Titre: Title:	Génération automatique de bases de connaissances floues pour les systèmes d'aide à la décision
Auteur: Author:	Sofiane Achiche
Date:	2000
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Achiche, S. (2000). Génération automatique de bases de connaissances floues pour les systèmes d'aide à la décision [Master's thesis, École Polytechnique de Montréal]. PolyPublie. https://publications.polymtl.ca/6942/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/6942/
Directeurs de recherche: Advisors:	Luc Baron, & Marek Balazinski
Programme: Program:	Unspecified

UNIVERSITÉ DE MONTRÉAL

GÉNÉRATION AUTOMATIQUE DE BASES DE CONNAISSANCES FLOUES POUR LES SYSTÈMES D'AIDE À LA DÉCISION

SOFIANE ACHICHE
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION
DU DIPLÔME DE MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE MÉCANIQUE)
NOVEMBRE 2000

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé:

GÉNÉRATION AUTOMATIQUE DE BASES DE CONNAISSANCES FLOUES POUR LES SYSTÈMES D'AIDE À LA DÉCISION

présenté par: ACHICHE Sofiane

en vue de l'obtention du diplôme de: Maîtrise ès sciences appliquées

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

M. MASCLE Christian, Doctorat, président

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

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

M. KAJL Stanislav, Ph.D., membre

REMERCIEMENTS

Mes remerciements vont naturellement à mon directeur de recherche Luc Baron et mon codirecteur de recherche Marek Balazinski, pour leurs qualités tant humaines que scientifiques, pour leur encadrement et leur patience.

Je tiens à remercier mes amis, toute ma famille, spécialement mes parents pour leur soutien continuel tout au long de mes études et ma soeur Kamila à qui je dois ce travail.

RÉSUMÉ

Ce mémoire présente l'automatisation, par le biais d'un algorithme génétique, du processus de construction de bases de connaissances pour les systèmes d'aide à la décision utilisant la logique floue. L'objectif principal de ce travail est de démontrer que par l'utilisation d'un algorithme génétique, il est possible de générer de façon automatique une base de connaissances sans avoir besoin des services d'un expert, qui doit, généralement, étudier le comportement du fichier de données pour créer son propre modèle.

Le premier chapitre de ce mémoire est une synthèse qui passe en revue, de manière générale, le travail fait dans les chapitres 2, 3 et 4 et présente de façon sommaire la méthode d'optimisation utilisée ainsi que le logiciel de logique floue utilisé comme support d'application à cette méthode.

Le deuxième chapitre consiste en un article dans lequel nous trouvons une revue des principes de base des logiciels d'aide à la décision utilisant la logique floue. L'algorithme génétique, qui est la méthode d'optimisation utilisée pour effectuer la génération automatique des bases de connaissances à partir de données numériques, y est présenté. Les différentes opérations d'évolution effectuées par l'algorithme génétique, la description des différents critères de performance, la structure de l'algorithme génétique, ainsi que la personnalisation de celui-ci par rapport au

problème étudié sont présentées. Une explication de la paramétrisation utilisée ainsi que des restrictions que nous nous sommes imposés sont mises en évidence, chose faite dans le but de simplifier le codage.

L'algorithme génétique utilisé dans ce travail évalue chaque nouvelle base de connaissances par rapport à deux critères de performance soit la précision de la base de connaissances quant à la reproduction des donnés numériques et la simplicité de celle-ci, qui, est exprimée par le plus petit nombre possible de règles floues.

Dans cette partie, les résultats de plusieurs essais faits sur des données représentant des surfaces théoriques ainsi que sur des données expérimentales sont présentés et discutés. Il est à noter que tous les exemples pris en considération sont des systèmes à deux entrées et une sortie ce qui les placent dans la catégorie des systèmes MISO—multiple input single output—.

Le troisième chapitre traite de l'influence des paramètres utilisés dans le processus d'évolution de l'algorithme génétique durant la construction des bases de connaissances. Une revue de ces critères ainsi qu'une classification dans deux catégories principales sont faites, soit: les critères d'optimisation et les critères de sélection. À cet effet, des essais sont effectués sur une base de données représentant une surface théorique pour mettre en évidence l'influence de ces critères sur la solution (base

de connaissances) finale.

Une application de l'algorithme génétique sur un problème de suivi d'usures des outils est le sujet du quatrième et dernier chapitre. Il s'agit toujours d'un système MISO, mais cette fois avec trois entrées et une sortie. Les résultats obtenus sont comparés avec ceux produits par un réseau neuronique, ainsi que ceux obtenus par le biais d'une construction manuelle de la base de connaissances réalisée par un expert en la matière.

De par les résultats obtenus lors des différentes applications, il ressort que l'une des limitations principales, est la discrétisation des paramètres qui limite la finesse des résultats obtenus. Une discrétisation plus fine permettrait sans doute de trouver des bases de connaissances plus simples et plus précises, mais cela serait fait au détriment du temps d'exécution, car l'espace de recherche (champs des solutions possibles) augmenterait considérablement.

L'algorithme génétique permet de trouver des solutions assez satisfaisantes sans avoir à explorer tout l'espace de recherche ce qui est un avantage certain et un attrait majeur de cette méthode.

ABSTRACT

This work presents a genetic-based method for the automatic generation of fuzzy knowledge bases used in decision support systems. The main objective in this research is to demonstrate, by the mean of a genetic algorithm, the possibility of generating automatically fuzzy knowledge bases without having to rely on an expert.

The first chapter of this these is a synthesis that present a general overview of the research work done in the chapters 2, 3 and 4, it also presents the optimization method developed through this work along with the fuzzy-logic software used as an application.

The second chapter is a paper containing a brief review of basic principles of fuzzy decision support systems. The genetic algorithm—which is the optimization method used to generate automatically the knowledge bases from numerical data—is presented. The different operations made by the genetic algorithm are described along with the different fitness values, the structure of the algorithm and it's customization operations—developed for the sake of the problem under consideration—. An explanation of the parameterization and the restrictions taken into consideration are exposed (the restrictions are made for the sake of coding sim-

plicity).

The genetic algorithm used in this work evaluates each knowledge base along two fitness values, namely: the approximation accuracy—or alternatively the approximation error—; and the level of complexity—the simplest knowledge bases are those with the fewest number of fuzzy rules—of a knowledge base. The results of several tests made on numerical data that represent different theoretical surfaces are presented and discussed. It is noted that all the examples are two inputs one output, which class them into the MISO category—multiple input single output—.

The third chapter studies the influence of the criteria used in the evolution process of the knowledge bases. A review of those criteria is made along with their classification in two main categories, namely the optimization and selection criteria. Different tests are made with numerical data coming from a theoretical surface in order to enhance the influence of these criteria on the final solution (knowledge base).

An application to a tool monitoring problem is the subject of the fourth and last chapter. It is a MISO system, but this time with three inputs and one output. A comparison is made of the automatically generated knowledge bases with those obtained from a neural network and a manual construction of a knowledge base by

an expert.

From the different results of the different applications, an important conclusion can be resorted, which is that the discretization of the parameters represents an important draw-back of this method. A refined discretization can obviously lead to more precise and more simple knowledge bases, but the counterpart is an important increase of the research space (the field of possible solutions).

The genetic algorithm allows certainly to find satisfactory solutions without having to explore the entire field of possible solutions, which is a major and very attractive feature of this method.

TABLE DES MATIÈRES

LISTE DES FIGURES

LISTE DES NOTATIONS ET DES SYMBOLES

AG : Algorithme génétique.

COG : Centre de gravité CENTER OF GRAVITY.

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

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

G : Génotype.

GA : Algorithme génétique GENETIC ALGORITHM.

HM : Méthode des hauteurs HEIGHT METHOD.

K : Nombre maximum de règles floues.

KB : Base de connaissances KNOWLEDGE BASE.

MISO : Entrées multiples sortie simple MULTIPLE INPUT SINGLE OUTPUT.

MOM : Moyenne des maximums MEAN OF MAXIMA.

N : Nombre de prémisses ou nombre d'entrées.

P : Taille de la population.

R : Agrégation des règles floues.

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

SAD : Système d'aide à la décision.

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.

p_1 : Probabilité de croisement simple SIMPLE CROSS-OVER.

p_2 : Probabilité de déplacement de sous-ensembles flous.

p_3 : Probabilité de réduction du nombre de règles floues.

p_4 : Probabililté de mutation.

t_1 : Probabilité de croisement simple SIMPLE CROSS-OVER.

t_2 : Probabilité relative de déplacement de sous-ensembles flous.

t_3 : Probabilité relative de reduction du nombre de règles floues.

ϵ_{RMS} : Erreur rms.

$\mathcal{L}$: Espace d'apprentissage.

ϕ_1 : Indice de performance d'approximation.

ϕ_2 : Indice de performance du niveau de complexité.

ϕ : Indice de performance pondéré.

ω : Pondération.

ω_o : Critère d'optimisation.

ω_s : Critère de selection.

$\wedge$: 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

INTRODUCTION

Ce mémoire est rédigé par articles, de ce fait deux des trois chapitres qu'il contient (chapitre 2 et chapitre 4), sont les versions originales des textes soumis à des journaux scientifiques. Afin de permettre au lecteur de mieux comprendre les différents concepts et le vocabulaire utilisés dans ce travail, l'introduction contient une explication des concepts de bases de la logique floue et des algorithmes génétiques, qui sont les deux principaux sujets de ce mémoire. Il est à noter que les chapitres 2 et 4 incluent une bibliographie propre à ces articles qui utilise une numérotation différente de la bibliographie présentée à la fin de ce mémoire.

Plusieurs problèmes de fabrication exigent souvent une expertise qui n'est pas facile à modéliser. Par exemple, les décisions sont très souvent prises dans un environnement où les contraintes et les conséquences ne sont pas toujours précisément connues. Pour gérer quantitativement l'imprécision, on peut généralement utiliser les concepts et techniques de la théorie des probabilités et les outils provenant de la théorie des décisions.

Pour le moment nous sommes incapable de fabriquer des machines qui puissent rivaliser avec l'homme pour l'exécution des tâches, telles que la connaissance des langues, la traduction des langues, la compréhension de l'intention, de l'abstraction,

de la généralisation, la prise de décision dans l'incertain, etc.

Dans une large mesure, notre capacité à fabriquer de telles machines s'explique par la différence fondamentale qui existe entre l'intelligence humaine, d'une part, et l'intelligence de la machine, d'autre part. Cette différence provient de l'aptitude du cerveau humain à penser et à raisonner en termes imprécis, "flou". Par flou, on comprend les types d'imprécision associés avec la théorie des sous–ensembles flous. Les sous-ensembles flous sont des groupes d'objets qui peuvent être caractérisés par des adjectifs comme : grand, petit, important, approximatif, etc.

Dans la vie courante, la majorité des sous-ensembles n'a pas de limite précise. On utilise les notions comme: Jean est petit, une belle femme, les petites voitures. Ces énoncés fournissent de l'information malgré leurs imprécisions.

Associer le mot flou avec le mot logique est choquant. La logique au sens propre du mot, est une conception des mécanismes de la pensée qui ne devrait jamais être floue, toujours rigoureuse et formelle. En approfondissant les mécanismes de la pensée, les mathématiciens se sont aperçus qu'il n'y a pas en réalité, une logique unique mais autant de logiques que l'on veut, tout dépend de l'axiomatique choisie. La logique booléenne est donc la logique 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 [1].

La figure 1 montre des sous-ensembles flous ainsi que les termes et les concepts les plus usuels.

Concept : Domaine auquel appartiennent les différents faits.

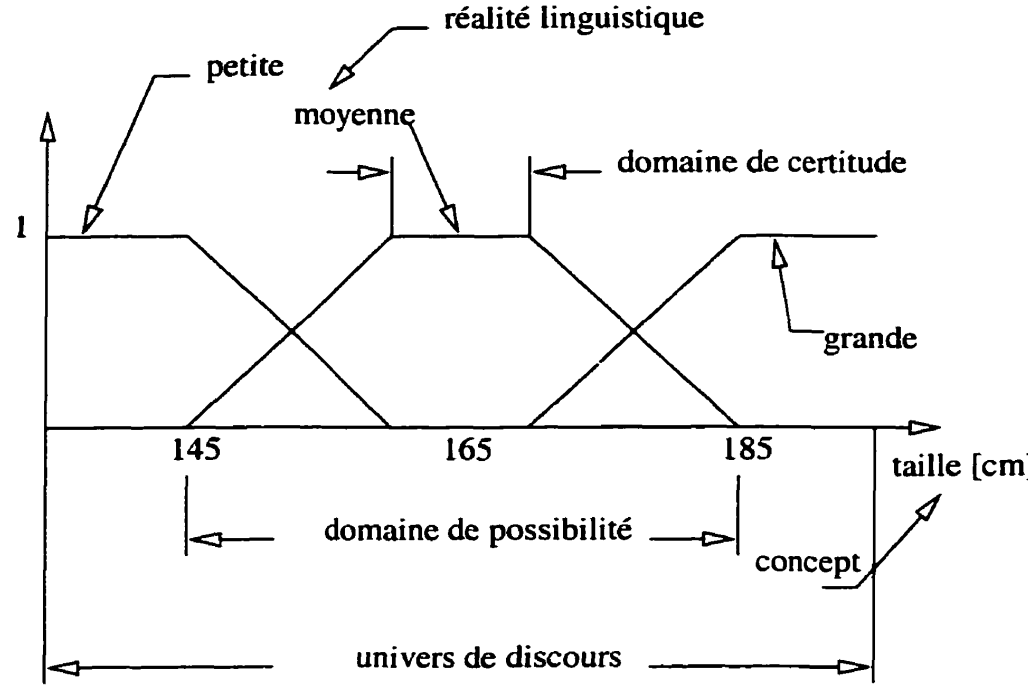

Figure 1 Termes et concepts en logique floue

ex. : la taille.

Réalité linguistique : Expression linguistique d'un fait.

ex. : pour le concept de la taille, on a des réalité linguistiques : petite, moyenne et grande.

Évaluation : évaluation faite par l'observation et le jugement d'un cas particulier.

ex. : on regarde une personne et on évalue sa taille à environ 170 cm.

Domaine de discours : champ de définition d'un concept.

ex. : la taille, dans notre exemple, est définie pour une personne d'âge adulte, de sexe masculin, vivant en Amérique du nord.

Possibilité 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 U, appelé référentiel ou univers de discours.

La logique floue, par sa spécificité, donne une réponse plus flexible aux données d'entrées. Les réponses ne sont plus régulières et continues. Elles sont très bien appropriées pour le contrôle des systèmes de positionnement ainsi que pour développer des systèmes d'aide à la décision dans des conditions incertaines, comme le logiciel Fuzzy-Flou développé à l'École Polytechnique de Montréal et l'Université de Technologie de Silésie à Gliwice (Pologne).

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" [2] et de choix de condition de coupe [3].

Récemment, les applications sont devenues de plus en plus nombreuses comme la

prédiction de maintenance préventive [4] et l'allocation des tolérances [5]. Ceci est dû au fait que la logique floue permet un contrôle là où il n'existe pas de modèles mathématiques du système à contrôler ou lorsque ceux qui existent sont trop compliqués, trop complexes ou ne s'appliquent que dans des circonstances très spécifiques.

Néanmoins, les systèmes d'aide à la décision utilisant la logique floue présentent un handicap certain, qui est le besoin d'un expert pour compiler les connaissances et construire de façon manuelle la base de connaissances (une base de connaissance comprend des sous-ensembles flous et des règles liant ces derniers). A cet effet beaucoup de travaux ont été faits dans un effort d'automatisation de ce processus afin de rendre le besoin d'un spécialiste moins déterminant. Ces travaux se sont, en premier lieu, concentrés sur la génération automatique des règles floues tout en admettant une répartition de sous-ensembles flous précise, et cela en utilisant des méthodes purement numériques [6], ou bien des algorithmes génétiques (AG) [7 – 11]. D'autres travaux ont été faits dans le cadre de la génération automatique des sous-ensembles flous (leurs répartitions) en utilisant des AGs [12, 13].

Un algorithme génétique est une méthode d'optimisation qui évalue une fonction objectif à un nombre fini de points. Cette méthode est basée sur l'analogie avec le mécanisme de génétique naturelle et imite l'approche Darwinienne de la sélection naturelle [14]. En général, un AG est caractérisé par

1. un codage de chaque solution possible, sous forme d'une chaîne de bits, (appelée chromosome);

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 croisement, mutation et sélection naturelle, afin de permettre l'évolution de la population de génération en génération.

Les AGs utilisent une amélioration itérative des solutions 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.

Le but principal de la recherche présentée dans ce mémoire est de montrer que l'utilisation d'un algorithme génétique pour l'automatisation complète du processus de construction des bases de connaissances peut mener à des résultats très satisfaisants. Plus spécifiquement, les objectifs sont de :

- contribuer à l'avancement des algorithmes génétiques dans le champs des systèmes d'aide à la décision utilisant la logique floue par l'étude des questions de *codage*, *opérateurs de reproduction* et *évaluation* des différentes bases de connaissances;

- produire un logiciel de construction automatique qui permette de réduire au minimum la dépendance par rapport à un expert;
- élargir le champs d'utilisation des systèmes d'aide à la décision utilisant la logique floue.

Le chapitre 1 est une synthèse qui passe en revue le travail fait dans ce mémoire. Le chapitre 2, présente le travail fait dans le cadre d'un article soumis au journal "International Journal of Approximate Reasonning" [15], dans lequel on trouve une explication de l'AG utilisé et plusieurs exemples d'application. Le chapitre 3 présente l'influence, des paramètres utilisés dans l'AG, sur la solution finale (bases de connaissances résultantes). Une application sur un problème de contrôle d'usure d'outils, ainsi qu'une comparaison avec d'autres méthodes d'intelligences artificielles sont présentées dans le chapitre 4. Un travail fait dans le cadre d'un article soumis au journal "Engineering Applications of Artificial Intelligence" [17]. Finalement, une conclusion générale et une discution des perspectives de travaux futurs est faite et elle est suivie d'une bibliographie générale du mémoire.

CHAPITRE 1

SYNTHÈSE DU MÉMOIRE

Comme ce mémoire est présenté "par articles", ce premier chapitre contient une synthèse générale, en français, de tout le mémoire, ainsi qu'une présentation des notions générales nécessaires à la compréhension des chapitres qui suivent. Les chapitres 2 et 4 contiennent chacun un article soumis à un journal scientifique. Le chapitre 3 effectue le lien entre les chapitres 2 et 4 et contient également un article présenté à une conférence.

1.1 Synthèse du chapitre 2

Le chapitre 2 contient l'article intitulé *"Fuzzy Decision Support System Knowledge Base Generation using a Genetic Algorithm"* [15] soumis à *"International Journal of Approximate Reasonning"*. Ce chapitre présente une méthode utilisant un algorithme génétique (AG) permettant la construction et l'optimisation de bases de connaissances floues pour des systèmes d'aide à la décision (SAD). La génération des bases de connaissances se fait donc de façon complètement automatique, c'est-à-dire sans avoir recours à un expert, et cela seulement à partir de données numériques. Le SAD utilisé est le "FDSS Fuzzy-Flou", un logiciel développé à l'École Polytechnique de Montréal et à l'Université de Technologie de

Silésie à Gliwice (Pologne) utilisant la logique floue.

Chaque base de connaissances construite est composée de deux prémisses d'entrée et d'une conclusion. Elle contient aussi les informations sur le nombre et la répartition des sous-ensembles flous sur chaque entrée et sur la conclusion (sortie), ainsi que les règles floues qui relient les sous-ensembles floues entre eux. Ces bases de connaissances sont composées d'un nombre minimal de sous-ensembles et de règles flous, il s'agit là d'une approche minimaliste de la méthode, qui construit des bases de connaissances selon deux critères de performance complètement contradictoire, à savoir minimiser les erreurs tout en maintenant le nombre de règles et d'ensemble flous le plus bas possible. Atteindre cet objectif permet l'obtention de bases de connaissances complètes et simples ce qui facilite un raffinement *a priori* par un expert.

Pour valider la méthode d'automatisation développée, différents essais sur des surfaces théoriques—dont le niveau de difficulté quant à la prédiction de la base de connaissances est graduelement augmenté tout au long du chapitre 2—ont été faits, ainsi qu'une application sur une base expérimentale de données de taille réduite, permettant de prédire les efforts de coupe en tournage, suivie d'une comparaison des résultats obtenus par AG avec ceux obtenus par une méthode mathématique généralement utilisée en usinage (surface de Taylor).

La structure du chapitre 2 peut être résumée par les points suivants:

- une introduction contenant une revue bibliographique;
- une présentation de la méthode développée basée sur un AG;
- une présentation du système d'aide à la décision "Fuzzy-Flou";
- les essais de validation de la méthode sur des surfaces théoriques et un ensemble de données expérimentales;
- une conclusion et une interprétation des résultats obtenus.

Tout les problèmes traités dans le chapitre 2 sont à deux entrées et une sortie, ce qui les classe dans le type MISO MULTIPLE INPUT SINGLE OUTPUT.

1.2 Synthèse du chapitre 3

Comme on l'a déjà mentionné, le chapitre 3 joue le rôle de lien entre les chapitres 2 et 4. Ce chapitre est en partie basé sur un travail publié dans le compte rendu de conférence *International Conference on Advanced manufacturing Technology* [16]. Il s'agit d'une analyse de l'influence des paramètres utilisés lors de l'évolution de l'AG, ces paramètres peuvent être subdivisés en deux groupes distincts :

1. paramètres (ou critères) de sélection;
2. paramètres (ou critères) d'optimisation.

Les paramètres de sélection permettent de choisir une base de connaissances parmi toutes celles proposées dans la population finale. Il est possible de choisir la base de connaissances selon plusieurs critères, c'est-à-dire: une faible erreur, un nombre minimal de règles floues ou une pondération des deux. Les paramètres d'optimisation sont eux utilisés par le mécanisme de reproduction. Ils constituent les probabilités d'application des différents types de mécanisme de reproduction, ainsi que la pondération entre les critères d'évaluations.

L'étude est appliquée à la surface modèle présentée à la figure 2.13. Il est important de préciser que dans ce chapitre le processus d'évolution de l'AG n'est pas complète, il est arrêté à un nombre de générations relativement bas. Ceci, permet de mieux apprécier les différences existant entre les individus d'une même population et de comprendre l'influence des probabilités appliquées aux différents types de mécanismes de reproduction, qui sont comme suit:

- p_1 : probabilité de croisement simple SIMPLE CROSS-OVER;
- p_2 : probabilité de déplacement de sous-ensembles flous;
- p_3 : probabilité de réduction du nombre de règles floues;
- p_4 : probabilité de mutation.
- ω_o : critère d'optimisation;
- ω_s : critère de sélection.

La raison du choix de l'arrêt du processus d'évolution de l'AG à des populations plus ou moins jeunes, est dû au fait que si l'évolution était poussée assez loin, il en résulterait des solutions trop similaires, d'un point de vue performance, et donc des résultats difficiles à interpréter dans le sens qu'il serait presque impossible de comprendre l'influence de chacune des probabilités, listées ci-dessus, sur le processus d'optimisation des bases de connaissances. Des solutions moins avancées permettent d'apprécier le rôle des critères d'optimisation et de sélection, car elle montrent de façon plus claire la tendance de l'évolution des bases de connaissances. Une analyse des différents résultats obtenus pour les différentes pondérations est faite. Ce qui a permit de mieux souligner le rôle de chaque paramètre et l'étendue de son influence sur la solution finale. Aussi, ce chapitre permet de donner une meilleure idée sur les ordres de grandeurs—valeurs standards—d'utilisation des paramètres d'optimisation et de sélection dans le processus d'évolution de l'AG. Le chapitre 3 permet donc de bien assimiler l'influence des pondérations utilisées dans l'AG et de ce fait facilite la compréhension d'une partie du travail présentée au chapitre 4. La structure de ce chapitre peut être résumée comme suit:

- le chapitre débute par la définition du problème que pose la pondération des différents critères d'évolution de l'AG, une brève présentation de ces derniers est faite;

- les différents essais ainsi que les résultats obtenus y sont relatés et les compi-

lations de ces résultats regroupées dans des tableaux;

- la dernière partie est une interprétation globale des résultats sous forme de conclusion.

