

2011.2835.9

Université de Montréal

Conception de systèmes experts d'identification de  
cibles à l'aide de la logique floue

par

Pascal Bouchard

Département de mathématiques et de statistique

Faculté des arts et des sciences

Mémoire présenté à la Faculté des études supérieures  
en vue de l'obtention du grade de  
Maître ès sciences (M.Sc.)  
en mathématiques

Orientation mathématiques appliquées

janvier 2001

© Pascal Bouchard, 2001



QA  
3  
L154  
2001  
N.002

**Université de Montréal**

Faculté des études supérieures

Ce mémoire intitulé

**Conception de systèmes experts d'identification de  
cibles à l'aide de la logique floue**

présenté par

**Pascal Bouchard**

a été évalué par un jury composé des personnes suivantes :

*Robert Brunet*

---

(président-rapporteur)

*Jean-François Angers*

---

(directeur de recherche)

*Éloi Bossé*

---

(co-directeur)

*Paul Arminjon*

---

(membre du jury)

Mémoire accepté le :

*janvier 2001*

---

## SOMMAIRE

---

Ce mémoire de maîtrise présente une des applications majeures de la logique floue : la conception de systèmes experts flous d'identification de cibles. Les bases de cette théorie, du raisonnement approché et les multiples facettes de l'identification de cibles sont abordées globalement. Les efforts ont été concentrés autour des moteurs d'inférence floue jugés les plus populaires dans le domaine du génie : les moteurs du produit et du minimum. Une légère tangente statistique a également été ajoutée au projet. Un tout nouveau moteur bayésien qui s'inspire grandement du domaine de la statistique a été conçu. Tous ces modèles sont scrutés systématiquement sous différents angles. En effet, une dizaine de simulations variées permettent d'analyser la panoplie de moteurs développés selon leur approche probabiliste et possibiliste respective. En résumé, il s'avère que le moteur d'inférence floue du produit avec opérateur d'union floue et fondement probabiliste est le plus performant. Il est suivi du moteur bayésien possibiliste qui surpasse légèrement celui probabiliste. Ces moteurs confirment sans équivoque l'intérêt marquant pour l'utilisation de la logique floue en identification de cibles : un rendement et une efficacité remarquables jumelés à une précision souvent fort étonnante. L'impact du nombre de senseurs, les appareils d'écoute électronique, sur le rendement global d'un système expert flou est également analysé. Il est jugé pertinent de restreindre le nombre de senseurs à trois pour tout type de moteur. L'optimalité n'est pas forcément une conséquence de l'emploi du nombre maximum de capteurs.

## REMERCIEMENTS

---

Avant de débiter, j'aimerais témoigner tout particulièrement de mon admiration à l'égard de mon directeur de recherche, Jean-François Angers. Tout au cours de mes études, j'ai pu côtoyer différents chercheurs et professeurs et j'affirme sans gêne qu'il est dans une classe à part. Son grand dévouement et sa disponibilité pour les étudiants font de lui un directeur de recherche hors pair que je recommande à tout futur étudiant. De plus, je profite de l'occasion pour souligner le travail remarquable du CES-MAT (Véronique Hussin, Paul Gauthier et Robert Brunet) qui a grandement facilité mon transfert universitaire. Je ne peux m'empêcher également de souligner la participation d'Éloi Bossé, du Centre de recherche pour la défense Valcartier, avec qui je suis en collaboration depuis près de quatre ans maintenant. Le support financier du FCAR et du CRNSG qui m'a permis de mener à bon port ce projet de maîtrise. De façon plus générale, je remercie mes parents de leur encouragement constant depuis le début de mes études. Ils m'ont toujours fourni l'essentiel et ainsi permis de mettre toutes les chances de réussite de mon côté. Ces études, je les ai faites pour moi, mais également un peu pour eux. En terminant, j'aimerais remercier tout spécialement Anne, mon amie, qui m'a supporté et encouragé au cours de ces deux années de maîtrise.

# Table des matières

---

<b>Sommaire</b> .....	iii
<b>Remerciements</b> .....	iv
<b>Table des figures</b> .....	ix
<b>Liste des tableaux</b> .....	xiv
<b>Introduction</b> .....	1
<b>Chapitre 1. Aperçu théorique</b> .....	3
1.1. Introduction .....	3
1.2. Théorie des ensembles flous .....	4
1.2.1. Concepts de base associés aux ensembles flous .....	5
1.3. Logique floue .....	8
1.3.1. Propriétés générales .....	8
1.3.2. Opérateurs logiques .....	9
1.3.2.1. Complément flou .....	9
1.3.2.2. Union floue : norme S .....	10
1.3.2.3. Intersection floue : norme T .....	11
1.3.2.4. Moyenne floue .....	12
1.3.3. Relation floue .....	13
1.3.4. Projection floue et extension cylindrique .....	14

