influencing variables, which can change significantly depending on the quality of the raw material (wood chips). However, there is very little knowledge about the relationships between the quality of the pulp obtained by the TMP process and the wood chips properties. The research proposed in this paper uses genetically-generated knowledge bases to model these relationships while using measurements of the wood chips quality, obtained from the chip management

system ($CMS^{©}$), and process parameters (bleaching materials).

*Keywords*: Pulp and Paper, Bleaching, Fuzzy Decision Support System, Knowledge Base, Learning, Genetic Algorithm.

## 5.2 Introduction

Pulp and paper quality depends on several factors that rely on the quality of the wood chips and on different physical properties such as: type of wood; bark percentage, presence of knots, etc. Exterior conditions (such as: weather variations, storage conditions, etc ...) can also influence the quality of the pulp and paper by altering the wood chips. Presently, there is no established knowledge concerning the co-influences of the several parameters surrounding the thermo-mechanical pulp and paper process (TMP). However, several works singled out the influence of some parameters as separate variables. For instance, the size of the chips influences its resistance and hence the energy level used in the transformation process can vary signficantly [9,12,25,26]. A degradation in the quality of the chips causes an increase in the use of bleaching material in the process [16–18,21] which therefore increases the cost of the pulp and paper. Learning about the behavior of the wood chips would improve the TMP process, however this knowledge is quite uncertain and the number of influencing parameters can change depending on the desired quality of the pulp, for example the importance of bark presence in the chips changes with

the process (KRAFT or thermo-mechanical). Knowing that the TMP process is a nonlinear and poorly understood phenomenon governed by an important number of influent variables, it is almost impossible to model it mathematically in order to predict the quality of the pulp beforehand (from wood chips characteristics). Fuzzy logic is known to be efficient when dealing with imprecise and/or incomplete information data which makes it an adequate approach to model the behavior of the transformation process from wood chips to pulp. As a prediction tool, the use of fuzzy decision support systems (FDSS) is very appropriate in this case, since it can deal with incomplete and/or imprecise knowledge applied to either linear or nonlinear problems. FDSS have already been successfully applied to many different problems such as: tool conditions monitoring [5], job dispatching [6], tolerance allocation [14] and surgery assitance [24]. However, in most cases, the build-up of the fuzzy model isn't a simple task, and in the particular case of this paper the task is even more complex due to the high number of influencing variables combined to the very poor knowledge of the TMP process behavior. Hence the need of an automatic generation tool of fuzzy knowledge bases (FKBs). This automatic generation tool includes in the learning process: the number of fuzzy sets and rules; the repartition of the fuzzy sets on premises and conclusions; and the fuzzy rules. A genetic-based learning algorithm (genetic algorithm) is used to automatically generate the FKBs. The genetic algorithm (GA) method includes all the above-mentioned knowledge aspects in the learning process [2]. In this paper, we use an FDSS software called

*Fuzzy-Flou*, developed at École Polytechnique Montreal (Canada) and the Technical University of Silesia in Gliwice (Poland).

The GA requires a set of numerical data to learn from in order to produce FKBs that will predict the pulp quality. This set of data is obtained through the application of several experiments executed according to an experience plan developed commonly by the Centre de Recherche Industrielle duQuébec (CRIQ) and the University of Québec at Trois-Rivières. The experience plan emulates a large variety of wood chips used or would be used in the pulp and paper industry. The influencing variables caracterizing the wood chips can be measured using two different approaches:

1. classic laboratory measurements;

2. artificial vision measurements using a chip management system $CMS^{©}$.

The main goal of this paper is to automatically generate fuzzy models to characterize wood chip quality online in order to optimize the TMP process and predict pulp quality using numerical data. The production settings would mainly take into account bleaching agents (hydrogen peroxyde–$H_2O_2$–and sodium hydrosulfite–$Na_2S_2O_4$–), and the FKBs obtained for both bleaching agents will be compared to assess the efficiency of each.

## 5.3 The Chips Management System

The CRIQ has developed the Chips Managment System $CMS^{©}$, an innovative device that measures chip brightness (Luminance). The $CMS^{©}$ was used to confirm the existence of a correlation between bleaching agent consumption and chip luminance. This correlation is an inverse relationship existing between the luminance factor measured with the $CMS^{©}$ and hydrosulfite consumption as shown in Fig. 5.1. The experiment, carried out in a pilot plan, uses the needed charges of hydrosulfite in order to achieve a certain predifined quality of the paper, while using wood chips of different ages (different luminance). This is a perfect example of

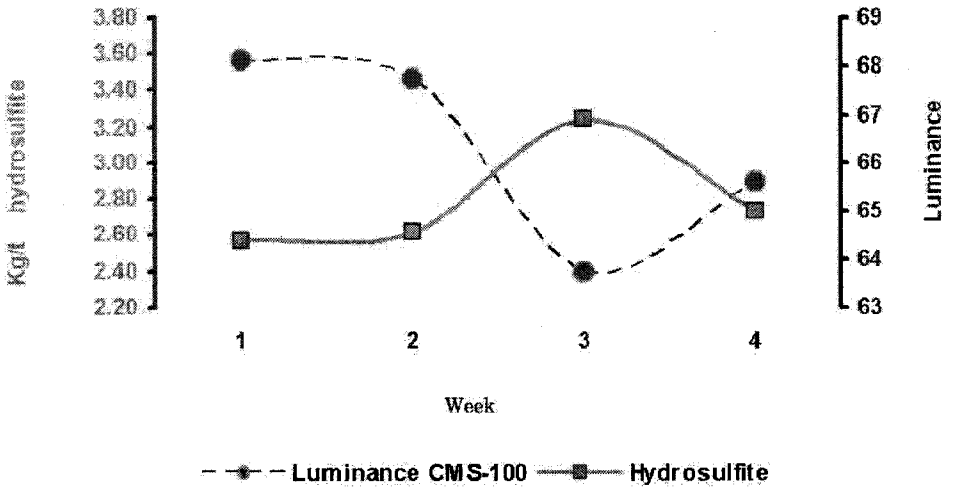


Figure 5.1 CMS Luminance Variation vs. Hydrosulfite consumption (during 4 weeks)

a correlation between a specific chip characteristic and a process parameter. This paper will introduce a new approach to modelise the TMP process, which will pro-

vide a mean to predict/control pulp properties using raw material caracteristics as inputs rather than reacting later on the basis of the obtained pulp. This approach of controlling input variables represents a significant and remarkable progress in the TMP field and beyond, providing a better understanding of chip characteristics and their impact on the pulp and paper quality. On one hand, it aims to improve stability in the TMP process through known or predictable bleaching chemical consumption and on the other hand, to reduce production costs by using the right amount of bleaching chemicals and avoiding precautionary overuse. Consequently, this approach reduces environment pollution, and increases paper quality by improving pulp quality.

The combination of the $CMS^{©}$ and the fuzzy logic technologies is unique and there is currently no integrated system in place that controls/predicts parameters as a function of chip characterization and qualification measurements through multiple sensors as proposed in this work.

The $CMS^{©}$ characterizes the wood chips online and is equipped with an artificial vision sensor (RGB color camera with a frame grabber) and a near-infrared sensor to measure chip brightness and moisture content. The $CMS^{©}$ was already used for tasks such as monitoring and organizing the chip piles [10] with success. The principal measurements taken from the $CMS^{©}$ are the following:

- **Chip Luminance**, in the $CMS^{©}$, the brightness of the black is defined as zero and the brightness of the white is set to 150. The RGB color camera is calibrated by a color checker made of black and white paperboard. The color of the wood chips is between black and white and so its brightness is between 0 and 150.

- **Chip average Moisture Content**, the near-infrared sensor measures the surface moisture of the wood chips. A phenomenological model has been developed by the CRIQ to compute the average moisture content from surface moisture content.

- **Auxiliary measurements**, several auxiliary sensors on the $CMS^{©}$, provide us with a multitude of data categorized in many different variables.

## 5.4 Experiment Plan for Data Collection

Two sets of experiments were conducted corresponding to two different blocks. In the first block, a potential mix of four species, black spruce, balsam fir, jack pine and white birch are used. The jack pine and white birch have been chosen because they represent a potential source of new resources. The trees have been selected, cut, barked and chipped in order to obtain standard chips with known and controlled age. In September 2001, outdoor stacks of each species of chips have been prepared. During the year, six samples were selected in order to conduct the ex-

perimental plan as described in table 1. In each sample, the experiment for 100% black spruce and 100% balsam fir were repeated twice in order to evaluate the experimental error (12 runs in each sample). The six samples allow us to evaluate the evolution of the quality, i.e., degradation of the chips in time, which is highly dependent on the storage temperature and several other parameters. The first four samples were evaluated at an interval of three weeks: very little changes were noticed. After that, a longer waiting time was used. The age of wood chips has a

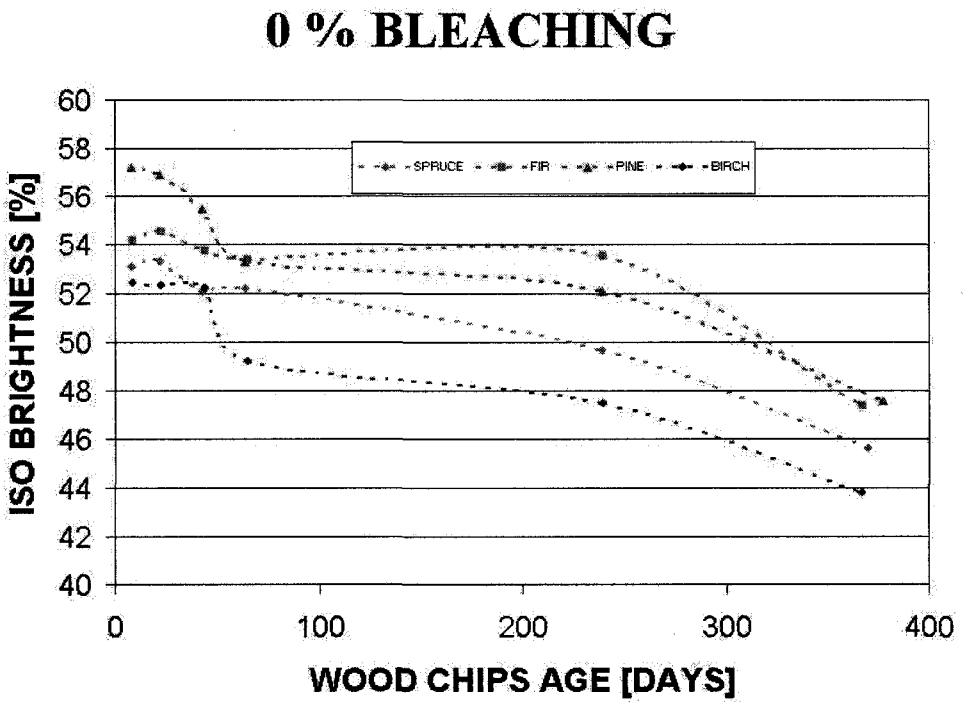


Figure 5.2 Pulp ISO Brightness vs. Age of Wood chips

strong influence on the pulp brightness (as shown in Fig. 5.2), and in the factories the wood chips are of different ages. To simulate this aspect the wood chips used

had the following ages: 9; 23; 44; 65; 240 and 371 days.

The second block of experiments was carried out to investigate the effects of other variables regarding pulp quality using: species (black spruce, balsam fir), density (large, small), initial dryness of the chips (fresh and dry), and thickness of the chips ($0-4$mm and $4-8$mm). However, this block isn't considered in this paper, since the variables can not be obtained from the $CMS^{©}$ and they represent only pure wood species, which is less representative of the reality.

The refining was conducted on the pilot unit Metso CD-300 of the Centre Intégré en Pâtes et Papier of Université du Québec at Trois-Rivières (CIPP of UQTR). Each Sample was washed and refined in two stages. The first one was conducted at a temperature of $128^{o}$C and the second one at atmospheric conditions. Pulps with freeness ranging from 200mL to 150mL were selected for further bleaching (using Peroxyde or Hydrosulfite). For more details on the experience plan refer to [19].

Different concentrations of peroxyde and hydrosulfite (bleaching chimicals) were used:

- Peroxyde from 0% to 5% (i.e.: 0%, 1%, 2%, 3% and 5%);
- Hydrosulfite from 0% to 10% (i.e.: 0%, 2%, 4%, 6%, 8% and 10%).

Table 5.1 Experimental plan for the aging of four different wood species

| Test # | Spruce % | Balsam fir % | Jack pine % | Birch% |
|---|---|---|---|---|
| 1 | 0 | 20 | 40 | 40 |
| 2 | 100 | 0 | 0 | 0 |
| 3 | 0 | 100 | 0 | 0 |
| 4 | 60 | 0 | 0 | 40 |
| 5 | 0 | 60 | 40 | 0 |
| 6 | 60 | 0 | 40 | 0 |
| 7 | 0 | 60 | 0 | 40 |
| 8 | 20 | 0 | 40 | 40 |
| The following are repetitions of tests 2 & 3 for experimental error determination | | | | |
| 9 | 100 | 0 | 0 | 0 |
| 10 | 0 | 100 | 0 | 0 |
| The following are additional tests | | | | |
| 11 | 0 | 0 | 100 | 0 |
| 12 | 0 | 0 | 0 | 100 |

Properties such as ISO brightness and color coordinates have been measured according to the PAPTAC standard.

The database obtained from the different experiences, taking into account all the measured variables (chip properties from the $CMS^{©}$, operation parameters of the TMP and the pulp quality characteristics) contains 178 variables (i.e. 178 columns of data). The additional runs used for error measurement are not a part of the data sets used as a learning base for the FKBs. The data sets for hydrosulfite and peroxyde are slightly different because of the different plateaus used for the bleaching agents, and also for the absence of some mixes when the hydrosulfite experiences have been processed. However, the differences are so slight that we

believe it doesn't have a big impact on the validity and comparability of the data sets.

## 5.5 Selection of the Influencing Variables

The selection of the input variables of the FKBs is a tedious task due to their large number. The FKBs should be able to predict pulp quality from raw material properties. Hence, the variables have to be chosen from wood chip properties obtained from either the $CMS^{©}$ or eventually laboratory measurements. The input variables can be categorized into:

1. Online variables: numerical luminance, color coordinates, humidity, etc; obtained from the $CMS^{©}$.

2. Offline variables: wood chips sizes, density, bark percentage, knots percentage, etc.; obtained from laboratory analysis.

Figure 5.3 shows all the measured variables (offline and online). The $CMS^{©}$ provides:

- the image properties: average of Hue (H), average of Saturation (S) and average of Luminance (L). The H,S and L are obtained from a color space conversion of the standard Red, Green, and Blue (RGB) color cooridnates;

- the average of Luminance obtained from the LAB color coordinates;

| CMS | Copeaux | Pâtes | Blanchiment | Procédé |
|---|---|---|---|---|
| IDBatch | CopeauxID | CopeauxID | CopeauxID | CopeauxID |
| IDSample | Rader8 | PatesID | PatesID | PatesID |
| DateTimeSample | Rader68 | EnergieStade2 | BlanchimentID | PressionLessiveurStade1 |
| CameraSimulation | Rader26 | CSFStade2 | Peroxyde | PressionBatiStade1 |
| FileName | RaderFin | Test_PatesID_1 | NaOH | VitesseRotationStade1 |
| HighLevel | RaderPoussiere | Test_PatesID_2 | PeroxydeResiduel | ChargeStade1 |
| LowLevel | Williams1_18 | EnergieSpecifique | BlancheurISO | DilutionDPStade1 |
| MeanLevel | Williams78_1_18 | IndiceEgouttage | L | DilutionDCStade1 |
| MeanLevelCorrected | Williams58_78 | RejetPulmac | A | EntreferDPStade1 |
| StdDeviation | Williams38_58 | LongFibreMoyArithm | B | EntreferDCStade1 |
| PctDarkChips | Williams18_38 | LongFibreMoyLong | MatieresExtractibles | ConsistanceStade1 |
| Red | WilliamsPoussiere | LongFibreMoyPoids | Lauric | ConsistanceConStade1 |
| Green | Siccite | LongFibreFineMoyArithm | Myristic | ProductionStade1 |
| Blue | DensiteHumide | LongFibreFineMoyPoids | Palmitoleic | ProductionMoyenneStade1 |
| H | DensiteSec | Grammage | Palmitic | EnergieStade1 |
| S | Ecorce | VolumeSpecifique | Linoleic | CSFStade1 |
| L | Noeuds | Densite | Linolenic | PressionLessiveurStade2 |
| MoistureSimulation | Carie | IndiceRupture | Oleic | PressionBatiStade2 |
| SurfaceMoisture | MatieresExtractibles | Allongement | Stearic | VitesseRotationStade2 |
| GlobalMoisture | Lauric | TEA | Pimaric | ChargeStade2 |
| EnvironmentSimulation | Myristic | IndiceEclatement | Sandaracopimaric | DilutionDPStade2 |
| Height | Palmitoleic | IndiceDechirure | Isopimaric | DilutionDCStade2 |
| ExteriorTemp | Palmitic | BlancheurISO | Palustric | EntreferDPStade2 |
| InteriorTemp | Linoleic | OpaciteISO | Levopimaric | EntreferDCStade2 |
| ExteriorHumidity | Linolenic | CoeffDiffusion | Dehydroabietic | ConsistanceStade2 |
| InteriorHumidity | Oleic | CoeffAbsoption | Abietic | ConsistanceConStade2 |
| Référence | Stearic | L | Neoabietic | ProductionStade2 |
| | Pimaric | A | Chlorodehydroabietic12 | ProductionMoyenneStade2 |
| | Sandaracopimaric | B | Chlorodehydroabietic14 | EnergieStade2 |
| | Isopimaric | | Dichlorostearic9_10 | CSFStade2 |
| | Palustric | | Dichlorodehydroabietic | |
| | Levopimaric | | Total | |
| | Dehydroabietic | | | |
| | Abietic | | | |
| | Neoabietic | | | |
| | Chlorodehydroabietic12 | | | |
| | Chlorodehydroabietic14 | | | |
| | Dichlorostearic910 | | | |
| | Dichlorodehydroabietic | | | |
| | Total | | | |

Figure 5.3 Online and offline measured variables

- the presence of dark chips in the wood chips samples;
- the surface humidity.

It is noteworthy that the H, S and L color space was chosen because it defines the colors more naturally: Hue (H) specifies the base color (red, blue, green, brown, etc.), the other two values (S and L) specify the saturation of that color and how bright the color is, the HSL represent the color parameters as perceived by the human eye. The variables HSL are obtained by image processing.

Before generating the FKBs one has to choose the independent variables (input/output) since choosing independent input variables makes the FKB a better representation of the process to model.

### 5.5.1 Selection of the output variable

The output variable of the fuzzy logic models must be a feature that gives a crisp information on the quality of the obtained pulp. Since the pulp brightness is one of the most important properties—high ISO brightness of the pulp translates into high quality standards for the paper—of the TMP process it is a natural choice to select it as the output variable.

### 5.5.2 Selection of the input variables

A large amount of variables can be measured by the mean of the different sensors. Some of these variables can greatly influence the TMP process, others not. Ones tend to beleive that wood spices should be a part of the input variables of a prediction tool of pulp quality. However as stated in [20], the ISO brightness can be predicted with no knowledge of the wood mixture. As for this work, since it concerns online prediction, only the parameters provided directly by the $CMS^{©}$ (using the RGB camera and near infrared sensor) are taken into account, as listed above, appart from the presence of dark chips which will be used later. As for the

process parameter, the bleaching concentration is considered. A data screening using the partial least squares method (PLS) on the variables provided the following results:

- Bleaching concentration (peroxyde or hydrosulfite) is the most influencing parameter.

The other parameters attain very close results in the following order:

1. Average of H;
2. Average of Luminance (luminance of the LAB color coordinates);
3. Average of S;
4. Average of L;
5. Average of surface moisture of the wood chips.

**Note:** The average of Luminance measured by the $CMS^{©}$ is directly correlated to luminance L of the HSL coordinates (a correlation close to $\approx; 90.00\%$), the linear relationship is shown in Fig. 5.4. Hence, only one of these two variables is conserned for the learning, i.e. the non-mapped average of Luminance. To summerize for the time being, we keep the following parameters as inputs to create the FKBs that will predict the TMP pulp quality: bleaching concentration (peroxyde or hydrosulfite); average of Luminance; average of H; average of S and average of surface moisture.