Le modèle traité dans le chapitre 3 est à trois entrées et une sortie, ce qui le classe dans le type MISO.

1.3 Synthèse du chapitre 4

Le chapitre 4 contient un article intitulé *"Tool Condition Monitoring Using Genetically-generated Fuzzy Knowledge Bases"* [17] soumis à *"Engineering Applications of Artificial Intelligence"*, qui présente l'application de méthodes d'intelligences artificielles (IA) sur un problème de suivi de l'usure des outils. La première méthode utilisée dans le chapitre 4 est un réseau neuronique conventionnel. Celle-ci est utilisée comme base de comparaison des performances des systèmes d'aide à la décision utilisant la logique floue. Ces systèmes fonctionnent via des bases de connaissances qui sont, généralement, construites manuellement (deuxième méthode). Néanmoins, un algorithme génétique (AG) peut être utilisé pour automatiser cette construction et remplace le travail de l'expert (troisième méthode).

Concernant la partie génération automatique des bases de connaissances, plusieurs exécutions sont faites, utilisant différents critères d'évolution de l'AG, critères présentés est développés au chapitre 3. Les trois méthodes sont appliquées sur

deux ensembles de données expérimentales afin de prédire l'usure des outils en tournage. Le premier ensemble, appelé "W5", est utilisé pour la construction des bases de connaissances (phase d'apprentissage) . Le deuxième ensemble, appelé "W7", est utilisé pour vérifier les performances de la base de connaissances, et ceci est fait en appliquant les solutions proposées par les différents systèmes intelligents sur l'ensemble "W7" et en analysant les états d'erreurs des résultats obtenus. La différence majeure entre les trois méthodes réside bien évidement dans l'aspect d'automatisation car dans la deuxième méthode, qui est manuelle, c'est un expert qui doit analyser l'ensemble de données "W5" et qui doit donc décider du contenu de la base de connaissances. Le chapitre 4 est structuré comme suit :

- une introduction mettant l'emphase sur l'importance du problème de contrôle de l'usure d'outils et contenant une revue de la littérature;
- une présentation du système à réseaux de neurones et du système d'aide à la décision "Fuzzy-flou";
- une explication de la méthode d'automatisation utilisée—la même que celle du chapitre 2—basée sur un AG;
- les résultats de l'application des trois méthodes d'intelligence artificielle dans le domaine de la prédiction et du suivi de l'usure des outils;
- une interprétation globale des résultats suivie d'une conclusion finale.

Tout les problèmes traités dans le chapitre 4 sont à trois entrées et une sortie, ce qui les classe dans le type MISO.

1.4 Notions générales sur la méthode utilisée (AG)

Chaque base de connaissances construite est composée de deux ou trois prémisses d'entrées et d'une conclusion. Elle contient aussi les informations sur le nombre et la répartition des sous-ensembles flous sur chaque entrée et sur la conclusion, ainsi que les règles floues qui relient les sous-ensembles flous entre eux.

La méthode utilisée est, comme citée ci-dessus, basée sur un AG. Les AGs 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, la reproduction, la mutation et la sélection naturelle, qui peuvent être définies comme suit:

Reproduction: L'évolution de la population à chaque génération est obtenue par la reproduction des meilleurs individus, basés sur leur habilité à survivre à la sélection naturelle. La reproduction est généralement faite par croisement du chromosome de deux parents, afin d'obtenir le chromosome de deux enfants. L'une des techniques de croisement suit le mécanisme suivant:

- les parents sont sélectionnés en se basant sur leur indice de performance, les meilleurs sont favorisés ;

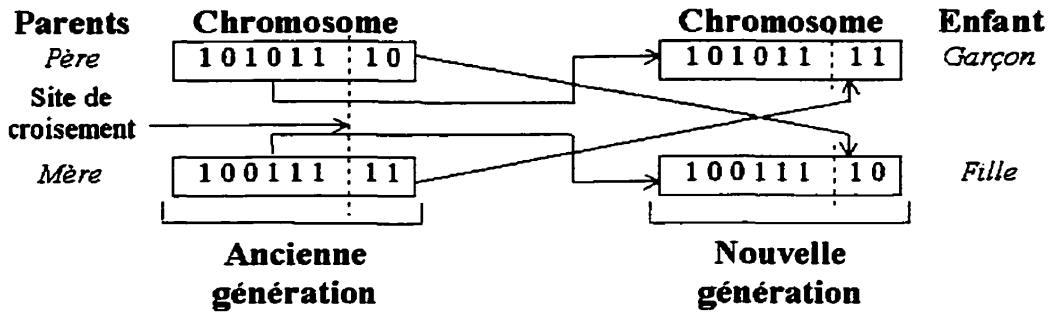

Figure 1.1 Croisement de deux individus.

- le chromosome des parents est subdivisé en deux parties à un site de croisement choisi aléatoirement;
- le chromosome des enfants est formé par la combinaison croisée des deux parties du chromosome des parents.

La figure 1.1 illustre ce mécanisme.

Mutation: La mutation est l'inversion aléatoire d'un bit dans le chromosome d'un individu lors de la reproduction. La mutation permet de considérer des solutions complètement différentes, et ainsi potentiellement trouver de meilleures solutions. La figure 1.2 montre la mutation d'un gène.

Sélection naturelle: La sélection naturelle est appliquée de façon à conserver les individus les plus prometteurs basés sur leurs indices de performance. Par simplicité, la taille de la population est conservée constante.

Une approche minimaliste est également suivie dans le processus d'automatisation afin que la base de connaissances satisfasse au mieux, deux critères d'optimisation

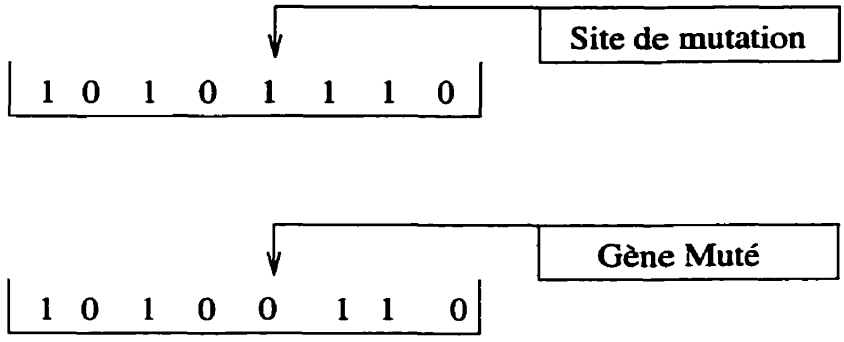

Figure 1.2 Opération de mutation.

contradictoires; à savoir approximer le mieux possible les données numériques d'entrée/sorties—les solutions les plus proches de l'optimum (environ 75%)—tout en utilisant un nombre minimal de règles floues—les solutions ayant le plus petit nombre de règles (environ 25%)—.

La première génération commence avec 100 individus (chaque individu représente une base de connaissances) crées aléatoirement, puis 100 individus supplémentaires sont crées par reproduction. Afin de conserver la taille de la population constante, la sélection naturelle est appliquée sur les 200 individus en les ordonnant selon chacun des deux critères d'évaluation. Ainsi les 50 premiers de chacun des deux critères (50+50=100) sont conservés tout en évitant la duplication. La figure 1.3 montre une vue d'ensemble du fonctionnement de l'algorithme. Il est à noter qu'une personnalisation du processus de reproduction de l'AG est aussi faite dans cet article. En plus du mécanisme de croisement simple présenté ci-dessus, deux autres mécanismes viennent s'ajouter, à savoir: le déplacement des sous-ensembles flous et

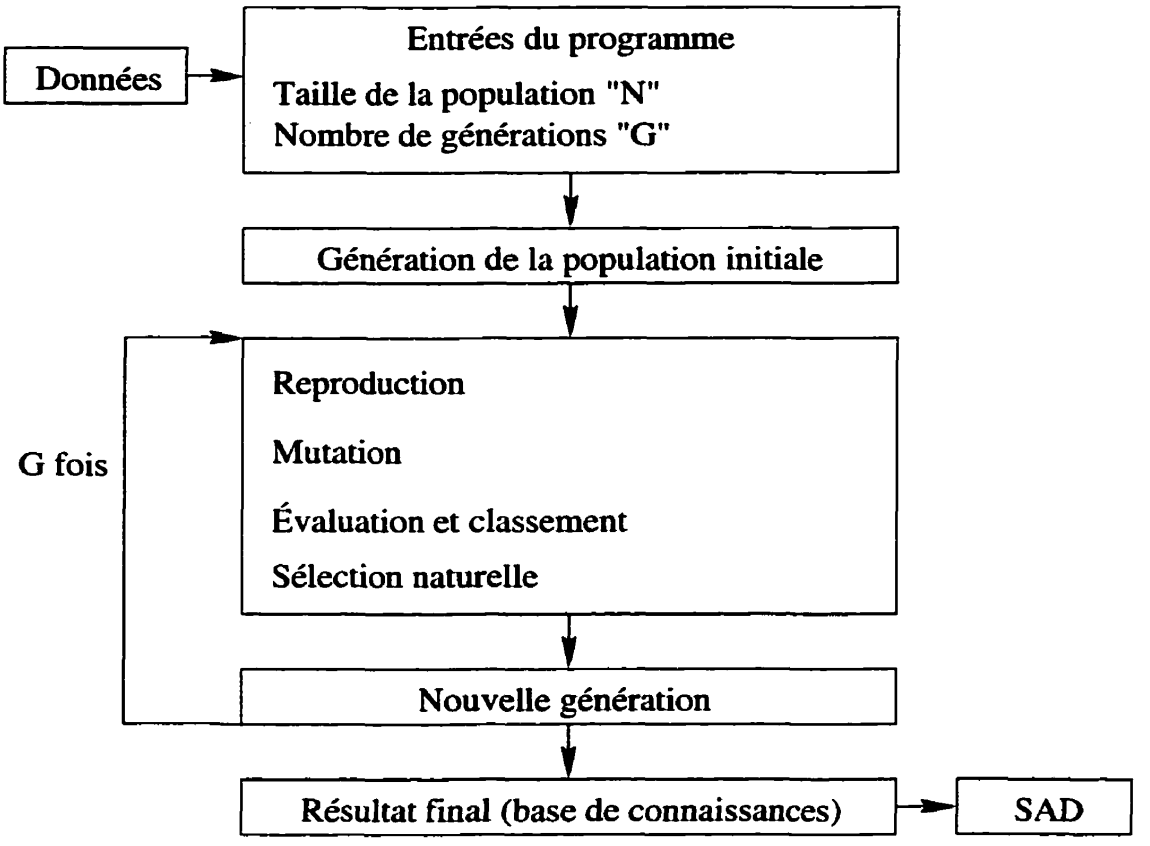

Figure 1.3 Vue d'ensemble du fonctionnement de l'AG

la réduction de règles floues. Ces deux opérations sont appliqués avec une certaine probabilité prédéterminée.

1.5 Préparation de la base de connaissances pour le SAD "Fuzzy-Flou"

Habituellement, les bases de connaissances sont construites manuellement par un expert. Conséquemment le temps de création et la qualité de la base de connaissances dépendent de l'habileté de l'expert. Celui-ci suit les étapes suivantes:

1. Définir les prémisses: il y a autant de prémisses que de variables d'entrées.

2. Définir les conclusions: il y a autant de conclusions que de variables de sorties.

3. Définir le nombre d'ensembles flous: dans chacune des prémisses d'entrées.

4. Définir le nombre d'ensembles flous: dans chacune des conclusions.

5. Définir la répartition des ensembles flous : leurs positions sur les prémisses puis sur les conclusions, ainsi que leurs formes en se basant sur les connaissances disponibles.

6. Définir les règles floues: qui représentent la relation entre les différents ensembles flous sur les différentes prémisses et conclusions.

Chaque individu contient le codage des sous-ensembles flous des prémisses X et Y, ainsi que la conclusion U. Dans notre cas, nous utilisons un maximum de 7 sous-ensembles flous pour chacune des prémisses (le chiffre 7 étant le nombre de sous-ensembles le plus communément utilisé en contrôle) et 8 sous-ensembles flous pour les conclusion. L'ensemble des règles ainsi que le nombre et la répartition des sous-ensembles flous sont optimisés par l'AG, afin de reproduire de façon optimale la surface demandée, tout en réduisant le nombre de règles. Une règle floue, pour un système à deux entrées et une sortie, s'exprime sous la forme suivante:

Règle 1

Si la valeur sur la prémisse X est de X_1 ;

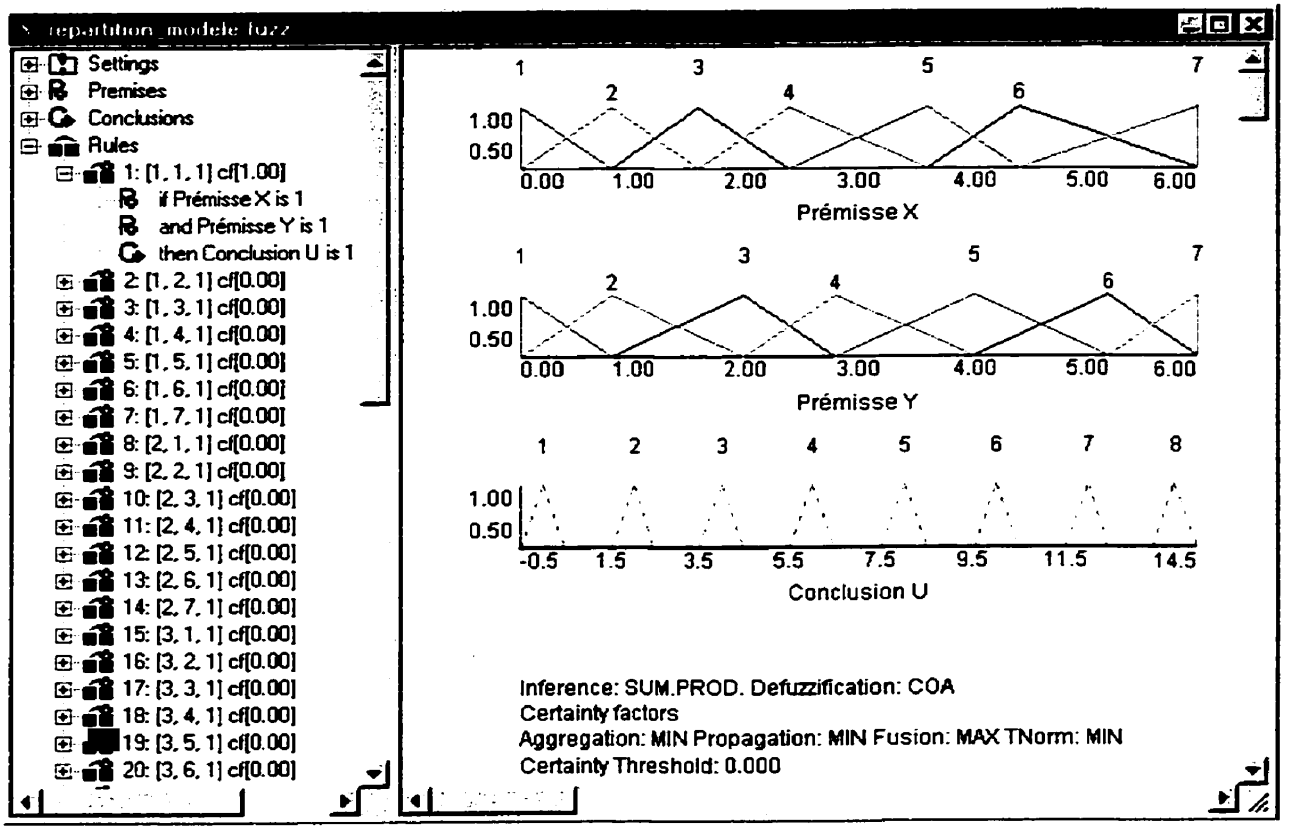

Figure 1.4 Interface graphique du SAD Fuzzy-Flou

et la valeur sur la prémisse Y est de Y_1 ;

alors la conclusion U est de U_1.

C'est ce que qui est communément appelé les règles "si-alors".

La figure 1.4 montre l'interface graphique du SAD avec une répartition maximale des sous-ensembles flous—système à deux entrées et une sortie—sur les prémisses X et Y et sur la conclusion U (colonne de droite). La colonne de gauche montre les prémisses X et Y, la conclusion U et l'ensemble des règles floues. Le SAD Fuzzy-Flou permet l'utilisation de fonctions d'appartenance de

formes trapézoïdales, comme le montre la fig.1.5. Néanmoins pour des raisons de simplicité de codage, nous considérons la forme triangulaire non symétrique pour les fonctions d'appartenance des deux prémisses d'entrées et triangulaire symétrique pour les fonctions d'appartenance de la conclusion. Comme montré à la fig.1.5, les valeurs modales $m1$ et $m2$ représentent le sommet du triangle ($m1$=$m2$), la valeur modale am représente la distance entre le sommet actuel et celui qui le précède, alors que la valeur modale bm est la distance entre le sommet actuel et celui qui lui succède, hm est la hauteur du triangle (fixée à 1).

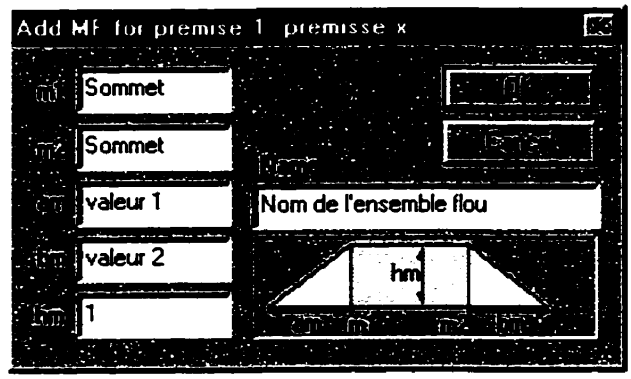

Figure 1.5 Fonction d'appartenance du SAD Fuzzy-Flou

1.6 Codage:

La nature possède sa propre méthode pour coder les phénotypes en chromosome, donc chaque problème d'optimisation doit aussi définir sa méthode de codage des paramètres en une chaîne de bits. Le nombre de bits alloués à chaque paramètre détermine sa résolution maximale. Dans ce mémoire, la position des sous-ensembles

flous et les règles floues sont codées comme suit:

Sous-ensembles flous: comme cité ci-dessus, nous utilisons des fonctions d'appartenance de forme triangulaire, dont la position du sommet est codée par 4 bits. Le sommet d'un triangle (fonction d'appartenance) est donc le début du prochain et ainsi de suite. Le choix d'une résolution de 16 (4 bits $\rightarrow 2^4 = 16$) peut être changé, mais au prix de l'augmentation du temps d'exécution de l'algorithme.

Règles floues: 4 bits sont alloués à chaque règle floue, le premier bit représente l'activation (si 1) ou la désactivation (si 0) de la règle, les trois autres bits qui restent constituent un pointeur vers le numéro d'un sous-ensemble flou sur la conclusion.

CHAPITRE 2

FUZZY DECISION SUPPORT SYSTEM KNOWLEDGE BASE GENERATION USING A GENETIC ALGORITHM

Soumis à *International Journal of Approximate Reasonning.*

Journal affilié à l'organisation NAFIPS.

North American Fuzzy Information Processing Society.

Edition *Elsevier Science.* [15]

Abstract

This paper presents a genetic algorithm (GA) that automatically constructs the knowledge base used by fuzzy decision support systems (FDSS). The GA produces an optimal approximation of a set of sampled data from a very small amount of input information. The main interest of this method is that it can be used to automatically generate (without the help of an expert) a fuzzy knowledge base—i.e., the fuzzy sets for premises, conclusions and the fuzzy rules—. This knowledge base is composed of the minimum number of fuzzy sets and rules. This *minimalist* approach produces fuzzy knowledge bases that are still manageable *a posteriori* by a human expert for fine tuning. The GA is validated through several examples of known behaviors and, finally, applied to experimental data.

Keywords: Fuzzy decision support system, knowledge base, learning, genetic algorithm.

2.1 Introduction

Nowadays fuzzy logic is increasingly used in decision-aided systems because it offers several advantages over other traditional decision-making techniques. The fuzzy decision support systems (FDSS) can easily deal with incomplete or/and imprecise knowledge applied to either linear or nonlinear problems. These systems have successfully been applied to many different problems such as: predictive maintenance [1], tool conditions monitoring [2], job dispatching [3] and tolerance allocation [4]. Unfortunately, all these cases require an expert in order to manually construct, from his own expertise, the fuzzy knowledge databases. Obviously, this *learning process* is lengthy. Moreover, the quality of the resulting knowledge base depends greatly on the objectivity and the teaching capacity of the expert. Consequently, many research works have been conducted toward the automatic generation of fuzzy knowledge bases. These works have first focused on different aspects of the automatic generation of fuzzy rules with either numerical methods [5] or genetic algorithms (GA) [6, 7, 8, 9, 10]. Although the number of rules increases exponentially with the number of fuzzy sets, GAs appear to be the most promising learning tool. Conversely, other works have focused on different aspects of the automatic generation of fuzzy sets with GAs [11, 12]. Membership functions

have been studied in terms of shape and probability, the quantity of fuzzy sets is user-defined, and hence, is not part of the learning process. An evolutionary paradigm of both fuzzy sets and rules using a genetic algorithm is proposed in [13]. The authors studied the membership probability on fuzzy sets and rules without including other fuzzy knowledge in the learning process. Clearly, there is a need for an automatic generation of fuzzy knowledge bases, which includes in the learning process: the quantity of fuzzy sets and rules; the repartition of the fuzzy sets on premises and conclusions; and the rules themselves. In this paper, we propose a genetic-based learning process of fuzzy knowledge bases to be used in a FDSS. Our method includes all the abovementioned knowledge aspects in the learning process. Here, we use *Fuzzy-Flou*, an FDSS software developed at École Polytechnique (Canada) and the Technical University of Silesia in Gliwice (Poland).

2.2 Problem Definition

First, let us present the FDSSs used in this research and the learning paradigm associated with fuzzy knowledge bases.

2.2.1 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 [14]. 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 [15]. 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:

$$R^k_{MISO} : \text{if } x_1 \text{ is } X^k_1 \text{ and } x_2 \text{ is } X^k_2 \text{ and } \cdots \text{ and } x_N \text{ is } X^k_N \text{ then } y \text{ is } Y^k, \qquad (2.1)$$

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

$$R = also_{k=1}^{K}(R_{MISO}^{k}). \tag{2.2}$$

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:

$$Y' = (X_1', X_2', \cdots, X_N') \circ R, \tag{2.3}$$

where $\circ$ denotes a compositional rule of inference (CRI), e.g., the *sup*-$\wedge$ or *sup-prod* (*sup*-$*$). Alternatively, the CRI of eq. (4.6) is easily computed as

$$Y' = X_N' \circ \cdots \circ (X_2' \circ (X_1' \circ R)). \tag{2.4}$$

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

$$Y' = \left\{ \begin{array}{c} \vee_{k=1}^{K} \\ \sum_{k=1}^{K} \end{array} \right\} \quad \underset{\{x_i \in X_i\}_{i=1}^{N}}{sup} \quad *_t(\ *_t(X'_N, \cdots, X'_2, X'_1)\ ,\ *_t(X_1^k, X_2^k, \cdots, X_N^k, Y^k)), \tag{2.5}$$

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. Although these use only two premises, our method is general and can be used with up to N premises.

2.2.2 FDSS Learning Paradigm

In general, FDSS requires a knowledge base in order to support the decision-making process of end-users. As shown in fig. 4.2, this knowledge base can be createdea manually by an human expert or automatically learned from a set of sampled data. This paper is concerned about the automatic learning process of FDSS knowledge base. Although contradictory, the overall objective is to find a knowledge base: 1) of minimum size; 2) that best approximates the set of sampled data. The first objective is aimed at producing small knowledge bases, i.e.,

knowledge bases that are still manageable by either a human expert or a computer. The second objective is aimed at accurately reproducing the set of learned data, while interpolating or extrapolating fair conclusions in other situations. The former allows for *a posteriori* tuning of the knowledge base by a human expert, while the latter allows for the learning of a very large set of sampled data and/or the handling of very complex decision-making problems. This *minimalist* approach is implemented through an automatic reduction of fuzzy rules and sets on the premises and on the conclusion, whenever the approximation error is not penalized too much by this reduction. Figure 4.2 presents our genetic-based FDSS knowledge base learning process.