1.3.5.	Composition de relations floues .....	15
1.3.6.	Principe d'extension .....	16
1.4.	Théorie des possibilités .....	18
1.4.1.	Principe général .....	20
1.4.2.	Traitement de l'incertitude .....	20
1.4.3.	Probabilité versus possibilité .....	21
1.5.	Inférence floue .....	23
1.5.1.	Variables linguistiques .....	24
1.5.1.1.	Conception des ensembles flous .....	26
1.5.2.	Base de règles floues si-alors .....	26
1.5.2.1.	Interprétation des règles floues si-alors .....	27
1.5.2.2.	Propriétés de la base de règles si-alors .....	29
1.5.3.	Étape de fuzzification .....	31
1.5.4.	Raisonnement approché .....	31
1.5.5.	Combinaison des règles si-alors .....	34
1.5.5.1.	Inférence basée sur la composition .....	34
1.5.5.2.	Inférence basée sur les règles individuelles .....	35
1.5.6.	Moteurs d'inférence floue .....	36
1.5.7.	Étape de défuzzification .....	40
<b>Chapitre 2. Modélisation et conception de systèmes experts flous</b>		<b>43</b>
2.1.	Introduction .....	43
2.2.	Définition du problème .....	44
2.2.1.	Sous-objectif théorique .....	45
2.2.2.	Sous-objectif pratique .....	45
2.3.	Caractéristiques des plates-formes .....	46
2.3.1.	Base de données de plates-formes .....	46
2.3.2.	Attributs .....	47

2.3.2.1.	Fuzzification .....	49
2.3.2.2.	Hypothèses et modélisation .....	51
2.3.3.	Identification hiérarchique .....	54
2.4.	Conception de systèmes experts flous .....	58
2.4.1.	Moteurs d'inférence floue privilégiés .....	59
2.5.	Transformation et application des moteurs d'inférence floue à l'identification de plates-formes .....	60
2.5.1.	Base de règles floues si-alors .....	60
2.5.2.	Pondération des capteurs .....	63
2.5.3.	Modus ponens généralisé .....	65
2.5.4.	Combinaison des règles si-alors .....	67
2.5.5.	Moteurs d'inférence floue résultant .....	70
2.5.5.1.	Moteur d'inférence floue du produit .....	71
2.5.5.2.	Moteur d'inférence du minimum .....	74
2.5.5.3.	Moteur d'inférence bayésien .....	76
2.6.	Raisonnement approché .....	80
2.6.0.4.	Logique probabiliste .....	81
2.6.0.5.	Logique possibiliste .....	82
<b>Chapitre 3.</b>	<b>Simulation et analyse .....</b>	<b>87</b>
3.1.	Introduction .....	87
3.2.	Fonctionnement des moteurs d'inférence floue .....	88
3.2.1.	Traitement non séquentiel .....	88
3.2.1.1.	Détermination du nombre de senseurs .....	89
3.2.1.2.	Type de détection .....	89
3.2.1.3.	Type de conflit .....	90
3.2.1.4.	Déroulement d'une simulation non séquentielle type .....	92
3.2.2.	Traitement séquentiel .....	94



3.2.2.1.	Types d'informations analysées .....	95
3.2.2.2.	Type de détections .....	95
3.2.2.3.	Type de conflit .....	95
3.2.2.4.	Critère d'arrêt .....	96
3.2.2.5.	Déroulement d'une simulation séquentielle type .....	97
3.3.	Résultats numériques .....	99
3.3.1.	Comportement non séquentiel des moteurs d'inférence floue ....	99
3.3.1.1.	Évaluation probabiliste .....	100
3.3.1.2.	Évaluation possibiliste .....	104
3.3.1.3.	Analyse des résultats .....	106
3.3.2.	Comportement séquentiel des moteurs d'inférence floue.....	107
3.3.2.1.	Évaluation probabiliste .....	109
3.3.2.2.	Évaluation possibiliste .....	113
3.3.2.3.	Analyse des résultats .....	115
3.3.3.	Impact du nombre de capteurs utilisés .....	117
3.3.3.1.	Effets sur le taux d'erreur.....	118
3.3.3.2.	Effets sur le nombre moyen de plateformes détectées .....	123
3.3.3.3.	Analyse des résultats .....	127
3.3.4.	Analyse globale des résultats.....	129
<b>Conclusion</b>	.....	<b>132</b>
<b>Bibliographie</b>	.....	<b>135</b>