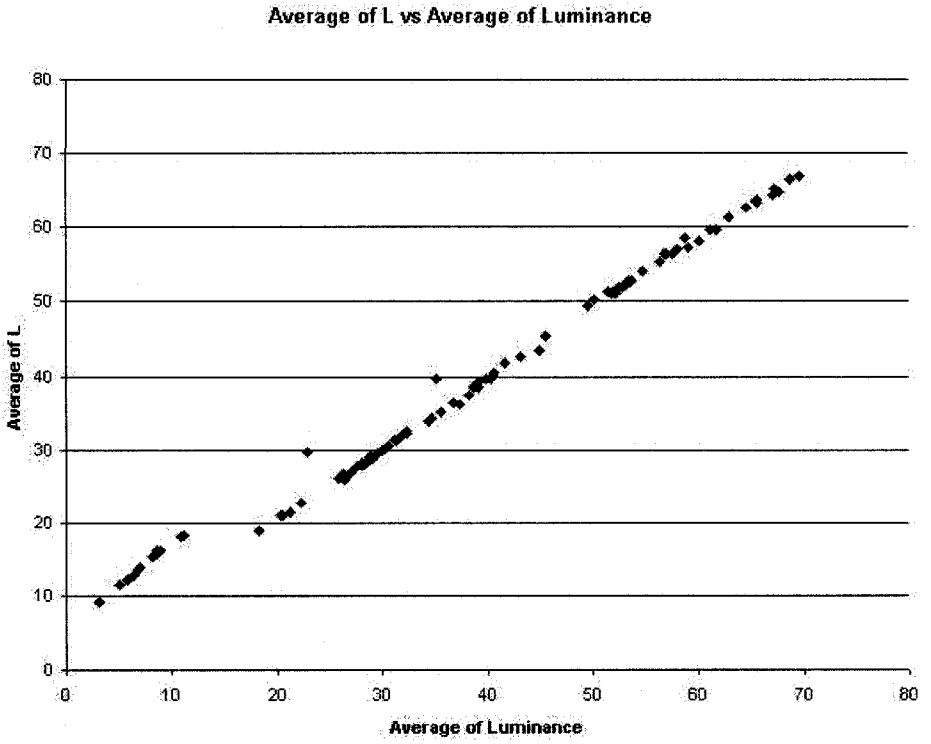


Figure 5.4 Average of Luminance vs Average of L

## 5.6 Fuzzy Decision Support Systems

A rule-based approach to decision making using fuzzy logic techniques may consider imprecise vague language as a set of rules linking a finite number of conclusions. The knowledge base of such systems consists of two components: a linguistic terms base and a fuzzy rules base [7]. The former is divided into two parts: the fuzzy premises (or inputs) and the fuzzy conclusions (or outputs). In this paper we consider FKBs of $N$ multiple inputs and one single output (MISO). Moreover, we consider only general triangular fuzzy sets on the premises and sharp-symmetric

triangular fuzzy sets on the conclusion. The representation of such imprecise knowledge by means of fuzzy linguistic terms makes it possible to carry out quantitative processing in the course of inference that is used for handling uncertain (imprecise) knowledge. This is often called approximate reasoning [27]. This knowledge, expressed by $(k = 1, 2, \cdots, K)$ finite heuristic fuzzy rules of the type MISO, may be written in the form:

$$R^k_{MISO} : \text{if } x_1 \text{ is } X^k_1 \text{ and } x_2 \text{ is } X^k_2 \text{ and } \cdots \text{ and } x_N \text{ is } X^k_N \text{ then } y \text{ is } Y^k, \quad (5.1)$$

where $\{X^k_i\}^N_{i=1}$ denote values of linguistic variables $\{x_i\}^N_{i=1}$ (conditions) defined in the following universe of discourse $\{\mathbf{X}_i\}^N_{i=1}$; and $Y^k$ stands for the value of the independent linguistic variable $y$ (conclusion) in the universe of discourse $\mathbf{Y}$. The global relation aggregating all rules from $k = 1$ to $K$ is given as

$$R = also^K_{k=1}(R^k_{MISO}). \quad (5.2)$$

where the sentence connective *also* denotes any t- or s-norm (e.g., *min* ($\wedge$) or *max* ($\vee$) operators) or averages. For a given set of fuzzy inputs $\{X'_i\}^N_1$ (or observations), the fuzzy output $Y'$ (or conclusion) may be expressed symbolically as:

$$Y' = (X'_1, X'_2, \cdots, X'_N) \circ R, \quad (5.3)$$

where $\circ$ denotes a compositional rule of inference (CRI), e.g., the *sup*-$\wedge$ or *sup-prod* (*sup*-$*$). Alternatively, the CRI of eq.(5.3) is easily computed as

$$Y' = X'_N \circ \cdots \circ (X'_2 \circ (X'_1 \circ R)). \tag{5.4}$$

The CRI mechanisms allow us to obtain different conclusions represented as the membership function $Y'$. In FDSS Fuzzy-Flou, there are three defuzzification methods: the center of gravity (COG); the mean of maxima (MOM); and the height method. All the results presented in this paper are obtained with the $\sum$-*sup*$*$-$*$-$*$ CRI and COG as defuzzification.

### 5.6.1 FDSS Learning Paradigm

In general, FDSS requires a knowledge base in order to support the decision-making process of end-users. The FKB can be created manually by a human expert or automatically learned from a set of sampled data. In this paper the automatic learning process of the FDSS knowledge base is automatic. The learning process is aimed at producing knowledge bases that are manageable by either a human expert or a computer. The FKBs must accurately reproduce the set of learned data, while interpolating or extrapolating fair conclusions in other situations. A *minimalist* approach is implemented through an automatic reduction of fuzzy rules and sets on the premises, whenever the approximation error is not penalized too

greatly by this reduction.

## 5.7 Genetic-Based Learning Process

GAs are powerful stochastic optimization techniques that are based on the analogy of the mechanics of biological genetics and imitate the Darwinian survival-of-the-fittest approach [8]. As shown in Fig. 5.5, each individual of a population is a potential FDSS Fuzzy-Flou knowledge base. Figure 5.5 presents the *encoding/decoding*

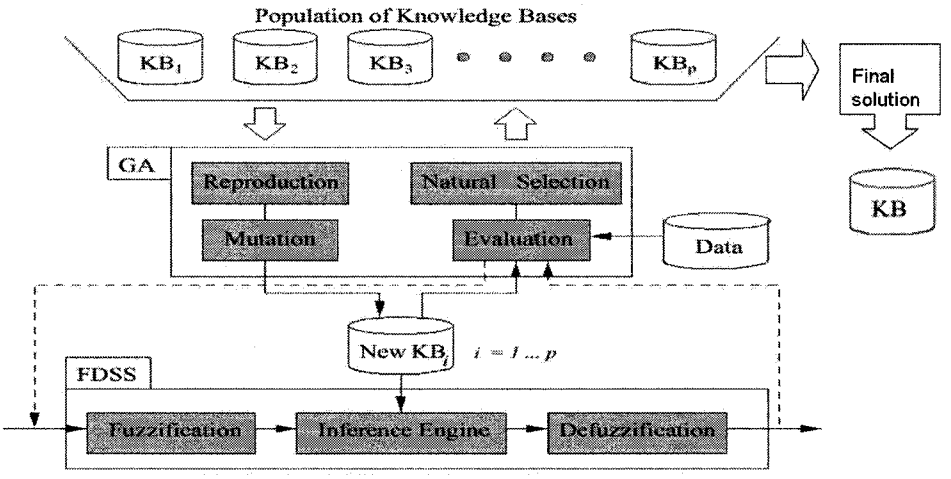


Figure 5.5 Genetic learning paradigm

*scheme* as well as the four basic operations, i.e.: *reproduction*, *mutation*, *evaluation* and *natural selection*, of the developed GA learning software [4]. The method uses iterative improvement of individuals at each generation to converge toward multiple optima simultaneously. When the number of unknown parameters increases, GA exhibits only a polynomial increase of the complexity [13, 22], while the other op-

timization techniques show an exponential increase. The genetic algorithm used in this paper is a Real/Binary like genetic algorithm (RBCGA) developed by the authors [2]. The RBCGA is a combination of a real coded genetic algorithm (RCGA) and a binary coded genetic algorithm.

### 5.7.1 Coding

The *genotype* corresponds to several independent sets of reals and a set of integers. Here, the *genotype* can be described as follows:

**Premises and Conclusion**

There are as many real number sets as there are premises in the problem and one set for the conclusion.

**Fuzzy Rules**

The fuzzy rules are coded as a set of integers representing an ordered list of the combination of the premises. The initial population of FKBs is composed of $P$ randomly generated FKBs. The *genotype* of each new solution contains all the sets mentioned above.

#### 5.7.1.1 Reproduction Mechanisms of the RBCGA

The reproduction of the FKBs in the RBCGA is performed through three principal crossover mechanisms, each one having its own purpose, as explained below.

### a) Multi Crossover

The multi-crossover is a combination of crossovers applied on different parts of the *genotype*.

### a.1) Premises/Conclusion Crossover

For this part of the FKB, the blended crossover $\alpha$ is used.

#### a.1.1) Blended Crossover $\alpha$

The blended crossover $\alpha$ is denoted as *BLX-*$\alpha$ [15], where $\alpha$ controls the exploitation/exploration level of the offspring obtained from the selected parents. As shown in Fig. 5.6, the $z_i$ values of the offspring are randomly selected in the interval [$min$, $max$].

### a.2) Fuzzy Rules Crossover

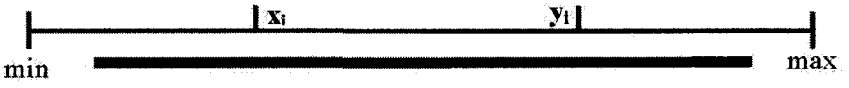


Figure 5.6 Blended crossover $\alpha$ – BLX-$\alpha$

Since the part of the *genotype* representing the fuzzy rule base is composed of integer numbers, the crossover on this part of the *genotype* is done by a *simple crossover* as shown in Fig. 5.7. These two mechanisms are governed by the initiating probability $P_c$.

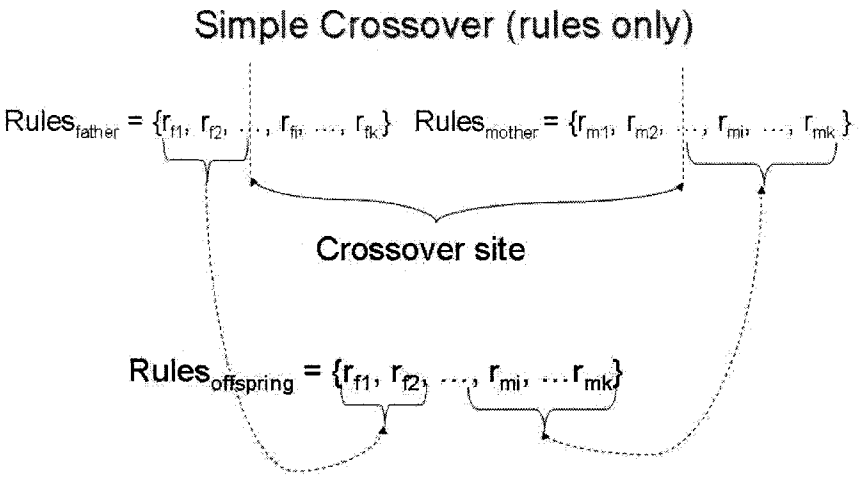


Figure 5.7 Simple crossover of the RBLGA

### b) Fuzzy Set Reducer

This mechanism is set to increase the simplicity of the FKBs by selecting a summit on each premise and erasing it from the respective sets. This mechanism is governed by the initiating probability $P_r$.

### c) Mutation

*Mutation* is the creation of a new individual by altering the gene of an existing one. The probability $P_m$ governs the occurrence of this mechanism. In this paper, uniform mutation [23] is applied to one randomly selected individual.

### d) Selection Mechanism

The selection mechanism used in this paper (to select the parents) is performed by respecting the Darwinian elitist philosophy based on the performance criteria.

## 5.8 Performance Criterion

The performance criteria allows one to compute the rating of each FKB. This performance rating is used by the RBCGA in order to perform natural selection. Here, the performance criterion is the accuracy level of a FKB (approximation error) in reproducing the outputs of the learning data. The approximation error is measured using the root-mean-square error:

$$\Delta_{rms} = \sqrt{\sum_{i=1}^{N} \frac{(RBCGA_{output} - data_{output})^2}{N}} \tag{5.5}$$

where $N$ represents the number of learning data. The fitness value is evaluated as a percentage of the output length ($L = z_{max} - z_{min}$) of the conclusion, i.e., $\phi_{rms} = \frac{L - \Delta_{rms}}{L} \times 100$.

## 5.9 Evolutionary Strategy

This strategy is used in order to ensure a good balance between exploitation and exploration throughout the evolution. The Multiple Crossovers Per Couple (MCPC) strategy is used on the reproduction mechanisms. Although we can use recombination strategy on all the parts of the FKB, the quality of the FKB is more sensitive to the quality of the repartition and the number of fuzzy sets [11], and hence, the evolutionary strategy is only applied to the premises. The MCPC used is the Exploration/Exploitation Balance Strategy studied by the authors [1]. The different

values given to $\alpha$ influence the exploration, exploitation or relaxed exploitation levels of the crossover mechanism. The three values are set as follow:

- Exploration: $\alpha = 1.00$ for total exploration;
- Relaxed Exploitation: $\alpha = 0.50$;
- Exploitation: $\alpha = 0.1$ for close to maximal exploitation—$\alpha \neq 0.00$, in order to make a difference between the BLX-$\alpha$ and the uniform mutation—.

The MCPC uses exploration at the early stages, then shifts to a relaxed exploitation for the evolution stage and finally switches to exploitation for the last stages of evolution. This order proved to be the most efficient [3]. The following was considered for the definition of the stages (for a maximal number of generations $N$):

- The first third (1/3) of $N$ is considered as being the early stages;
- The second thrid (from the end of the first third to the beginning of the last one, limits not included) is considered as the evolution stage.
- The last third of the $N$ is considered as being the last stages;

### 5.9.1 Evolution parameters

The evolution parameters of the RBCGA are the parameters that allow the algorithm to reach near optimal solutions. The parameters to set at the beginning of

the evolution process are:

- crossover probability of the premises $P_c$;
- reduction probability of the fuzzy sets $P_r$;
- mutation probability $P_m$;
- maximal complexity allowed : the maximal number of fuzzy sets allowed on each premise (this number sets also the maximum of possible fuzzy rules);
- population size;
- number of generations or a stopping condition.

These variables must be set manually and they are not part of the evolution. However, the maximal complexity is allowed to decrease during the evolution and this is done with the fuzzy sets reduction mechanism.

Using our experiences from previous research works and some pre-runs of the RBCGA on the experimental data, the following values were chosen for the above parameters:

- $P_c = 85\%$, the same probability is applied for the fuzzy rules crossover mechanism;
- probability $P_r = 15\%$;

- probability $P_m = 10\%$;
- population size is 100;
- maximal number of generations is set to 500;
- the maximal complexity is set as follows:
  - the premises can't contain more than 6 fuzzy sets ($NP_{FS}$). Different premises can contain a different number of fuzzy sets: $2 \leq NP_{FS} \leq 6$;
  - 32 fuzzy sets is the maximum # of conclusions ($NC_{FS}$): $1 \leq NC_{FS} \leq 32$;
  - the maximal number of fuzzy rules is equal to $6^{n}$, $n$ being the number of premises.

## 5.10 Learning the FKBs for Brightness Prediction

Before applying the RBCGA on the set of experimental data, this data is divided into two different sets:

- a set of data for the learning;
- a set of data for testing the FKBs, once the learning is completed.

The separation of the initial set of data into two sets (learning and testing) is made automatically using a "pulling without hand-over" algorithm. One tenth (10%) of the data is hidden from the learning; this set will be used later in order to test the

generalization of the FKBs. The remaining 90% are used in the learning, and it is noteworthy that the limit values of the input variables are kept in the learning set of data in order to define the variable's range of application. The set of learning data will be named *FLearn* and the file containing the remaining 10%, *FTest*.

### 5.10.1 Application to the *FLearn*

To serve as a reminder, the *FLearn* file contains the following input variables: bleaching concentration (peroxyde or hydrosulfite); average of Luminance; average of H; average of S; average of surface moisture, and the output variable being the pulp ISO brightness.

The learning file containing the peroxyde is named *FLearnP*, and the one containing the hydrosulfite is named *FLearnH*. The testing file containing the peroxyde is named *FTestP*, and the one containing the hydrosulfite is named *FTestH*.

#### 5.10.1.1 Results obtained for the *FLearnP*

The application of the RBCGA to the *FLearn*, produced an FKB that approximates the ISO brightness with an RMS error of 2.13% $\Delta_{rms}$. The approximated FKB has the following structure:

- 2 fuzzy sets on the *mean level luminance*, *mean of H* and *mean of S* premises;
- 3 fuzzy sets on the *mean of surface humidity* and the *% of Peroxyde*;

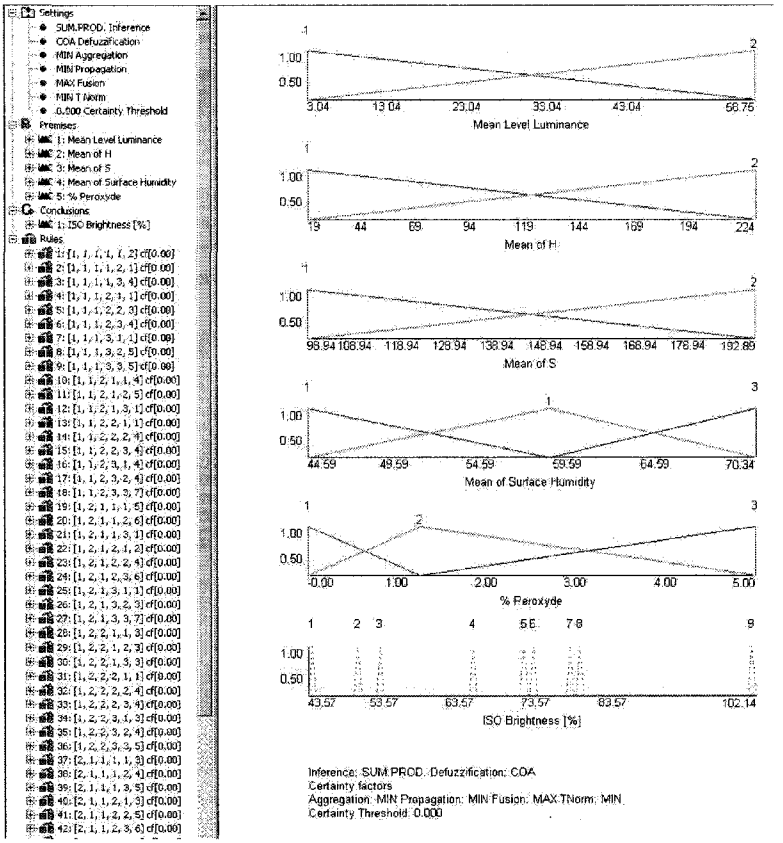


Figure 5.8 Genetically approximated FKB for the *FLearnP*

- 9 fuzzy sets on the conclusion *ISO Brightness*;
- 72 fuzzy rules (rather than the possible 7776 fuzzy rules).

Figure 5.8 presents a screen print-out of the FKB that was obtained, opened with FDSS Fuzzy-Flou software. The *FtestP* is proceeded as an observation file through the FKB shown in Fig. 5.8. The predicitons provided by the FKB allow one to define the generalization level of the FKB while answering a set of data not included in the learning. Figure 5.9 shows the difference between the predictions of the FKB and the experimental values.

The $\Delta_{rms}$ obtained for the testing file is 2.45%, which is very satisfactory since it

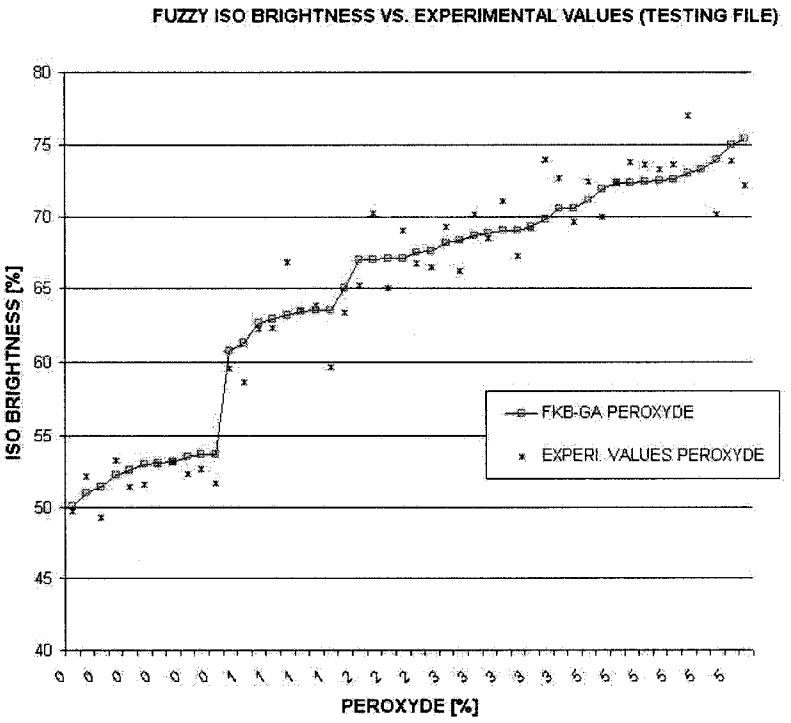


Figure 5.9 Results obtained by the GA-FKB for the *FtestP*

remains at the same level as the the learning $\Delta_{rms}$. The errors obtained for both learning and testing files are below the experimental error of the experimental plan that have been evaluated at approximately 5%.

#### 5.10.1.2 Results obtained for the *FLearnH*

The application of the RCBGA to the *FLearnH* produced a similar FKB to the one obtained for the *FLearnP* in terms of the level of complexity. The FKB that

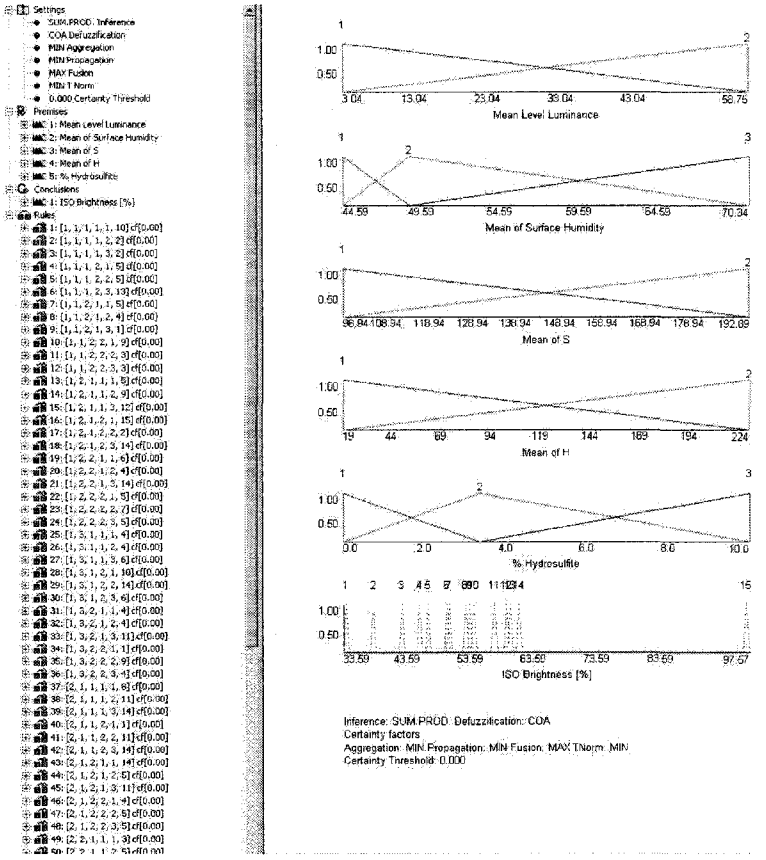


Figure 5.10 Genetically approximated FKB for the *FLearnH*

was obtained is illustrated in Fig. 5.10 and has the following structure:

- 2 fuzzy sets on the *mean level luminance*, *mean of H* and *mean of S* premises;
- 3 fuzzy sets on the *mean of surface humidity* and the *% of Hydrosulfite*;
- 15 fuzzy sets on the conclusion *ISO Brightness*;
- 72 fuzzy rules (rather than the possible 7776 fuzzy rules).