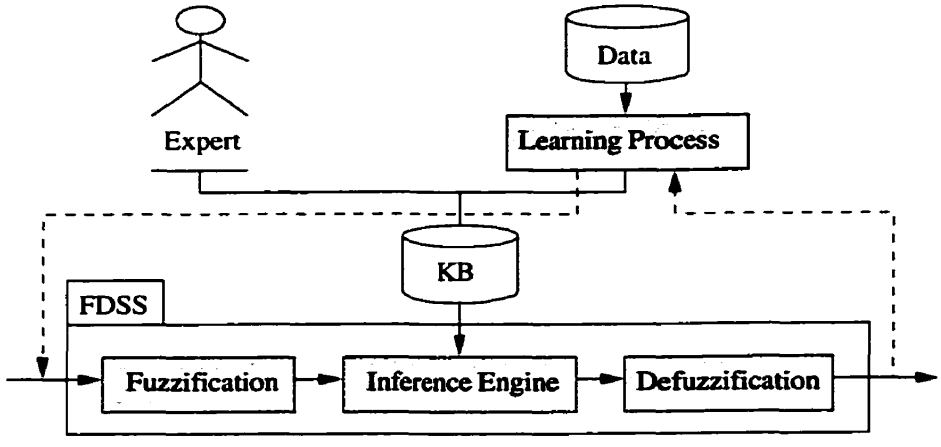

Figure 2.1 The learning paradigm of FDSS Fuzzy-Flou

2.3 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 [16]. As shown in fig. 4.3, each individual of a population is a potential FDSS Fuzzy-Flou knowledge base. The method uses iterative improvement of individuals at each generation to converge toward multiple optima simultaneously. 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—. It is noteworthy that

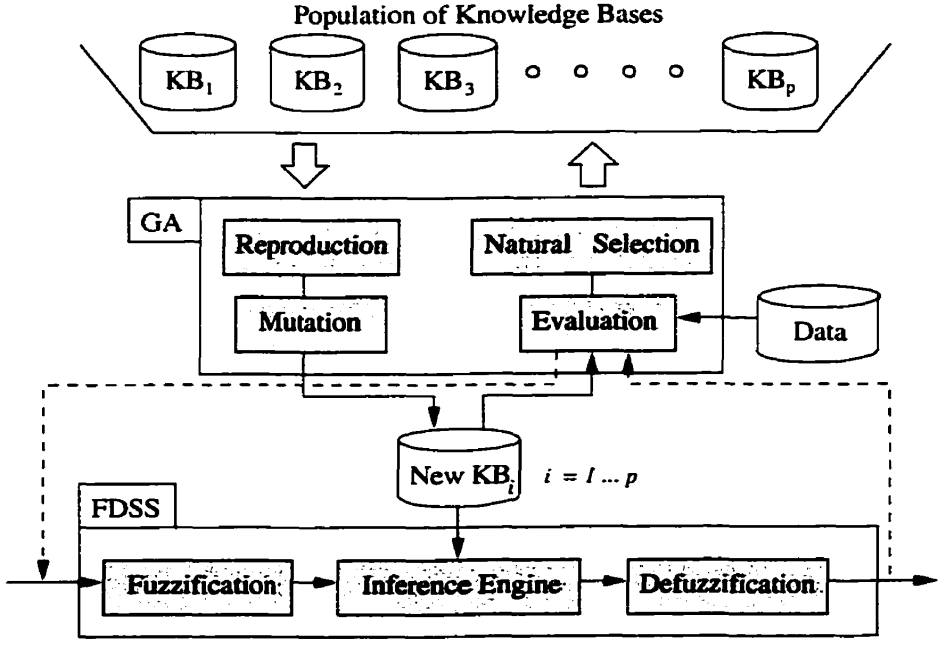

Figure 2.2 The GA learning process of an FDSS Fuzzy-Flou knowledge base

the coding of several parameters into bit strings is crucial in GA. When the number

of unknown parameters increases, GA exhibits only a polynomial increase in the size of the search space, while the other optimization techniques show an exponential increase. Figure 4.3 presents the *encoding/decoding scheme* as well as the four basic operations, i.e.: *reproduction*, *mutation*, *evaluation* and *natural selection*, of the developped GA learning sofware [17].

2.3.1 Encoding/Decoding Scheme

The genotype of an individual p member of a population of size P is defined as

$$G^p \equiv \{ G^p_{sets}, G^p_{rules} \}, \tag{2.6}$$

where G^p_{sets} and G^p_{rules} are respectively the genotypes of the fuzzy sets and rules. For the sake of brevity, the indice p is omitted in the following equations. However, it must be clear that all the following genotypes apply to any individual p.

Fuzzy sets:

The genotype of the fuzzy sets must contain all the information on the position of the fuzzy sets on the premises and the conclusion, i.e.:

$$G_{sets} \equiv \{G_{X_1}, G_{X_2}, \cdots, G_{X_N}, G_Y\}, \tag{2.7}$$

where G_v is the genotype of the n_v fuzzy sets on v, i.e.,

$$G_v \equiv \{\underbrace{10...01}_{g_v^1}\underbrace{11...10}_{g_v^2}\cdots\underbrace{01...11}_{g_v^{n_v}}\}, \quad \forall v \in \{X_1, X_2, \cdots, X_N, Y\}. \tag{2.8}$$

As shown in fig. 4.4, the fuzzy sets are made of sharp symmetric triangles on the conclusion—to have an equal weighting—and non-symmetric triangles on premises—to allow overlapping, and hence, a reasoning process—. The phenotype p_v^i expresses

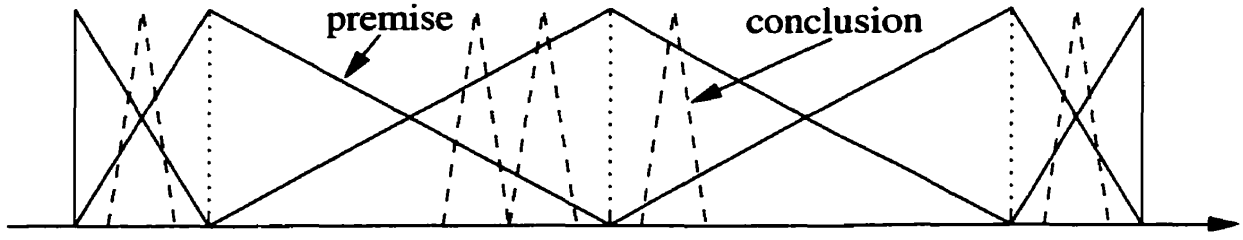

Figure 2.3 Fuzzy sets on a premise and a conclusion

the location of the summit of a triangle on the premise or the conclusion field v. For each premise, there are always two half-triangles located at p_v^{min} and p_v^{max}, and hence it is not necessary to encode their positions in G_v (also not counted in n_v).

The number of the bit, denoted b_v, allocated to each basic genotype g_v is chosen in such a way as to obtain a desired resolution r_v on the positioning of the fuzzy sets along p_v between p_v^{min} and p_v^{max}. The encoding of the basic phenotype p_v^i into its corresponding genotype g_v^i is given as

$$g_v^i = f(p_v^i), \quad \forall i = 1, ..., n_v, \quad \textit{and} \quad \forall v \in \{X_1, X_2, \cdots, X_N, Y\}, \tag{2.9}$$

where the *encoding scheme* $f(\cdot)$ is defined as

$$f(p_v^i) \equiv \frac{p_v^i - p_v^{min}}{r_v}, \quad g_v^i \in \{0, 1, \cdots, 2^{b_v} - 1\}, \quad p_v^{min} \leq p_v^i \leq p_v^{max}, \tag{2.10}$$

with the resolution r_v on the phenotype p_v computed as

$$r_v = \frac{p_v^{max} - p_v^{min}}{2^{b_v} - 1}. \tag{2.11}$$

The decoding of the basic genotype g_v^i into its corresponding phenotype p_v^i is given as

$$p_v^i = f^{-1}(g_v^i), \quad \forall i = 1, ..., n_v, \quad \textit{and} \quad \forall v \in \{X_1, X_2, \cdots, X_N, Y\}, \tag{2.12}$$

where the *decoding scheme* $f^{-1}(\cdot)$ is defined as

$$f^{-1}(g_v^i) \equiv r_v g_v^i + p_v^{min}. \tag{2.13}$$

While the number of premises N and the limits p_v^{min} and p_v^{max} are parameters determined by the problem to be solved using the FDSS Fuzzy-Flou, the maximum number of fuzzy sets n_v and the number of bits of resolution on the positioning of these fuzzy sets are important parameters for the learning process because it determines the size of the fuzzy-sets-search space. Since the number bits of g_v^i is

b_v, the function $size(\cdot)$ is thus defined as $size(g_n^i) \equiv b_v$. Consequently, the number of bits of G_v and G_{sets} are given as

$$size(G_v) = n_v b_v, \tag{2.14}$$

$$size(G_{sets}) = size(G_Y) + \sum_{i=1}^{N} size(G_{X_i}) = n_Y b_Y + \sum_{i=1}^{N} n_{X_i} b_{X_i}. \tag{2.15}$$

Fuzzy rules:

The genotype of fuzzy rules must contain information about all the possible combinations of connecting a fuzzy set of the conclusion to a fuzzy set of each premises. The maximum number of rules, K, is given as the total number of combinations, i.e.,

$$K = (n_{X_1} + 2)(n_{X_2} + 2) \cdots (n_{X_N} + 2), \tag{2.16}$$

where the $+2$ is required because of the presence of the two half triangles, located at $p_{X_i}^{min}$ and $p_{X_i}^{max}$, that are not counted in n_{X_i}. Without lost of generality, we can assign fuzzy sets $p_{X_i}^{min}$ to indice 0 and $p_{X_i}^{max}$to indice $n_{X_i} + 1$ such that we have:

$$p_{X_i}^0 \equiv p_{X_i}^{min}, \quad p_{X_i}^{n_{X_i}+1} \equiv p_{X_i}^{max}, \quad i = 1, ..., n_{X_i}. \tag{2.17}$$

The fuzzy rules are coded in an ordered list of combination of the premises, each having an enable/disable bit, denoted e, together with a conclusion fuzzy set num-

ber, i.e.,:

$$G_{rules} \equiv \{\underbrace{e10...01}_{g_0}\underbrace{e11...10}_{g_1}\cdots\underbrace{e01...11}_{g_{K-1}}\}. \tag{2.18}$$

For a specific combination of fuzzy set premises $\{f_1, f_2, \cdots, f_N\}$ with $f_i \in \{0, 1, ..., n_{X_i}+1\}$, the indice k of g_k can be computed as r_1 by the recursive equation

$$r_j = f_j + (n_{X_j} + 1)r_{j+1}, \quad j = 1, ..., (N-1), \quad r_N = f_N. \tag{2.19}$$

The number of bits b_r allocated to each g_k must have sufficient space to refer to n_Y conclusion fuzzy sets, plus one enable/disable bit, i.e.,

$$2^{(b_r-1)} \geq n_Y. \tag{2.20}$$

Although it is not the minimum size, the fuzzy rules coding of eq.(4.19) is chosen because of its simplicity and constant genotype size, i.e.,

$$size(G_{rules}) = Kb_r. \tag{2.21}$$

2.3.2 Reproduction

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 by any of the three following operators based on a different initiating

probability.

Simple Crossover:

In general, the reproduction is mainly done by simple crossover (with a probability t_1) of the *genotype* of two parents to produce the genotype of two children. The

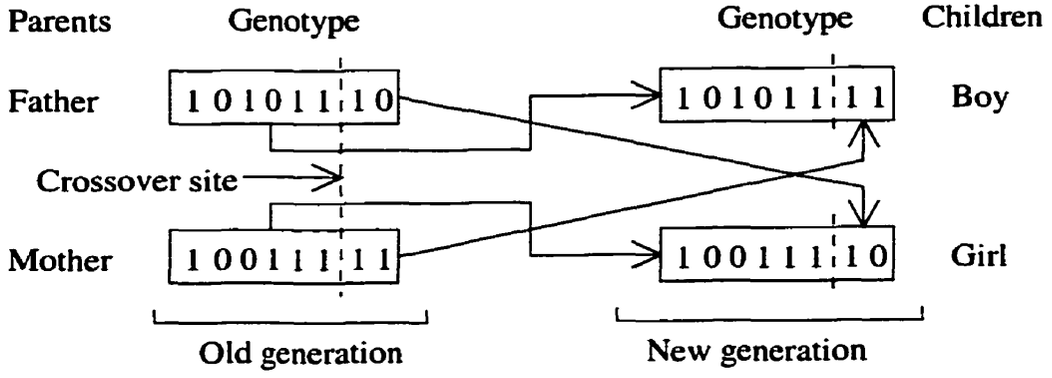

Figure 2.4 Simple crossover of the genotypes of two parents

simplest way to implement this operation is as follows: the parents are selected based on their ability; the genotype of the parents is split in two parts at a randomly selected crossover site; the genotype of the children is formed by recombining one part of the genotype of each of their parents, as shown in fig. 4.5.

Fuzzy-Sets Displacement:

The displacement of the fuzzy sets is performed (with a probability t_2) by randomly selecting a fuzzy set on a premise. The selected fuzzy set is then moved by one step of resolution toward the left or right, with an equal probability. This reproduction operator has the virtue of trying different fuzzy set repartitions, while decreasing

the number of fuzzy sets by superimposing two or more fuzzy sets.

Fuzzy-Rules Reduction:

The reduction of the number of fuzzy rules is performed, with a probability t_3 given by

$$t_3 = (1 - t_1)(1 - t_2). \tag{2.22}$$

One of the K fuzzy rules is randomly selected, and the bit e is set to disable. Obviously, this reproduction operator does not always give rise to a reduction in the number of fuzzy rules, but gradually it works in that direction. At the same time, when a fuzzy set is disabled the fuzzy set number is assigned to zero.

2.3.3 Mutation

Mutation is a random inversion of a bit in the genotype of a new member of the population. Mutation makes it possible to try a completely different solution. The probability of mutation t_4 should be kept very small in order to let the population improve itself mainly by reproduction. This way of seeking completely different solutions allows the algorithm to jump out of a local optima, and potentially fall into more promising regions.

2.3.4 Evaluation

The capacity of each individual to survive natural selection is evaluated through two objective functions. The first objective function evaluates the capacity of a knowledge base to approximate the set of sampled data. This fitness value, denoted ϕ_1, is defined as

$$\phi_1 \equiv \frac{\delta - \epsilon_{RMS}}{\delta}, \quad \delta \equiv p_Y^{max} - p_Y^{min}. \tag{2.23}$$

The root-mean-square error, ϵ_{RMS}, between the FDSS Fuzzy-Flou decision Y_i and the conclusion values y_i of the sampled data is computed as

$$\epsilon_{RMS} = \sqrt{\frac{\sum_{i=1}^{n}(Y_i - y_i)^2}{n}}, \tag{2.24}$$

where n is the number of points in the sampled data. The second objective function evaluates the complexity of a knowledge base through its number of active fuzzy rules. This fitness value, denoted ϕ_2, is defined as

$$\phi_2 \equiv \frac{K - n_a}{K} \tag{2.25}$$

where K is the maximum number of fuzzy rules and n_a is the number of active rules of the knowledge base under evaluation. In order to chose between these two contradictory objectives, we use the following weighted sum of the two objective

functions, i.e.:

$$\phi = w\phi_1 + (1 - w)\phi_2, \tag{2.26}$$

where w is usually set to around 75%.

2.3.5 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. In this paper, the first generation starts with 100 knowledge base and 100 additional are generated by reproduction and mutation. These brand new knowledge bases are then evaluated. The natural selection is applied on the 200 knowledge bases by ranking them based on ϕ and ϕ_1. We keep the first 50 non identical knowledge bases of the two lists.

2.4 Numerical Validation

The learning performances of this GA are now investigated using several examples of known behaviors for which it is easy to manually produce a knowledge base. All these examples have $N = 2$ input premises, namely X_1 and X_2, and one output conclusion Y. The known behaviors are chosen as 3D surfaces of type: $y = f(x_1, x_2)$, where the nodes are the learning set of sampled data. Moreover, a

maximum of 7 fuzzy sets is used on each premise ($n_{X_1} = n_{X_2} = (7-2) = 5$) and a maximum of 8 fuzzy sets on the conclusion ($n_Y = 8$). Therefore, the maximum number of fuzzy rules is given by eq.(4.17) as $K = 7 \times 7 = 49$. Furthermore, each input premise is discretized into 16 different fuzzy set positions, and hence, requires $b_{X_1} = b_{X_2} = 4$ bits. Using eq.(4.20) with $n_Y = 8$, $b_r = 4$ bits are required for each item of the ordered list of fuzzy rules. Finally, the size of the genotype of the learning problem of this FDSS is

$$\begin{aligned} size(G) &= size(G_{sets}) + size(G_{rules}) \\ &= Kb_r + n_Y b_Y + n_{X_1} b_{X_1} + n_{X_2} b_{X_2} = 268, \end{aligned} \tag{2.27}$$

which means that the size of the learning space $\mathcal{L}$ is given by

$$size(\mathcal{L}) = 2^{size(G)} = 2^{268} = 4.7 \times 10^{80}. \tag{2.28}$$

Under the assumption that the evaluation of one knowledge base by a Pentium II-350 MHz requires about 1 msec, the total computing time to evaluate all the knowledge bases of $\mathcal{L}$ will require 1.5×10^{70} years, which is totally unacceptable! It is thus clear that we need a learning process that requires the evaluation of a very small percentage of $\mathcal{L}$, while proposing near optimal knowledge bases. This is what we will obtain using GAs.

2.4.1 Example 4.1: Horizontal planes

The two theoretical surfaces 4.1 are horizontal planes at two different heights, i.e.,

$$\begin{array}{l} a)\ y = 7.75 \\ \text{with} \\ b)\ y = 7.33 \end{array} \quad \begin{array}{l} 0 \le x_1 \le 10 \\ \\ 2 \le x_2 \le 12 \end{array}, \quad 1 \le y \le 10, \tag{2.29}$$

where $Y = 7.75$ is right on one of the discrete locations of Y, while $Y = 7.33$ is off of these. As shown in the fig. 2.5, 2.6, 2.7 and 2.8, the learning process proposed a

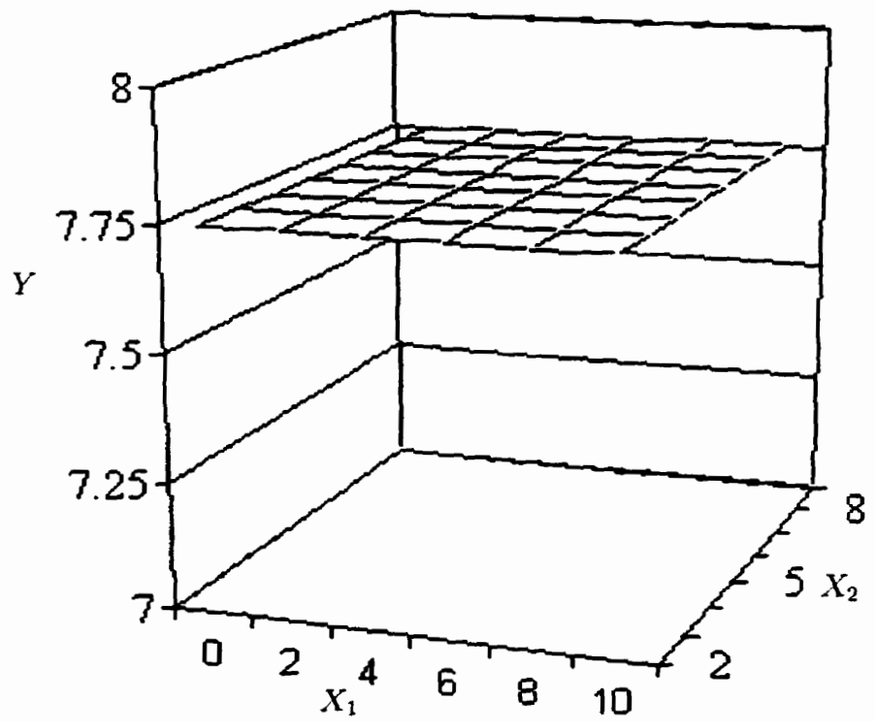

Figure 2.5 Theoretical surface 4.1a (horizontal plan)

knowledge base that allows an exact approximation of eq.(2.29) with a minimum of 2 of fuzzy sets on each premise. As expected on the conclusion, one fuzzy set at

$Y = 7.75$ is proposed in the first case and 2 fuzzy sets at $Y = 7.1875$ and $Y = 7.75$ in the second case.

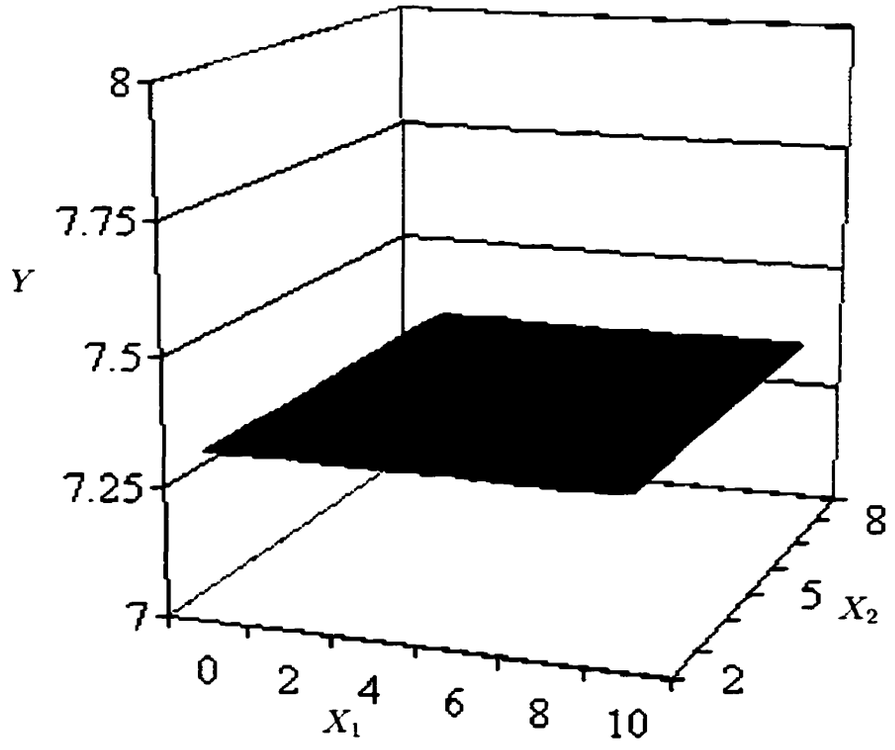

Figure 2.6 Theoretical surface 4.1b (horizontal plan)

Moreover, the learning process has automatically reduced the number of fuzzy rules from a maximum of 49 to the a minimum of 4. In the first case, all 4 rules point to the same conclusion fuzzy set $Y = 7.75$, while in the second case, 3 rules point to $Y = 7.1875$ and only one to $Y = 7.75$, since the output value of $Y = 7.33$ is closer to $Y = 7.1875$ than $Y = 7.75$.