## Table des figures

---

1.2.1	Un ensemble flou $A$ .....	5
1.2.2	Représentation mathématique d'un ensemble classique $A$ .....	7
1.2.3	Représentation mathématique d'un ensemble flou $A$ .....	7
1.3.1	Exemple 1.3.1 : les ensembles flous $A$ et $B$ .....	19
1.5.1	Diagramme fonctionnel d'un moteur d'inférence floue.....	24
1.5.2	Exemple de variable linguistique.....	25
1.5.3	Variable linguistique de sortie <i>accélération</i> .....	29
2.3.1	Distribution de l'attribut <i>vitesse</i> pour la plate-forme numéro trois....	55
2.3.2	Distribution de l'attribut <i>nombre d'émetteurs</i> pour la plate-forme numéro trois.....	55
2.3.3	Arbre de détection pour l'attribut <i>vitesse</i> pour les plates-formes de la base de données simplifiée.....	57
2.3.4	Arbre de détection pour l'attribut <i>nombre d'émetteurs</i> pour les plates- formes de la base de données simplifiée.....	57
2.5.1	Probabilités de détection primaires des plates-formes suite à des rapports sur la <i>vitesse</i> (a), le <i>nombre d'émetteurs</i> (b) et leur <i>type</i> (c) favorisant la plate-forme numéro trois.....	64
2.5.2	Probabilités de détection intermédiaires au niveau de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et finalement globales de	

plates-formes (d) déduites de l'application du modus ponens généralisé avec opérateur d'union floue .....	68
2.5.3 Probabilités de détection intermédiaires au niveau de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et finalement globales de plates-formes (d) déduites de l'application du modus ponens généralisé avec opérateur d'intersection floue .....	69
2.5.4 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du produit avec union floue .....	73
2.5.5 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du produit avec intersection floue.....	74
2.5.6 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du minimum avec union floue .....	77
2.5.7 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du minimum avec intersection floue.....	78
2.5.8 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence bayésien.....	80
2.6.1 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du produit avec union floue et fondement possibiliste.....	84
2.6.2 Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du minimum avec union floue et fondement possibiliste ...	85

2.6.3	Détections intermédiaires de la <i>vitesse</i> (a), du <i>nombre d'émetteurs</i> (b) et de leur <i>type</i> (c) et inférence globale (d) suite à l'emploi du moteur d'inférence bayésien avec fondement possibiliste .....	86
3.3.1	Effets du moteur séquentiel d'inférence floue du produit de fondement probabiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b) .....	109
3.3.2	Effets du moteur séquentiel d'inférence floue du minimum de fondement probabiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b) .....	110
3.3.3	Effets du moteur séquentiel d'inférence floue bayésien de fondement probabiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b) .....	110
3.3.4	Effets du moteur séquentiel d'inférence floue du produit de fondement possibiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b) .....	114
3.3.5	Effets du moteur séquentiel d'inférence floue du minimum de fondement possibiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b) .....	115
3.3.6	Effets du moteur séquentiel d'inférence floue bayésien de fondement possibiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b) .....	116
3.3.7	Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du produit de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b) .....	119
3.3.8	Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du minimum de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b) .....	120

- 3.3.9 Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue bayésien de fondement probabiliste... 120
- 3.3.10 Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du produit de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b) ..... 121
- 3.3.11 Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du minimum de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b) ..... 122
- 3.3.12 Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue bayésien de fondement possibiliste.... 122
- 3.3.13 Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du produit de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b).... 124
- 3.3.14 Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du minimum de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b).... 125
- 3.3.15 Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur d'inférence floue bayésien de fondement probabiliste..... 126
- 3.3.16 Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du produit de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b) .... 127
- 3.3.17 Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du minimum de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b) .... 128

3.3.18 Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur d'inférence floue bayésien de fondement possibiliste .....	129
--	-----

## Liste des tableaux

---

2.3.1	Base de données de plates-formes simplifiée .....	48
2.3.2	Fuzzification de l'attribut <i>vitesse</i> pour les plates-formes de type <i>surface</i>	50
2.3.3	Fuzzification de l'attribut <i>vitesse</i> pour les plates-formes de type <i>air</i> ..	50
2.3.4	Fuzzification de l'attribut <i>nombre d'émetteurs</i> pour tout type de plate- forme.....	51
2.3.5	Fuzzification des attributs <i>vitesse</i> et <i>nombre d'émetteurs</i> pour les plates-formes de la base de données simplifiée .....	52
2.3.6	Exemples de rapport de senseurs concernant la base de données simplifiée.....	58
2.5.1	Résultats de l'identification de la cible numéro trois.....	70
2.5.2	Résultats de l'identification de la cible numéro trois avec moteur d'inférence floue du produit.....	74
2.5.3	Résultats de l'identification de la cible numéro trois avec moteur d'inférence floue du minimum.....	77
2.5.4	Résultats de l'identification de la cible numéro trois avec moteur d'inférence floue bayésien .....	80
2.6.1	Résultats de l'identification de la cible numéro trois avec moteurs d'inférence floue du produit, du minimum et de Bayes avec opérateur d'union floue (deux premiers cas) fondée sur une approche possibiliste	84