The FKB approximates the data in the *FLearnH* with a 3.58% $\Delta_{rms}$ RMS error on the ISO brightness. As done for the peroxyde file, the *FTestH* is screened through

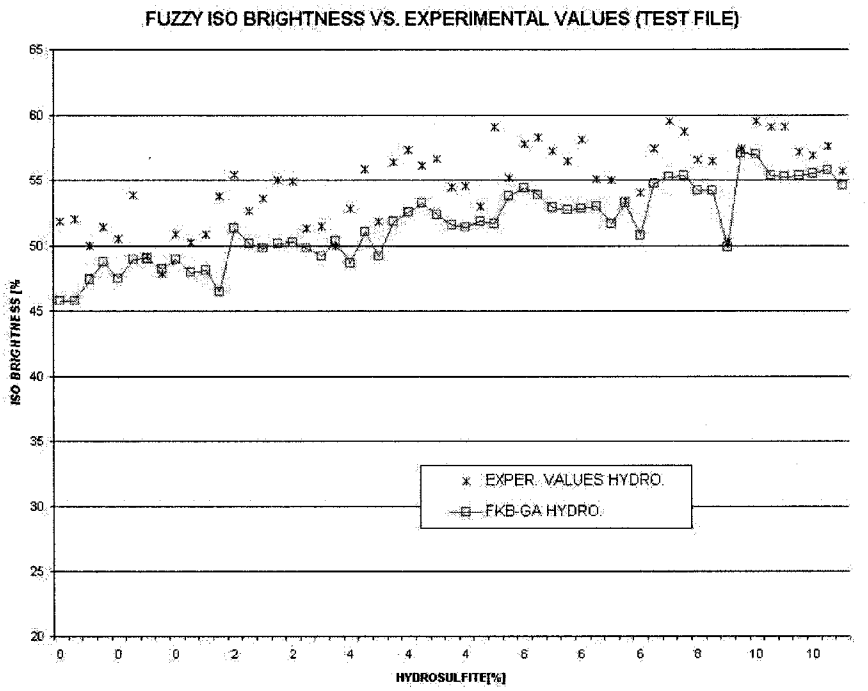


Figure 5.11 Results obtained by the GA-FKB for the *FTestH*

the new FKB, and a 3.77% $\Delta_{rms}$ is obtained (see Fig. 5.11). The learning and testing errors are very close. The $\Delta_{rms}$ error obtained for both learning and testing is below the experimental error of the experimental plan that has been evaluated at 5%.

### 5.10.2 Discussion and results

In this section, we will discuss some of the results obtained by the genetically generated FKBs. One has to analyze the influence/relationships of the different input/output variables. Both FKBs obtained for peroxyde and hydrosulfite are satisfactory when it comes to $\Delta_{rms}$. Because of the similarities of the FKBs and

the peroxyde FKB achieving a higher prediction of the ISO brightness, only the peroxyde FKB is considered in the following sections.

#### 5.10.2.1 Range of the input and output variables : Peroxyde

The range of the input variables is predefined and fixed, even though the exploration option used in the reproduction mechanism can produce values outside this range. A fonction was added, whithin the RBCGA, to fit the values if they overpass the original input range. Meanwhile, the COG (center of gravity) is used on the conclusion as a deffuzification procedure which may produce fuzzy sets out of the experimental data range. These values are not eliminated since they can help to map the predictions, and in some cases, they also allow one to respond to a set of observations that are not included in the learning set.

To analyze the differences between the fuzzy and the experimental prediction ranges, we consider the input variables worth the maximal and minimal ISO brightness of the learning sets, and we screen these inputs through the FKB. The result is presented in Table 5.2.

One can observe that the values are at the same level, however the maximal and minimal values are not reached by the FKB, which can be considered as a negative aspect since the FKB does not cover the complete range of the experimental data.

Table 5.2 Ranges of the output based on experimental data

| | MIN | MAX |
|---|---|---|
| Experimental values (FLearnP) | 43.79% | 79.70% |
| FKB values (FLearnP) | 49.18% | 75.71% |
| Experimental values (FTestP) | 50.12% | 75.41% |
| FKB values (FTestP) | 49.70% | 72.20% |

However, one has to consider the fact that the set of inputs translated into the maximal and minimal values is a very small percentage of the learning and testing files, which can explain why the RBCGA overlooked them. Now the results are sorted based on the answers of the FKB rather than the testing file. The results are shown in Table 5.3. The highest value provided by the FKB is 80.2%, which

Table 5.3 Ranges of the output based on the FKB's predictions

| | MIN | MAX |
|---|---|---|
| Experimental values (FLearnP) | 47.56% | 79.70% |
| FKB values (FLearnP) | 47.67% | 75.70% |
| Experimental values (FTestP) | 51.44% | 73.25% |
| FKB values (FTestP) | 49.30% | 80.20% |

is close to the highest experimental value of 79.70% and the input values giving these two results are quite close, while, the lowest fuzzy value of 47.67% is higher than the minimal value of the experimental data (43.79%). However, in analyzing the FKB illustrated in Fig. 5.8, we can see that the range of the output is between 43.57% (fuzzy set # 1) and 102.14% (fuzzy set # 9). The first fuzzy set is used several times by the fuzzy rule base and it's value is close to what we expect as a minimum (from the learning set point of view). The last fuzzy set is called only

once by the fuzzy rule base and even if the 102.14% is an absurd value (more than 100%) it remains very helpful since it allows the FKB to predict higher values of ISO brightness while using new combinations of wood chips properties.

## 5.11 Learning the FKBs using laboratory variables

The FKBs developed have exclusively used $CMS©$ variables as input variables. However, some very important and influent variables are missing from these FKBs, namely; the percentage of bark, the percentage of the knots and other impurities. The problem with these kinds of variables is that they have to be measured in a laboratory. The sampling of the wood chips has to be analyzed in order to evaluate the percentage of bark, knots and other dark impurities. This necessity of having to analyze samples presents two major drawbacks:

- one can't predict the quality of the pulp online, since the measure is processed offline;
- the quality of the measurements depends on the quality of the sampling choice.

The $CMS©$ provides a variable named the *% of dark chips*, which represents the % of dark spots in the image obtained from the vision system incorporated in the $CMS©$. We assume that the *% of dark chips* represents the amount of barks, knots, decay and other impurities present in the wood chips. This new variable

will replace the color analysis (to avoid evident correlations). However, the average of Luminance is kept due to its high importance. The new input variables are:

- the average of Luminance;
- the % of dark chips;
- % of bleaching concentration agent (peroxyde or hydrosulfite).
- average of surface moisture;

The output remains the ISO brightness of the pulp.

### 5.11.1 Learning of the FKBs for Brightness Prediction using Dark Chips %

The same method of deviding data set files is used and the proportions of the testing and the learning files are kept identical. The set of learning data is named *FLearnDC* and the file containing the remaining 10%, *FTestDC*.

#### 5.11.1.1 Application to the *FLearnDC*

The learning file containing the peroxyde is named *FLearnDCP*, and the one containing the hydrosulfite is named *FLearnDCH*. The testing file containing the peroxyde is named *FTestDCP*, and the one containing the hydrosulfite is named *FTestDCH*.

## A. Results obtained for the *FLearnDCP*

The application of the RBCGA to the *FLearnDCP*, produced an FKB that approximates the values with a 2.19% $\Delta_{rms}$ error on the ISO brightness, which is comparable to the one obtained for the *FLearnP*. The obtained FKB has the fol-

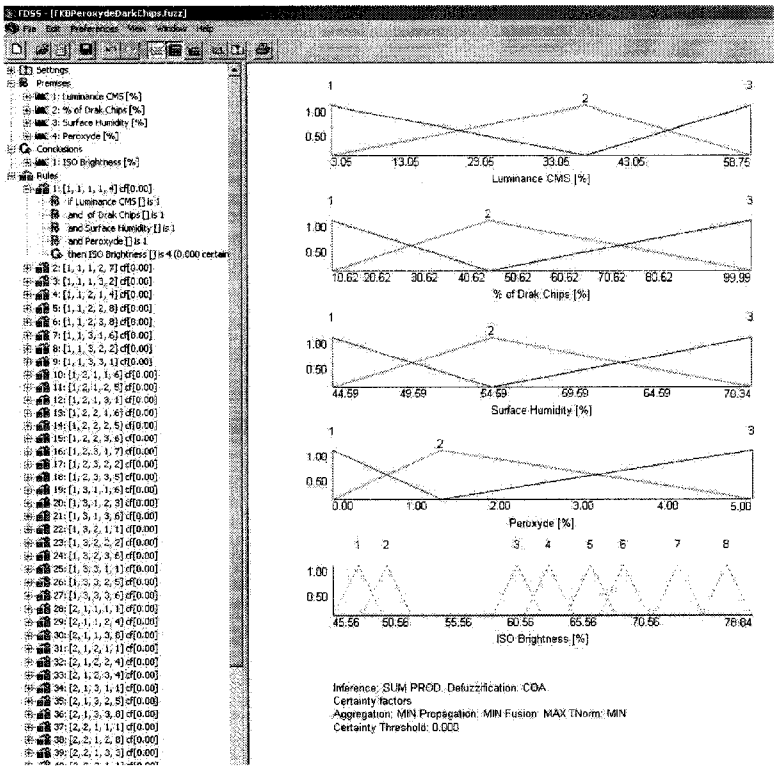


Figure 5.12 Genetically approximated FKB for the *FLearnDCP*

lowing structure:

- 3 fuzzy sets on each input premise;
- 9 fuzzy sets on the conclusion *ISO Brightness*;
- 81 fuzzy rules (rather than the suggested 1293 fuzzy rules).

Figure 5.12 shows a screen print-out of the FKB that was obtained, and opened with FDSS Fuzzy-Flou software. Using this FKB, we proceed the *FTestDCP* test file as an observation file. Figure 5.13 shows the differences between the predicted ISO brightness and the experimental values.

The $\Delta_{rms}$ obtained for the testing file is 1.65%, which is very satisfactory and

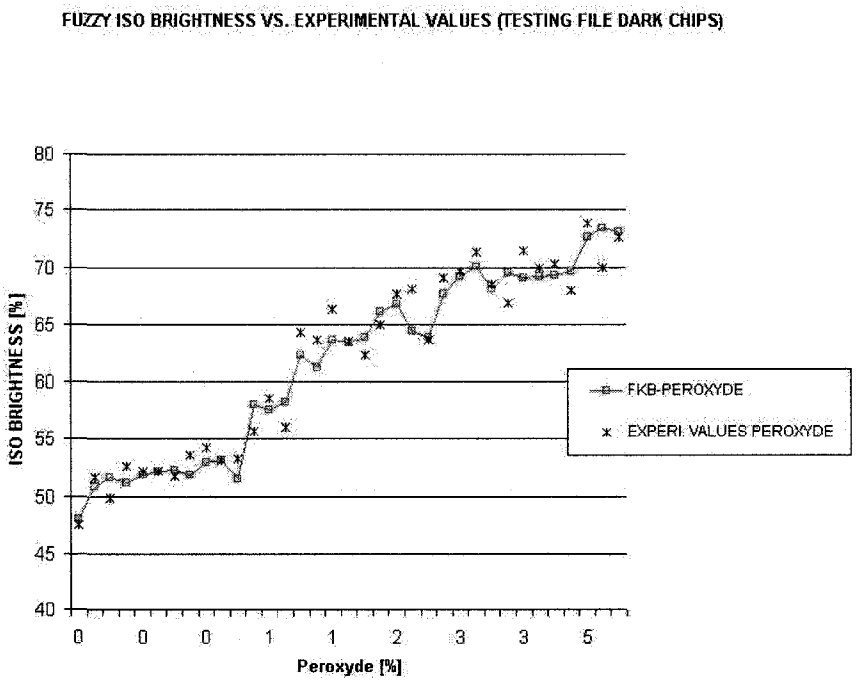


Figure 5.13 Results obtained by the GA-FKB for the *FtestDCP*

surprisingly lower than the learning error (generally the learning is more precise), this result can be explained by the different sizes of the files (the learning file being much larger than the testing file). However, both learning and testing errors are of the same magnitude. The error obtained for both learning and testing files is

below the experimental error of 5%.

### B. Results obtained for the *FLearnDCH*

The application of the RBCGA to the *FLearnDCH* produced a similar FKB to the one obtained for the *FLearnDCP*, in terms of the level of complexity and the distribution of the fuzzy sets. The obtained FKB as illustrated in Fig. 5.14 has the

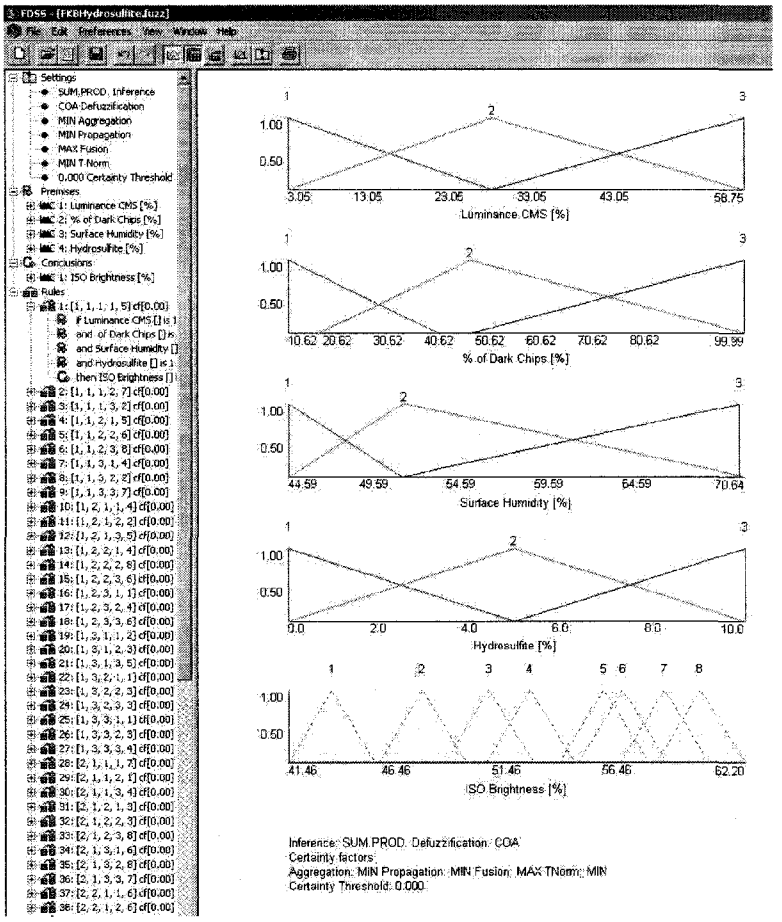


Figure 5.14 Genetically approximated FKB for the *FLearnDCH*

following structure:

- 3 fuzzy sets on each premise;

- 8 fuzzy sets on the conclusion *ISO Brightness*;
- 81 fuzzy rules (rather than the suggested 1293 fuzzy rules).

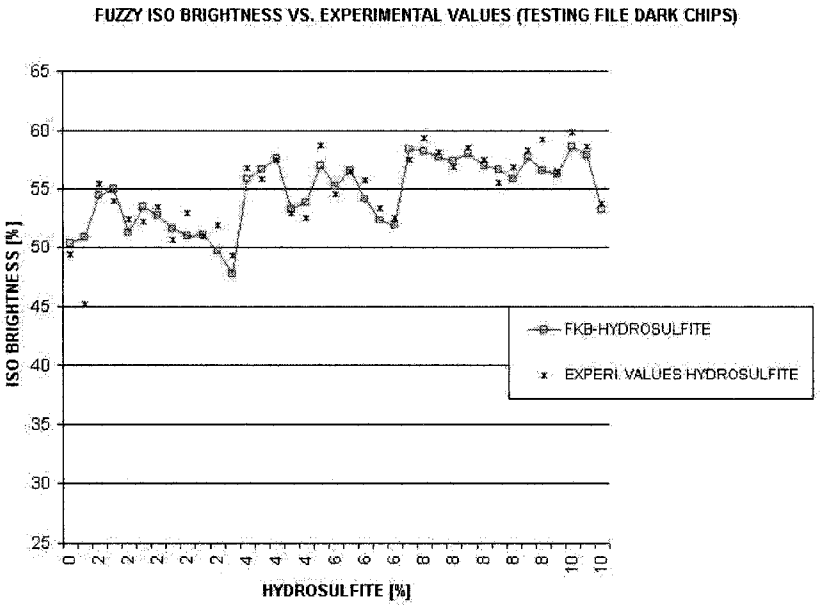


Figure 5.15 Results obtained by the GA-FKB for the *FTestDCH*

The FKB approximates the data in the *FLearnDCH* with a 1.34% $\Delta_{rms}$ on the ISO brightness which is less than half the error obtained for the *FlearnH*. The hydrosulfite testing file is screened through the new FKB. Figure 5.15 shows the difference between the testing and the experimental values of the ISO brightness. An $\Delta_{rms}$ of 1.45% is obtained. The testing and learning errors are at the same level, in this case the testing error is slightly higher which is more predictable since generally the FKBs react better to the learning than to the testings files. The RMS obtained for both learning and testing is below the experimental error of 5%.

As for the peroxyde, using the % of dark chips as an input data for the learning improved significantly the FKB behavior.

### 5.11.2 Discussion and results

Because of the similarity of the peroxyde and hydrosulfite FKBs we only analyze the peroxyde one.

#### 5.11.2.1 Range of the input and output variables : Peroxyde

Taking into account the set of input values giving the maximal and minimal experimental ISO brightness and their equivalent given by the FKB, the results that were obtained are summarized in Table. 5.4. The results show that the values are

Table 5.4 Ranges of the output based on experimental data

| | MIN | MAX |
|---|---|---|
| Experimental values (FLearnPDC) | 43.79% | 79.70% |
| FKB values (FLearnPDC) | 47.88% | 75.71% |
| Experimental values (FTestPDC) | 48.04% | 73.37% |
| FKB values (FTestPDC) | 47.54% | 70.00% |

still comparable, however the maximum and minimum values are not reached by the FKB, same as in the previous section and identical conclusions can be drawn. The data is now sorted based on the answers of the FKB rather than the set of experimental data, the results shown in Table 5.5. The highest value predicted by the FKB is 77.00%, which is closer to the 79.70%, but still a bit far, and the same goes for the minimum value of 47.54% when compared to 43.79%. However the set

Table 5.5 Ranges of the output based on the FKB's predictions

| | MIN | MAX |
|---|---|---|
| Experimental values (FLearnPDC) | 45.04% | 77.00% |
| FKB values (FLearnPDC) | 47.70% | 74.40% |
| Experimental values (FTestPDC) | 48.04% | 72.65% |
| FKB values (FTestPDC) | 47.54% | 73.80% |

of input data providing these extreme values represents a very small percentage of the experimental data. For instance, the combination giving 79.70% ISO brightness appears only once in approximately 500 lines. From the FKB illustrated in Fig. 5.12, one can see that the range of the output is between 45.56% (fuzzy set # 1) and 78.84% (fuzzy set # 9), both values being close to the minimum/maximum ISO brightness of the experimental data and the set of input variables resulting into those two values are of close range to the one giving the experimental extrema. Hence, the FKBs can be considered as satisfactory.

## 5.12 Conclusion

The Real Binary-like Coded Genetic Algorithm (RBCGA), using the proposed evolutionary paradigm, has shown efficiency in producing fuzzy knowledge bases (FKBs) to predict ISO brightness of the pulp. The RBCGA produced precise yet simple FKBs which is a contradictory paradigm. The produced FKBs showed a high stability level of generalization, when taking into account the very small changes in the root mean square error (RMS) between the learning and the testing data, and this for both sets of input variables. From a structure point of view

(fuzzy sets repartition, fuzzy rules, number of fuzzy rules etc.), both FKBs using peroxyde and hydrosulfite are very similar which leads one to believe that both bleaching materials have comparable effects on the bleaching process. However, simply by observing the universe of discourse of the FKBs on the conclusions, we can conclude that the peroxyde allows us to obtain higher levels of ISO brightness for both sets of input variables.

The use of the *% dark chips* rather than the hue and saturation in the learning lead to a much lower RMS error for both the learning and testing files, which can be explained by the importance of the presence of barks, knots and other dark impurities in the wood, knowing that the quality of the wood chips influences its color and hence the *% of dark chips*. The FKBs obtained in this research can't be used yet, in their actual state, to control the bleaching process since their first role is to support a decision system (i.e. to predict). However with a solution search algorithm, one can easily use the genetically generated FKBs to transform any of the inputs into an output by reconstructing the hyperplane allowing to map an input prediction variable into a controlled one.

**Acknowledgement**

The authors want to acknowledge the partners of the Centre d'Excellence pour la Valorisation des Copeaux (MNRFQ, PRECARN, UQTR, CRIQ, ABITIBI div.

BELGO, KRUGER, Trois-Rivière and BOWATER Gatineau). Financial support from Centre de Recherche Industrielle du Quebec and the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618 and RGPIN-105518 is gratefully acknowledged.

# REFERENCES

[1] Achiche, S., Balazinski, M. and Baron, L., "Multi-combinative strategy to avoid premature convergence in genetically-generated fuzzy knowledge bases", *Journal of Theoretical and Applied Mechanics*, No. 3, Vol 42, 2004 pp. 417–444, 2004.

[2] Achiche, S., Baron, L. and Balazinski, M., "Real/Binary-Like Coded Versus Binary Coded Genetic Algorithms to Automatically Generate Fuzzy Knowledge Bases: A comparative study", *Engineering Applications of Artificial Intelligence*, Vol. 17, No. 4, pp. 313–325, 2004.

[3] Achiche, S., Baron, L. and Balazinski, M., "Scheduling Exploration/Exploitation Levels in Genetically-Generated Fussy Knowledge Bases", *IEEE/NAFIPS International Conference on Fuzzy Systems*, Banff, 2004.

[4] Achiche, S., Balazinski, M. and Baron, L., "Génération automatique par un algorithme génétique de bases de connaissances pour un systèmes d'aide à la décision", *Proceeding of the* $3^{rd}$ *International Conference on Integrated Design and Manufacturing in Mechanical Engineering*, Montreal, 2000.

[5] Balazinski, M. and Jemielniak, K., "Tool Conditions Monitoring Using Fuzzy-Decision Support", *Proceedings of the V International Conference on Monitoring and Automatic Supervision In Manufacturing*, pp. 115–122, 1998.

[6] Balazinski, M., Kops, L. and Massicotte, P., "Job Dispatching Using Priority Factors And Fuzzy Logic", *ICME 98 CIRP International Seminar on Intelligent Computation in Manufacturing Engineering, Capri, Italy.*, pp. 147–153, July 1998.

[7] Balazinski, M., Bellerose, M. and Czogala, E., "Application of fuzzy logic techniques to the selection of cutting parameters in machining processes", *Fuzzy sets and systems*, Vol. 61, pp. 301–317, 1994.