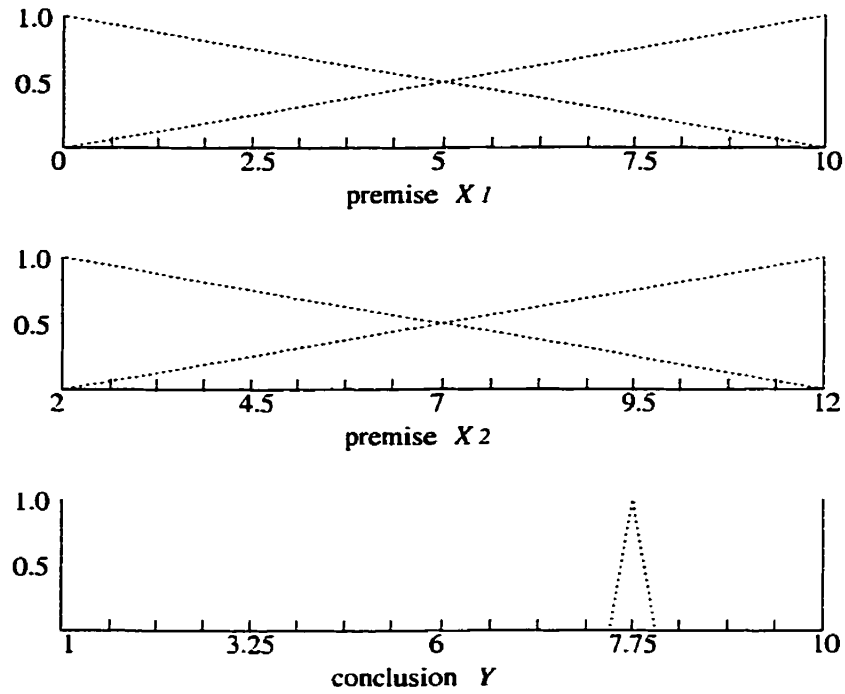

Figure 2.7 Computed fuzzy sets of theoretical example 4.1a (horizontal plan)

2.4.2 Example 4.2: Three planes

The theoretical surface 4.2 is made of three plans, i.e.,

$$y = \begin{cases} x_2 & : \ 0 \le x_2 < 10 \\ 10 & : \ 10 \le x_2 < 30 \\ x_2 - 20 & : \ 30 \le x_2 < 40 \end{cases} \quad \text{with} \quad \begin{array}{l} 10 \le x_1 \le 20 \\ 0 \le x_2 \le 40 \\ 0 \le y \le 20 \end{array} . \tag{2.30}$$

As shown in fig. 2.9 and 2.10, the learning process proposed a knowledge base that allows an approximation of eq.(2.30) with a 2% error using a minimum of 2 and 4 fuzzy sets on the premises X_1 and X_2, respectively, and 3 fuzzy sets on the conclusion. As shown in fig. 2.11 and 2.12, the repartition of the computed fuzzy

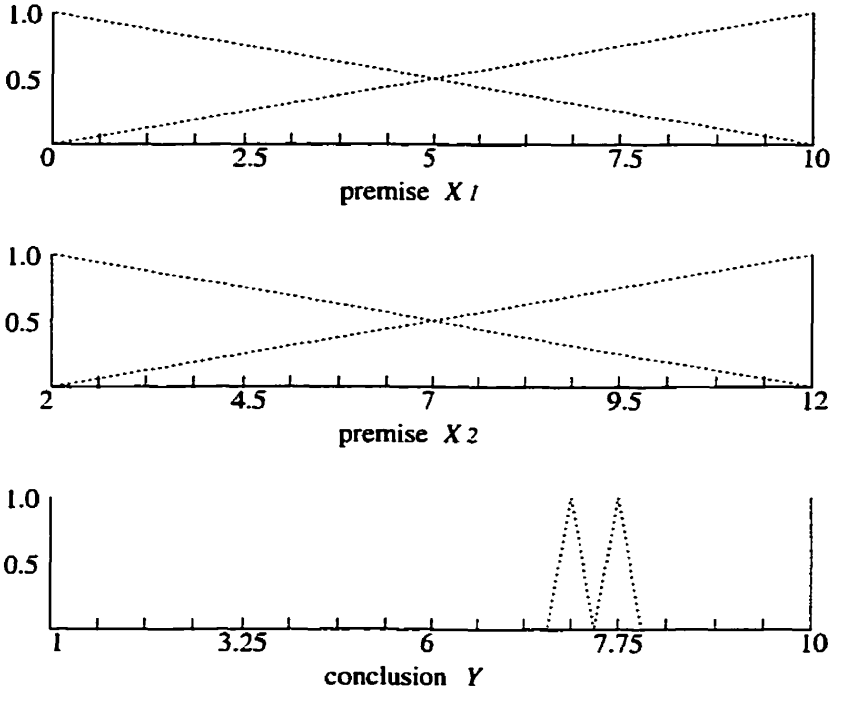

Figure 2.8 Computed fuzzy sets of approximated example 4.1b (horizontal plan)

sets is almost identical to the theoretical one manually proposed by an expert. The only difference comes from the discretization error on the locations of the fuzzy sets on the conclusion, which does not allow, in this case, an exact approximation of the theoretical surface 4.2. As expected, the number of computed fuzzy rules is automatically reduced from 49 to 5, thus corroborating the 5 fuzzy rules manually proposed by the expert.

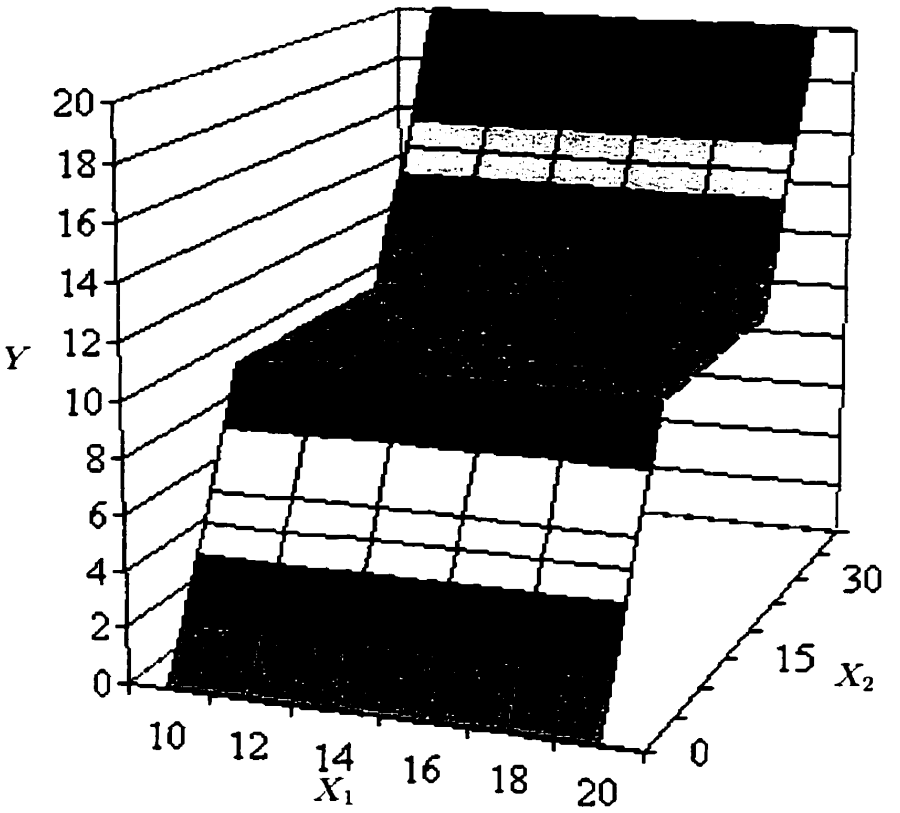

Figure 2.9 Theoretical surface 4.2 (three planes)

2.4.3 Example 4.3: Curved surface

The theoretical surface 4.3 is defined as

$$y = \frac{x_2 + 0.5}{x_1} \text{ with } \begin{matrix} 0.25 \leq x_1 \leq 5.0 \\ 0.5 \leq x_2 \leq 4.5 \end{matrix}, \quad 0 \leq y \leq 20. \tag{2.31}$$

As shown in fig. 2.13 and 2.14, the learning process proposed a knowledge base that allow an approximation of eq.(2.31) with a 2% error using 6 and 2 fuzzy sets on the premises X_1 and X_2, respectively, and 5 fuzzy sets on the conclusion, as

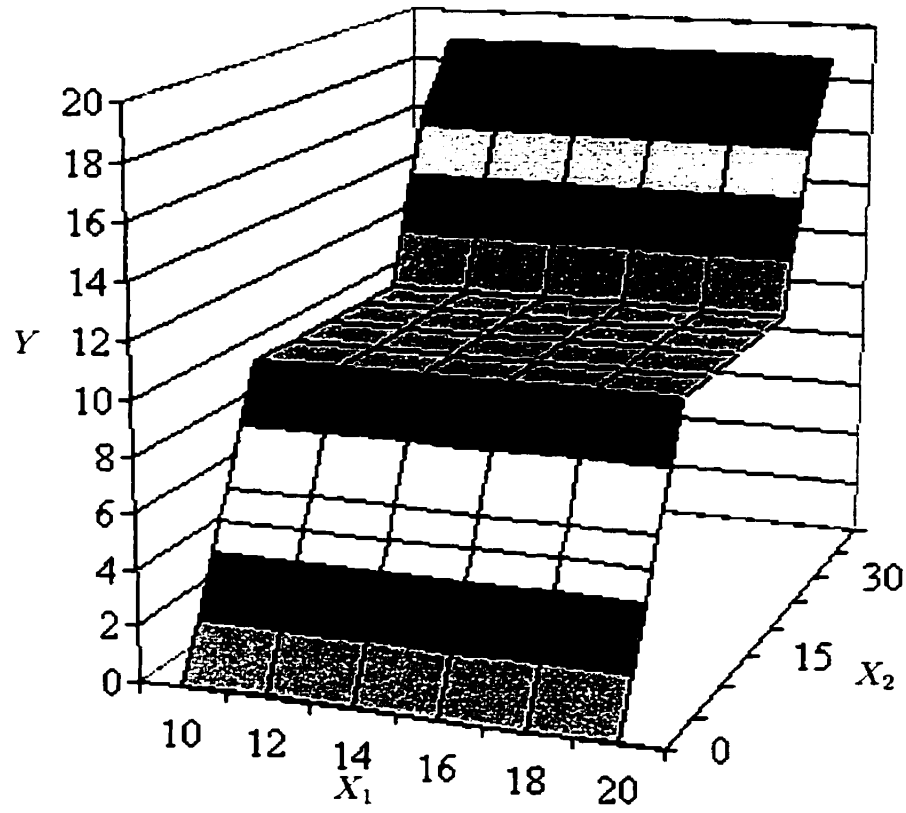

Figure 2.10 Approximated surface 4.2 (three planes)

shown in fig. 2.15. The number of computed fuzzy rules is automatically reduced from 49 to 10.

2.4.4 Example 4.4: Concave surface

The theoretical surface 4.4 is defined as

$$y = exp(x_1^2 + x_2^2) \text{ with } \begin{matrix} -1.5 \leq x_1 \leq 1.5 \\ -2 \leq x_2 \leq 2 \end{matrix}, \quad 1 \leq y \leq 600. \tag{2.32}$$

As shown in fig. 2.16 and 2.17, the learning process proposed a knowledge base

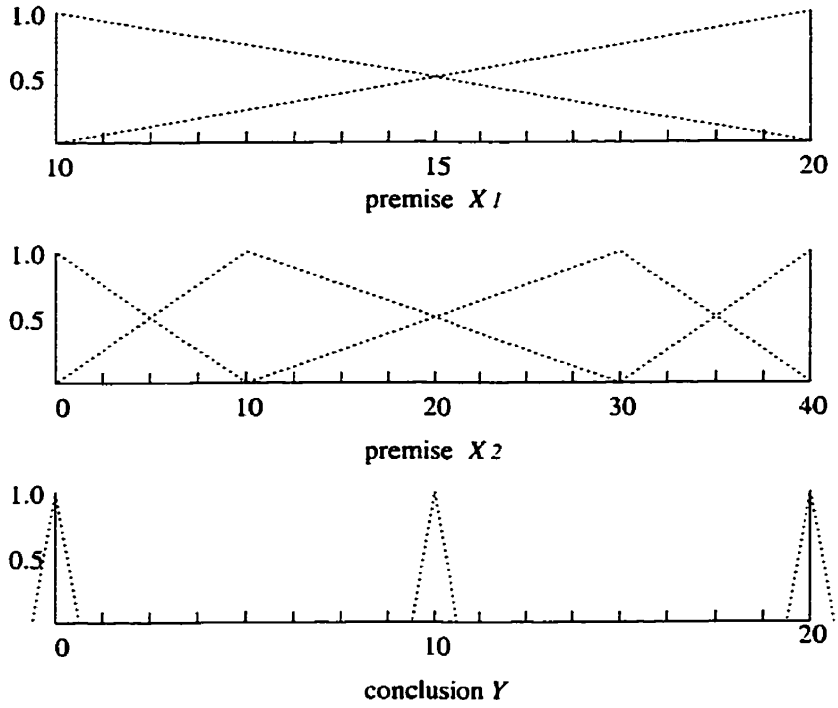

Figure 2.11 Theoretical fuzzy sets of example 4.2 (three planes)

(fig. 2.18) that allows an approximation of eq.(2.31) with a 3% error using 7 and 4 fuzzy sets on the premises X_1 and X_2, respectively, and 6 fuzzy sets on the conclusion. The number of computed fuzzy rules is automatically reduced from 49 to 22.

Apparently the fuzzy sets are distributed, on the conclusion field, proportionally to the vertical density of nodes on the theoretical surface. Although we have a fully symmetric theoretical surface with respect to x_1 and x_2, the distribution of the fuzzy sets on these two premises are slightly different. This difference could be explained by the fact that the learning process does not provide an optimal knowledge base, but rather only near optimal knowledge bases.

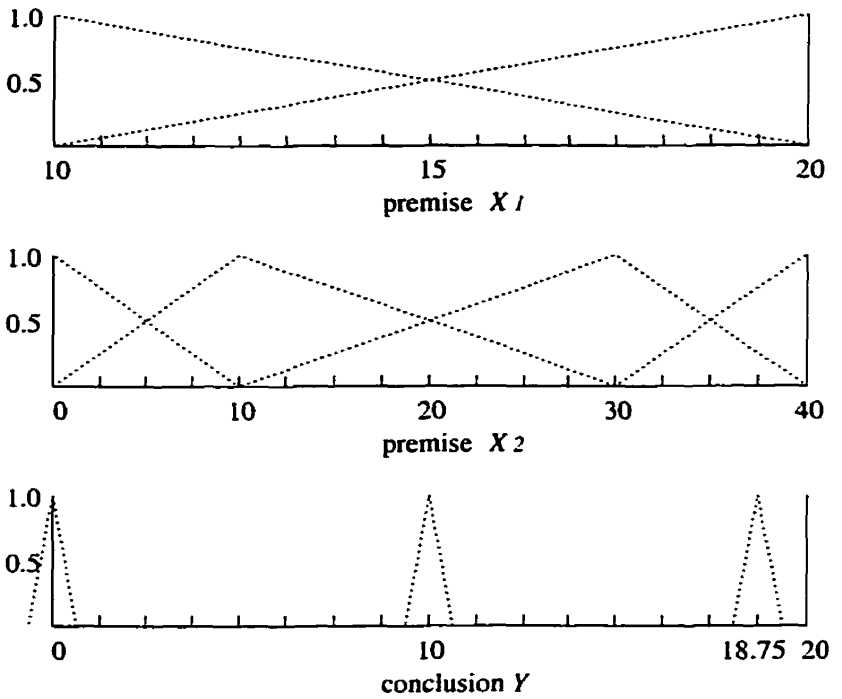

Figure 2.12 Computed fuzzy sets of example 4.2 (three planes)

2.5 Experimental Data

In this section, we use the learning process on a rather incomplete set of experimental data used to predict the cutting force F based on a measure of the feed rate f and the cutting depth d during bar-turning operations. The measurements were taken on a 5kW lathe equipped with a bi-directional dynanometer, a high speed steel cutting tool, a cutting speed of 32 m/min. The machined part was XC48 with Rm = 80 daN/mm^2 [18]. The bar-turning cutting forces were measured under two experimental conditions: a constant feed rate with a variable cutting depth (Table

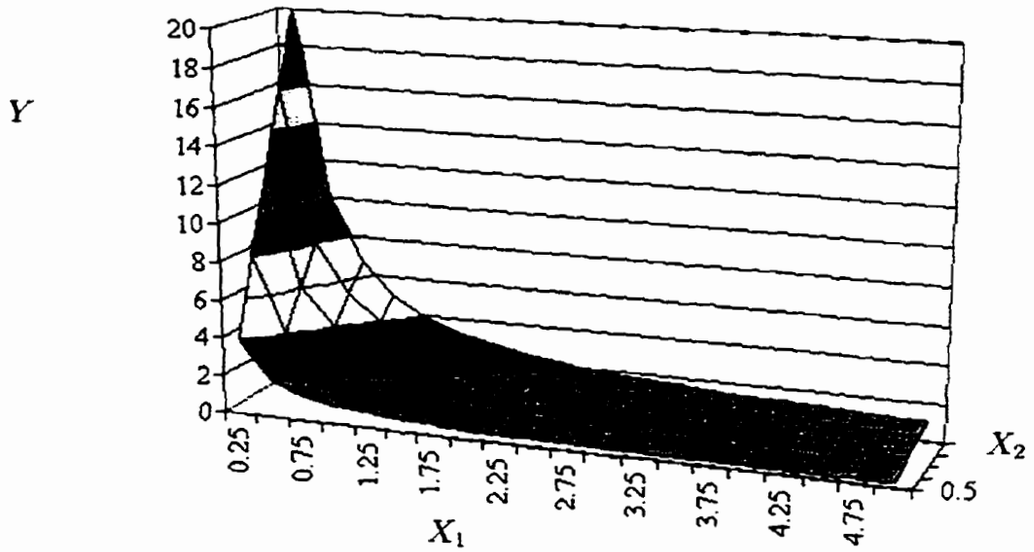

Figure 2.13 Theoretical surface 4.3 (curved surface)

4.1); and a constant cutting depth with a variable feed rate (Table 2.3).

For the sake of comparison, the standard Taylor equation is used here to predict

Tableau 2.1 Cutting force vs feed rate for a 5 mm depth of cut

Test #	1	2	3	4	5	6
f[mm/rev]	0.05	0.1	0.2	0.3	0.4	0.5
F[N]	1000	1600	2800	3600	4300	4950

Tableau 2.2 Cutting force vs feed rate for a 5 mm depth of cut

the cutting forces as commonly done in the metal cutting industry, i.e.,

$$F = Cf^{\alpha}d^{\beta} \text{ with } C = 1589,\ \alpha = 0.7,\ \beta = 0.994, \tag{2.33}$$

where C, α and β are computed as nonlinear least-square estimates of eq.(2.33)

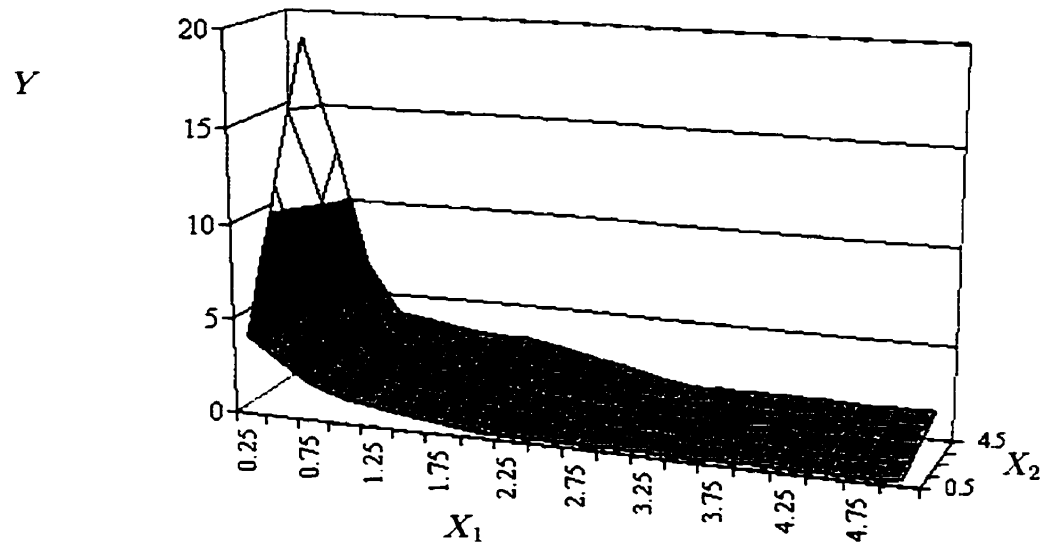

Figure 2.14 Approximated surface 4.3 (curved surface)

Tableau 2.3 Cutting force vs depth for a 0.4 mm/rev feed rate

Test #	7	8	9	10	11	12
d[mm]	1	2	3	4	5	6
F[N]	800	1900	2400	3400	3900	4750

from the experimental data of Tables 1 and 2. Figure 2.19 shows the Taylor surface and fig. 2.20 shows the corresponding FDSS Fuzzy-Flou approximated surface of the experimental data.

Using the knowledge base produced by our learning process, the Taylor surface is approximated by the FDSS Fuzzy-Flou with an error of 3%. This result is acceptable considering the 4 bit resolution on the position of the fuzzy sets on the conclusion field. However, there are still inconsistencies in the experimental data (see test 5 and 11), which explains the impossibility of having zero approximation error.

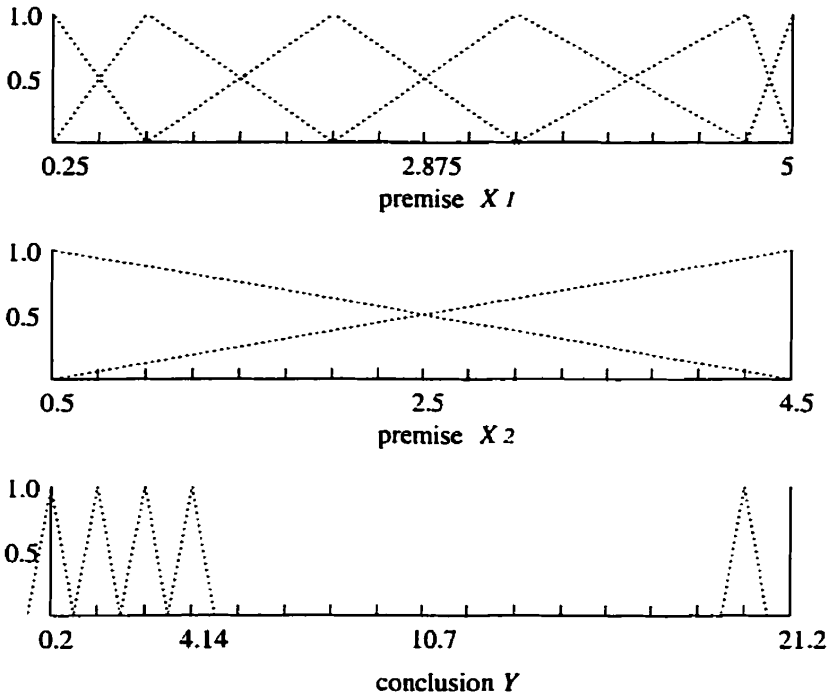

Figure 2.15 Computed fuzzy sets of example 4.3 (curved surface)

Figure 2.21 and 2.22 compare the experimental data with the results obtained by the Taylor equation and the GA-FDSS with the knowledge base automatically computed by our GA.

Apparently the results obtained with GA-FDSS are closer to the experimental data than those obtained with the theoretical Taylor equation.

As shown in fig. 2.23, only two fuzzy sets are generated on each premise and three on the conclusion field. Here again, the number of fuzzy rules is automatically reduced from 49 to 4.

Even with a very small amount of data our learning process allows the automatic construction of a knowledge base which enables FDSS Fuzzy-Flou to approximate

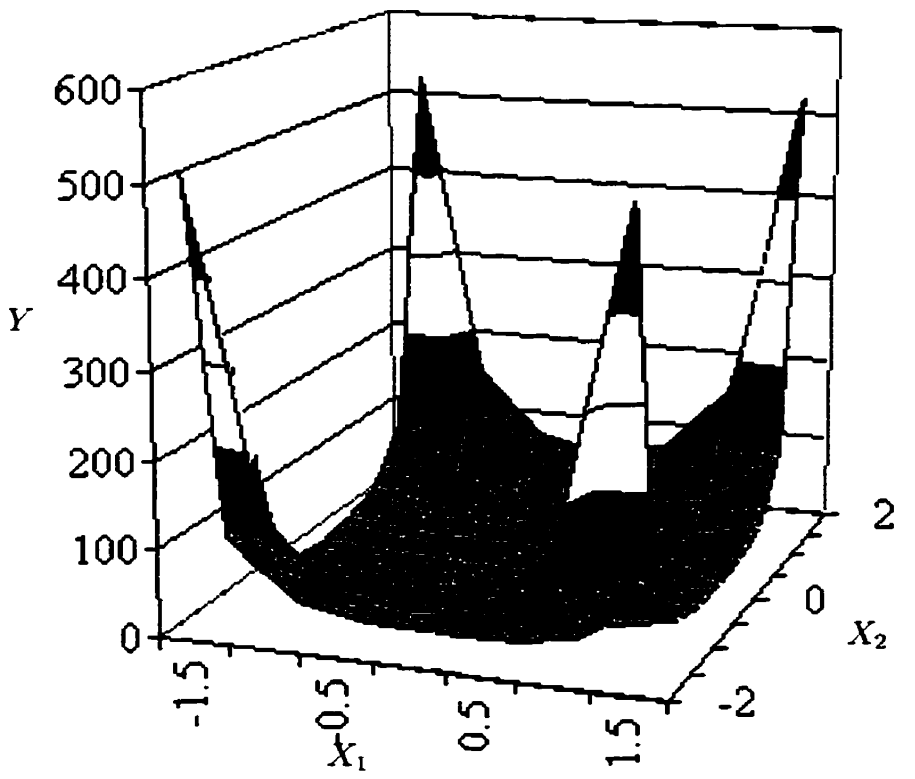

Figure 2.16 Theoretical surface 4.4 (concave surface)

the experimental data with a very low error level of together with the smallest possible number of fuzzy sets and rules. This provides a method to cover the full range of possible inputs without having to perform a large number of expensive experiments. Moreover, the GA automatically discards superfluous fuzzy sets and rules, and hence, provides a simple knowledge base that can be manually handled by an expert for fine tuning.

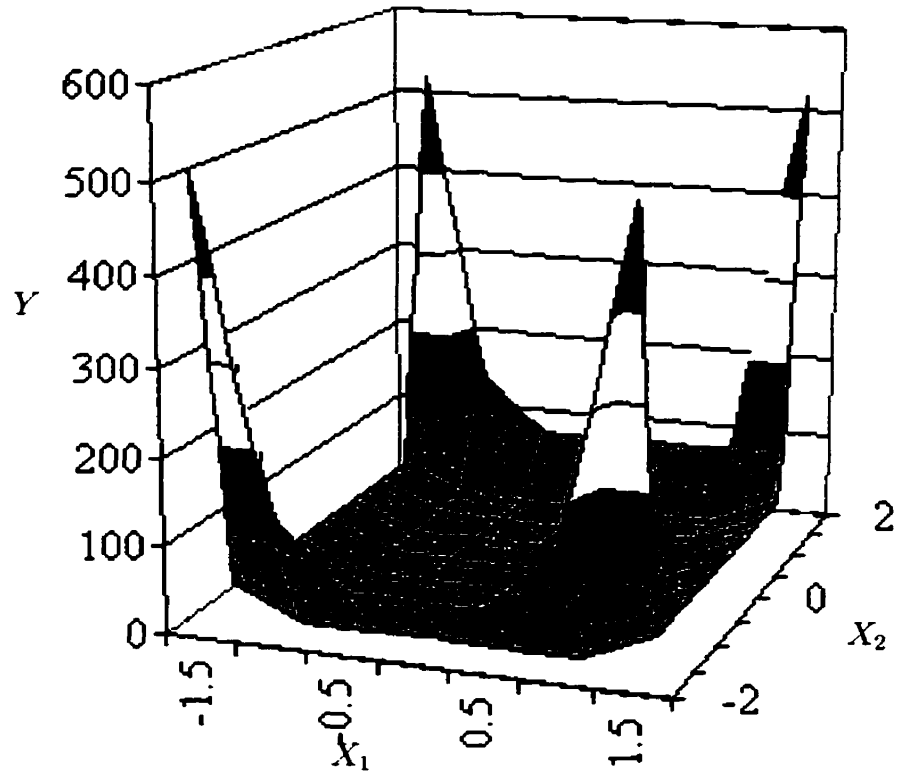

Figure 2.17 Approximated surface 4.4 (concave surface)

2.6 Conclusion

The need of an expert to construct a knowledge base for each given problem is the most important drawback of decision-aided systems. However, we have presented, in this paper, a genetic algorithm that can automatically construct a knowledge base. Besides the reduction of subjectivity related to the manually construction of a knowledge base, this GA has the virtue simultaneously reducing the approximation error, while eliminating superfluous fuzzy sets and rules. In all examples, our GA automatically computed a very good knowledge base in only a few minutes for a search space of size 10^{80}, on a Pentium II-350 MHz.

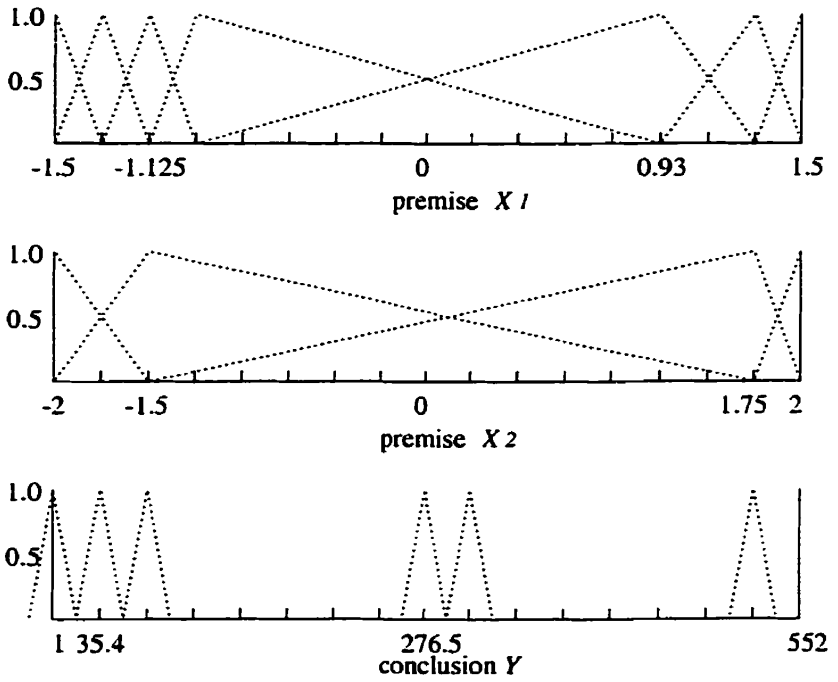

Figure 2.18 Computed fuzzy sets of example 4.4 (concave surface)

Acknowledgement

The financial support from the Natural Sciences and Engineering Research Council of Canada under grants of RGPIN-203618 and RGPIN-105518 is gratefully acknowledged.

References

[1] Balazinski, M. and Klim, Z., "Etude sur l'application de la logique floue pour la prédiction de la maintenance préventive", *International Industrial Engineering Conference*, Vol. II, pp. 1133–1142, October 1995.

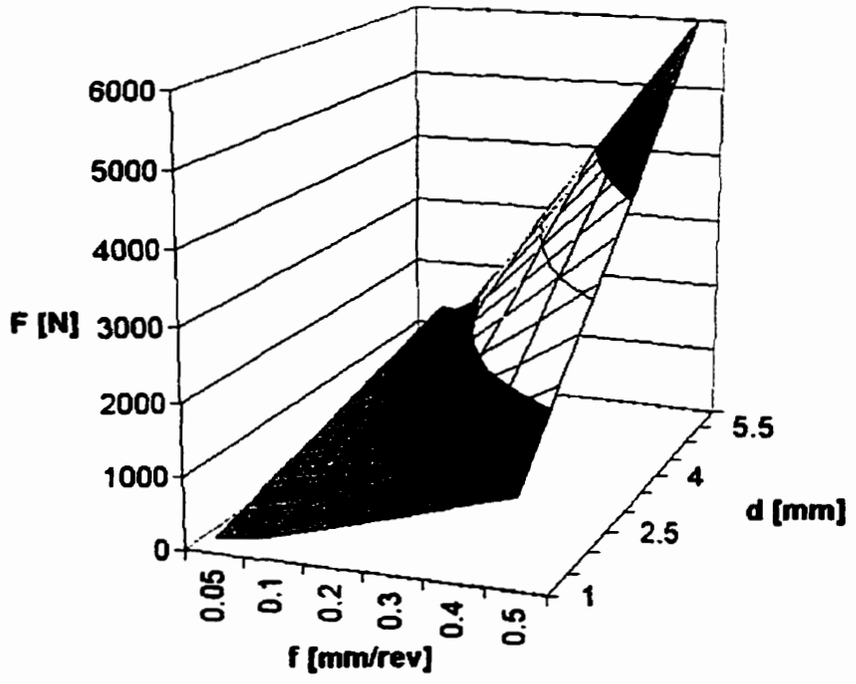

Figure 2.19 Taylor surface of predicted cutting forces

[2] 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.

[3] 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.

[4] Dupinet, E., Balazinski, M. and Czogala, E., "Tolerance Allocation based on fuzzy logic and simulated annealing", *Journal of Intelligent Manufacturing*, Vol. 7,

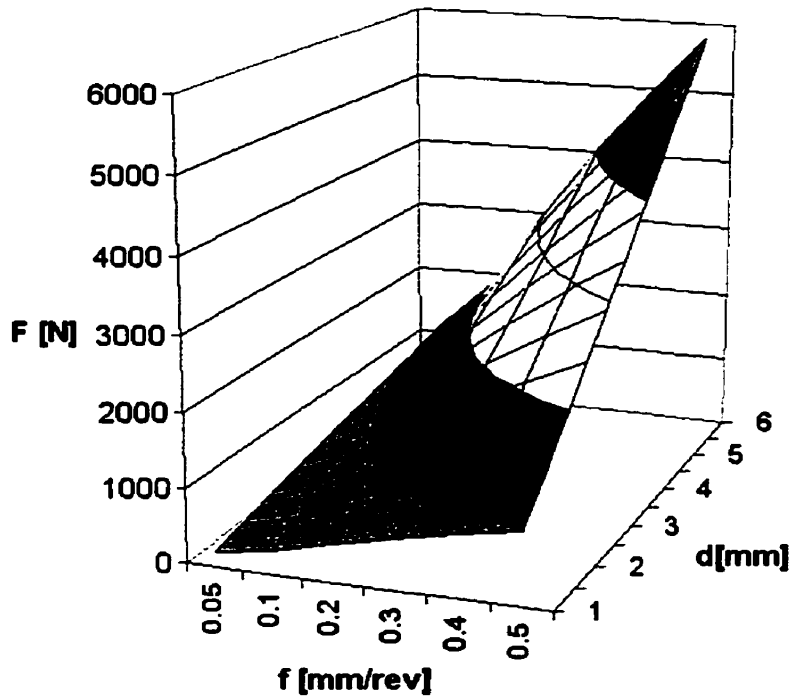

Figure 2.20 FDSS Fuzzy-Flou approximated surface of predicted cutting forces

pp. 487–497, 1996.

[5] Sugeno, M. and Yasukawa, T., "A fuzzy-logic-based approach to qualitative modeling", *IEEE Transactions on Fuzzy Systems*, Vol. 7, pp. 7–31, 1993.

[6] 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*, pap. 680–684, 1999.

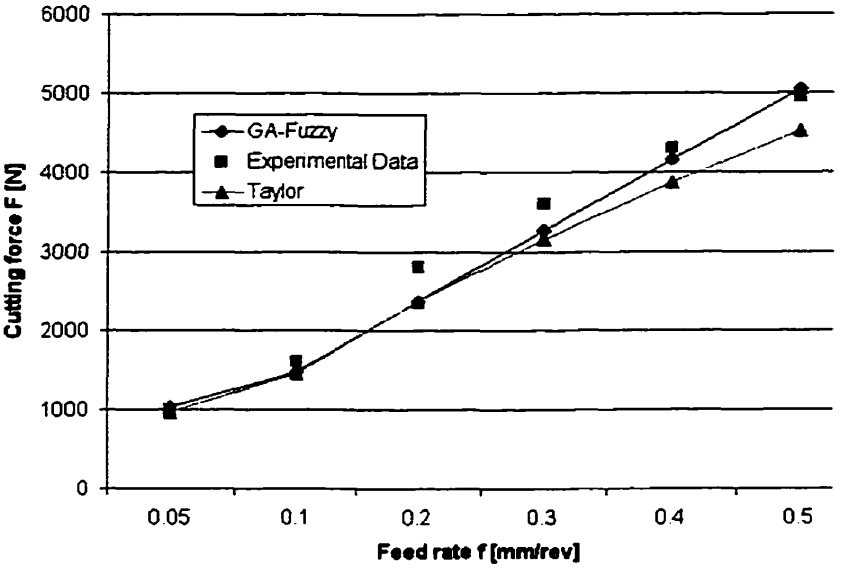

Figure 2.21 Taylor vs GA-FDSS predicted cutting force for 5 mm depth of cut

[7] 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.

[8] Yuan, Y. and Zhuang, H., " Using a genetic algorithm to generate fuzzy classification rules." *In Proceedings. Third European Conference on Intelligent Techniques and Soft Computing (EUFIT'95)*, pp. 458–462, 1995.

[9] 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.

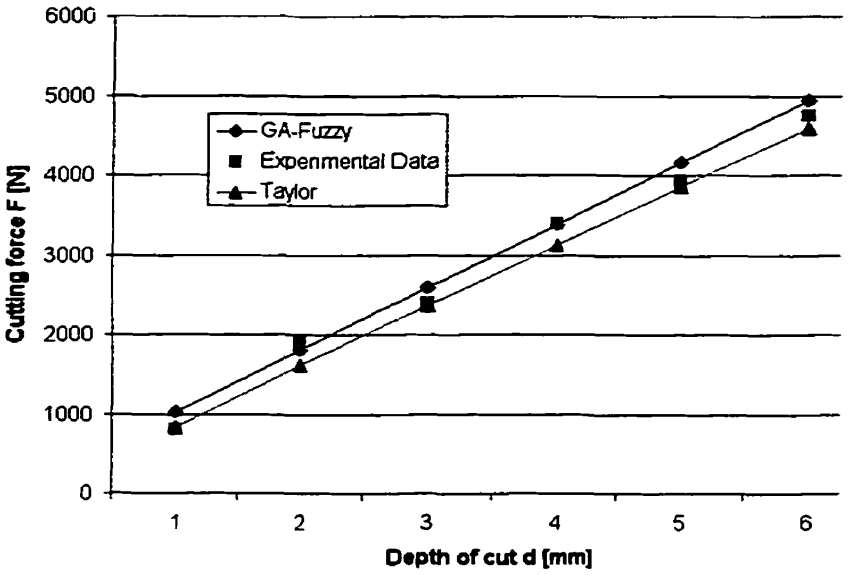

Figure 2.22 Taylor vs GA-FDSS predicted cutting force for a 0.4 mm/rev feed rate

[10] Janikow, C.Z., "A genetic algorithm for optimizing fuzzy decision trees", *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 421–428, 1995.

[11] 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.

[12] 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.

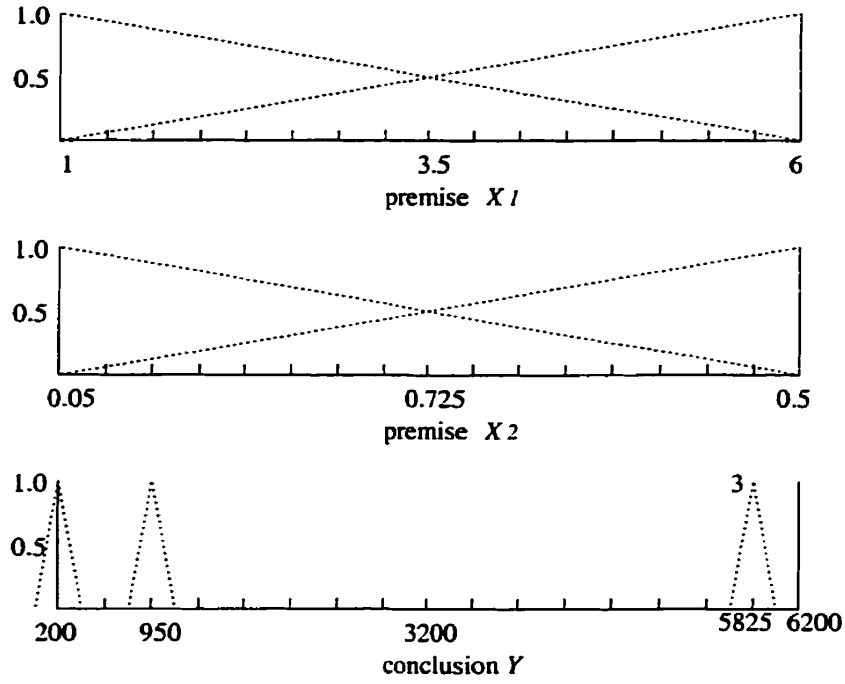

Figure 2.23 Computed fuzzy sets of the experimental data of cutting forces estimation

[13] Carse, B., Fogarty, T.C. and Munro, A., "Evolving fuzzy rule based controllers using genetic algorithm, *Fuzzy Sets and Systems*, Vol. 80, pp. 273–293, 1996.

[14] 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.

[15] 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.

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

[17] 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, Mai 2000.

[18] Vergnas, J., "Usinage. Technologie et pratique", *Edition Dunod*, 1992.

CHAPITRE 3

INFLUENCE DES PARAMÈTRES D'OPTIMISATION ET DE SÉLECTION.

Le travail de recherche présenté dans ce chapitre a été publié dans le compte rendu de la conférence "*International Conference on Advanced manufacturing Technology*", Jahor Bahru, Malaisie, Août 2000. [16].

3.1 Définition du problème

Dans ce chapitre, il est question de l'influence des paramètres de sélection et d'optimisation sur la génération automatique d'une base de connaissance par algorithme génétique (AG).

Le but est d'étudier les poids entre les différents critères et de ce fait, montrer les différentes évolutions résultantes.

L'AG est appliqué sur des données numériques et génère une base de connaissances qui les reproduit avec une erreur d'approximation minimale tout en utilisant le plus petit nombre possible de règles floues.

3.2 Système d'aide à la décision

Le système d'aide à la décision utilisé, est le SAD (FDSS) Fuzzy-Flou, dont les détails sont cités au chapitre 1. Il est à noter que les paramètres utilisés sont comme suit :

- $\sum *$ (somme poduit) comme moteur d'inférence;
- COG (centre de gravité) comme méthode de defuzzification.

3.3 Paramètres de l'AG

Comme cité au chapitre 1, l'AG fait évoluer la population de bases de connaissances, en utilisant la reproduction, la mutation, l'évaluation et la sélection naturelle. Ceci dit, ces derniers mécanismes sont régis par des pondérations qui sont citées et expliquées ci-dessous :

Reproduction

La reproduction est divisé en trois mécanismes principaux:

- Croisement simple

  Il s'agit là, du principal mécanisme de reproduction, il est régit par la probabilité p_1.

- Déplacement des sous-ensembles flous

  Cette partie du mécanisme est gouverné par la probabilité p_2.

- Réduction de règle floue

  La réduction de règle floue est appliquée avec une probabilité p_3 qui est calculée par l'équation :

$$p_3 = 1 - (p_1 + p_2) \tag{3.1}$$

Ce paramètre est généralement maintenu à un niveau assez faible.

Mutation

La mutation est aussi appliquée avec un pourcentage de probabilité p_4 très faible, généralement maintenu autour de 5% pour tout nos essais.

Evaluation

La capacité de chaque individu de survivre à la sélection naturelle est, comme déjà cité au chapitre 1, faite selon deux fonctions objectif. La première fonction objectif, évalue la capacité d'une base de connaissances à approximer les données utilisées. Cet indice de performance, dénoté ϕ_1, est calculé avec la méthode des moindres carrés.

La deuxième fonction objectif, évalue la complexité d'une base de connaissances à travers son nombre de règles floues, les bases de connaissances les plus simples sont celles avec le plus petit nombre de règles floues, cet indice de performance est dénoté ϕ_2.

La combinaison de ces deux critères contradictoires est faite par le biais de la

pondération suivante :

$$\phi = \omega_o \, \phi_1 + (1 - \omega_o) \, \phi_2, \tag{3.2}$$

où le critère d'optimisation ω_o est le poids associé à ϕ_1.

Sélection naturelle

À la fin de l'évolution, le meilleur individu est selectionné dans la population finale utilisant un nouvel indice de performance dénoté Φ donné comme suit :

$$\Phi = \omega_s \, \phi_1 + (1 - \omega_s) \, \phi_2, \tag{3.3}$$

où le critère de selection ω_s représente le poids associé à ϕ_1.

3.4 Paramètres de sélection et d'optimisation

Une distinction peut être faite, entre les critères de sélection et d'optimisation. Les premiers sont ceux utilisés par le processus d'optimisation pour converger vers la population de bases de connaissances finale (p_1, p_2, p_3, p_4 *et* ω_o), alors que les derniers sont ceux utilisés pour choisir une base de connaissances parmi la population finale (ω_s).

Dans cette étude, l'AG est appliqué sur des données numériques représentant la surface théorique de la figure 2.13, donnée par l'équation 2.31 (voir chapitre 1).

Chaque noeud de la surface représentant une donnée.

Pour cette partie, la complexité maximale de la base de connaissances a été fixée comme suit :

- 7 sous-ensembles flous sur chacune des prémisses d'entrées (X et Y);
- 8 sous-ensembles flous sur la conclusion;
- 49 ($7 \times 7 = 49$) règles floues.

Le nombre de 7 sous-ensembles flous a été choisi car c'est le nombre le plus communément utilisé en contrôle.

3.4.1 Influence des paramètres d'optimisation

Dans cette section, deux parties différentes sont prise en compte. La première concerne les paramètres de reproduction (ex. p_1, p_2, p_3 et p_4) et la deuxième concerne le critère d'évaluation (ω_o).

3.4.1.1 Paramètres de reproduction

La valeur de ω_o est fixé à 0.8, la mutation est établie à $p_4 = 0.05$ et la sélection des meilleurs individus est faite avec un $\omega_s = 0.8$. L'optimisation est faite avec les valeurs suivantes : $p_1 = 0.7; p_2 = 0.21,\ 0.27$ et $p_1 = 0.9; p_2 = 0.07,\ 0.09$.

À la fin de l'évolution, la moyenne des charactéristiques des cinq meilleurs individus

est prise en compte. Les résultats obtenus sont présentés au tableau 3.1.

Sachant que "# règles" représente le nombre de règles floues encore actives dans

Tableau 3.1 Influence des paramètres de reproduction

ϕ_1	ϕ_2	# règles[1]	ϕ_1	ϕ_2	# règles[1]
$p_1 = 0.7$ *et* $p_2 = 0.27$			$p_1 = 0.9$ *et* $p_2 = 0.07$		
0.8599	0.8285	8.4	0.9077	0.7510	12.2
$p_1 = 0.7$ *et* $p_2 = 0.21$			$p_1 = 0.9$ *et* $p_2 = 0.09$		
0.8576	0.7837	10.6	0.9135	0.7673	11.4
1: Le nombre de règles floues est la moyenne des cinq meilleurs individus					

la base de connaissances. Il apparaît que la croissance de ϕ_1 est proportionnelle à l'augmentation de p_1, pour p_1=0.9 on atteint une valeur d'approximation de 91% des données numériques. Néanmoins, ceci est fait au détriment de la simplicité de la base de connaissances, car le nombre de règles floues augmente simultanément.