[8] Baron, L., "Genetic Algorithm for Line Extraction", *Rapport technique EPM/RT-98/06*, École Polytechnique Montréal, 1998.

[9] Becker, E., "The Effects of Chip Thikness for Kraft Cooking Conditions on Kraft Pulp Propreties", *TAPPI Pulping Conference*, pp. 561–565, 2001.

[10] Bédard, P., Ding, F. and Benaoudia, M., "Amélioration de la cours à bois par la caractérisation en ligne des copeaux", *Congrès francophone du papier*, pp. 11–16, 2003.

[11] Bonissone, P.P., Khedkar, P.S. et Chen, Y.T., "Genetic Algorithms for Automated Tunning of Fuzzy Controllers, a transportation application", *Fifth International Conference on Fuzzy Systems (FUZZ-IEEE'96)*, pp. 674–680, 1996.

[12] Braaten , K.R, Palm, A. and Omlhot, K., "Wood Classification Leads to More Uniform TMP", *Proceeding of the International Mechanical Pulping Conference"*, pp. 23–31, 1993.

[13] Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T., "A Fast and Elitist Multi-Objective Genetic Algorithm: NSGA-II", *IEEE Transactions on Evolutionary Computation*, V. 6, pp. 182–200, 2000.

[14] Dupinet, E., Balazinski, M. and Czogala, E., "Tolerance Allocation based on fuzzy logic and simulated annealing", *Journal of Intelligent Manufacturing*, Vol. 7, pp. 487–497, 1996.

[15] Herrera, F. and Lozano, M., "Gradual Distributed Real-Coded Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 4, pp. 43–63, 2000.

[16] Koran, Z., "The Effects of Density and CSF on the Tensil Strength of Paper", *TAPPI Journal*, Vol. 77:6, pp. 167–170, 1994.

[17] Koran, Z. and Nombi, K., "PTM Pulping Characteristics of Budworm Killed Trees". *WPDD Fiber Science*, Vol. 26, No. 4, pp 489–495, 1994.

[18] Kranks, N.E., "The Effects of Using Budworm-damadged Wood on Mechanical Quality and Yeild", *Proceedings: Recent Advances in Spruces Picea-Fir Abies*

*Utilization Technology*, Society of American Foresters, No. 83–13, pp. 143-149, 1983.

[19] Lanouette, R., Bedard, P. and Benaoudia, M., "Effect of Woodchips Characteristics on the Pulp and Paper Properties by the use of PLS Analysis", $90^{th}$ *Annual Meeting - Pulp and Paper Technical Association of Canada*, pp. 39–43, 2004.

[20] Lanouette, R., Benaoudia, M. and Bedard, P., "Amélioration de la Stabilité des Raffinneurs et de la Qualité de la Pâte par un Suivi des Copeaux", Congrès Francophone du Papier, pp. 10–16, 2003.

[21] Lanouette, R., Larochelle, G. and Nader, J.A., "Impact de la carie de sapin et de l'èpinette blanche sur la mise en pâte par le procédé thermomécanique", *Conférence Technicnologique Estivale*, pp. 77-77, 2000.

[22] Lobo F. G., Goldberg D. E.and Pelikan M., "Time complexity of genetic algorithms on exponentially scaled problems", *GECCO-2000: Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 151–158, 2000.

[23] Michalewicz, Z., *Genetic Algorithms + data structure = evolution programs*, New York: Springer, 1992.

[24] Nault, M.L., Aubin, C.E, Achiche, S. and Labelle, H., "Fuzzy Logic to Assist the Planning in Adolescent Idiopathic Scoliosis Instrumentation Surgery"

*North American Fuzzy Information Processing Society (NAFIPS) International Conference on Fuzzy Systems*, pp. 401–407, 2004.

[25] Shariff, A. and Luu, N.H., "Defining Optimum Chip Size for Market Softwood Kraft Mill using Empirical Model Developped from Layered Cooking of Chips", *87th Annual Congress PAPTAC*, pp. 13–17, 2001.

[26] Wood, J.A., "Chip Quality Effects in Mechanical Pulping - A selected Review", *Proceedings TAPPI Pulping Conference*, pp. 561–565, 1992

[27] Zadeh, L.A., "Outline of new approach to the analysis of complex systems and decisions processes", *IEEE Transactions of Systems, Man and Cybernetics*, Vol. 3, pp. 28–44, 1973.

# CHAPITRE 6

# DISCUSSION GÉNÉRALE

L'objectif général de cette thèse était de proposer une méthode de génération automatique de bases de connaissances floues (BC) fonctionnant selon le paradigme contradictoire qui réside dans l'obtention de BC précises (erreur définie) et à complexité minimale. En premier lieu, cet objectif a nécessité le développement d'un algorithme capable de traiter simultanément la génération des deux parties principales composant les BC, soit : la base de faits (BF) et la base de règles floues (BR), sans pour autant disposer de connaissances sur le phénomène à modéliser en BC autres que le nombre d'entrées\sorties et les données numériques d'apprentissage. Aussi, l'idée était de pouvoir intégrer à même l'algorithme la distinction qui existe entre le type de variables composant ces deux parties, soit : des nombres réels pour la BF et des nombres entiers pour la BR. L'algorithme d'optimisation présenté dans cette thèse est un algorithme métaheuristique, en l'occurrence un algorithme génétique (AG). Le choix de l'utilisation de l'AG se basait sur le fait que le problème à résoudre est un problème de grande taille qui peut éventuellement contenir plusieurs objectifs (multi-objectifs). Aussi, un AG est assez flexible quant à la modification de ses mécanismes de reproduction, de mutation, etc. L'aspect évolutif des AG fut aussi un point prépondérant dans le choix de celui-ci. Une fois

le choix de l'algorithme fait, il reste le choix du paradigme de codage à utiliser, ce choix ne représente pas un simple artifice numérique, mais plutôt une orientation de l'évolution de l'AG du fait que les mécanismes de croisement sont souvent très différents d'un paradigme à l'autre. Dans le cadre de l'article 1, il s'agit d'un AG hybride, soit un AG codé aux nombres réels et binaires (noté AGCRB) qui traite la BF et la BR simultanément, mais utilise des génotypes indépendants : un génotype de nombres réels pour la BF et un génotype de nombres entiers pour la BR. Il s'agit là de la première combinaison de ces deux paradigmes de codage dans un AG.

Les performances de l'AGCRB ont été comparées à celles de l'AG codé en binaire (noté AGB), sur plusieurs critères importants. L'AGCRB a été particulièrement performant quand il a fait face à des formes de surfaces complexes, avantage dû au fait que le génotype du AGCRB n'est pas transformé en phénotype contrairement à celui de l'AGB (contrainte de résolution sur les solutions). Dans presque tous les cas, l'AGCRB a produit des BC avec un indice de performance plus élevé que celle proposée par l'AGB. Concernant le temps d'apprentissage, l'AGB s'exécute plus vite pour une même population et un même nombre de générations, ce qui était prévisible vu le type de codage (les paramètres réels (des doubles) utilisant plus d'espace mémoire que les 4 bits alloués à chaque paramètre dans l'AGB). Néanmoins, comme montré au chapitre 3, pour que l'AGB atteigne un même niveau de performance que l'AGCRB, il lui faut consommer plus de temps d'appren-

tissage. Les deux approches ont des comportements comparables quant au critère de simplification des BC. Simplifier la BC tout en gardant un niveau de précision définie, est contradictoire, dans la première version du AGB proposée dans [4,16], une pondération est utilisée entre deux critères de performance : l'indice de performance de précision et l'indice de performance de simplicité. L'existence de ces deux critères fait que la recherche d'une BC n'est plus une solution unique mais un ensemble de solutions, connu comme l'ensemble de solution Pareto Optimal (problème multi-objectif). Une fois l'ensemble de solutions Pareto Optimal obtenu, il reste au décideur de choisir laquelle des solutions il veut utiliser. Pour régler ce problème, le deuxième critère de performance basé sur la simplicité a été tout simplement éliminé dans l'AGCRB et à la place nous avons utilisé un mécanisme de croisement, soit : la réduction des sous-ensembles flous. Il s'agit là d'une approche originale qui consiste à transformer un problème Pareto Optimal en un problème uni-objectif, sans utiliser aucune pondération des critères de performances, en respectant l'évolution naturelle des solutions vers un optimum global et en prenant garde du double paradigme de simplicité et de précision des BC. Dans le cadre restreint de l'article, des surfaces synthétiques en trois dimensions ont été utilisées, ce qui donne lieu à des BC à deux entrées et une sortie ; par contre, dans le cadre plus général de cette recherche, les deux AGs peuvent s'appliquer à n'importe quel système du type *Multiple Input Single Ouptput*(MISO).

L'AGCRB est un meilleur choix et une approche plus performante que l'AGB. Néanmoins, l'AGCRB souffre du même défaut que presque tous les algorithmes métaheuristiques : la convergence prématurée. Dans le chapitre 4, nous traitons des deux raisons principales qui causent la convergence prématurée, soit :

- le manque de diversité dans la population de solutions ;
- le mauvais balancement entre l'exploration et l'exploitation dans la population de solutions.

Le manque de diversité dans la population de solutions a été traité par les stratégies de combinaison linaire et d'application simultanée, qui ont donné de bons résultats en augmentant l'indice de performance des BC générées, tout en utilisant des population de petite taille. Néanmoins, d'un point de vue de temps de calcul, la stratégie de combinaison linaire est bien plus rapide, vu qu'elle ne produit que deux enfants par paire de parents contrairement à l'application simultanée qui en produit 6 différents. Pour l'approche du contrôle de balancement de l'exploitation et de l'exploration, l'hypothèse prise était que la gestion de cet équilibre devrait dépendre du besoin de changement dans la population de solutions à un moment donné de son évolution. Dans la plupart des cas, l'exploitation relaxée ($\alpha = 0.5$) est appliquée tout au long de l'évolution, ce qui traite les différents stages de l'évolution des solutions d'un pied égal. Cette monotonie de l'apprentissage n'est pas, dans le sens de l'auteur, la meilleure des stratégies d'évolution car il y a une différence entre les différents stages d'apprentissage même pour un algorithme. C'est, entre

autres, cette piste qui est exploitée dans le chapitre 4.

Afin de déterminer quel ordre de balancement d'évolution était le plus efficace, deux approches ont été proposées selon deux ordres différents, soit :

- commencer par l'exploitation au début de l'évolution puis l'exploitation relaxée pendant la majorité de l'évolution, suivie par l'exploration vers la fin de l'évolution ;
- commencer par l'exploration, puis l'exploitation relaxée suivie de l'exploitation.

Une question a émergé de ces deux approches : quelles parties de l'évolution peuvent être considérées comme étant son début et sa fin (pour un nombre de génération donné) ? La proportion utilisée était du 1/3, ce choix n'a pas été discuté dans le chapitre 4, mais il a été beaucoup plus développé dans le travail fait par l'auteur en [2] (l'article est présenté en annexe 1), duquel est ressorti que la meilleure distribution était aux environs de 1/3 de proportion, donc : le premier 1/3 est considéré comme le premier stade de l'évolution ; le dernier 1/3 comme le dernier stade de l'évolution et entre ces deux se situe le stade de l'évolution.

Des tests effectués avec les deux stratégies de balance de l'exploitation/exploration, il est très clairement ressorti que l'ordre préférentiel était celui de commencer par explorer puis relaxer pour finir par exploiter les connaissances acquises. Aussi, ces

résultats prouvent l'influence positive que peut avoir un bon équilibre entre l'exploitation et l'exploration dans le mécanisme de croisement choisi, sur l'évolution de l'AGCRB. L'application au problème de tournage a prouvé l'efficacité des stratégies évolutives quant à un apprentissage sur des données expérimentales.

Au chapitre 4, plusieurs stratégies évolutives ont été proposées, l'une d'entre elles est sélectionnée et utilisée au chapitre 5, soit : le changement de l'exploration et de l'exploitation du mécanisme blended crossover $\alpha$, avec le bon ordre pour la distribution de l'évolution. Dans le chapitre 5, il s'agit de l'application principale de l'AGCRB, soit : la génération automatique de BC pour la prédiction de la qualité de la pâte thermomécanique, à partir de caractéristiques des copeaux de bois. Avant de générer les BC, il fallait choisir les variables d'entrées et de sortie. La variable de sortie devait représenter la qualité de la pâte produite, le choix s'est rapidement porté sur la blancheur ISO de celle-ci. Pour les variables d'entrées, le travail était plus difficile, le choix s'est fait en se basant sur l'évaluation de l'influence des paramètres aux moindres carrés partiels sur la sortie selectionnée. Les choix ont été validés par les experts du Centre de Recherche Industrielle du Québec (CRIQ). Le premier ensemble d'entrées contenait les valeurs moyennes de la luminosité, de l'humidité surfacique, des paramètres d'image Hue (H) et Saturation (S), ainsi que la concentration de l'élément de blanchiment. Comme cité dans le chapitre 5, il s'agit là de paramètres exclusivement synthétiques, tous captés grâce aux senseurs

dont le *Chips Managment System* ($CMS^{©}$) est doté. Les BC ont été construites en utilisant l'hydrosulfite et le peroxyde comme éléments de blanchiment. Les BC obtenues sont satisfaisantes d'un point de vue erreur aux moindres carrés (RMS), elles ont aussi confirmé l'efficacité supérieure du peroxyde par rapport à l'hydrosulfite à produire de pâtes plus blanches. Par contre, le peroxyde est bien plus coûteux que l'hydrosulfite, de ce fait le fabricant peut choisir, en fonction de la blancheur voulue, lequel des deux éléments de blanchiment choisir en prenant en compte les propriétés des copeaux de bois à l'intrant du procédé.

Le deuxième volet du chapitre 5 explorait la possibilité de remplacer des variables obtenues hors-ligne (en laboratoire), comme le pourcentage de noeuds, le pourcentage d'écorce et le pourcentage de caries dans les copeaux de bois, par des variables synthétiques, dans la perspective de les utiliser dans la prédiction en ligne de la qualité de la pâte, vu que ces variables influent fortement le procédé de transformation du bois en pâte puis en papier. Le $CMS^{©}$ fournit une information sur le pourcentage de copeaux bruns (appelé *% of Dark Chips*), variable obtenue à partir d'analyse d'images. C'est cette variable qui a été considérée. Dans son état actuel, le pourcentage de copeaux bruns considère les différents contaminants comme un ensemble. Les BC obtenues en utilisant cette variable (à la place du H et du S) sont d'une grande précision et d'une grande stabilité (comportement face aux données cachées à l'apprentissage), ce qui porte à croire au bien-fondé de cette approche

originale. Dans le cadre de l'article, c'étaient les seules combinaisons qui ont été testées, mais dans un cadre plus général, d'autres pistes de combinaisons de variables d'entrées ont été explorées, comme par exemple une BC obtenue à partir des paramètres de traitement d'image uniquement (les paramètres HSL), ce qui a donné lieu à l'article [1] présenté en annexe 2. Toutes ces bases de connaissances sont en cours d'implémentation à même le $CMS^{©}$ par le CRIQ, le manufacturier pourra choisir la BC à utiliser de par les variables disponibles.

# CHAPITRE 7

# RECOMMANDATIONS

Dans ce chapitre, quelques recommandations seront énoncées. Elles visent à proposer des idées pour la poursuite de la recherche initiée dans cette thèse.

- Les résultats présentés dans cette thèse ont été menés pour des bases de connaissances (BC) du type *Mutliple Input Single Output* (MISO). Une extension de ce travail vers la génération de BC à plusieurs entrées et plusieurs sorties est souhaitable (Multiple Input Multiple Output—MIMO—). Néanmoins, la construction de la base de règles (BR) pour des systèmes MIMO est souvent très complexe, du fait de la difficulté de lier plusieurs sorties complètement indépendantes par les mêmes règles et la même base de faits (BF). Aussi, la BR d'une BC de type MISO est souvent considérée plus représentative de l'information (les entrées versus la sortie). C'est pourquoi il serait préférable de transformer plusieurs BC de type MISO en une BC de type MIMO. L'algorithme effectuant cette transformation devra prendre en compte et exploiter les similarités déjà existantes entre les différentes BF et BR. Un algorithme métaheuristique serait un choix judicieux pour accomplir

cette tâche, du fait de la taille du problème.

- Les algorithmes génétiques (AG) développés dans cette thèse sont évolutifs, mais les paramètres d'évolution sont définis manuellement par l'utilisateur. Automatiser l'apprentissage des paramètres d'évolution est un sujet délicat, du fait de l'utilisation d'un algorithme pour en faire fonctionner un autre. Cette double imbrication peut engendrer la divergence de l'algorithme en le perdant dans trop de directions d'exploration de l'espace de recherche. Néanmoins, il est possible de trasnformer les AGs proposés dans cette thèse en AGs autoadaptatifs. Pour ce faire, il est conseillé et possible d'utiliser un algorithme de recuit simulé (RS) qui, grâce à sa fonction exponentielle d'exploration aléatoire de l'espace d'état (espace de recherche), permettra des directions de descente sans interdire les remontées (ne pas buter sur un optimum local). Il sera possible d'inclure les paramètres d'évolution de l'AG dans la fonction objectif du RS. Le RS s'exécutera en parallèle à l'AG et les paramètres d'évolution changeront en fonction de l'influence qu'ils auront sur la nouvelle génération obtenue par l'AG. Au départ de l'apprentissage, les paramètres d'évolution peuvent être choisis aléatoirement ou bien en se basant sur des connaissances de l'utilisateur, ce qui peut, éventuellement, permettre d'accélérer la convergence vers des valeurs optimales, et ce pour un meilleur fonctionnement de l'AG auto-adaptatif.

- Le deuxième article qui composait cette thèse a proposé de nouvelles pistes pour réduire (ou éviter) la convergence prématurée dans les AG. D'autres techniques peuvent être explorées dans le cadre de la génération automatique de BC, par exemple, la création de plusieurs populations de départ indépendantes (avec des tailles diverses). L'évolution de chacune des populations se fait indépendamment des autres en utilisant des mécanismes de reproduction différents. Au cours de l'évolution, des mécanismes de migration d'une population à l'autre sont appliqués afin d'augmenter la diversité. Certaines règles et directions de migrations sont à prendre en compte, comme celles proposées en [36].

- Un aspect important dans l'optimisation du design d'un système d'aide à la décision flou (SADF) est la sélection de l'ensemble des variables les plus informatives du problème en étude. L'algorithme qui établit une solution à ce probléme est appelé "Algorithme de sélection". Les algorithmes de sélection sont généralement caractérisés par les trois points suivants :

  1. Un *algorithme de recherche* qui explore le sous-ensemble des variables en présence, la taille de ce sous-ensemble est de $2^d$ où $d$ est le nombre de variables. Les points dans cet espace sont généralement appelés *états.*
  2. Une *fonction d'évaluation* qui donne la mesure de l'acceptabilité d'un

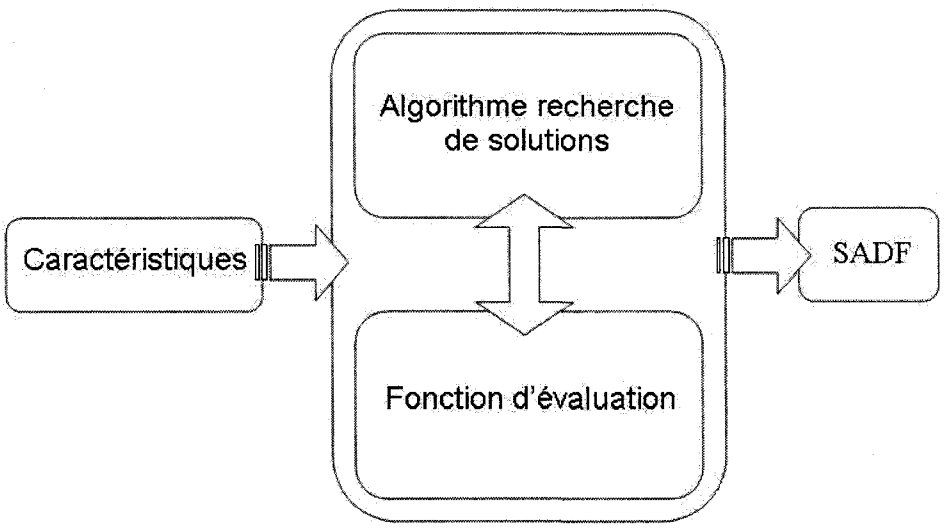


FIG. 7.1 Stratégie de contrôle Filtre

certain sous-ensemble de caractéristiques lors du processus de recherche, elle prend comme entrée un état et donne comme sortie une évaluation numérique, le but de l'algorithme de recherche est de maximiser cette fonction.

3. Un *modèle SADF* ou une *fonction de performance* qui détermine la validité du sous-ensemble obtenu.

Deux approches sont généralement utilisées pour faire la sélection de caractéristiques :

1. stratégie de contrôle de type *Filtre*;
2. stratégie de contrôle de type *Wrapper*.

Comme le montrent les figures 7.1 et 7.2, la différence principale entre les deux approches est l'intervention du modèle du SADF dans le processus.

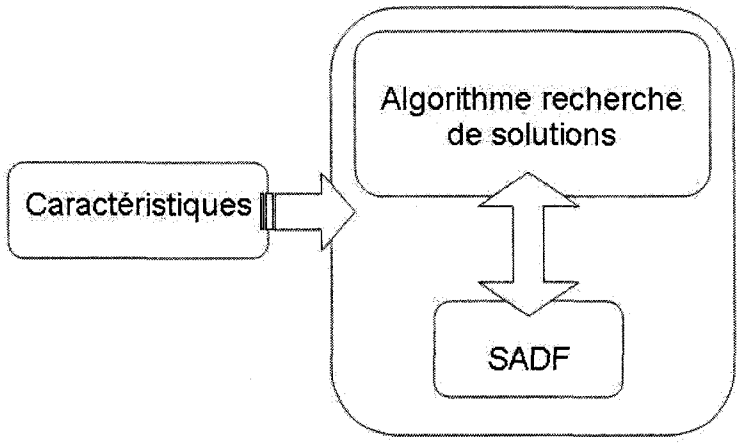


FIG. 7.2 Stratégie de contrôle Wrapper

Pour l'approche "Filtre", le sous-ensemble des caractéristiques est construit avant que le modèle du SADF ne soit consulté; quant à l'approche "Wrapper", le modèle SADF joue le rôle de la fonction d'évaluation.