Augmenter la valeur de p_2 mène à la diminution du nombre de règles floues actives (8 et 10 règles floues pour p_2=0.27, 0.21). La contre-partie est une perte de précision quant à l'approximation des données numériques (environ 80%). Il est aussi bien apparant que p_2 joue un rôle plus important que p_3 quant à la réduction du nombre de règles floues. La cause peut être expliquée comme suit :

- p_3 est généralement appliqué avec un faible pourcentage, car l'augmentation abusive de ce paramètre mène à une divergence ou à une très lente convergence de l'AG, à cause de l'importante présence d'individus de qualité inférieure;

- p_2 gère le déplacement latéral des sous-ensembles flous, et de ce fait leur superposition, ce qui a pour effet de réduire le nombre de règles floues, et conséquemment, augmente la valeur de ϕ_2;

- le mécanisme géré par p_3—la désactivation aléatoire de règles floues–peut ne produire aucun effet si la règle était déjà inactive.

3.4.1.2 Paramètres d'évaluation

Dans cette partie, l'optimisation est faite en changeant la valeur de ω_o, tous les autres paramètres étant fixés. Les valeurs utilisées lors des différentes exécutions sont, $p_1 = 0.8, p_2 = 0.16$ et $p_4 = 0.05$. À la fin de l'évolution, les cinq meilleurs individus sont sélectionés en utilisant le paramètre de sélection ω_s égal à ω_o. Le tableau 3.2 montre la moyenne des caractéristiques des cinq meilleurs individus.

Il est assez évident que la croissance de la valeur de ω_o produit des individus avec

Tableau 3.2 Influence des paramètres d'évaluation

ϕ_1	ϕ_2	# règles
$\omega_o = 0.25$ *et* $\omega_s = 0.25$		
0.6813	0.9388	3
$\omega_o = 0.50$ *et* $\omega_s = 0.50$		
0.9060	0.8571	7
$\omega_o = 0.75$ *et* $\omega_s = 0.75$		
0.9265	0.8286	8.4
$\omega_o = 1.00$ *et* $\omega_s = 1.00$		
0.9496	0.7469	12.4

une meilleure approximation des données numériques, mais avec de plus en plus

de règles floues actives. La précision varie de 0.68 à 0.95, alors que le nombre de règles floues augmente de 3 à 12.

3.4.2 Influence du paramètres de sélection ω_s

Tous les paramètres d'optimisation sont fixées. La variation se fait seulement sur ω_s. Les valeurs utilisées sont $\omega_o = 0.8$, $p_1 = 0.8$, $p_2 = 0.16$ et $p_4 = 0.05$. Les exécutions sont faites avec trois différentes valeurs de $\omega_s = 0$, 0.5 et 1.

Comme dans les cas précédents, les valeurs moyennes des charactéristiques des cinq meilleurs individus sont présentées dans le tableau 3.3.

Tableau 3.3 Influence des paramètres de sélection

ϕ_1	ϕ_2	# règles
$\omega_s = 0.0$		
0.9174	0.8000	9.8
$\omega_s = 0.50$		
0.9225	0.7959	10
$\omega_s = 1.00$		
0.9478	0.7306	13.2

Le paramètre de sélection ω_s est utilisé pour sélectionner un certain individu dans la population finale. Comme on peut facilement le voir sur le tableau 3.3, $\omega_s = 0$ produit l'individu avec le plus petit nombre de règles floues actives ($\approx$ 9), mais aussi la plus mauvaise approximation ($\approx$ 92%) des données numériques. $\omega_s = 1.00$ produit l'individu avec la meilleure approximation ($\approx$ 95%) de données

numériques, mais avec le plus grand nombre de règles floues ($\approx$ 13).

3.5 Conclusion

La base de connaissances utilisée par un système d'aide à la décision peut être construite automatiquement par algorithme génétique à partir de données numériques. Le but est donc de trouver une base de connaissances qui peut satisfaire au mieux deux objectifs contradictoires, à savoir: minimiser l'erreur d'approximation tout en essayant de garder le nombre de règles floues actives le plus bas possible. De là, ressort toute l'importance de la pondération sur les indices de performance. Un poids de 100% met toute l'emphase sur la précision de la base de connaissances, alors qu'un poids de 0% met l'emphase sur la réduction du nombre de règles floues. Les poids d'optimisation et de sélection sont généralement pris égaux sans, par contre, que cela ne soit une règle absolue.

La reproduction est principalement gérée par le mécanisme de croisement (70-100%), le reste étant exécuté presque entièrement par le mécanisme de déplacement des sous-ensembles flous, alors que la mutation est quant à elle maintenue à un bas niveau d'environ 5%.

CHAPITRE 4

TOOL WEAR MONITORING USING GENETICALLY-GENERATED FUZZY KNOWLEDGE BASES

Soumis à *Engineering Applications of Artificial Intelligence.*

Journal affilié à l'organisation IFAC (International Federation of Automatic Control).

Edition *Elsevier Science.* [17]

Abstract

In this paper, two fuzzy logic systems are compared with a neural network system for application of tool wear monitoring. Although the three artificial intelligent systems are equivalently accurate for tool wear estimation, they greatly differ from their learning point of view. The manual construction of a fuzzy knowledge base from a set of experimental data is time consuming, while requiring a human expert. Alternatively, the fuzzy knowledge base can be automatically constructed by a genetic algorithm from the same set of experimental data in a shorter period of time than the one required to train the neural network, and this, without requiring any human expertise. The fuzzy logic system with a genetically-constructed

knowledge base is thus recommended for factory floor implementation of tool wear monitoring.

Keywords: Artificial intelligence, Fuzzy decision support system, knowledge base, tool condition monitoring, genetic algorithm, neural network.

4.1 Introduction

Since tool wear has a direct effect on the quality of machined parts, on-line wear monitoring is one of the most important challenges in manufacturing. The tool wear influences a variety of machining phenomena, and thus, number of monitoring systems use, e.g., the increase of the cutting force or other related quantities, as a mean for tool wear estimation [1, 2, 3]. Systems developed in laboratories are often multi-sensor systems embodying artificial intelligent (AI) methods in order to make more reliable estimation of the state of the tool, and consequently, of the machined parts themselves [1, 4, 5]. Usually, a set of experimental tests involving different cutting conditions, e.g., different feed rate and depth of cut, is repeatedly performed on a typical part. During the machining, the cutting and feed forces are recorded, while the tool wear is manually measured after each test. These experimental data are then cast into a knowledge base (KB) through a learning process. Finally, this KB is used by an AI method to predict the tool wear. Among these methods, fuzzy logic (FL) systems, neural networks (NN) and neuro-fuzzy (NF) systems are the

most frequently chosen AI methods, for this type of application [5, 6, 7, 8]. The aim of this paper is to compare performances of two FL-based monitoring systems relative to those of a NN-based system, for application to tool wear estimation. The first FL system, called FL-MA, uses a KB manually constructed by a human expert from a set of experimental data, while the second FL system, called FL-GA, uses a KB automatically constructed by a genetic algorithm (GA) from the same set of experimental data.

Below, each of these three AI methods are briefly presented with emphasis on the GA used to automatically-construct the KB, since the later is a relatively new method [9]. The experimental conditions of tool wear estimation are then described together with the learning and operating conditions. Finally, performances and required resources are then compared and discussed.

4.2 Monitoring Systems

Since the tool wear monitoring requires multiple input information to predict the tool wear, this type system can be cast into the class of multi-inputs and single output (MISO) systems.

4.2.1 Neural Network

Multi-layer NN are one of the most well-known types of AI systems. In this paper, a NN of three layers is considered for tool wear monitoring. As shown in

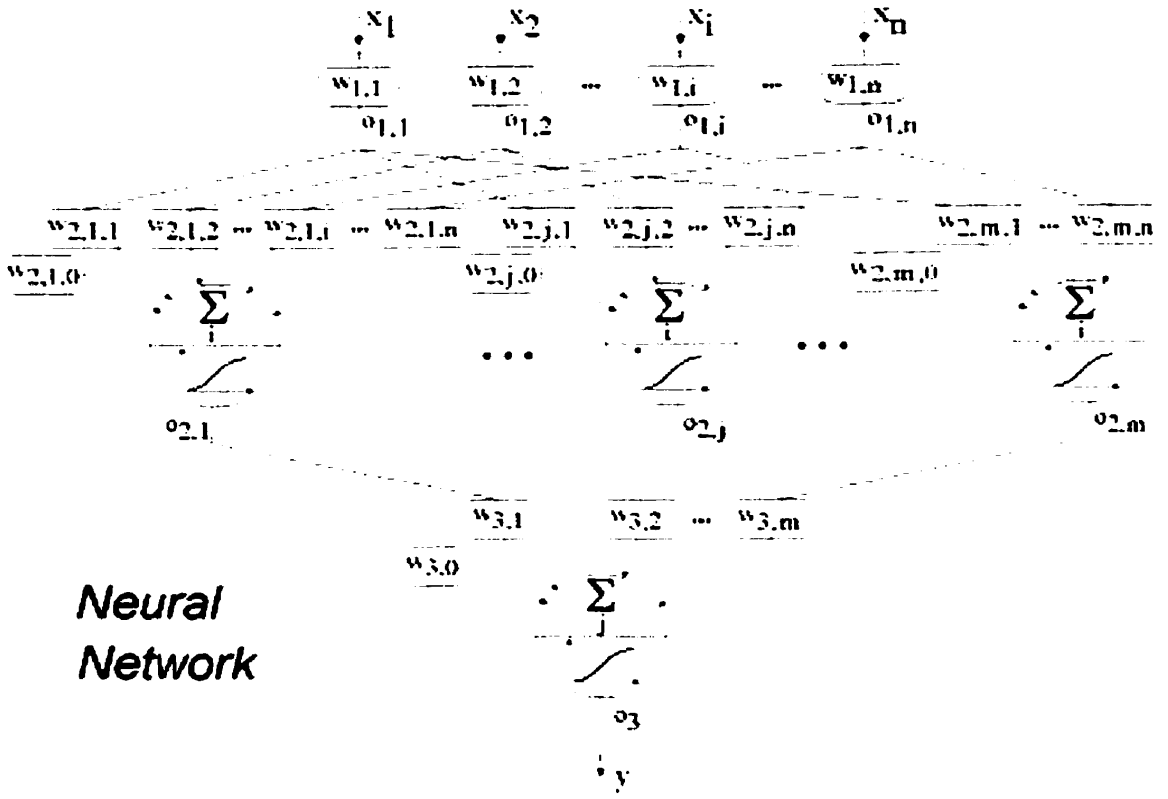

Figure 4.1 Neural Network with three layers

Fig. 4.1, layer 1, also called the *input layer*, contains n neurons, one corresponding to each input x_i, layer 2, also called the *hidden layer*, contains m neurons, while layer 3 contains only one neuron, since it is a MISO system. The output signal of the neurons of layer 1 is readily computed as

$$o_{1,i} = w_{1,i}x_i, \quad i = 1, ..., n, \tag{4.1}$$

where $w_{1,i}$ and x_i are, respectively, the weight and input signal of neuron i of layer 1. The input signal of each neuron j of layer 2 is computed as a weighted sum of

the output signal of the neurons of layer 1, i.e.,

$$s_{2,j} = \sum_{i=0}^{n} w_{2,j,i} o_{1,i}, \quad j = 1, ..., m, \tag{4.2}$$

where $w_{2,j,i}$ is the weight corresponding to the output signal $o_{1,i}$ and $o_{1,0} \equiv 1 - bias$. The output signal of each neuron j of layer 2 is computed with the sigmoid activation function as

$$o_{2,j}(s_{2,j}) = \frac{1}{1 + e^{-s_{2,j}}}, \quad j = 1, ..., m. \tag{4.3}$$

Finally, the input and output signals of the single neuron of layer 3 are computed similarly.

4.2.2 Fuzzy Logic System

A rule-based approach to decision making using FL 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 [10]. The former is divided into two parts: the fuzzy premises (or inputs) and the fuzzy conclusions (or outputs). For the sake of simplicity, we consider only non-symmetric triangular fuzzy sets on the n inputs (called in this context premises) and sharp-symmetric triangular fuzzy sets on the single

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 [10]. 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:

$$R^k_{MISO} : \text{if } x_1 \text{ is } X^k_1 \text{ and } x_2 \text{ is } X^k_2 \text{ and } \cdots \text{ and } x_n \text{ is } X^k_n \text{ then } y \text{ is } Y^k, \qquad (4.4)$$

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

$$R = also^K_{k=1}(R^k_{MISO}), \qquad (4.5)$$

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:

$$Y' = (X'_1, X'_2, \cdots, X'_n) \circ R, \tag{4.6}$$

where $\circ$ denotes a compositional rule of inference (CRI), e.g., the *sup*-$\wedge$ or *sup-prod* (also denoted *sup*-$*$). Alternatively, the CRI of eq.(4.6) is easily computed as

$$Y' = X'_n \circ \cdots \circ (X'_2 \circ (X'_1 \circ R)). \tag{4.7}$$

Usually, 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

$$Y' = \left\{ \begin{array}{c} \vee_{k=1}^{K} \\ \sum_{k=1}^{K} \end{array} \right\} \quad \underset{\{x_i \in X_i\}_{i=1}^{n}}{sup} \quad *_t\big(\, *_t(X'_n, \cdots, X'_2, X'_1)\,,\; *_t(X_1^k, X_2^k, \cdots, X_n^k, Y^k)\big), \tag{4.8}$$

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'. Additionally, there are usually 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. The fuzzy KB must be either manually constructed by a human expert or automatically constructed by a genetic algorithm as it is explained below.

4.2.3 Genetic Algorithm

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 [11]. As shown in Fig. 4.2, each individual of a population is a potential KB. The method uses iterative improvement of individuals at each

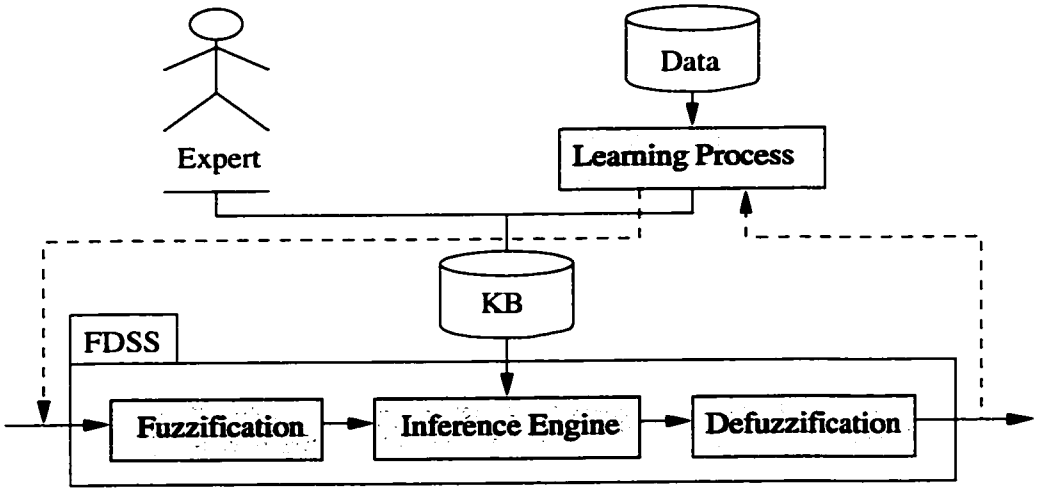

Figure 4.2 The learning paradigm of FDSS Fuzzy-Flou

generation to converge toward multiple optima simultaneously. This evolutionary process operates directly on the *genotype*—i.e., the coded physical characteristics

into bit string—of individuals rather than on the *phenotype*—i.e., the physical characteristics themselves—. It is noteworthy that the coding of several parameters into bit strings is crucial in GA. When the number of unknown parameters increases, GA exhibits only a polynomial increase in the size of the search space, while the other optimization techniques show an exponential increase. Figure 4.3 presents the *encoding/decoding scheme* as well as the four basic operations, i.e.: *reproduction*, *mutation*, *evaluation* and *natural selection*, of the developped GA learning sofware [12].

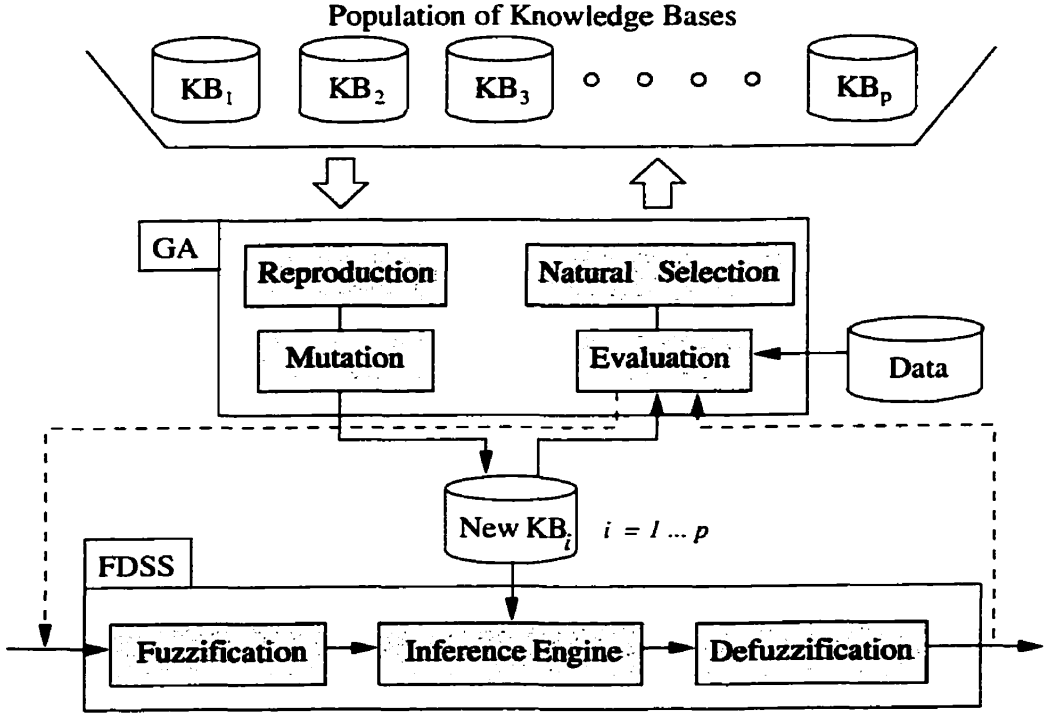

Figure 4.3 The GA learning process of and FDSS Fuzzy-Flou knowledge base

4.2.3.1 Encoding/Decoding Scheme

The genotype of an individual p member of a population of size P is defined as

$$G^p \equiv \{\ G^p_{sets},\ G^p_{rules}\ \}, \tag{4.9}$$

where G^p_{sets} and G^p_{rules} are respectively the genotypes of the fuzzy sets and rules. For the sake of brevity, the indice p is omitted in the following equations. However, it must be clear that all the following genotypes apply to any individual p.

Fuzzy sets:

The genotype of the fuzzy sets must contain all the information on the position of the fuzzy sets on the premises and the conclusion, i.e.:

$$G_{sets} \equiv \{G_{X_1}, G_{X_2}, \cdots, G_{X_n}, G_Y\}, \tag{4.10}$$

where G_v is the genotype of the n_v fuzzy sets on v, i.e.,

$$G_v \equiv \{\underbrace{10...01}_{g_v^1}\ \underbrace{11...10}_{g_v^2} \cdots \underbrace{01...11}_{g_v^{n_v}}\}, \quad \forall v \in \{X_1, X_2, \cdots, X_n, Y\}. \tag{4.11}$$

As shown in Fig. 4.4, the fuzzy sets are made of sharp symmetric triangles on the conclusion—to have an equal weighting—and non-symmetric triangles on premises—to allow overlapping, and hence, a reasoning process—.

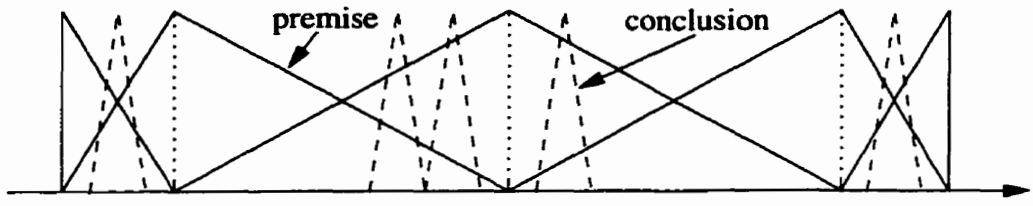

Figure 4.4 Fuzzy sets on a premise and a conclusion

The phenotype p_v^i expresses the location of the summit of a triangle on the premise or the conclusion field v. For each premise, there are always two half-triangles located at p_v^{min} and p_v^{max}, and hence it is not necessary to encode their positions in G_v (also not counted in n_v).

The number of bits, denoted b_v, allocated to each basic genotype g_v is chosen in such a way as to obtain a desired resolution r_v on the positioning of the fuzzy sets along p_v between p_v^{min} and p_v^{max}. The encoding of the basic phenotype p_v^i into its corresponding genotype g_v^i is given as

$$g_v^i = f(p_v^i), \quad \forall i = 1, ..., n_v, \quad \text{and} \quad \forall v \in \{X_1, X_2, \cdots, X_n, Y\}, \tag{4.12}$$

where the *encoding scheme* $f(\cdot)$ is defined as

$$f(p_v^i) \equiv \frac{p_v^i - p_v^{min}}{r_v}, \quad g_v^i \in \{0, 1, \cdots, 2^{b_v} - 1\}, \quad p_v^{min} \leq p_v^i \leq p_v^{max}, \tag{4.13}$$

with the resolution r_v on the phenotype p_v computed as

$$r_v = \frac{p_v^{max} - p_v^{min}}{2^{b_v} - 1}. \tag{4.14}$$

The decoding of the basic genotype g_v^i into its corresponding phenotype p_v^i is given as

$$p_v^i = f^{-1}(g_v^i), \quad \forall i = 1, ..., n_v, \quad and \quad \forall v \in \{X_1, X_2, \cdots, X_n, Y\}, \tag{4.15}$$

where the *decoding scheme* $f^{-1}(\cdot)$ is defined as

$$f^{-1}(g_v^i) \equiv r_v g_v^i + p_v^{min}. \tag{4.16}$$

Fuzzy rules:

The genotype of fuzzy rules must contain information about all the possible combinations of connecting a fuzzy set of the conclusion to a fuzzy set of each premises. The maximum number of rules, K, is given as the total number of combinations, i.e.,

$$K = (n_{X_1} + 2)(n_{X_2} + 2) \cdots (n_{X_n} + 2), \tag{4.17}$$

where the +2 is required because of the presence of the two half triangles, located at $p_{X_i}^{min}$ and $p_{X_i}^{max}$, that are not counted in n_{X_i}. Without lost of generality, we can

assign fuzzy sets $p_{X_i}^{min}$ to indice 0 and $p_{X_i}^{max}$to indice $n_{X_i}+1$ such that we have:

$$p_{X_i}^{0} \equiv p_{X_i}^{min}, \quad p_{X_i}^{n_{X_i}+1} \equiv p_{X_i}^{max}, \quad i = 1, ..., n_{X_i}. \tag{4.18}$$

The fuzzy rules are encoded into an ordered list of combination of premises, each having an enable/disable bit, denoted e, together with a conclusion fuzzy set number, i.e.:

$$G_{rules} \equiv \{\underbrace{e10...01}_{g_0}\underbrace{e11...10}_{g_1}\cdots\underbrace{e01...11}_{g_{K-1}}\}. \tag{4.19}$$

The number of bits b_r allocated to each g_k must have sufficient space to refer to n_Y conclusion fuzzy sets plus one enable/disable bit, i.e.,

$$2^{(b_r-1)} \geq n_Y. \tag{4.20}$$

4.2.3.2 Reproduction

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 by one of the following three operators based on an initiating probability.

Simple Crossover:

In general, the reproduction is mainly performed (with a probability p_1) by simple

crossover of the *genotype* of two parents to produce the genotype of two children. The simplest way to implement this operation is as follows: the parents are selected based on their ability; the genotype of the parents is split in two parts at a randomly selected crossover site; the genotype of the children is formed by recombining one part of the genotype of each of their parents, as shown in Fig. 4.5.