L'avantage de la stratégie Wrapper est qu'elle évite les problèmes que peut engendrer l'utilisation d'une fonction d'évaluation dont le biais pourrait différer du modèle SADF, d'où sa supériorité comparativement à la stratégie Filtre[40], des résultats empiriques confirment cette hypothèse [6]. Néanmoins, le processus est alourdi car il faut construire une BC pour chaque état. C'est, tout de même, la stratégie de contrôle Wrapper qui semble la plus appropriée. Pour l'algorithme de recherche, un algorithme simple, sélectionnant le nombre de caractéristiques à prendre en considération, appartenant à la catégorie des algorithmes stochastiques de type ***Monte-Carlo*** est approprié au travail

à faire, et ce du fait de l'aspect aléatoire et probabilistique de la tâche à accomplir.

- Le troisième article composant cette thèse concerne l'application au domaine des pâtes et papiers. Les BC générées, pour cette application, se limitaient à la prédiction de la blancheur ISO de la pâte. D'autres aspects du procédé de pâtes thermomécaniques (PTM) peuvent être pris en compte dans le but d'optimiser la transformation des copeaux de bois en pâte puis en papier, l'un de ces paramètres est l'énergie nécessaire à la production de la pâte. Aussi, l'influence d'autres variables d'entrées telles que : la taille des copeaux, la densité des copeaux, etc., peuvent être incluses dans l'apprentissage des BC. Pour définir des sous-ensembles de variables d'entrées, il est conseillé d'utiliser un système de sélection de caractéristiques.

- Récemment, plusieurs développements ont eu lieu au niveau de la théorie de la logique floue, dont la logique floue de type II [46]. La nouvelle approche réside dans l'augmentation de la dimension des sous-ensembles flous qui arborent une largeur leur permettant d'y ajouter une probabilité d'appartenance. Il serait donc intéressant d'explorer les avantages de ce type II, par rapport au type conventionnel et aussi d'explorer les performances des AG sur des systèmes d'aide à la décision flous de type II.

# CONCLUSION

Le travail présenté dans cette thèse a démontré plusieurs faits qui seront énumérés dans ce chapitre.

- L'utilisation d'algorithmes génétiques pour la tâche spécifique de génération automatique de structures de données floues est un choix adéquat.
- L'approche de codage hybride, préconisée dans cette thèse, a démontré l'avantage de personnaliser les algorithmes génétiques en fonction des spécificités du problème à optimiser.
- L'algorithme génétique hybride est plus perfomant que l'approche traditionnelle de codage binaire, et ce sur tous les critères de comparaisons étudiés dans cette thèse.
- Les algorithmes génétiques proposés ont été à même de construire des bases de connaissances au mieux de deux critères contradictoires, à savoir : minimiser l'erreur et minimiser la complexité.
- La personnalisation des mécanismes de croisement dans l'algorithme hybride (réduction de sous-ensembles flous) est une alternative à la résolution du problème Pareto Optimal posé par le besoin d'optimiser des fonctions multi-objectifs (bi-objectifs dans le cas de cette thèse).
- Le travail fait autour de la problématique de la convergence prématurée a fait ressortir deux points importants, soit :

  - les stratégies multicombinatoires améliorent la diversité au sein de la population de solutions d'un algorithme génétique ;
  - une meilleure diviersité dans un algorithme génétique réduit les effets de la convergence prématurées sur ses perfomances ;
  - un bon choix d'équilibre entre l'exploitation et l'exploration le long de l'évolution d'un algorithme génétique améliore ses performances dans son exploration de l'espace de recherche.
- L'usage des caractéristiques des copeaux de bois comme moyen de prédiction de la qualité de la pâte obtenue par le procédé des pâtes thermomécaniques (PTM), est possible par le biais de l'utilisation de bases de connaissances floues générées génétiquement.
- Les bases de connaissances développées, utilisant des variables obtenues à partir d'analyse d'images et d'un capteur aux infra rouges, ont prédit de façon satisfaisante la blancheur ISO de la pâte. Ces résultats apportent une preuve de la possibilité de mise en oeuvre de ce type de systèmes de prédiction utilisant des informations en aval du procédé PTM (données sur les copeaux de bois) pour prédire des valeurs propres à la qualité du produit obtenu (blancheur ISO de la pâte).
- La possibilité de remplacement de certaines variables traditionnellement obtenues en laboratoire—contaminants—par une variable synthétique—*% of dark chips*—a été explorée avec succès.

# RÉFÉRENCES

[1] Achiche, S., Balazinski, M., Baron, L. et Benaoudia, M., "Chips Image Processing to Predict Pulp Quality Using Fuzzy Logic Techniques", *91st Annual Meeting of the Pulp and Paper Technical Association of Canada*, 07-10 février 2005.

[2] Achiche, S., Baron, L. et Balazinski, M., "Scheduling Exploration/Exploitation Levels in Genetically-Generated Fussy Knowledge Bases", *North American Fuzzy Information Processing Society (NAFIPS)*, pp. 34–37, 2004.

[3] Achiche, S., Balazinski, M. et Baron, L. "Real/Binary-Like Coded Genetic Algorithm to Automatically Generate Fuzzy Knowledge Bases", *IEEE Fourth International Conference on Control and Automation*, pp. 799–803, 2003.

[4] Achiche, S., Balazinski, M., Baron, L. et Jemielniak, K., "Tool Condition Monitoring using Genetically-generated Fuzzy Knowledge Bases", *Engineering Applications of Artificial Intelligence*, 2002.

[5] Achiche, S., Balazinski, M. et Baron, L., "Génération automatique par algorithme génétique de bases de connaissances pour un système d'aide à la décision", *International Conference on Integrated Design and Manufacturing in Mechanical Engineering*, 2000.

[6] Aha, D. W., et Bankert, R. L., "Feature Selection for Case-based Classification of Cloud Types : An Empirical Comparison.", *In D. W. Aha (Ed.) Case-Based Reasoning : Papers from the 1994 Workshop (Technical Report WS-94-01). Menlo Park, CA : AAAI Press.*, 1994.

[7] Ayel, M. et Rousset, M.C., "La cohérence dans les bases de connaissances", *Editions Cepadues*, Toulouse, 1984.

[8] Badiru, A.B. et Cheung, J.Y., "Fuzzy Engineering Expert Systems with Neural Network Applications", *John Wiley & Sons, Inc.*, New York, 2002.

[9] Balazinski, M., Czogala E. et Sadowski T., "Modelling of Neural Controllers with Application to the Control of a Machining Process", *Fuzzy Sets and Systems 56*, pp. 273–280, 1993.

[10] Balazinski, M., Bellerose, M. et Czogala, E., "Decision Support System for Cutting Parameters Selection in Machining Processes using Fuzzy Logic Knowledge", *Proceeding of the Fifth IFSA World Congres*, Vol. 2, pp. 798–801, 1993.

[11] Balazinski, M. et Klim, Z., "Étude sur l'application de la logique floue pour la prédiction de la maintenance préventive", *International Industrial Engineering Conference*, Vol. 2, pp. 1133–1142, 1995.

[12] Balazinski, M., Achiche, S. et Baron, L., "Influence of Optimization and Selection Criteria on Genetically-Generated Fuzzy Knowledge Bases", *International Conference on Advanced Manufacturing Technology*, pp. 159–164, 2000.

[13] Balazinski, M., Achiche, S. et Baron, L., "Application of Genetically-Generated Fuzzy Knowledge Bases in Manufacturing", *9th International Fuzzy Systems Association World Congress*, 2001.

[14] Balazinski, M., Achiche, S., Baron, L., Elektorowicz, M., et El-Agoudy, A., "Investigation of Constructed Wetlands Efficiency in Mercury Removal Using Genetically-Generated Fuzzy Knowledge Bases", *9th Canadian Society of Civil Engineering, 2001 Annual Conference*, 2001.

[15] Baron, L., "Genetic Algorithm for Line Extraction", Rapport technique EPM/RT-98/06, École Polytechnique de Montréal, 1998.

[16] Baron, L., Achiche, S. et Balazinski, M., "Fuzzy Decision Support System Knowledge Base Generation using a Genetic Algorithm", *International Journal of Approximate Reasoning*, NAFIPS, 2001.

[17] Bezdek, J.C., "Fuzzy Models - What are They, and Why?", *IEEE Transactions on Fuzzy Systems*, Vol. 1, No. 1, pp. 1–5, 1993.

[18] Bonissone, P.P., Khedkar, P.S. et Chen, Y.T., "Genetic Algorithms for Automated Tunning of Fuzzy Controllers : A Transportation Application", *Fifth International Conference on Fuzzy Systems (FUZZ-IEEE'96)*, New Orleans, pp. 674–680, 1996.

[19] Carraway, R.L., Morin, T.L. et Moskowitz, H., "Generalized Dynamic Programming for Stochastic Combinatorial Optimization", *Operations Research*

*Journal*, Vol. 37, No. 5, pp. 819–829, 1990.

[20] Casillas, J., Cordon, O., et Herrera, F., "A Methodology to Improve Ad Hoc Data-driven Linguistic Rule Learning Methods by Including Cooperation among Rules", *Technical Report #DECSAI-000101*, 2000.

[21] Chiu, S., "Fuzzy Model Identification Based on Cluster Estimation", *Journal of Intelligent Fuzzy Systems*, Vol. 2, pp. 267–278, 1994.

[22] Cornelius, T.L., "Knowledge-Based Systems : Techniques and Applications", *Academic Press*, Vol. 1, 2000.

[23] Cordón, O., Herrera, F., Villar, P., "Applying Genetics to Fuzzy Logic", *AI Expert*, Vol. 6, No. 3, pp. 38–43, 1991.

[24] Cordón, O., Herrera, F. et Villar, P. "Analysis and Guidelines to Obtain a Good Uniform Fuzzy Partition Granularity for Fuzzy-rule Based Systems using Simulated Annealing", *International Journal of Approximate Reasoning*, pp. 187–216, 2000.

[25] De Jong, K. A.. "An Analysis of the Behavior of a Class of Genetic Adaptive Systems", *PhD Dissertation*, University of Michigan, 1975.

[26] Diederich, J. and Renaud, F., "A Fuzzy classifier using genetic algorithms for biological data", *Proceedings of the 8th International Conference of the North American Fuzzy Information Processing Society - NAFIPS - New York, BEE*, pp. 680–684, 1999.

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

[28] Dupinet, E., Balazinski, M. et Czogala, E., "Tolerance Allocation Based on Fuzzy Logic and Simulated Annealing", *Journal of Intelligent Manufacturing*, Vol. 7, pp. 487–497, 1996.

[29] Eshelman, L. J. et Schaffer, J. D., "Real- Coded Genetic Algorithms and Interval-Schemata, Foundations of Genetic Algorithms 2", *Morgan Kaufman Publishers, San Mateo*, pp. 187–202, 1993.

[30] Farmerie, W.G.; Hammer, J., Li Liu et Schneider, M., "Having a BLAST : Analyzing Gene Sequence Data with BlastQuest", *Proceedings. 14th International Workshop on Database and Expert Systems Applications*, pp. 37–41, 2003.

[31] Goldberg, D.E. , "Genetic Algorithm in Search, Optimization, and Machine Learning", *Adison-Wesley*, New York, 1989.

[32] Hagras, H., Callaghan, V., Colley, M. and Carr-West, M., "A Fuzzy-Genetic Based Embedded-Agent Approach to Learning & Control in Agricultural Autonomous Vehicles", *Proceedings of the 1999 IEEE International Conference on Robotics & Automation, Detroit, Michigan*, pp. 1005-1010, 1999.

[33] Halgamuge, S. et Glesner, M., "Neural Networks in Designing Fuzzy Systems for Real World Applications", *Fuzzy Sets and Systems*, Vol. 65, No. 1, pp. 1–12,

1994.

[34] Hayashi, I., Nomura, H., Yamasaki, H. et Wakami, N., "Construction of Fuzzy Inference Rules by NDF and NDFL", *International Journal of Approximate Reasonning*, Vol. 6, pp 241–266, 1992.

[35] Herrera, F., Lazano, M. et Verdegay, J.L., "Tuning Fuzzy Controllers by Genetic Algorithms", *International Journal of Approximate Reasoning*, Vol. 12, pp. 299–315, 1995.

[36] Herrera, F. et Lozano, M., "Gradual Distributed Real-Coded Genetic Algorithms", *IEEE Transactions on Evolutionary Computation*, 2000.

[37] Ishibushi, H., Nozaki, K., Tanaka, H., Hosaka, Y. et Matsuda, M., "Empirical study on learning in fuzzy systems by rice test analysis",*Fuzzy Sets and Systems 64*, pp. 129–144, 1994.

[38] Ishibushi,H., Nozaki K., Yamamoto, N. et Tanaka, N.H., "Selection if-then Rules for Classification Problems using Genetic Algorithms",*IEEE Transactions on Fuzzy Systems*, Vol. 3, No. 3, pp. 260–270, 1995.

[39] Janikow, C.Z., "A Genetic Algorithm for Optimizing Fuzzy Decision Trees", *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 421–428, 1995.

.

[40] John, G., Kohavi, R., et Pfleger, K., "Irrelevant Features and the Subset Selection Problem", *Proceedings of the 11th International Machine Learning Conference*, 1994.

[41] Karr, C.L. et Gentry, E.J., "Fuzzy Control of pH using Genetic Algorithms", *IEEE Transactions of Fuzzy Systems*, Vol. 1, pp. 46–53, 1993.

[42] Lucero, J., "Fuzzy Systems Methods in Structural Engineering", *Ph.D. Dissertation*, University of New Mexico, Department of Civil Engineering, Albuquerque, NM, 2004.

[43] Lee M. A. et Tagaki H., "Integrating Design Stages of Fuzzy Systems using Genetic Algorithms", *Proceedings of the 2nd IEEE International Conference on Fuzzy Systems*, pp. 612–617, 1993.

[44] Mühlenbein, H. et Schleirkamp-Voosen D., "Predictive Models for the Breeder Genetic Algorithms I : Continuous Parameter Optimization", *Evolutionary Computation 1*, pp. 25–49, 1993.

[45] Nault, M.L., Aubin, C.E, Achiche, S. et Labelle, H., "Fuzzy Logic to Assist the Planning in Adolescent Idiopathic Scoliosis Instrumentation Surgery", *North American Fuzzy Information Processing Society (NAFIPS) International Conference on Fuzzy Systems*, pp. 401–407, 2004.

[46] Mendel, J.M., "Uncertain Rule-Based Fuzzy Logic Systems : Introduction and New Directions", *Prentice Hall PTR*, Upper Saddle River, NJ, 2000.

[47] Nomura, H., Hayashi, I. et Wakami, N., "A Self-tuning Method of Fuzzy Reasoning by Genetic Algorithm", *Proceedings of the International Fuzzy Systems and Intelligent Control Conference*, pp. 236–245, 1992.

[48] Pedrycz, W., Gudwin, R. et Gomide,F., "Nonlinear Context Adaptation in the Calibration of Fuzzy Sets", *Fuzzy Sets and Systems*, Vol. 88, pp. 91–97, 1997.

[49] Przybylo, A., Achiche, S., Balazinski, M. et Baron, L., "Enhancing Fuzzy Learning with Data Mining Techniques", *Artificial Intelligence Methods*, 17-19 novembre 2004.

[50] Radakovic, Z.R., Milosevic, V.M. et Radakovic, S.B., "Application of Temperature Fuzzy Controller in an Indirect Resistance Furnace", *Applied Energy*, pp. 167–182, 2002.

[51] Radcliffe, N.J. , "Non-linear Genetic Representations", *Parallel Problem Solving from Nature 2*, R. Männer and B.Manderick (Ed.), Amsterdam, pp. 259–268, 1992.

[52] Ross, T., "Fuzzy Logic with Engineering Applications", *McGraw-Hill*, New York, 1995.

[53] Ross, T., "Fuzzy Logic with Engineering Applications : Second Edition", *John Wiley and Sons, Ltd*, England, 2004.

[54] Roubos, H. et Setnes, M.,"Compact Fuzzy Models Through Complexity Reduction and Evolutionary Optimization",*Proceedings of the Ninth IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 762–767, 2000.

[55] Saaty, T.L, "Measuring the Fuzziness of Sets", *Journal of Cybernetics*, Vol. 4, No. 4, pp. 53–61, 1974.

[56] Sayin, S. et Karabati. S., "A Bicriteria Approach to the Two-machine Flow Shop Scheduling Problem", *European Journal of Operational Research*, pp. 435–449, 1999.

[57] Shannon, C.E., "A Symbolic Analysis of Relay and Switching Circuits", *Transactions of the AIEE*, Vol. 57, pp. 713–723, 1938.

[58] Setnes, Babuska, R., Kaymak, U. et Van Nauta-Lemke, H.R., "Similarity Measures in Fuzzy Rule Base Simplification", *IEEE Transactions on Systems, Man and Cybernetics - Part B : Cybernetics*, Vol28, pp. 376–386, 1998.

[59] Setnes, M. et Hellendoorn, H., "Orthogonal Transforms for Ordering and Reduction of Fuzzy Rules", *Proceeding of the Ninth IEEE International Conference on Fuzzy Systems*, Vol. 2, pp. 700–705, 2000.

[60] Sugeno, M. et Yasukawa, T., "A Fuzzy-logic-ased Approach to Qualitative Modelling", *IEEE Transactions on Fuzzy Systems*, Vol. 7, pp. 7–31, 1993.

[61] Thomas, G.M., Gerth, H., Valesco, T. et Rabelo, L.C., "Using Real-coded Genetic Algorithms for Weibull Parameter Estimation", *Computers & Industrial*

*Engineering*, Vol. 29, No. 1–4, pp. 377–381, 1995.

[62] Ulungu, E. L. et Teghem, J., "The Two Phase Method : An Efficient Procedure to Solve Bi-objective Combinatorial Optimization Problems". *Foundations of Computing and Decision Sciences*, Vol. 20, pp. 149–165. 1995.

[63] Valenzuela-Rendon, M., "The Fuzzy Classifier System : A Classifier System for Continuously Varying Variables", *Proceedings of the 4th International Conference on Genetic Algorithms*, pp. 346–353, 1991.

[64] Valesco,J.R., López, S. et Magdalena, L., "Genetic Fuzzy Clustering for the Definition of Fuzzy Sets", *Proceeding of the Sixth IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'97)*, Barcelone, Vol. 3, pp. 1665–1670, 1997.

[65] Visée, M., Teghem, J., Pirlot, M. et Ulungu, E.L., "Two-phases Method and Branch and Bound Procedures to Solve the Bi-objective Knapsack Problem". *Journal of Global Optimization*, Vol. 12, pp. 139–155., 1998.

[66] Wolpert, D.H. et Macready, W.G, "No Free Lunch Theoremes for Optimization", *IEEE Transactions on Evolutionary Computation 1*, pp. 67–82, 1997.

[67] Youssef, H., Sait, S.M. et Adiche, H., "Evolutionary Algorithms, Simulated Annealing and Tabu Search : A comparative Study", *Engineering Application of Artificial Intelligence*, Vol. 14, pp. 167–181, 2001.

[68] Yuan, Y. and Zhuang, H., " Using a Genetic Algorithm to Generate Fuzzy Classification Rules." *In Proceedings of the 3rd European Conference on Intelligent Techniques and Soft Computing (EUFIT'95)*, pp. 458–462, 1995.

[69] Zadeh, L.A., "Outline of New Approach to the Analysis of Complex Systems and Decision Processes", IEEE Transactions of Systems, Man and Cybernetics, Vol. 3, pp. 28–44, 1973.

[70] Zadeh, L., "A Rational for Fuzzy Control" , *Journal of Dynamic Systems, Measurement, and Control, ASME*, Vol. 94, pp. 3–4.

[71] Zheng, L., "A Practical Guide to Tune Proportional and Integral (PI) like Fuzzy Controllers", *Proceedings of the First IEEE International Conference on Fuzzy Systems (FUZZ-IEEE'92)*, San Diego, pp. 187–216, 1992.

# Appendix I

## Scheduling Exploration/Exploitation Levels in Genetically-Generated Fuzzy Knowledge Bases

*North American Fuzzy Information Processing Society (NAFIPS), pp. 34–37, Banff, Alberta, Canada, 2004.*

S. Achiche, L. Baron et M. Balazinski

Department of Mechanical Engineering

École Polytechnique de Montréal

Montréal, Québec, Canada

sofiane.achiche@polymtl.ca, luc.baron@polymtl.ca, marek.balazinski@polymtl.ca

### I.1 Abstract

In this paper we study the influence of the exploration/exploitation balance on the performances of a real binary/like genetic algorithm in automatically generating fuzzy knowledge bases from a set of numerical data. The influence is explored through different scheduling of crossover strategies throughout the evolution process. The aim of this paper is to prove the influence of a good balance between exploration and exploitation levels on the performances of the optimization algorithm used, along with the influence of a good definition of the early stages versus

the late stages of the evolution.

## I.2 Introduction

Decision Support Systems using fuzzy logic are often used to deal with complex and large decision making problems. However they are drawn back by the need of an expert to manually construct the fuzzy knowledge bases (FKBs). Genetic algorithms (GAs) proved to be successful in solving this problem[1]. GAs are powerful stochastic optimization techniques that are based on the analogy of the mechanics of biological genetics[2]. The GA used in this paper is a real/binary-like coded genetic algorithm (RBCGA) developed by the authors[3].

An FKB contains the following entities/information:

- the number of premises (inputs) and a single conclusion (output);
- the number of fuzzy sets and their repartition;
- the number of fuzzy rules and the fuzzy rule base.

The RBCGA uses three operations: crossover, mutation and natural selection. Most of the evolution is controlled by crossover. A single point crossover is used for the fuzzy rules and the blending crossover $\alpha$ (BLX-$\alpha$)[4] is used for the premises and the conclusion. The value of $\alpha$ determines the exploitation/exploration level of the offspring obtained from the selected parents. Setting $\alpha$ to 0.5 defines what

is commonly called relaxed exploitation.

In this paper we study the influence of the exploration/exploitation balance on the performances of the RBCGA while automatically generating FKBs from a set of numerical data. We will test the influence of the following crossover strategies:

1. Uniform scheduling of the exploration/exploitation levels;

2. Non-Uniform scheduling.

An influent variable has to be taken into account: At which stage of the evolution process the balanced crossovers occur? Knowing that the evolution stage is set by the generation number, what can be considered an early stage of the evolution? Finding the more adequate number of generations for each stage (for each level of exploration/exploitation) is also discussed.

This paper contains a brief description of the FDSS software Fuzzy-Flou, developed at Ecole Polytechnique (Canada) and University of Silesia (Poland)[5] used for the validation tests. Then, it presents a description of the RBCGA used, explaining the specificities of the reproduction and mutation mechanisms, and finally validation results (through theoretical surfaces) are presented with the adequate comparisons.

## I.3 Fuzzy Decision Support System

In this section we present a rule-based approach to decision making using fuzzy logic techniques, based on the compositional rule of inference (CRI). This approach is used to handle imprecise knowledge and was developed in the sixties by L.A. Zadeh[6]. Such knowledge can be collected and delivered by a human expert (e.g. decision-maker, designer, process planner, machine operator, etc.). The CRI may be written in the form:

$$U = (C \times \cdots \times B \times A) \circ R \tag{I.1}$$

where $R$ represents the global relation that aggregates all the rules, ($A$, $B$, $\cdots$, $C$) represents the inputs (observations) and $U$ represents the output (conclusion). The symbol $\circ$ represents the CRI operator. In this paper, the center of gravity (COG) is used for the defuzzification. Figure I.1 shows a screen printout of the FDSS Fuzzy-Flou software used as validation tool for the genetically generated FKBs.

## I.4 Automatic Generation of FKBs

The automatic generation of FKBs is performed with the help of the RBCGA. The Coding of the RBCGA is explained in the next sections.

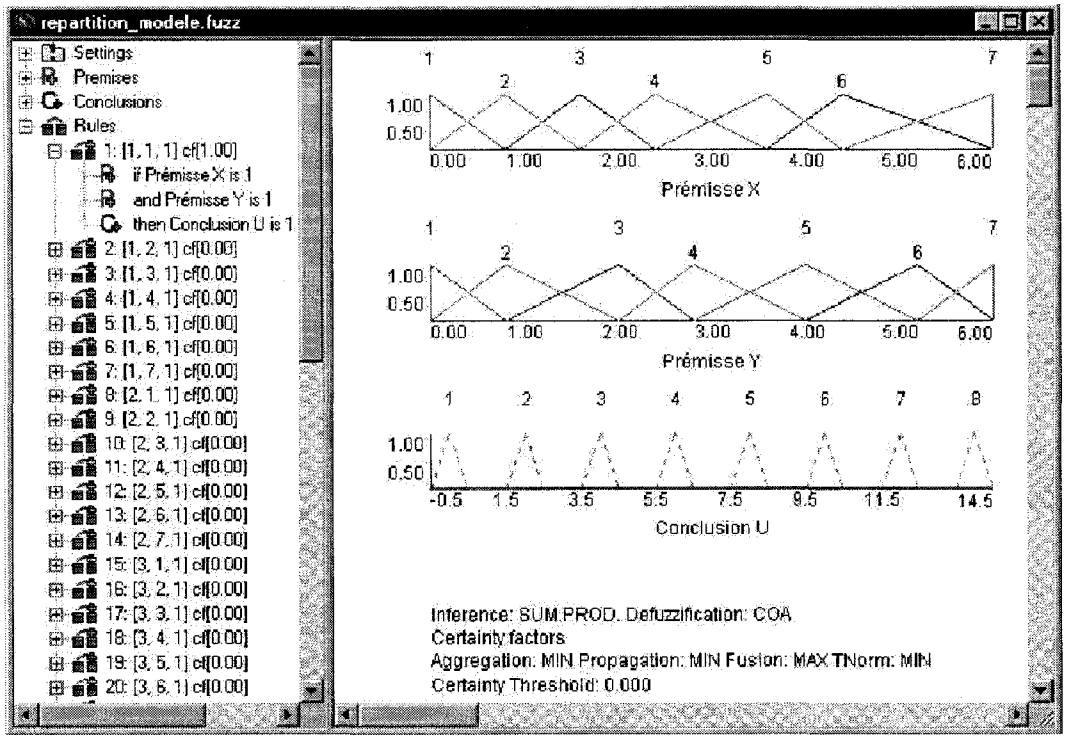


Figure I.1 Graphic Interface of the FDSS Fuzzy-Flou

### I.4.1 Coding

The genotype of an FKB is the coding of its parameters into chromosomes and corresponds to several independent sets of real numbers and a set of integers. The genotype contains the following items:

1. Input/Output Premises: A set of real numbers. For the sake of coding simplicity, we consider only non-symmetrical-overlapping triangular fuzzy sets for premises and symmetrical triangular fuzzy sets for the conclusion. Therefore, the position of each fuzzy set is given by the position of the summit of the triangle.

2. Fuzzy Rules: The genotype of fuzzy rules contains information about all the possible combinations of connecting one fuzzy set on each premise to a fuzzy set on the conclusion. For $N$ input premises and $K_i$ fuzzy sets on premises $i$, the maximum number of fuzzy rules $K$ is computed as:

$$K = K_1 \times K_2 \times \cdots \times K_N \tag{I.2}$$

Each number of the set represents a conclusion fuzzy set number.

### I.4.2 Multi-Crossover Mechanisms

The evolution of a population of FKBs at each generation is achieved by the reproduction of the "best" individuals, based on their abilities to survive *natural selection.* Reproduction is performed by crossover of the genotype of the parents to obtain the genotype of an offspring. Multi-crossover is composed of a *premises/conclusion crossover* and a *fuzzy rules crossover.* These two mechanisms are governed by the initiating probability $p_2$.

a. Premises/Conclusion Crossover

The mechanism used is called *blending crossover* $\alpha$ $(BLX - \alpha)$[4], where $\alpha$ determines the exploitation/exploration level of the offspring. Exploitation means using the interval between the values of the two parents, the exploration uses an interval outside of these two limits (Fig. I.2). The $BLX - \alpha$ works as follows:


Figure I.2 Blended crossover $\alpha$ $(BLX - \alpha)$

If $A = \{x_1, x_2, \cdots, x_i, \cdots, x_n\}$ and $B = \{y_1, y_2, \cdots, y_i, \cdots, y_n\}$ represents the two selected parents, $C$ the offspring obtained by the crossover of $A$ and $B$ then $C = \{z_1, z_2, \cdots, z_i, \cdots, z_n\}$, where $z_i$ are randomly selected in the interval $[min_i$ -I $\alpha$, $max_i$ + I $\alpha]$ where:

- $max_i$ = maximum $\{x_i, y_i\}$;
- $min_i$ = minimum $\{x_i, y_i\}$;
- I = $\{max_i - min_i\}$.

b. Fuzzy Rules Crossover

Since the part of the genotype representing the fuzzy rule base is composed of integer numbers, the crossover on this part of the genotype is done by a simple crossover. The operation is performed by inverting the end part of the sets (containing the fuzzy rules) of the parents at a randomly selected crossover site, as shown in Fig. I.3.

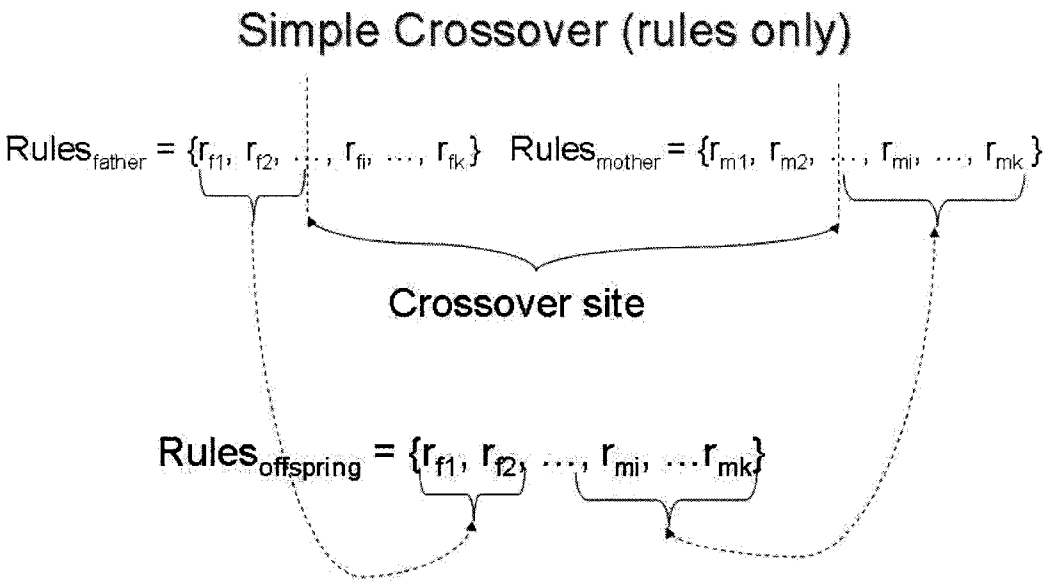


Figure I.3 Simple crossover

### I.4.3 Mutation

Mutation is the creation of an individual from an existing one by altering one gene. Mutation makes it possible to try completely different solutions. The probability $p_2$ of mutation is fixed at a low level to let the population improve mainly by multi-crossover. The mutation used is a uniform mutation[7].

### I.4.4 Natural Selection

Natural selection is performed on the population by keeping the "most" promising individuals, based on their fitness. The first generation begins with $P$ FKBs; the same number of additional FKBs is generated by crossover and mutation. To keep the population constant, we apply natural selection on the $2 \times P$ FKBs by ordering

them according to the performance criterion and keeping the $P$ first FKBs. The number $P$ has to be fixed depending on the computer performances.

## I.5 Performance Criterion

The performance criteria allow one to compute the rating of each FKB. This performance rating is used by the RBCGA in order to perform the natural selection. Here, the performance criterion is the accuracy level of a FKB (approximation error) in reproducing the outputs of the learning data. The approximation error $\Delta_{rms}$ is measured using the rms error method:

$$\Delta_{rms} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(GA_{output_i} - data_{output_i})^2} \tag{I.3}$$

where, $M$ represents the number of learning data. The fitness value is evaluated as a percentage of $L$, the output length base ($L = U_{max} - Umin$) of the conclusion, i.e.,

$$\phi_{rms} = \frac{L - \Delta_{rms}}{L} \times 100\,, \tag{I.4}$$

## I.6 Test Functions

The learning performances of the RBCGA are investigated using three examples of known behavior in term of 3D surfaces of the type $z = f(x, y)$. We have used four different surfaces of different complexities inspired from *De Jong* test

environment[8]. The evolution and selection criteria are set to the following values:

- $pr_1 = 100.0\%$;
- $pr_2 = 5.0\%$.
- Maximal complexity: 6 fuzzy sets (including the limits) on each premise and 10 on the conclusion. This sets the maximal number of the fuzzy rules to 36.

These numbers were chosen based on performance tests applied on the surfaces. The evolution is completely governed by the *multi-crossover* reproduction mechanism. The number of fuzzy sets can decrease when two summits overlap, hence, the reducing of the fuzzy rule base. The theoretical surfaces used are:

- Spherical surface (Fig. I.4):

The spherical surface is defined as:

$$z = x_2 + y_2. \text{ with } \begin{array}{l} -5.12 \leq x \leq 5.12 \\ \\ -5.12 \leq y \leq 5.12 \end{array}, \tag{I.5}$$

- Stochastic surface (Fig. I.5):

The stochastic surface is defined as:

$$z = x_4 + y_4 + random(0,1). \text{ with } \begin{array}{l} -1.28 \leq x \leq 1.28 \\ \\ -1.28 \leq y \leq 1.28 \end{array}, \tag{I.6}$$

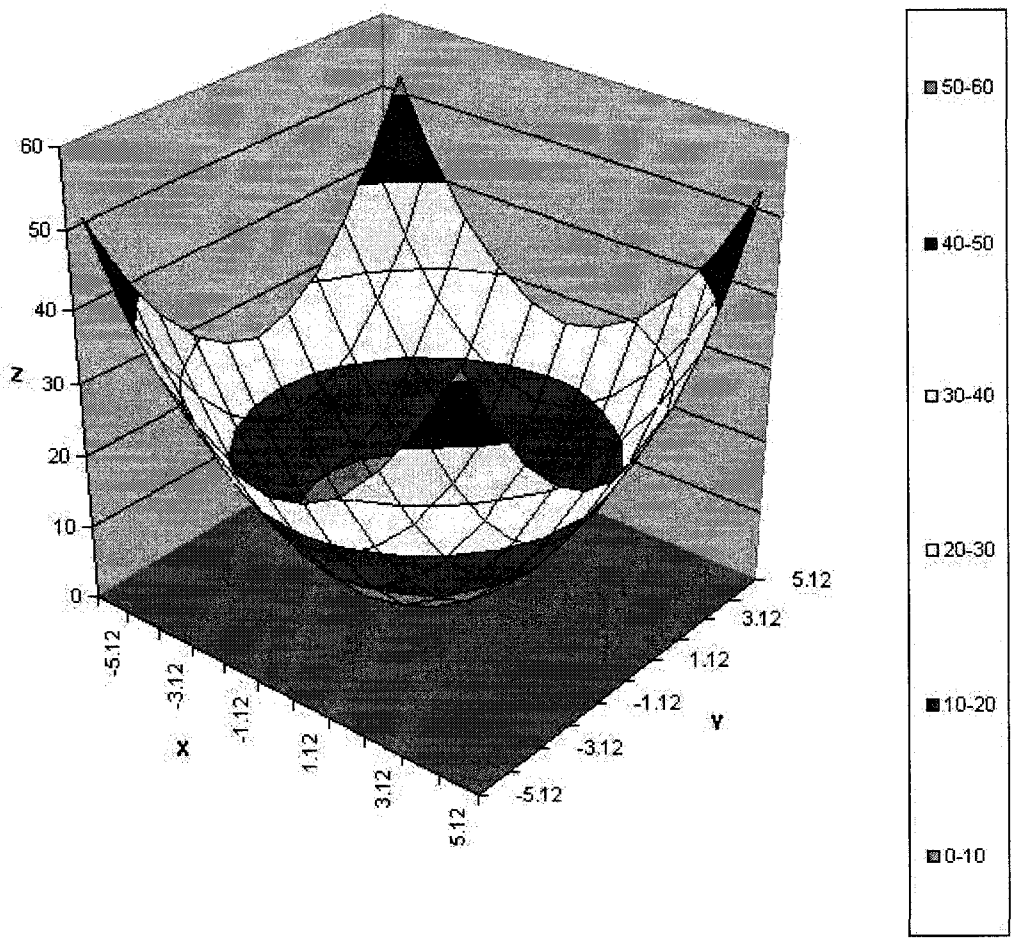


Figure I.4 Spherical surface

where $random(0,1)$ provides a real number between 0 and 1.

- Inclined plane surface:

The inclined plane surface is defined as (Fig. I.6):

$$z = x + y. \text{ with } \begin{matrix} -5.12 \leq x \leq 5.12 \\ -5.12 \leq y \leq 5.12 \end{matrix}, \tag{I.7}$$

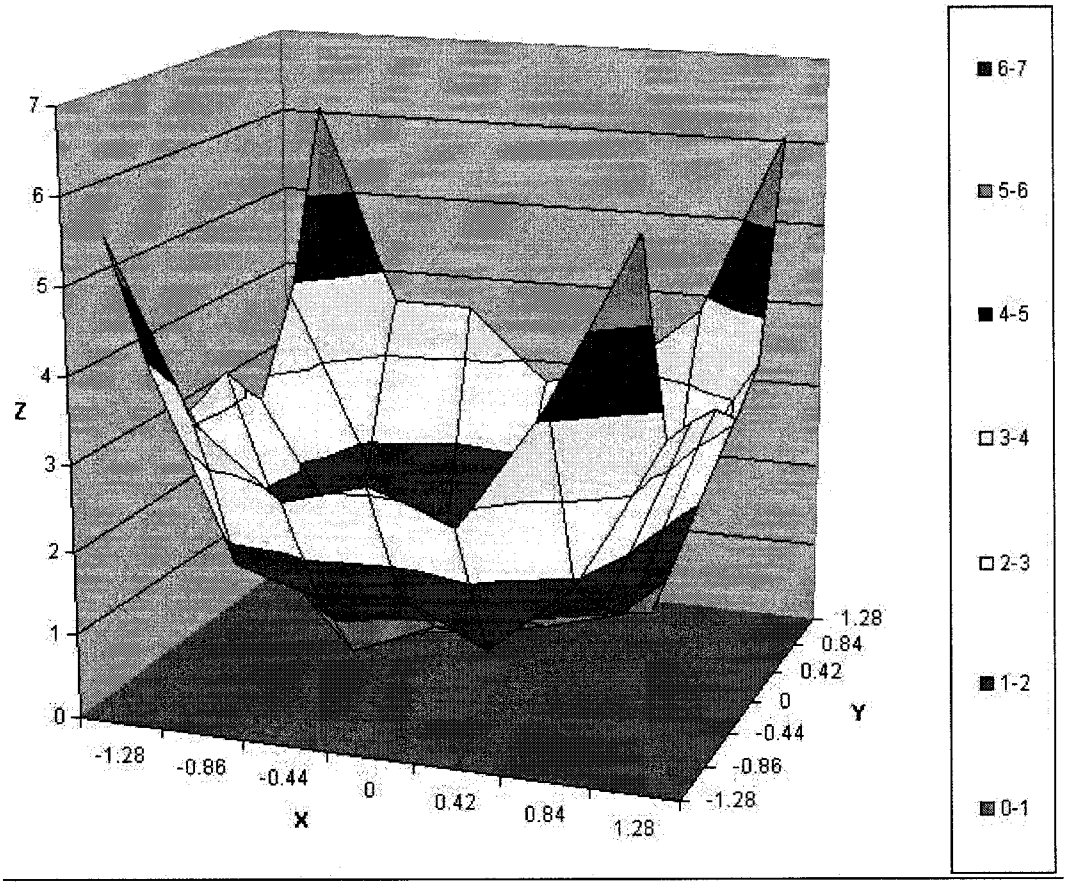


Figure I.5 Stochastic surface

- Non-symmetric surface (Fig. I.7):

The non-symmetric surface is defined as:

$$z = x\ EXP(x_2\ -\ y_2).\ \text{with}\ \begin{matrix} -2.5 \leq x \leq 2.5 \\ -2.5 \leq y \leq 2.5 \end{matrix}, \tag{I.8}$$

In order to measure the fitness (accuracy levels) of the RBCGA in generating FKBs, and for the sake of comparison, several runs have been made on the surfaces.

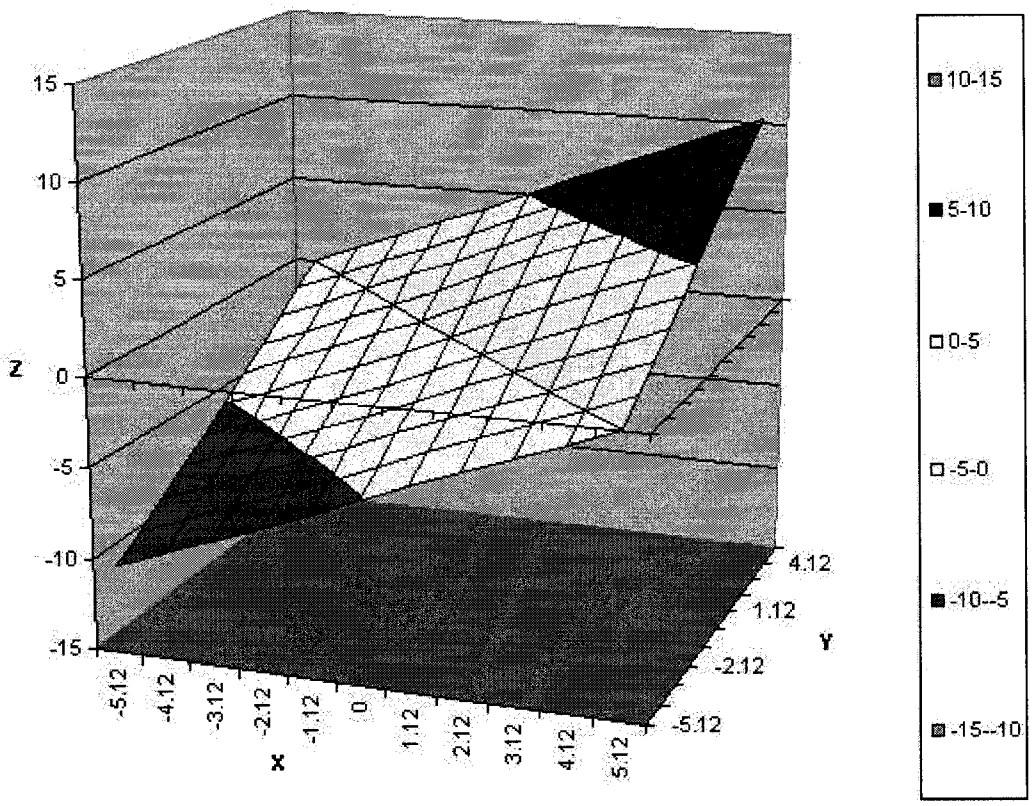


Figure I.6 Inclined plane surface

The population size $P$ was set to 20 individuals. Runs were performed three times for 100, 500 and 2500 generations. The fitness of the best individual is taken into account at the last generation for each surface. The average value of the results obtained for each theoretical surface was computed. The runs are performed using different balances of the exploitation/exploration level.

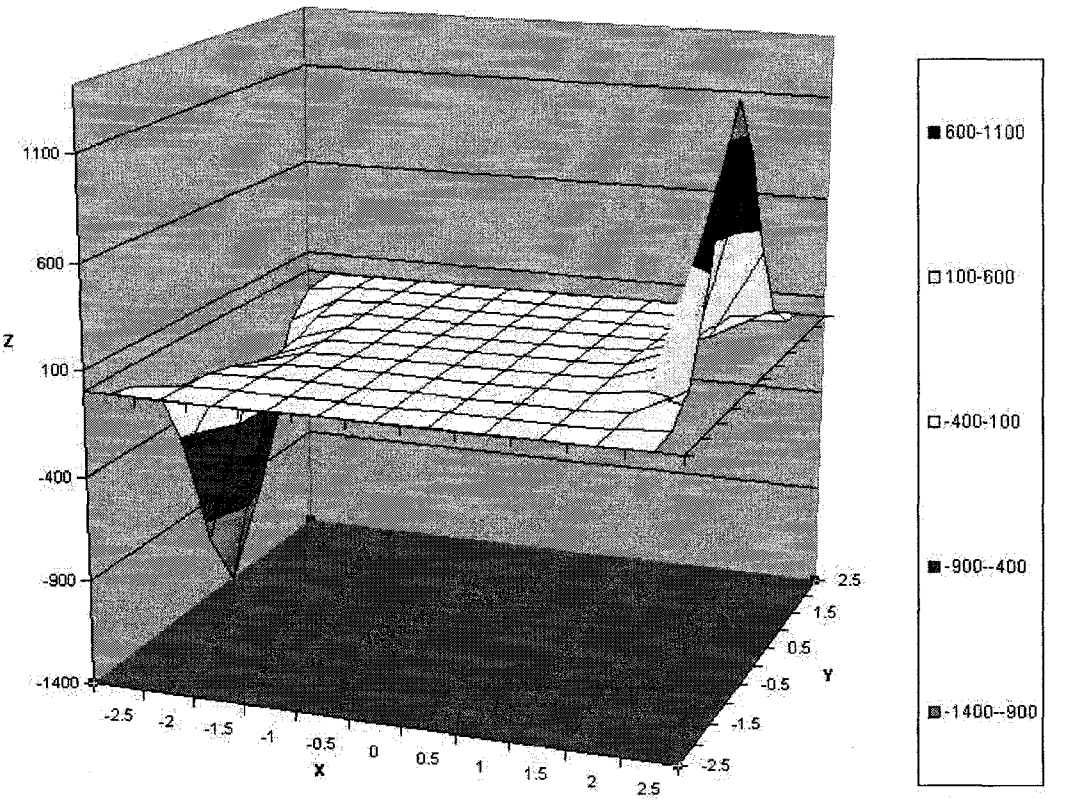


Figure I.7 Non-symetric surface

## I.7 Scheduling Exploitation/Exploration Levels

There are two main reasons that affect the performances of the GAs: the loss of diversity in the population and the bad balance between exploration and exploitation throughout the evolution process. In this section, we study the influence of the exploration/exploitation balance ($EEB$) on the performances of the RBCGA aiming to find the best combination from the following:

- Exploiting the individuals at the early stages of the evolution, applying relaxed exploitation through the main part of the evolution and then exploring

the individuals at the late stages;

- Exploring at the early stages of the evolution, applying relaxed exploitation through the main part of the evolution and then exploiting at the late stages of the evolution.