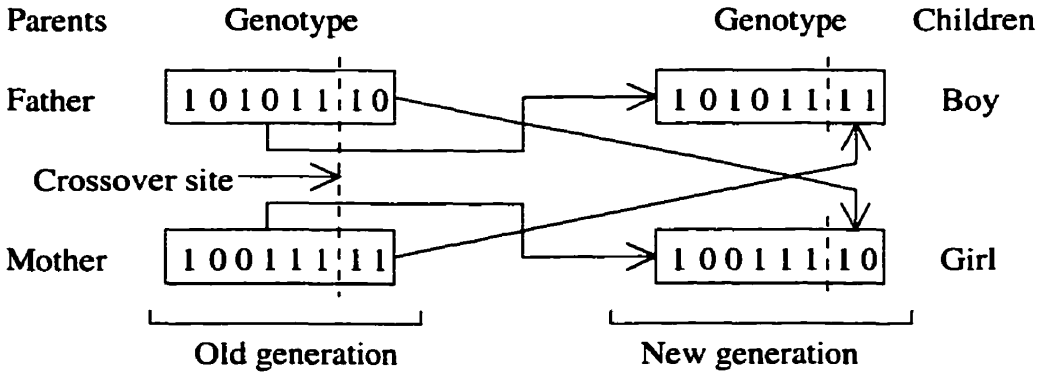

Figure 4.5 Simple crossover of the genotypes of two parents

Fuzzy-Sets Displacement:

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

Fuzzy-Rules Reduction:

The reduction of the number of fuzzy rules is performed (with remaining probability p_3) computed as

$$p_3 = 1 - p_1 - p_2. \tag{4.21}$$

One of the K fuzzy rules is randomly selected, and the activation bit e is disable. Obviously, this operator does not always give rise to a reduction in the number of fuzzy rules, but rather gradually works toward that direction.

Mutation:

Mutation is a random inversion of a bit in the genotype of a new member of the population as shown in Fig. 4.6. Mutation makes it possible to try completely dif-

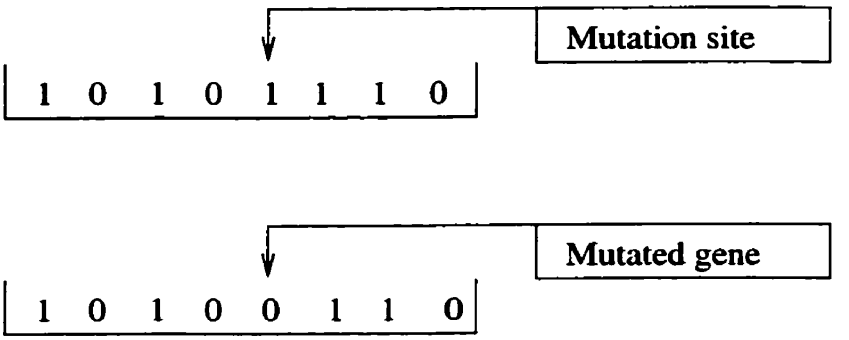

Figure 4.6 Mutation of a genotype

ferent solutions. The probability of mutation p_4 should be kept very small in order to let the population improve itself mainly with the other types of reproduction

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

4.2.3.3 Natural Selection

The capacity of each KB to survive natural selection is measured by two objective functions. The first objective function, denoted ϕ_1, evaluates the capacity of a KB to approximate the set of experimental data, i.e.:

$$\phi_1 \equiv \frac{\delta - \epsilon_{RMS}}{\delta}, \quad \delta \equiv p_Y^{max} - p_Y^{min}, \tag{4.22}$$

where δ is defined as the range on the conclusion Y and ϵ_{RMS} the root-mean-square error between the FL conclusion Y_i and the desired conclusion y_i, for the N experimental data, i.e.,

$$\epsilon_{RMS} = \sqrt{\frac{\sum_{i=1}^{N}(Y_i - y_i)^2}{N}}. \tag{4.23}$$

The second objective function, denoted ϕ_2, evaluates the complexity of a KB through its number of active fuzzy rules, i.e.,

$$\phi_2 \equiv \frac{K - n_a}{K}, \tag{4.24}$$

where K is recalled to be the maximum number of fuzzy rules and n_a the actual number of active fuzzy rules. In order to deal with these two contradictory

objectives, a weighted sum of the two former is used, i.e.,

$$\phi = \omega_o \ \phi_1 + (1 - \omega_o)\phi_2, \tag{4.25}$$

where the weight ω_o is usually set around 75%. The influence of ω_o, p_1, p_2 and p_3 are extensively discussed in [13]. Natural selection is performed on the population by keeping the *most* promising KB along a single fitness value. This is equivalent to using solutions that are closest to the optimum. In this work, the size of the population is kept constant to 100 KB. At each generation, the reproduction produces 100 brand new KB that are evaluated and ranked. The natural selection applies on the resulting 200 KB by keeping the 50 best non-identical KB along ϕ and ϕ_1, for a total of 100 KB.

4.3 Knowledge Base Learning and Results

In this paper, the tool wear, denoted VB, is estimated from only $n = 3$ input information, i.e.: the feed rate, denoted f; the feed force, denoted F_f; and the cutting force, denoted F_c. This choice of input variables is based on the following two observations (see Fig. 4.7 and 4.8). Force F_f is independent of f, but rather depends on VB and the depth of cut, denoted d. Moreover, F_c depends on d and f, while being only weakly dependent on VB. This gives one the interesting opportunity to use f and the measurement of F_c to determine d, and using the

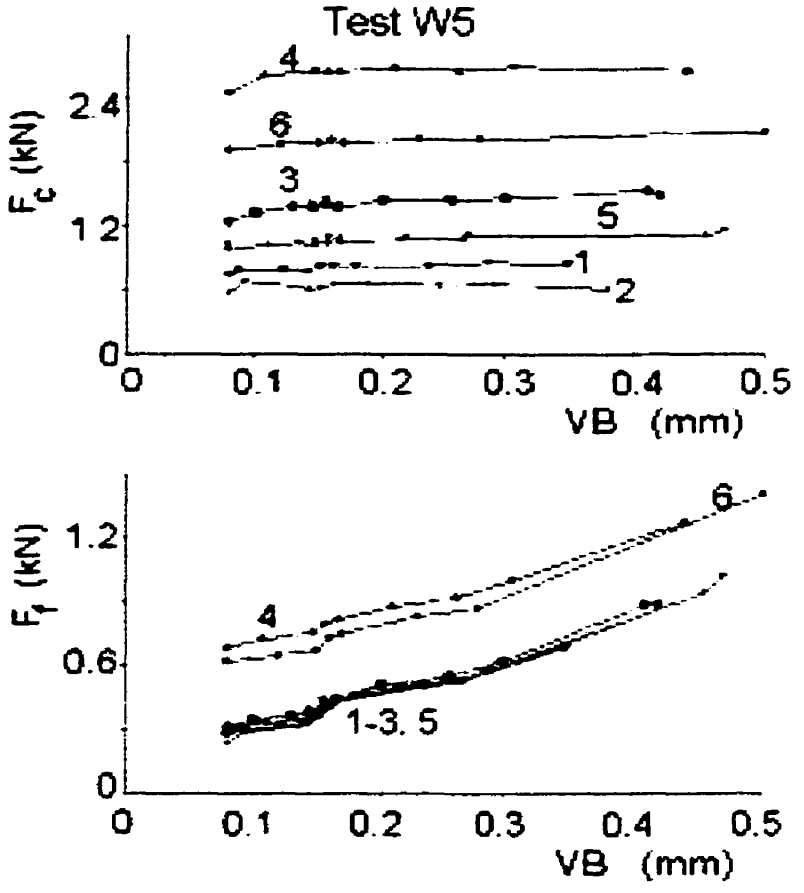

Figure 4.7 Cutting force components vs Tool wear (set of data W5)

measurement F_f to estimate VB without requiring d as input variable. Two sets of experimental tests are repeatedly performed until a tool failure occurred. Test W5 is intented to be use for KB learning, while test W7 is used to verify the performances of the different monitoring systems. In order to simulate factory floor conditions, a typical part is machined on a conventional lathe under six different cutting conditions, such as shown in Fig. 4.9. The cutting speed of each operation is selected to ensure approximately the same share in tool wear. Tool wear is manually measured after carrying out each cycle, and the VB of each single cut

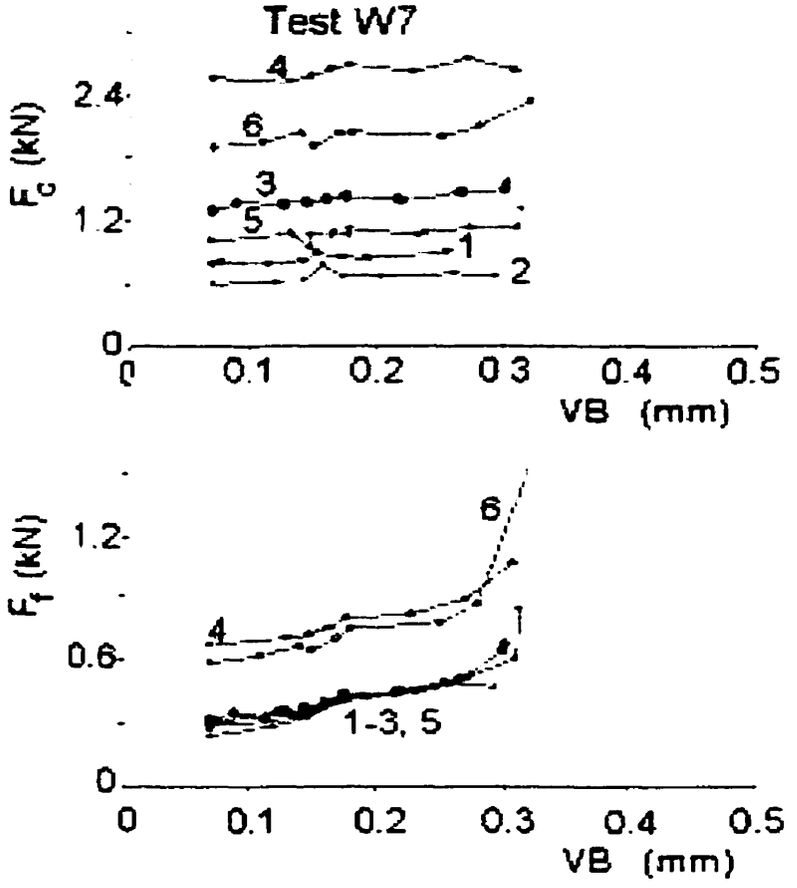

Figure 4.8 Cutting force components vs Tool wear (set of data W7)

is linearly interpolated. For each cut, f and F_c were measured using a Kistler 9263 dynanometer during 5 second intervals while the cut was executed. Since the inserts used in the experiments had soft, cobalt-enriched layer of substrate under the coating, the tool life had a tendency to end suddenly after this coating wore through. In test W5, ten cycles were performed, until a sudden rise of the flank wear VB occurred, reaching approximately 0.5 mm. In test W7, failure of the coating resulted in chipping of the cutting edge at the end of the 9-th cycle, where flank wear was about 0.35 mm. The results of tests W5 and W7 are shown in

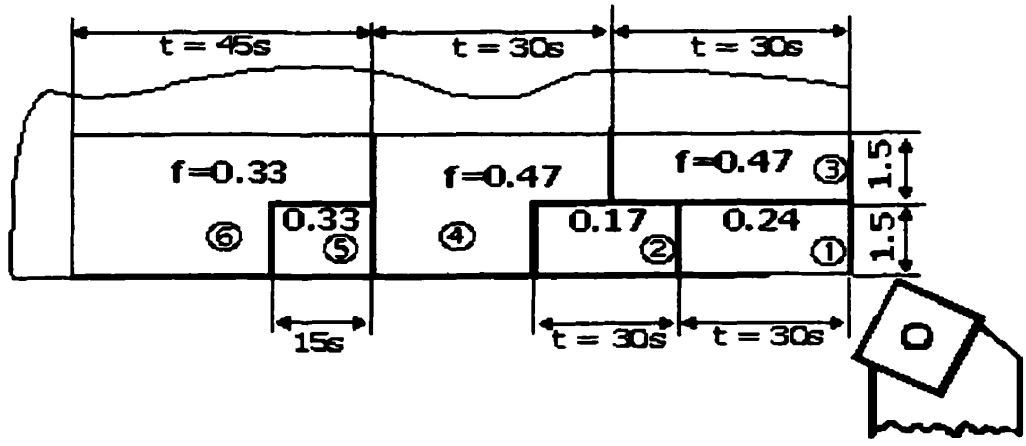

Figure 4.9 Sets of cutting parameters

Fig. 4.7 and 4.8. The approximation error of the monitoring systems are measured using the root-mean-square error

$$\Delta_{rms} = \sqrt{\sum_{i=1}^{N} \frac{(VB_m - VB_e)^2}{N}}, \tag{4.26}$$

and the maximum error

$$\Delta_{max} = max(VB_m - VB_e), \tag{4.27}$$

where VB_m and VB_e are, respectively, the measured and estimated VB, and N is the the number of patterns in the experimental test (i.e., $N = 71$ for W5; and $N = 66$ for W7).

4.3.1 Neural Network

The training of the NN is made up to 200 000 iterations with the results of test W5, while using from $m = 2$ to 10 cells in the hidden layer. As shown in Table 4.1, the approximation errors are almost identicals for network with 3 or more hidden cells. Therefore, $m = 5$ cells is arbitrarily chosen for the hidden layer. For the two sets of experimental data, the approximation errors are both

Tableau 4.1 Approximation errors of the Neural Network method

		W5 (training)	W7 (testing)
# of hidden cells	Training time [min]	average error [mm]	average error [mm]
2	7.33	0.0158	0.0370
3	10.43	0.0160	0.0367
4	12.92	0.0140	0.0362
5	15.37	0.0149	0.0363
6	18.03	0.0149	0.0362
7	20.78	0.0150	0.0362
8	23.17	0.0149	0.0362
9	25.72	0.0145	0.0361
10	28.83	0.0144	0.0362

acceptable. Of course, the Δ_{max} and Δ_{rms} for test W7 are larger than those of the learning test W5. Moreover, the Δ_{max} is extremely large. This error occurred only once, i.e., for the last cutting force measurement just after chipping of the cutting edge. The chipping did not cause an increase of VB but resulted in an increase in the F_f. Since both values are high for VB and the chipped edge mean tool failure, the output of the network should not be considered erroneous. Without the last results, the Δ_{max} and Δ_{rms} would be 0.081 mm and 0.029 mm, respectively.

Obviously, the learning time depends on the computer used. On a Pentium II-350 MHz, the learning time is about 15 min for 5 hidden cells and 29 min for 10 hidden cells. Since the network needs retraining from time to time, this long delay can be considered as an important limitation to the use of this type of monitoring system on the factory floor. Apparently, the weights become approximately constants after 100 000 iterations. This means that the NN is not sensitive to over-training, i.e., it did not fit too closely the learning set of experimental data, which can lead to a loss of generalization ability. For factory floor practice, small changes should not be shown to the operator, since they are too complex to deal with. Despite the useless of this longer learning, the number of iteration has been kept to 200 000 to ensure conservative results.

4.3.2 Manually-Constructed Fuzzy Knowledge Base

A certain level of experience and expertise is required in order to develop the fuzzy knowledge base (FKB) from the set of experimental data, since some conditions may be uncertain and incomplete, and hence, must be estimated. The quality of the FKB depends on the quality of the data and the skills of the expert. The usual approach is to choose the number and location of the fuzzy membership functions on each premises and the conclusion, and finally, to determine the fuzzy rules. If the membership functions are wisely-chosen, only a small number of fuzzy rules are usually needed.

Apparently from Fig. 4.7 and 4.8, the relationship between F_f and VB is roughly linear, and thus, only two fuzzy sets are necessary on this premise. The same approximately linear relationship can be observed between F_c and VB, and between F_f and VB, and hence, only two fuzzy sets are necessary for those two premises, as shown on the right of Fig. 4.10.

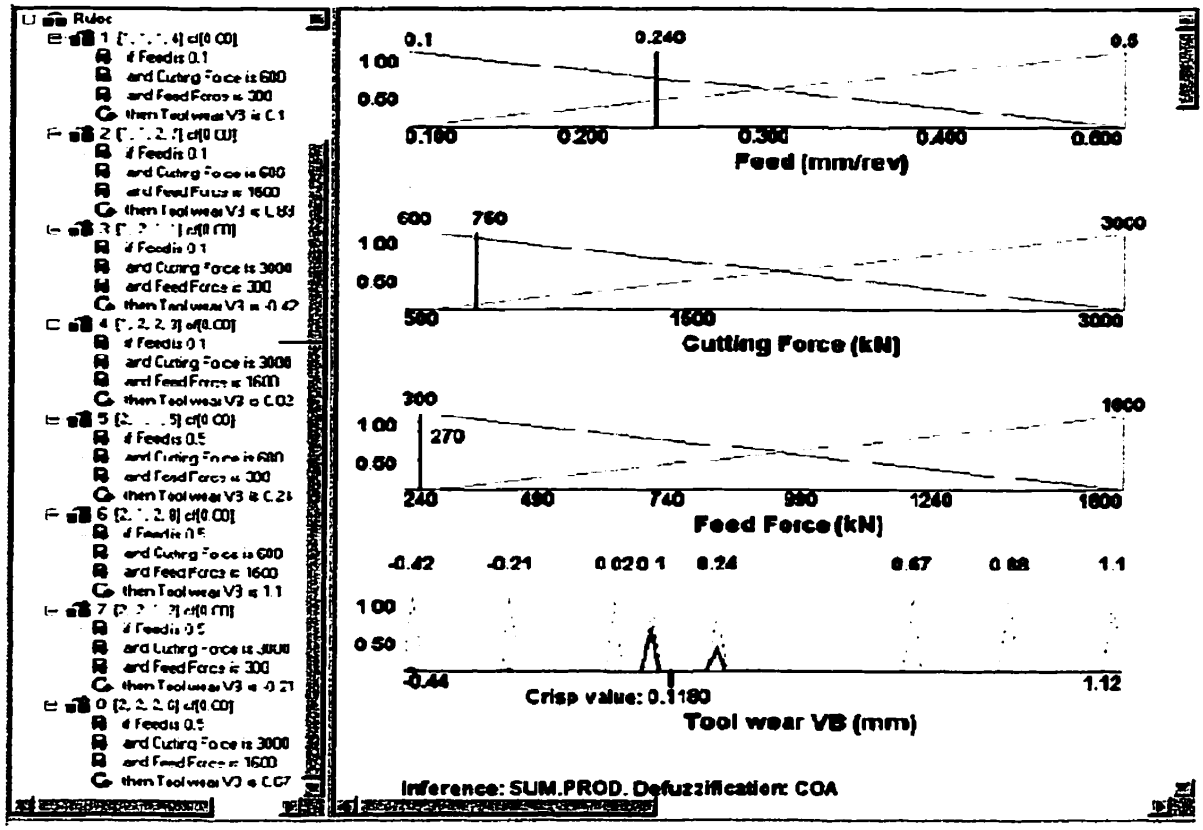

Figure 4.10 Screen printout of the manually constructed knowledge base

It is noteworthy that the extreme values of the two fuzzy sets on premise f are 0.1 and 0.5 mm/rev. A range wider then the one used by the experiments (i.e. 0.17 and 0.47 mm/rev). This widening of the feed rate range ensures a proper

working of the monitoring system in cases when shop floor feed rate values exceed the experimental range.

There are a maximum of eight possible fuzzy rules with two fuzzy sets on three premises (i.e. $2^3 = 8$). From Fig. 4.7 of test W5, the first fuzzy rule can be directly established as: If f is 0.1 mm/rev and F_c is 600 kN and F_f is 300 kN then VB is 0.1 mm; and shown in the left side of the Fig. 4.10. In such a rule, the fuzziness is expressed by the membership function. The strongest conclusion arise at the maximum degree of memberships of each premises (i.e., 0.1, 600 and 300, respectively). The conclusion value diminishes as the observations move away from their maximum degree of memberships. Problems occur when the expert try to define a second fuzzy rule, because there is no experimental measure of VB, for $f = 0.1$ mm/rev, $F_c = 600$ kN and $F_f = 1600$ kN. However, it is possible to extrapolate the actual measurements up to an $F_c = 600$ kN. In this case, we obtain an approximate value of $VB = 0.9$ mm. Obviously, this value of VB does not have any physical meaning. It only serves to complete the integrity of the FKB. As shown for the fuzzy rules 3 and 7 of Fig. 4.10, the VB can even be negative for this purpose. Once all the fuzzy rules are defined, the expert can performed a tuning of the location of the fuzzy sets on the premises together with a revision of the fuzzy rules themselves in the aim of reducing the approximation errors. The complexity of this tuning process depends on the number of fuzzy sets and rules. For a simple FKB, only minor adjustment should be necessary. In our case here,

the location of the conclusion fuzzy set of the fuzzy rule 2 has been moved from $VB = 0.9$ mm to $VB = 0.88$ mm.

Figure 4.10 shows a screen printout of the software *Fuzzy-Flou* developed at École Polytechnique de Montréal and the Warsaw University of technology with the manually-tuned FKB. On the left side, the fuzzy rules in numerical and linguistic forms are presented. While on the right side, the fuzzy sets are shown on the three premises. One can see an example of VB estimation, for the following inputs: $f = 0.24$ mm/rev, $F_c = 750$ kN and $F_f = 270$ kN, a crisp value (center of gravity of conclusions $VB = 0.1$ mm and 0.24 mm) of estimated VB is 0.118 mm.

The performance results of the FL-MA system are presented in Table 4.2. As

Tableau 4.2 Δ_{rms} of the three AI methods

	W5 (training)	W7 (testing)
AI method	Δ_{rms} (mm)	Δ_{rms} (mm)
neural network	0.015	0.029
FL-MA	0.024	0.034
FL-GA	0.02	0.037

in the previous case, a large value of Δ_{max} results from chipping of the cutting edge, and therefore, the answer of the FL-MA should not be considered erroneous. Without this last result Δ_{max} is 0.056 mm and Δ_{rms} is 0.034 mm. Unlike the NN, which is a kind of *black-box*, the FKB presented above is *transparent* and *understandable*. Nevertheless, the manual construction of the FKB requires knowledges and experiences, which can seldom be expected from a machine tool operator.

Therefore, FL in its *pure* form is not recommended for small batch production, but rather for mass production, where some operations are carried out repeatedly over an extended period of time, at least several months.

4.3.3 Genetically-Constructed Fuzzy Knowledge Base

Alternatively, a genetic algorithm can be used to automatically generate the FKB, from the same set of experimental data (test W5). As previously explained, the GA uses a set of probabilities (i.e., p_1, p_2, p_3, p_4) in order to control the occurrences of the different reproduction operators. Obviously, a different set of values drive the GA toward an FKB with different behaviors. Moreover, the weight ω_o between the two contradictory objectives ϕ_1 (approximation error) and ϕ_2 (number of fuzzy rules) produces FKBs with completely different behaviors.

In general, a different value of weight can be used at each iteration of the learning process (called ω_o) than the weight used at the end of the learning process to select the final FKB (called ω_s). Four FKB has been automatically constructed from test W5 with the following parameters:

- Run 1 : $\omega_o = 0.8,\ p_1 = 0.85,\ p_2 = 0.13$;
- Run 2 : $\omega_o = 0.8,\ p_1 = 1.00,\ p_2 = 0.00$;
- Run 3 : $\omega_o = 1.0,\ p_1 = 0.85,\ p_2 = 0.13$;
- Run 4 : $\omega_o = 1.0,\ p_1 = 1.00,\ p_2 = 0.00$;

It is noteworthy that a $\omega_o = 1.0$ puts all the emphasis on the approximation accuracy, while an $\omega_o = 0.0\%$ puts the emphasis on a reduction of the number of fuzzy rules. The other parameters are fixed as follow:

- $p_4 = 0.05$;
- $\omega_s = 1.00$.

The operation must define the maximal limits of complexity of the desired FKB. These limits are chosen for this problem as:

- a maximum of 7 fuzzy sets on each premise;
- a maximum of 8 fuzzy sets on the conclusion.