The first combination is called $EEB1$, while the second is called $EEB2$. The control of the exploration/exploitation balance is based on the variation of $\alpha$. The different values given to $\alpha$ influence the exploration, exploitation or relaxed exploitation levels of the crossover mechanism. The three values are:

- Exploration: $\alpha = 1.00$ for total exploration;
- Relaxed Exploitation: $\alpha = 0.50$;
- Exploitation: $\alpha = 0.10$ for close to maximal exploitation ($\alpha \neq 0.00$, in order to keep a difference between the $BLX - \alpha$ and the uniform mutation).

What is the best stage of the evolution process where the three variations above should occur? The stage of the evolution i s defined by the generation number. Considering a maximal number of generations $G$, we tested the following:

- The first and last fifths (1/5) of $G$ are considered the early and late stages respectively;
- The first and last quarters (1/4) of $G$ are considered the early and late stages respectively;

- The first and last thirds (1/3) of $G$ are considered the early and late stages respectively;

- The remaining part is considered the evolution stage.

### I.7.1 Applying $EEB1$ and $EEB2$

This section presents the results obtained for the $EEB1$ and $EEB2$ strategies using the three different values setting the late/early stages of the evolution.

Table I.1 and Table I.2, shows the results obtained for the 100, 500 and 2500 generations runs. The first column contains the balance used in defining the early and late stages of the evolution, the second one the number of generations of the RBCGA, the third represents the best $\phi_{rms}$ obtained. The last column represents the size of the fuzzy rules base (number of rules).

A high number of fuzzy rules translate into a high complexity level of the FKB. If two FKBs achieve the same accuracy, the simplest is considered the better one, since it is easier to fine-tune a simple FKB manually and also simple FKBs possesses better generalization qualities[9].

#### I.7.1.1 Analyzing the $EEB_1$ strategy results

Table I.1 shows that the fitness results are close (from a fitness standpoint) for each of the three runs (1/3, 1/4 and 1/5). The $\phi_{rms}$ improved with the increase of the generation number, which is predictable. The FKBs obtained have similar accuracy levels but they are still different, since the size of the fuzzy rule base (# of rules) differs. The fuzzy rule base size decreases with the increase of the number of generations and decreases also with the decrease of the early/late stages balance. The one third (1/3) balance produced the best $\phi_{rms}$ for the smallest number of generations (100), while keeping the number of fuzzy rules at a very respectable number (considering that the maximal complexity of the FKBs contains 36 fuzzy rules).

Table I.1 Results obtained for $EBB_1$ (1/3, 1/4, 1/5)

| $EBB_1$ | # Generation | Best $\phi_{rms}$ | # Rules |
|---|---|---|---|
| 1/3 | 100 | 91.44 % | 23 |
| 1/3 | 500 | 93.91 % | 19 |
| 1/3 | 2500 | 93.94 % | 17.75 |
| 1/4 | 500 | 90.91% | 22 |
| 1/4 | 1000 | 93.23% | 19 |
| 1/4 | 2500 | 93.94% | 17.75 |
| 1/5 | 100 | 90.79 % | 20.5 |
| 1/5 | 500 | 93.24 % | 17.75 |
| 1/5 | 2500 | 93.88 % | 16.5 |

Table I.2 Results obtained for $EBB_2$ (1/3, 1/4, 1/5)

| $EBB_2$ | # Generation | Best $\phi_{rms}$ | # Rules |
|---|---|---|---|
| 1/3 | 100 | 88.58 % | 34.5 |
| 1/3 | 500 | 88.48 % | 36 |
| 1/3 | 2500 | 90.38 % | 34.5 |
| 1/4 | 500 | 87.82% | 36 |
| 1/4 | 1000 | 88.21% | 34.5 |
| 1/4 | 2500 | 90.50% | 34.5 |
| 1/5 | 100 | 88.63 % | 36 |
| 1/5 | 500 | 88.21 % | 34.5 |
| 1/5 | 2500 | 90.08 % | 33 |

#### I.7.1.2 Analyzing the EEB2 strategy results

Table I.2 shows that the fitness results are close (from a fitness standpoint) for each of the three runs (1/3, 1/4 and 1/5). However, the $\phi_{rms}$ doesn't always improve with the increase of the generation number (see case of the 1/3 balance). The size of the fuzzy rule base (# of rules) doesn't seem to follow any pattern (decreasing with the increase of the number of generations or the decrease of the early/late stages balance). There is almost no simplification of the FKBs since the number of fuzzy rules is close to the maximum possible (i.e., 36).

#### I.7.1.3 Comparing $EEB_1$ and $EEB_2$

The $EBB_1$ strategy performed better than the $EBB_2$, using the exploration at first and exploitation at the late stages produced more accurate and simpler FKBs; even thought better accuracy of FKBs generally translates into a higher number of fuzzy rules (high complexity level).

Figure I.8 illustrates the evolution of $\phi_{rms}$ versus the increase of the early/late stages balance for the 100 generations runs. The $EEB_1$ strategy proved to be a

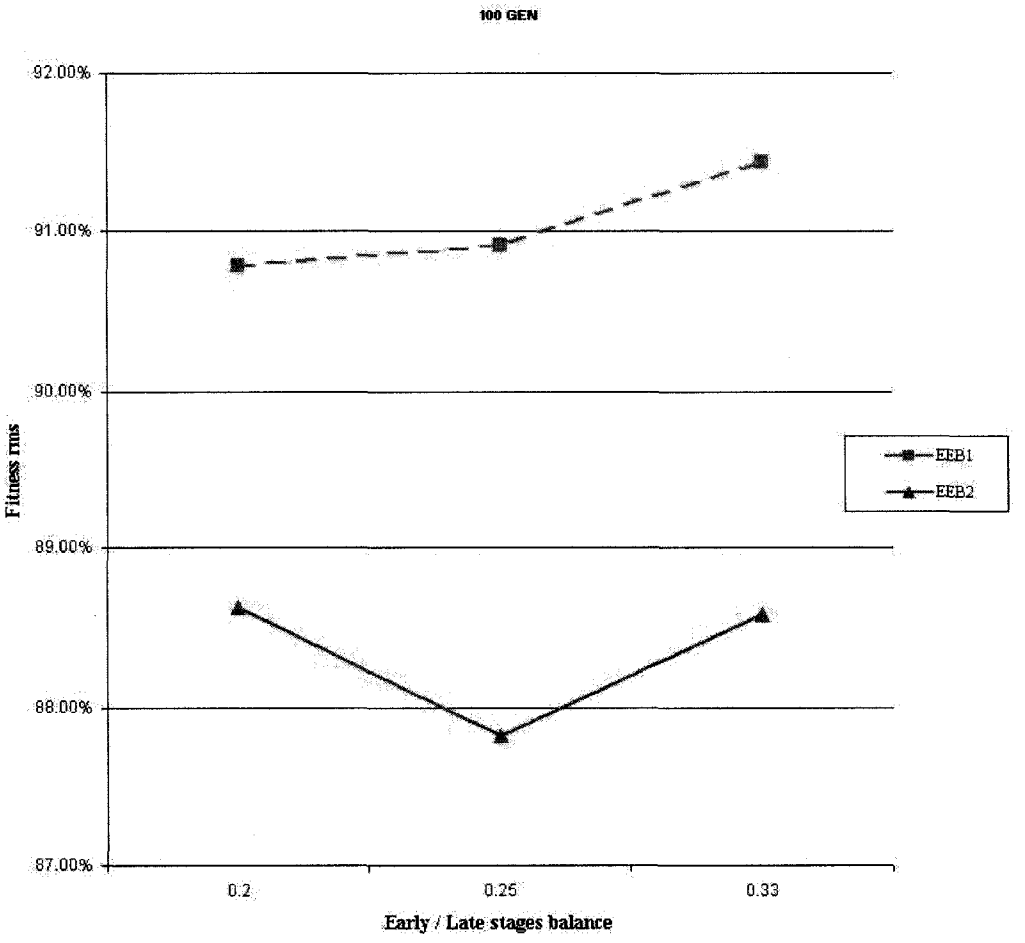


Figure I.8 $\phi_{rms}$ for $EEB_1$ and $EEB_2$ (100 generations)

better way for scheduling the exploration/exploitation levels throughout the evolution process. It seems more natural to apply exploration in the early stages (which improves diversity) and to apply exploitation at the late stages as a fine tuning.

## I.8 Non-Uniform Scheduling of Exploitation/Exploration Levels

In this section, a non-uniform balance of the early/late stages distribution using EEB1 is explored.

In the previous section $EEB_1$ proved to be the better choice, when it comes to exploration/exploitation scheduling through the evolution. In this section the choice of the early/late stages balance is randomly selected. A new balance is produced for each generation. The mechanism works as follow:

- For each generation an integer number $e$ is randomly generated (with a normal distribution), $e$ is contained in the interval [3, 10].
- For each value of $e$: the individuals of the first $\frac{P}{e}$ generations are produced using $BLX-1.0$ and the individuals of the last $\frac{P}{e}$ generations are produced using $BLX-0.1$, the rest is produced using the $BLX-0.5$ (relaxed exploitation).

Table I.3 Results obtained for $EBB_2$ (Random)

| $EBB_2$ | # Generations | Best $\phi_{rms}$ | # Rules |
|---|---|---|---|
| Random | 100 | 90.87 % | 12 |
| Random | 500 | 92.22 % | 13.25 |
| Random | 2500 | 92.36 % | 14.25 |

From Table I.3 and Table I.1, we can see that fixed early/late stages balance achieved better accuracy than the randomly generated scheduling (non-uniform

$EEB_1$). However the non-uniform $EEB1$ produced simpler FKBs. The size of the fuzzy rule base varies between 12 and 14 fuzzy rules while the fixed scheduling gave between 16 and 23 fuzzy rules. Since we consider the accuracy as the most important criterion to satisfy, we can conclude that the uniform $EBB_1$ can be preferred to the non-uniform one.

## I.9 Conclusion

The different ways to create new individuals from the same pair of parents' genes is a good way to improve diversity. We can also conclude that exploration/exploitation balance influences the performances of the RBCGA. The rule to follow is using exploration at the early stages of the evolution, relaxing in the middle and exploitation at the end. This order achieved much better results than using exploitation at the early stages. One of the reasons is the lack of diversity created by heavy exploitation during the early stages, due to the fact that the offspring's genotype is generated too close to its parents' average (definition of the exploitation).

The non-uniform exploration/exploitation strategy proved to be quite efficient especially for the fuzzy rules reduction. If a very simple FKB is needed this strategy can be used.

## I.10 Acknowledgment

Financial support from the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618, RGPIN-105518 and STPGP-269579 is gratefully acknowledged.

# RÉFÉRENCES

[1] L. Baron., S. Achiche, and M. Balazinski, "Fuzzy decisions system knowledge base generation using a genetic algorithm", *International Journal of Approximate Reasoning*, pp. 25–148, 2001.

[2] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning", *Addison-Wesley*, 412 pages, 1989.

[3] S. Achiche, M. Balazinski and L. Baron, "Real/Binary-Like coded genetic algorithm to automatically generate fuzzy knowledge bases", *The 4th International Conference on Control and Automation*, pp. 799–803, 2003.

[4] F. Herrera and M. Lozano, "Gradual distributed real-coded genetic algorithms", *IEEE Transactions on Evolutionary Computation*, Vol. 4, pp. 43–63, 2000.

[5] M. Balazinski, M. Bellerose, and E. Czogala, "Application of fuzzy logic techniques to the selection of cutting parameters in machining processes", *International Journal for Fuzzy Sets and Systems*, Vol. 61, pp. 307–317, 1993.

[6] L.A. Zadeh, "Outline of new approach to the analysis of complex systems and decisions processes", *IEEE Transactions of Systems, Man and Cybernetics*, Vol. 3, pp. 28–44, 1973.

[7] Z. Michalewicz, "Genetic algorithms + data structure = evolution programs",

*New York: Springer*, 1992.

[8] K.A. De Jong, "An Analysis of the Behavior of Genetic Adaptive Systems", *Doctoral Dissertation*, University of Michigan. Dissertation Abstracts International, 36(10), 5140B, 1975.

[9] M. Balazinski, S. Achiche, and L. Baron, "Influence of optimization and selection criteria on genetically-generated fuzzy knowledge bases", *(ICAMT2000) International Conference on Advanced manufacturing technology*, pp. 159–164, 2000.

# Appendix II

## Chips Image Processing to Predict Pulp Quality using Fuzzy Logic Techniques

*91 Annual Meeting of the Pulp and Paper Technical Association of Canada, Montréal, Canada, pp. D173-D176, 2005.*

S. Achiche[1], L. Baron[1], M. Balazinski[1] et M. Benaoudia[2]
Department of Mechanical Engineering[1]
École Polytechnique de Montréal
Montréal, Québec, Canada
Centre de Recherche Industrielle du Québec[2]
Ste-Foy, Québec, Canada.
sofiane.achiche@polymtl.ca, luc.baron@polymtl.ca, marek.balazinski@polymtl.ca
mokhtar.benaoudia@criq.qc.ca

### II.1 Abstract

The quality of the thermomechanical pulp and paper (TMP) process is influenced by a large number of variables. The influencing variables are generally chosen by the process maker, and they change depending on the raw material feeding the TMP process. In this paper, a genetically generated fuzzy knowledge base (FKB) is used to model the relationship between the wood chips characteristics and the quality of the resulting pulp; measured by the pulp ISO brightness. The learning of the FKBs uses measurements obtained from the chip management system ($CMS^{©}$).

## II.2 Introduction

In the paper industry, the quality of the thermomechanical pulp is influenced by a large number of variables. One of the most important standards used to evaluate the pulp quality is its brightness. In order to reach high brightness values, one has to deal with numerous parameters, such as: the wood chips condition; wood chips origins (tree spice), humidity, the bleaching process, etc.

Several works have singled out the influence of some parameters on the quality of the pulp. The influence of the chip size is presented in [4]. A degradation of the chip quality influences the bleaching process[3, 7]. However, the behavior of the wood chips characteristics influence on the thermomechanical pulping process (TMP) is somehow unknown. Usually, the manufacturers adjust some variables of the process to account for the variation of the chips characteristics. For example, the consumption of bleaching agents is adjusted at level sufficiently high to cope with the worst case of chips quality. The objective of this paper is to model the relationship between the wood chips characteristics and the quality of the resulting pulp. In order to measure the changes in chip quality, we use an image processing system called the chip management system ($CMS^{©}$). A fuzzy decision support system (FDSS) Fuzzy-Flou developed at École Polytechnique (Montreal) and University of Silesia (Poland)[2] is used as the prediction tool. The fuzzy knowledge

base (FKB), used by the FDSS, is automatically generated by a real binary like genetic algorithm (RBCGA)[1] from a set of experimental data. In this paper we use the $CMS^{©}$ information, to predict the pulp ISO brightness for a certain amount of peroxide charges.

## II.3 Prediction of the Pulp ISO Brightness

### II.3.1 The Chip Management System

The $CMS^{©}$ is an innovative device that allows online measurements of several chip characteristics such as the chip brightness (or luminance). The $CMS^{©}$ was used to confirm the existence of a correlation between bleaching agent consumption and the chip luminance. The main sensors of the $CMS^{©}$ include an artificial vision sensor (an RGB camera and a frame grabber), which provide the Hue (H), Saturation (S) and Luminance (L). As shown in Fig. II.1, the HSL can be represented as a cone, whose base corresponds to the lowest luminance and the top to the highest. For a given level of luminance, the distance from the cone center indicates the saturation of the color, and the axis compared to the horizontal axis gives the hue of the color.

### II.3.2 Selection Input/Output of the FKB

Pulp quality depends on different physical properties such as: wood type; bark percentage, presence of knots, etc. Exterior conditions (such as: the weather vari-

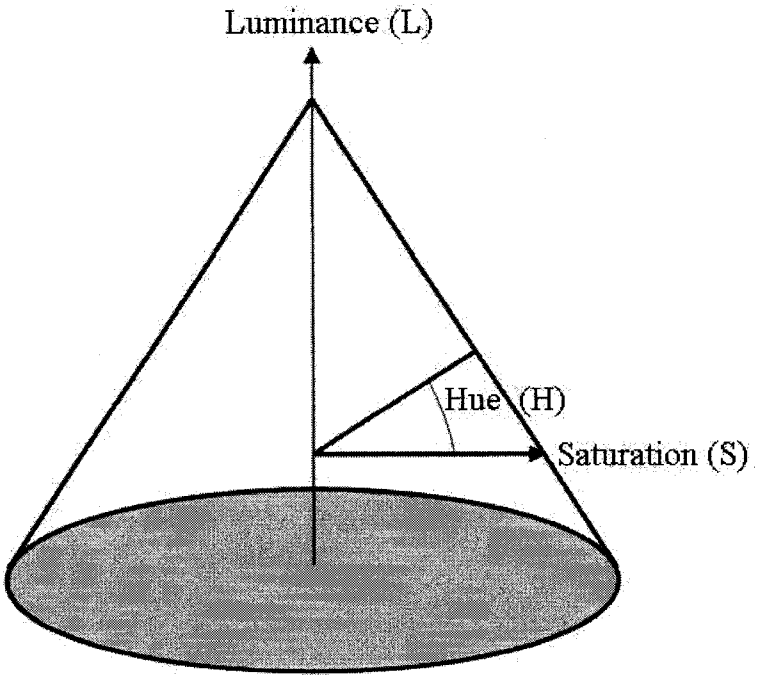


Figure II.1 Cone representation of the HSL

ations, the storage conditions, etc.) influence the quality of the pulp. As stated in [6], optical measures can be translated into quality information of the paper. The changes of the quality of the wood chips are translated into a change in its color, and hence, of its HSL parameters, which are natural variables to describe a color. For these reasons, the parameters H, S, and L are the only parameters taken into account for the learning, along with the peroxide (4 input variables). The output variable is the ISO brightness of the pulp. Hence, the genetically generated FKBs are of the multiple input single output (MISO) type.

### II.3.3 Fuzzy Decision Support System

The rule-based approach to decision making is done using fuzzy logic techniques, based on the compositional rule of inference. This approach was developed in the

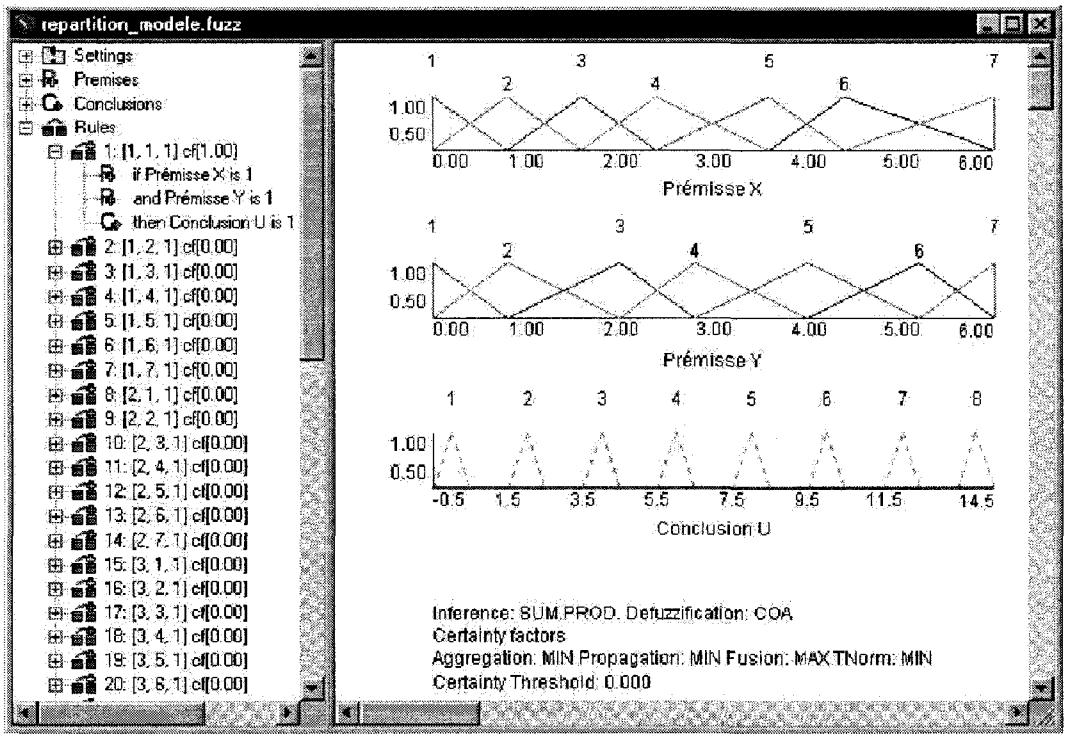


Figure II.2 Print-out of the FDSS Fuzzy-Flou showing an FKB

sixties by L.A. Zadeh [8]. Figure II.2 shows a screen printout of the FDSS Fuzzy-Flou software used as a validation tool for the genetically generated FKBs. For more information on the Fuzzy-Flou software, please refer to [2].

## II.4 Automatic Learning

The automatic learning of FKBs is performed with the help of the RBCGA from a set of experimental measurements. The coding of the FKBs in the RBCGA is explained in the next sections.

### II.4.1 Coding

The genotype of an FKB is the coding of its parameters into chromosomes and corresponds to several independent sets of real numbers and a set of integers. The genotype contains the following items:

- Input/Output Premises: A set of real numbers. For the sake of coding simplicity, we consider only non-symmetrical-overlapping triangular fuzzy sets for premises and symmetrical triangular fuzzy sets for the conclusion.

- Fuzzy Rules: The genotype of fuzzy rules contains information about all the possible combinations of connecting one fuzzy set on each premise to a fuzzy set on the conclusion.

### II.4.2 Multi-Crossover Mechanisms

The evolution of a population of FKBs at each generation is achieved by the reproduction of the "best" individuals, based on their abilities to survive natural selection. Reproduction is performed by crossover of the genotype of the par-

ents to obtain the genotype of an offspring. Multi-crossover is composed of a premises/conclusion crossover and a fuzzy rules crossover.

### II.4.2.1 Premises/Conclusion Crossover

The mechanism used is called blending crossover $\alpha$ (BLX-$\alpha$)[5], where $\alpha$ determines the exploitation/exploration level of the offspring. The parameter $\alpha$ changes from 0.10 for the first quarter of the evolution, 0.50 for the second and third quarter and finally 1.00 for the last quarter. This shift in the balance reduces premature convergence through the evolution[1].

### II.4.2.2 Fuzzy Rules Crossover

The crossover on this part of the genotype is performed by a single point crossover, as shown in Fig. II.3.

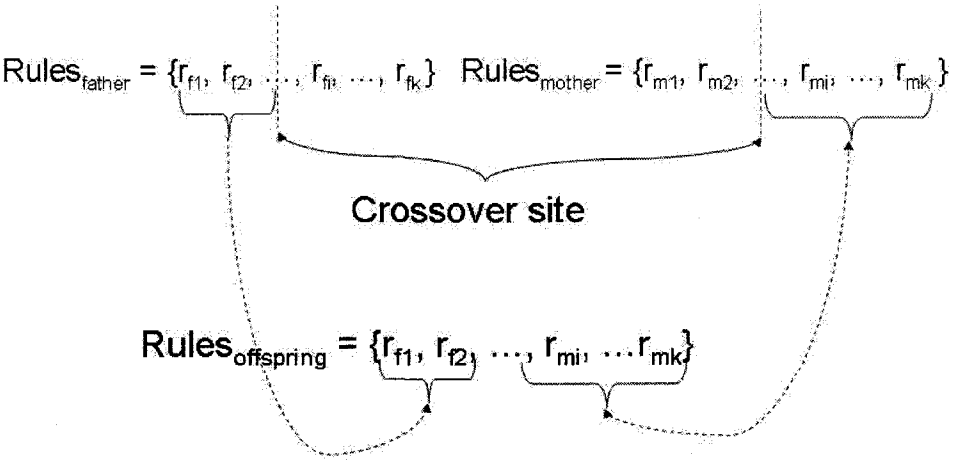


Figure II.3 Single point crossover

### II.4.3 Mutation

Mutation is the creation of an individual from an existing one by altering one gene. The mutation used is a uniform mutation[5].

### II.4.4 Natural Selection

Natural selection is performed on the population by keeping the "most promising" individuals, based on their fitness. The population size is fixed throughout the evolution.

### II.4.5 Performance criterion of the RBCGA

The performance criterion is the accuracy level of a FKB (approximation error) in reproducing the outputs of the learning data. The approximation error $\Delta_{rms}$ is defined as the root-mean-square (rms) error between the predicted and experimental data, i.e.

$$\Delta_{rms} = \sqrt{\frac{1}{M}\sum_{i=1}^{M}(GA_{output_i} - data_{output_i})^2} \qquad \text{(II.1)}$$

where, $M$ represents the number of learning data. The fitness value is evaluated as a percentage of $L$, the output length base ($L = U_{max} - Umin$) of the conclusion, i.e.,

$$\phi_{rms} = \frac{L - \Delta_{rms}}{L} \times 100\,, \qquad \text{(II.2)}$$

## II.5 Optimal FKB of Pulp ISO Brightness

The set of experimental data is split into two different sets for learning and testing. The testing file is used to verify the generalization level of the FKB. The proportions of the sets are the following: 10% of the data are mapped into a testing file and the remaining 90% are used to learn the FKBs.

In this paper, the maximum complexity of a FKB learned by the RBCGA is limited to 6 triangular, overlapping, fuzzy sets on each premise and 16 isosceles triangles on the conclusion. Therefore, the maximal number of fuzzy rules is $6^{numberofpremises}$ $= 6^4 = 1296$. The population size is fixed to 100, and the number of generations is 500.

### II.5.1 Genetic learning and testing of the FKBs

The application of the RBCGA to the learning file produced the FKB illustrated in Fig. 4. The FKB contains 2 fuzzy sets on the average of H, average of S and average of L premises, three fuzzy sets on the peroxide concentration premise, and 10 sharp triangular fuzzy sets on the conclusion, i.e., the ISO brightness. Finally a total of 24 if-then fuzzy rules (rather than the maximum of 1296 initially proposed).

The fitness level $\phi_{rms}$ attained approximately 94.30% for this FKB, which corre-

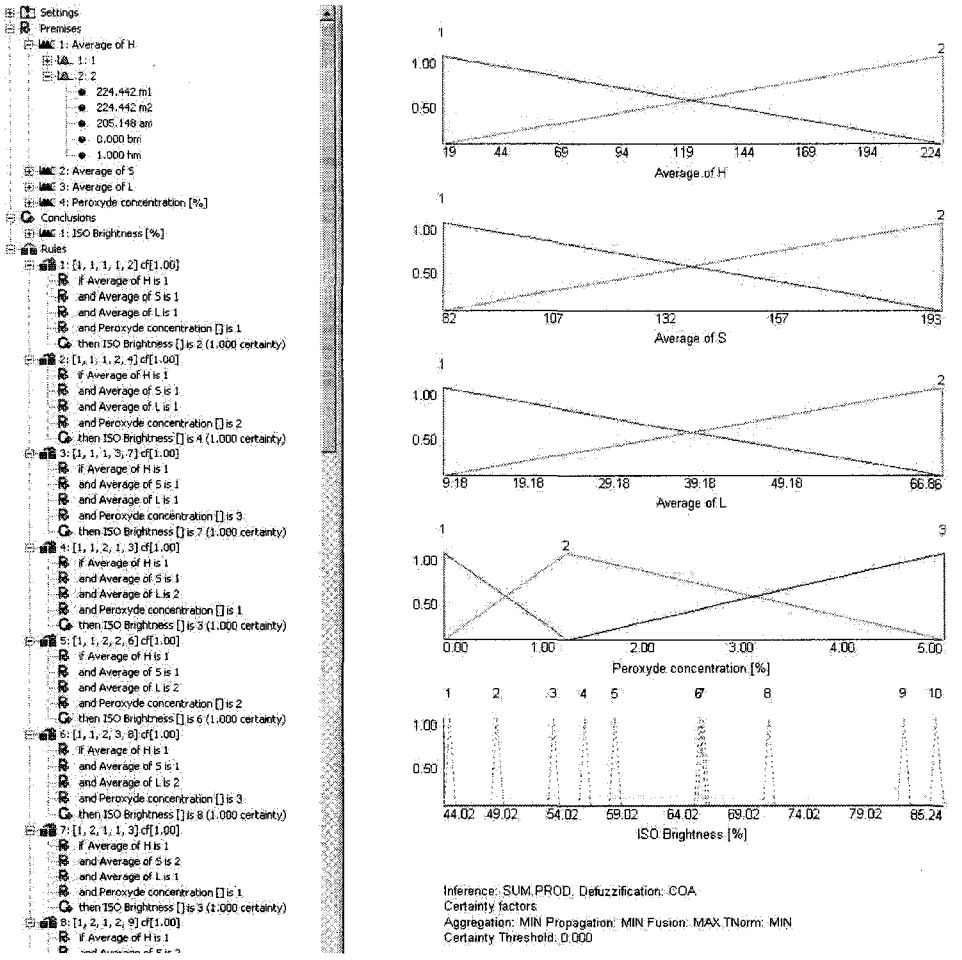


Figure II.4 The FKB to predict pulp ISO brightness from H, S, and L and peroxide concentration.

sponds to a 2.10% $\Delta_{rms}$, a very satisfactory error level for the learning. Processing the test file (the 10% of data hidden from the learning) through the obtained FKB, produced approximately 2.21% $\Delta_{rms}$. The learning and testing errors are comparable, which reflects the stability and the good generalization level of the genetically generated FKB. Figure II.5, illustrates the superimposition of the experimental ISO brightness (testing values) and the prediction of the genetically generated FKB (the

line represents a theoretical perfect prediction). The minimal and maximal values of the ISO brightness were reached by the obtained FKB, since its universe of discourse covers the interval [$\approx 44.50\%$ $\approx 84.70\%$] while the interval containing the experimental values, is [$\approx 43.80\%$ $\approx 80.00\%$].

Now, let us take into account the part of the learning file corresponding to the

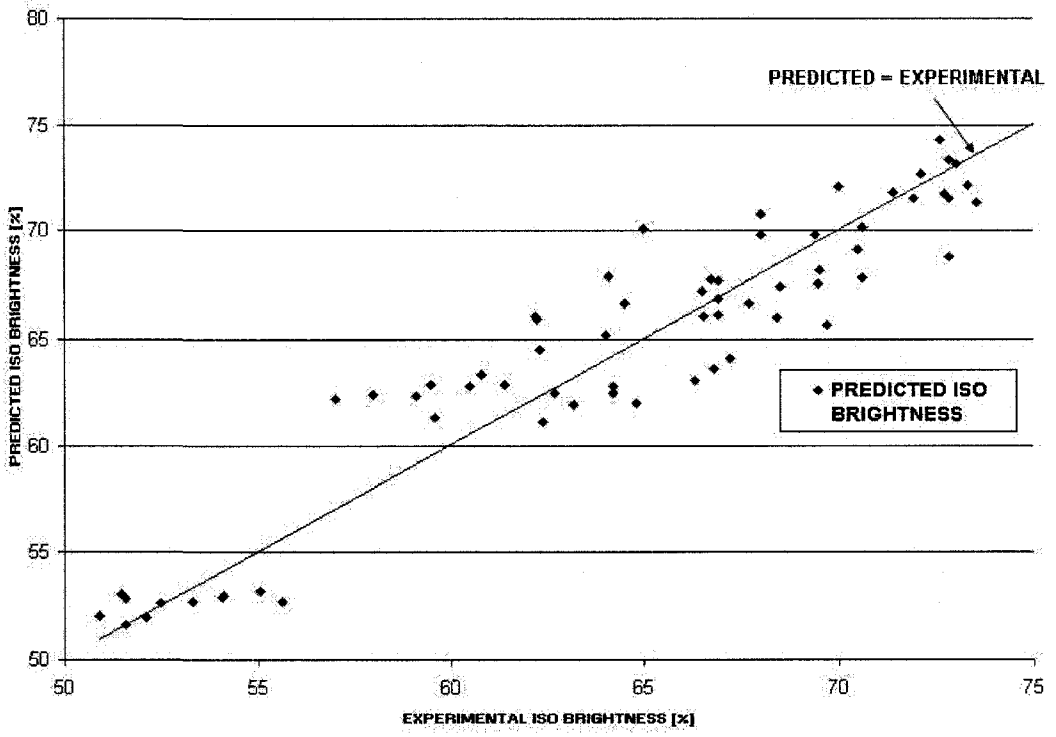


Figure II.5 Predicted vs. Experimental ISO Brightness.

0% peroxide charge. These values correspond the non bleached pulp can lead us to draw some relationship rules between raw material qualities and pulp quality. The

data file is sorted twice, according to:

1. the predicted ISO brightness;

2. the experimental ISO brightness;

The curves representing the average values of H, S and L versus the experimental ISO brightness is shown in Fig. II.6, while Fig. II.7 shows the variations of the H, S and L parameters versus the predicted ISO brightness.

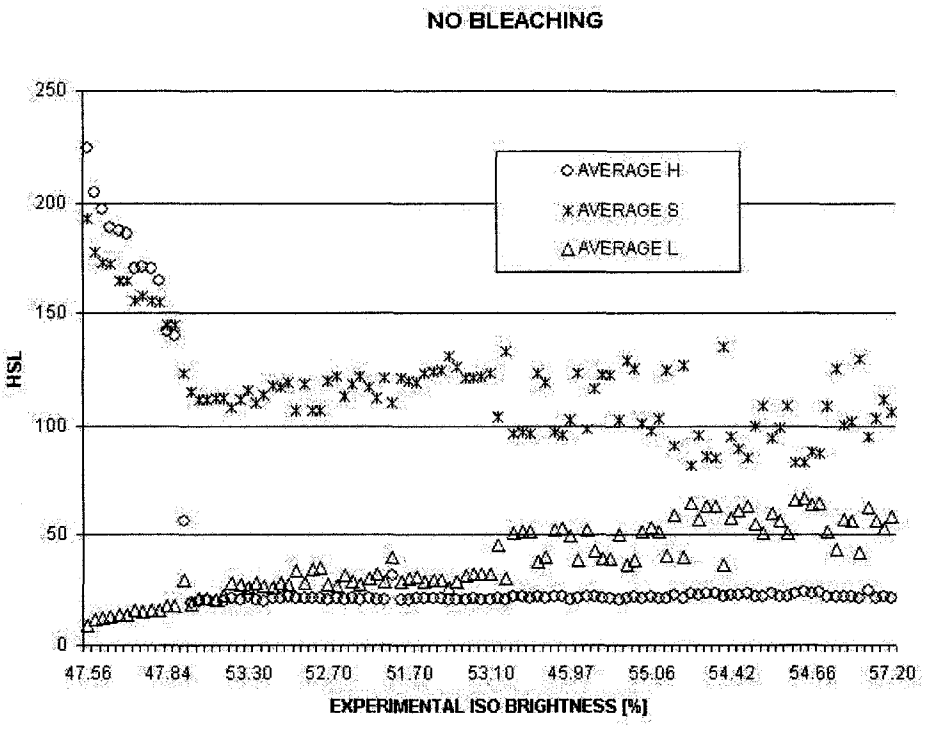


Figure II.6 Averages of H, S and L versus experimental ISO brightness.

From Fig. II.6 and II.7, one can easily notice, from the similar shape of the curves, that the influence of the H, S and L parameters on both the experimental and predicted ISO brightness is almost identical, which proves, again, that the genetically-

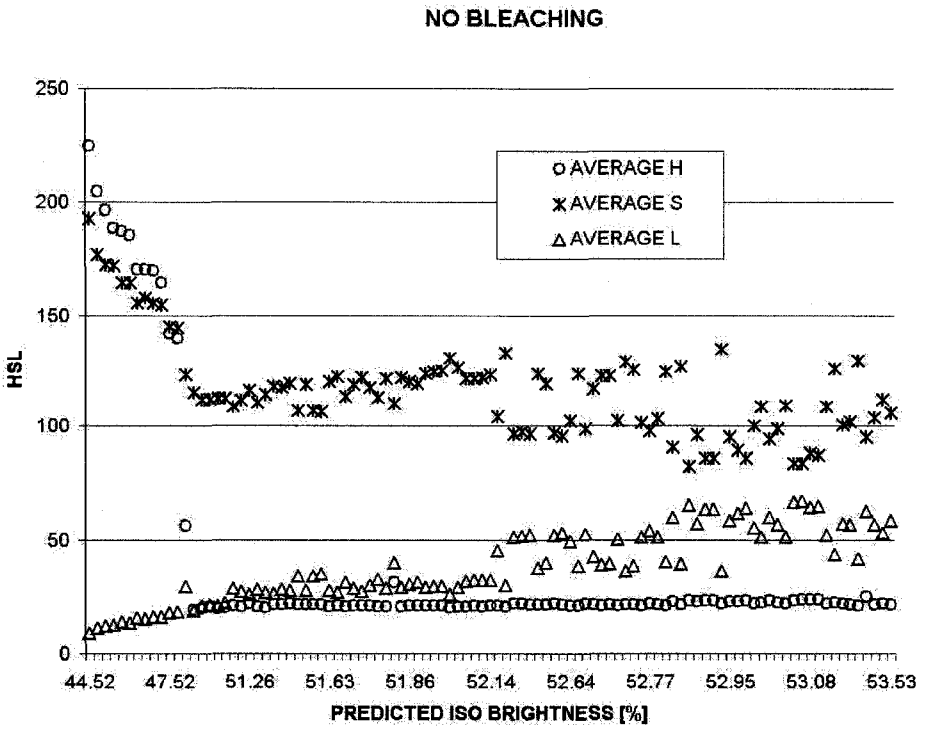


Figure II.7 Averages of H, S and L versus predicted ISO brightness.

generated FKB respected the information structure (co-influences between the different parameters) contained in the data, rather than producing averaged values.

The highest ISO brightness is obtained for relatively low values of S, around the mean values of H (H doesn't vary significantly) and the highest values of L. However, the worst ISO brightness is obtained for the lowest average of L, the highest averages of H and S, in this case the variation of average of L is the less significant especially compared to the averages of H and S, which shows their importance in the prediction of the ISO brightness.

Now, let us take into account the part of the learning file corresponding to the 5% peroxide charge (the highest bleaching charge giving the highest ISO brightness level). The variations of the predicted and the experimental ISO brightness are presented in Fig. II.8, where one can see that the shapes of the two curves are very close; another proof of the stability of the genetically-generated FKB.

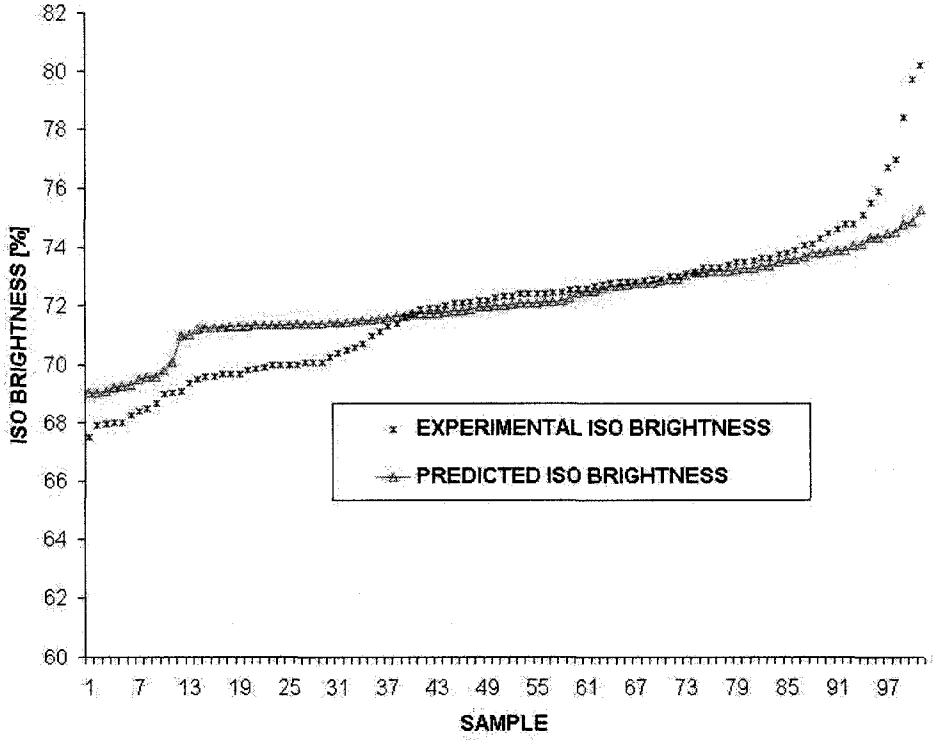


Figure II.8 ISO brightness for a 5% peroxyde.

## II.6 Conclusion

The RBCGA produced satisfactory FKBs for the prediction of pulp ISO brightness, while using exclusively variables obtained from image/color analysis, namely H, S and L, along with the % of peroxide concentration, which adds to the general knowledge about the relationship existing between wood chips specifications and pulp quality. These results can be of very high interest, since it allows optimizing the consumption of bleaching material, without compromising the quality of the pulp and the paper. However, one has to take these results carefully, because they only predict ISO brightness and the learning didn't take into account variables that can greatly influence the TMP process such as: chips size, humidity, presence of barks, knots, etc. However, if a fast learning using a simple model, this HSL FKBs can be very useful, since the genetically generated FKB presented a very stable performance when confronted with a set of hidden data, which was shown by the very little change between the learning and the testing rms error. The RBCGA was able to reach a very good balance between the accuracy of the FKB and its simplicity, reflected by the very good fitness value and the low number of fuzzy sets and fuzzy rules.

## II.7 Acknowledgments

The authors want to acknowledge the partners of the Centre d?Exellence pour la Valorisation des Copeaux (MNRFQ, PRECARN, UQTR, CRIQ, ABITIBI div. BELGO, KRUGER Trois-Rivières and BOWATER Gatineau). Financial support from the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618, RGPIN-105518 and STPGP-269579 is gratefully acknowledged.

# RÉFÉRENCES

[1] Achiche, S., Balazinski, M. and Baron, L., "Multi-combinative strategy to avoid premature convergence in genetically-generated fuzzy knowledge bases", *Journal of Theoretical and Applied Mechanics*, No. 3, Vol. 42, pp. 417–444, 2004.

[2] Balazinski, M., Bellerose, M. and Czogala, E., "Application of fuzzy logic techniques to the selection of cutting parameters in machining processes", *International Journal for Fuzzy Sets and Systems*, Vol. 61, pp. 307–317, 1993.

[3] Ding, F., Benaoudia, M., Bedard, P., Lanouette, R., Lejeune, C. and Gagne, P., "Wood Chip Physical Quality Definition and Measurement", *International Mechanical Pulping Conference*, pp. 367–373, 2003.

[4] Lanouette, R., Bedard, P. and Benaoudia, M., "Effect of woodchips characteristics on the pulp and paper properties by the use of PLS analysis", $90^{th}$ Annual Meeting - Pulp and Paper Technical Association of Canada (PAPTAC), pp 39–43, 2004.

[5] Michalewicz, Z., "Genetic algorithms + data structure = evolution programs", *New York: Springer*, 1992.

[6] Popson, S.J., Malthouse, D.D. and Robertson, P.C., "Applying Brightness, Whiteness, and Color Measurements to Color Removal", *TAPPI Journal*, Vol. 80,No. 9, pp. 137–146, 1997.

[7] Strand, B.C., "The effect of Refiner Variation on Pulp Quality", *International Mechanical Pulping Conference Proceedings*, pp.125-130, 1995.

[8] Zadeh, L.A., "Outline of new approach to the analysis of complex systems and decisions processes", *IEEE Transactions of Systems, Man and Cybernetics*, Vol. 3, pp. 28–44, 1973.