As a result, the maximum number of fuzzy rules is given as: $7 \times 7 \times 7 = 343$, while the maximum of 7 fuzzy sets per premises is the most frequently chosen limits for the type of problems. As shown in Table 4.3, Δ_{rms} and Δ_{max} are both acceptable, for the two experiments (W5 and W7). The Δ_{max} is slightly high because of the last

Tableau 4.3 Δ_{rms} and Δ_{max} errors for training and testing sets of data

		Run1	Run 2	Run 3	Run4
Number of fuzzy rules		20	4	28	38
W5 (training)	Δ_{rms} (mm)	0.041	0.040	0.050	0.020
	Δ_{max} (mm)	0.110	0.110	0.170	0.070
W7 (testing)	Δ_{rms} (mm)	0.040	0.050	0.064	0.037
	Δ_{max} (mm)	0.090	0.140	0.170	0.100

measurement of VB, a fact that was also noticed with the other systems. The time

of the execution is around 14 minutes for each run (on Pentium II-350 Mhz), which makes the method very attractive for factory floor use. The best approximation accuracies are obtained with the Run 4, where both Δ_{rms} and Δ_{max} are relatively low. However, the FKB requires 38 fuzzy rules. Conversely, Run 2 provides an FKB with only 4 fuzzy rules with of course a higher approximation error. It is noteworthy that 38 fuzzy rules is still a manageable number of fuzzy rules by a human expert. It is recalled that this number of fuzzy rules has been reduced by the GA from the maximum of 343 fuzzy rules.

4.4 Comparison and remarks

All the three AI methods used to estimate tool wear give satisfactory results. There is a slight difference in the approximation errors, depending on the set of control data used (W5 and W7) . The testing set (W5) is approximated with a lower level of error, which was predictable, since it is the set used for the training.

However, an interesting point can be resorted from the automatically constructed FKB (see Table 4.3). Even if the Run 4 gives the best approximation error, since the Δ_{rms} is the lowest for the W7, the difference between the Δ_{rms} of W5 and W7 increases with the complexity of the FKB. Run 1 and 2 being the simplest, providing less fuzzy rules, 3 and 4 the more complex (see Table 4.3). This can be explained by a too close approximation of the training set, which is essentially the

case in Run 4, that leads to generate a specific FKB, rather than a model that can be used for other sets of data.

Figure 4.11 and 4.12 shows respectively the knowledge bases corresponding to Run 1 and Run 4.

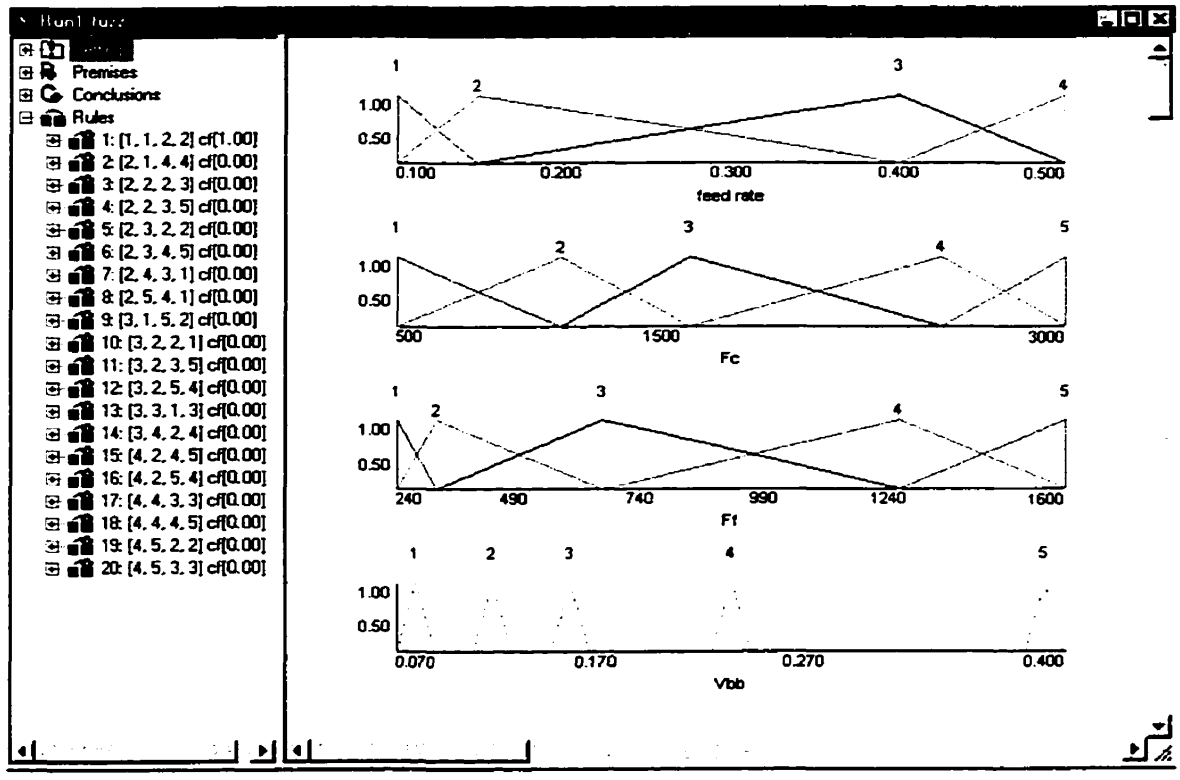

Figure 4.11 Knowledge base obtained from Run 1

If a fuzzy knowledge base has to be selected from the four runs, Run 4 can be an evident and easy choice since it provides the lowest Δ_{rms}. However, it is more interesting to use the results of Run 1 instead of the others, since it offers a better balance between simplicity and accuracy of the FKB. We can, also, notice it's stability regarding to the Δ_{rms}, since it stays at the same value for W5 and W7.

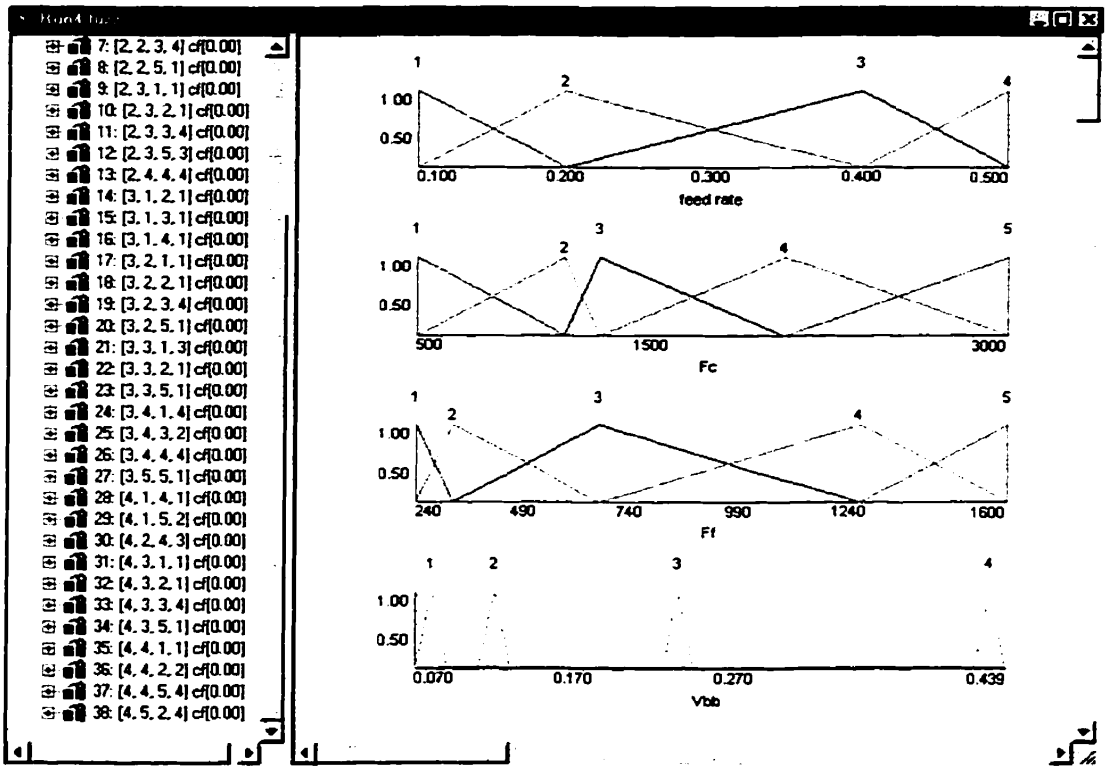

Figure 4.12 Knowledge base obtained from Run 4

Run 2, which gives the simplest FKB, is not reliable, since it uses a small number of fuzzy rules, leaving an important part of the input domain uncovered by fuzzy rules (lack of information), even if it provides good results for both sets of data (W5 and W7). Such a small amount of fuzzy rules remains a risky choice. As shown in Figs. 4.13 and 4.14, all three monitoring systems give similar and acceptable results for the prediction of tool wear, from a measure of the three premises.

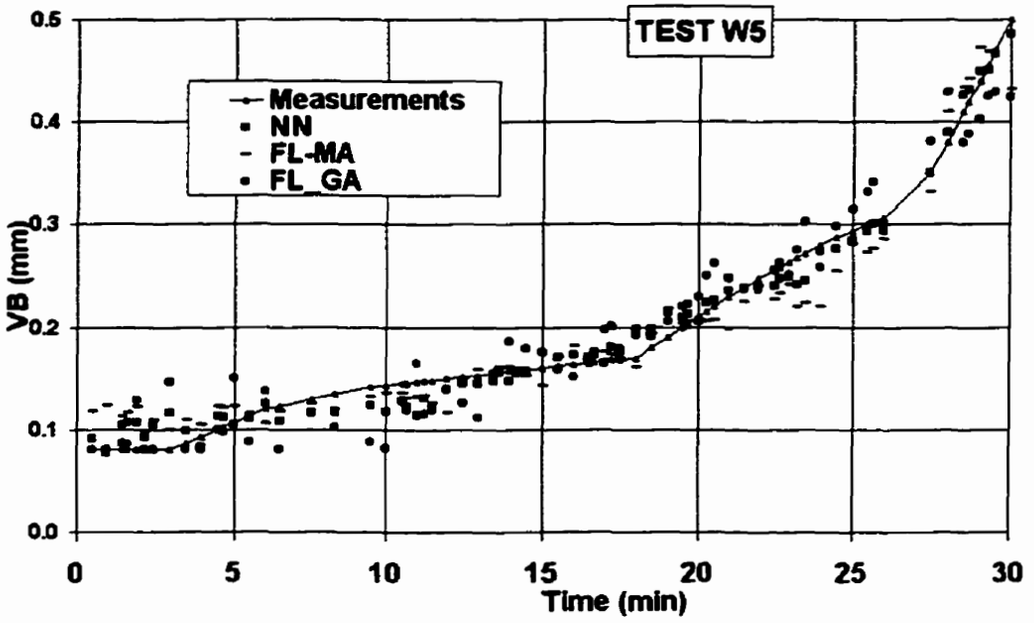

Figure 4.13 Tool wear vs time for training values

4.5 Conclusion

The three monitoring systems are equivalently accurate. Important differences remain in their internal structures which are irrelevant for the operator. However, a major difference in their usage is a critical factor. The construction of an FKB necessitates skills and expertise. The operator has to analyze the dependence of F_c on VB, which means that the results of preliminary experiments have to be presented to the operator in a convenient and understandable form. This makes FL systems, rather difficult, for practical implementation in it's manual form. However, this problem is solved by using a GA to automatically construct the FKB. The operator has no longer to analyze the experimental data. He only needs to

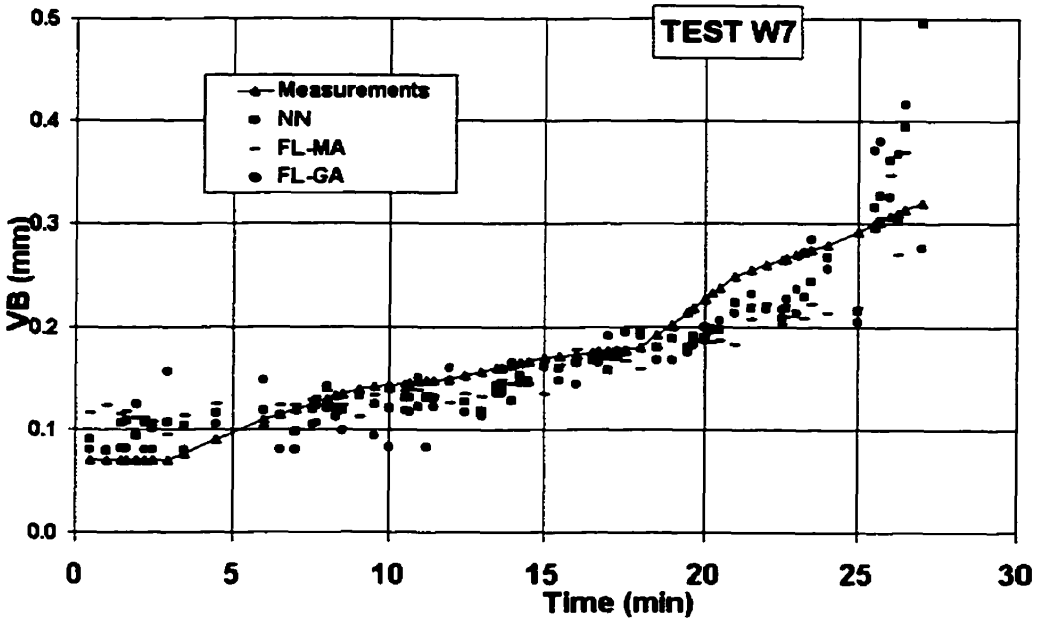

Figure 4.14 Tool wear vs time for testing values

setup the maximum level of complexity he wants to consider. This can even be pre-seted in order to avoid such interaction if desired. The learning time of the FL-GA method is very convenient, since among the three methods FL-GA is the shortest one, making it particularly attractive for shop floor uses. Moreover, one can specify the maximum level of complexity (as the maximum number of fuzzy rules) together with how much emphasis the GA must place on the increase of approximation accuracy relative to the reduction of the complexity level. Finally, the FKBs are transparent and understandable KBs relative to the *black-box*, where the NN stores the KBs.

References

[1] Byrne, G., Dornfeld, D., Inasaki, I, Ketteler, G and Teti R., "Tool Condition Monitoring (TCM)- The Status of Research and Industrial Application", *Annals of the CIRP*, 44/2, pp. 541–567, 1995.

[2] Du, R., Elbestawi, M.A. and Wu, S.M.,"Automated Monitoring of Manufacturing Process", *Journal of Engineering for Industry*, Part 1 & 2, 117, pp. 121–141, 1995.

[3] Jemielniak, K. and Kosmol, J., "Tool and Process Monitoring -State of Art and Future Prospects", *Scientific papers of the Inst. of Mech. Eng. and Automation of the Technical University of Wroclaw*, 61, pp. 90–112, 1995.

[4] 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.

[5] Leski, J. and Czogala, E., "A new fuzzy inference system with moving consequents in if-then rules. Application to pattern recognition" *Bulletin of the Polish*

Academy of Sciences, 45/4, pp. 643–655, 1997.

[6] Jemielniak, K., "Commercial Tool Condition Monitoring Systems", *Int. J. Adv. Manuf. Technol.*, Vol. 15, pp. 711–721, 1999.

[7] Monostori, L., "A Step towards Intelligent Manufacturing: Modelling and Monitoring of Manufacturing Processes through Artificial Neural Networks", *Annals of CIRP*, 42/1, pp. 485–488, 1993.

[8] Monostori, L., "Connectionist and neuro -fuzzy techniques in manufacturing", *The first World Congress on Intelligent Manufacturing*, pp. 940–949, 1995.

[9] Baron, L., Achiche, S. and Balazinski, "Fuzzy Decisions System Knowledge Base Generation Using a Genetic Algorithm", *submitted to The International Journal of Approximate Reasoning*, Mars 2000.

[10] 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.

[11] Baron, L., "Genetic Algorithm for Line Extraction", *Rapport technique EPM/RT-*

98/06, École Polytechnique Montréal, 1998.

[12] 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 3rd International Conference on Integrated Design and Manufacturing in Mechanical Engineering*, Montreal, Canada, May 2000.

[13] Balazinski, M., Achiche, S. and Baron, L., "Influences of Optimization and Selection Criteria on Genetically-Generated Fuzzy Knowledge Bases", (ICAMT2000) *International Conference on Advanced manufacturing Technology*, pp. 159–164, Jahor Bahru, Malaysia, August 2000.

CONCLUSION

Dans le travail de recherche présenté dans ce mémoire, nous nous sommes intéressés au problème d'automatisation du processus de construction de bases de connaissances pour les systèmes d'aide à la décision utilisant la logique floue. Le logiciel utilisé est le FDSS Fuzzy-Flou développé à l'École Polytechnique de Montréal et l'Université de Technologie de Silésie à Gliwice (Pologne).

Cette automatisation a été faite par le biais d'un algorithme génétique qui, à partir d'une base de données numériques quelconques, crée une base de connaissances complète.

De par les résultats obtenus, les AGs semblent bien adaptés à ce genre de problématique, puisqu'ils permettent de proposer des solutions (bases de connaissances) simplement à partir de données numériques, ce qui diminue considérablement le rôle accordé à l'expert dans le domaine des systèmes d'aide à la décision.

L'AG construit des bases de connaissances au mieux de deux critères contradictoires, à savoir: minimiser l'erreur et le nombre de règles floues—dans un soucis de simplicité, une base de connaissances plus simple est plus facilement gérable par un opérateur—.

L'espace de recherche de solutions est généralement très vaste. Prenons l'exemple

cité au chapitre 2, un système à deux prémisses, une conclusion et une complexité maximale de 7 sous-ensembles flous sur chacune des prémisses, 8 sous-ensembles flous sur la conclusion, et donc 49 ($7 \times 7 = 49$) règles floues, chaque prémisse d'entrée et de sortie étant discrétisée en 16 positions différentes, ce qui requiert un chiffre de 4 bits, et chaque règle floue est codée par un chiffre de 4 bits. Ceci génère le nombre impressionnant de $2^{((7-2)+(7-2)+8+49)\times 4} = 4.7 \times 10^{80}$ solutions possibles et si l'on considère que l'évaluation d'une base de connaissance prends 1 msec, alors il faudrait 1.5×10^{70} années pour tout évaluer ce qui est bien évidemment impossible. Néanmoins l'AG évalue qu'une infime partie de cet espace et arrive tout de même à proposer des solutions acceptables.

Notre automatisation a cependant certaines limites. Par exemple, la méthode ne prend pas en considération le choix des paramètres d'optimisation et de sélection qui sont traités au chapitre 3. Actuellement, ces valeurs sont choisies de façon manuelle par l'opérateur et dépendent bien évidemment des donnés numériques utilisées, ainsi que de la base de connaissances désirée, c'est-à-dire un choix adéquat entre simplicité (nombre de règles floues moindre) et précision (faible niveau d'erreur). Une automatisation du choix de ces paramètres pourrait être faite en ayant recours à plusieurs évolutions incomplètes préliminaires à l'AG. Les résultats obtenus seront utilisés pour procéder à l'évolution et à la recherche de solutions. Il est aussi possible de choisir des valeurs moyennes que nous avons obtenus par des

essais préliminaires.

La discrétisation choisie est aussi une limitation certaine de l'AG. Il est évident qu'augmenter cette dernière permettrait une répartition des sous-ensembles flous bien plus précise, mais conséquemment l'espace de recherche augmente considérablement. C'est donc un compromis qu'il faut essayer de trouver entre les deux. Afin de résoudre, du moins en partie, ce problème nous pouvons passer du codage binaire au codage en nombres réels avec des bornes limites, le problème que pose cette option est le passage d'un espace de solution fini à un espace infini, en plus de l'écart fait par rapport au modèle naturel de la génétique qui impose des génotypes de longueur finie.

Le mécanisme de reproduction utilisé dans notre AG est une composition entre un croisement simple et deux systèmes particuliers cités aux chapitres 1 et 2. Pour minimiser la perte d'information, les connaissances sur la structure même du codage des paramètres devraient être utilisées. Par exemple: changer le site de croisement en fonction de la position de certains paramètres dans le génotype; ou bien, choisir plusieurs sites de croisement différents et faire des changements sur ces chromosomes. La même chose pourrait être faite pour le mécanisme de mutation qui reste tout de même appliqué à un très faible pourcentage pour les raisons déjà men-

tionnées.

Même si la méthode mise en place n'a pas de limite théorique. Il est évident que le codage lui, pose des problèmes de limitation par la puissance de la machine utilisée. Actuellement l'AG est construit pour des systèmes à deux ou trois prémisses et une conclusion seulement. L'augmentation du nombre de prémisses accroît automatiquement le nombre de règles floues. Ce qui pourrait être fait c'est de limiter le nombre maximum de règles, et de ce fait, nous n'assisterions pas à une explosion combinatoire. Ce maximum permettrait d'augmenter le nombre de prémisses et même de raffiner la discrétisation sans pour autant trop accroître l'espace de recherche.

Il faudrait envisager aussi d'intégrer un module d'étude des bases de données que nous traitons de façon à pouvoir éliminer les points parasites (e.g. erreurs de mesure, erreurs de saisie, etc.) et aussi uniformiser les données qui gravitent autour d'une même valeur et les remplacer, par exemple, par une moyenne, cela éviterait de sommer ces écarts multiples surtout lorsqu'il s'agit de bases de données de volume important, ce qui peut être souvent le cas dans certaines applications de contrôle, comme par exemple le contrôle de la température des fours.

RÉFÉRENCES

[1] Czogala, E. et Hirota, K., "Probabilistic Sets: Fuzzy and Stochastique Approach to Decision, Control and Recognition Processes", *Verlag TUV Rheinland*, Cologne, 1986.

[2] Balazinski, M., Czogala E. et Sadowski T., "Modelling of Neural controllers with Application to the Control of a Mchining Process", *Fuzzy Sets and Systems 56*, Hollande, Amsterdame, pp. 273–280 1993.

[3] 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*, Séoul, Corée, Vol. II, pp. 798–801, juillet 4-9 1993.

[4] Balazinski, M. et Klim, Z., "Etude sur l'application de la logique floue pour la prédiction de la maintenance préventive", *International Industrial Engineering Conference*, Vol. II, pp. 1133–1142, octobre 1995.

[5] 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.

[6] Sugeno, M. et Yasukawa, T., "A fuzzy-logic-based approach to qualitative modeling", *IEEE Transactions on Fuzzy Systems*, Vol. 7, pp. 7–31, 1993.

[7] Diederich, J. et 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.

[8] Hagras, H., Callaghan, V., Colley, M. et 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.

[9] Yuan, Y. et Zhuang, H., " Using a genetic algorithm to generate fuzzy classification rules." *Proceedings. Third European Conference on Intelligent Techniques and Soft Computing (EUFIT'95)*, pp. 458–462, 1995.

[10] 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.

[11] Janikow, C.Z., "A genetic algorithm for optimizing fuzzy decision trees", *Proceedings of the 6th International Conference on Genetic Algorithms*, pp. 421–428, 1995.

[12] Valesco, J.R., Lopez, S. et 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.

[13] Nomura, H., Hayashi, I. et Wakami, N., "A self-tuning method of fuzzy reasoning by genetic algorithm", *Proceedings International fuzzy Systems and Intelligent Control Conference*, pp. 236–245, 1992.

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

[15] Baron, L., Achiche, S. et Balazinski, M., "Fuzzy Decision Support System Knowledge Base Generation using a Genetic Algorithm", soumis à *International Journal of Approximate Reasonning*, journal affilié à *North American Fuzzy Information Processing Society*, NAFIPS, Edition *Elsevier Science*, Mars 2000.

[16] Balazinski, M., Achiche, S. et Baron, L., "Influence of Optimization and Selection Criteria on Genetically-Generated Fuzzy Knowledge Bases", (ICAMT2000) *International Conference on Advanced manufacturing Technology*, pp. 159–164, Jahor Bahru, Malaisie, Août 2000.

[17] Achiche, S., Balazinski, M., Baron, L. et Jemielniak, K., "Tool Condition Monitoring Using Genetically-generated Fuzzy Knowledge Bases", soumis à *En-*

gineering Applications of Artificial Intelligence, journal affilié à *International Federation of Automatic Control*, IFAC, Edition *Elsevier Science*, Septembre 2000.