3.3.1 Résultats obtenus aux simulations pour les moteurs non séquentiels d'inférence floue du produit, du minimum et de Bayes de fondement probabiliste.....	102
3.3.2 Résultats obtenus aux simulations pour les moteurs non séquentiels d'inférence floue du produit, du minimum et de Bayes de fondement possibiliste .....	105



# INTRODUCTION

---

«Le temps est venu pour une nouvelle branche des mathématiques radicalement différente, les mathématiques du flou non basées sur les théories probabilistes actuelles» déclarait Lofti A. Zadeh en 1962 (voir Wang, 1997, section 1). Par ces quelques mots, il venait de jeter les bases d'une toute nouvelle théorie, appelée logique floue, destinée à l'étude de l'incertitude et de l'imprécision légèrement délaissées par les mathématiques classiques. Après une décennie plus sombre marquée de scepticisme et de doute, elle sut enfin s'imposer grâce à l'apparition des premières applications pratiques. Les fondements de la théorie étaient désormais tangibles. Depuis ce temps, la technique s'est développée dans de nombreux domaines de la science et ses applications foisonnent. Le présent mémoire de maîtrise aborde d'ailleurs l'une d'entre elles : la conception de systèmes experts flous d'identification de cibles.

Le premier chapitre présente les bases de la logique floue, de la théorie des ensembles flous aux opérateurs flous les plus couramment employés en passant par toutes les subtilités des systèmes d'inférence floue, coeur même du présent mémoire. Le second chapitre, quant à lui, marque la véritable entrée en matière de ce mémoire, soit l'application de la logique floue et des moteurs d'inférence floue à l'identification de cibles. En plus de préciser la nature des travaux de recherche, il aborde différentes facettes de l'identification de plates-formes : les attributs et leur modélisation, les techniques de conception de systèmes experts flous ainsi que les fondements du raisonnement approché. Globalement, les efforts

ont été concentrés autour des moteurs d'inférence floue jugés les plus populaires dans le domaine du génie, soit les moteurs du produit et du minimum. Cependant une tangente statistique a également été imprégnée au projet. En effet, un tout nouveau moteur bayésien a été conçu, qui s'inspire grandement du domaine de la statistique. Finalement, les systèmes experts flous sont scrutés sous tous les angles au troisième et dernier chapitre. Une dizaine de simulations variées permettent d'analyser les moteurs d'inférence floue développés selon leur approche probabiliste et possibiliste respective. Une étude comparative détaillée suit afin d'analyser les performances de chacun et d'établir leurs forces et faiblesses. Une autre catégorie de simulations ne concernant plus spécifiquement le comportement des moteurs d'inférence floue, mais plutôt les senseurs, ces appareils d'écoute électronique, est réalisée. L'impact de leur nombre lors d'une simulation sur le rendement d'un système expert flou est alors systématiquement analysé. Tous ces divers résultats sont abondamment illustrés et rigoureusement analysés afin de statuer sur le bien-fondé de l'utilisation de la logique floue en technologie de l'information.

# Chapitre 1

---

## APERÇU THÉORIQUE

### 1.1. INTRODUCTION

Une des différences fondamentales entre la machine et l'être humain réside dans la faculté de celui-ci de raisonner en termes imprécis, vagues et parfois flous. Est-ce une lacune pour la machine? Est-ce un avantage pour l'être humain? D'autres sciences répondront à ces questions. Mais pour les spécialistes de l'intelligence artificielle, humaniser la machine, cela signifie lui apprendre à penser et à réagir comme un être humain, lui apprendre à prévoir l'imprévisible.

Très tôt au début des années 60, les faiblesses des théories mathématiques classiques, pour traiter et analyser les systèmes complexes deviennent flagrantes : le besoin d'une nouvelle théorie mathématique s'impose. En 1962, Lofti A. Zadeh, plus tard reconnu comme le père de la logique floue, déclarait : «Le temps est venu pour une nouvelle branche des mathématiques radicalement différente, les mathématiques du flou non basées sur les théories probabilistes actuelles». Quelques années plus tard, il formalisait ces idées sur les ensembles aux limites imprécises tels le langage, la parole ou la perception humaine dans le premier papier du genre intitulé *Fuzzy Sets* (voir Zadeh, 1965).

Après une décennie plus sombre marquée par le scepticisme et le doute, la logique floue sut enfin s'imposer grâce à l'apparition des premières applications pratiques, les fondements de la théorie étaient désormais posés.

Depuis ce temps, la logique floue est employée dans de nombreux domaines de la science et les applications foisonnent : robot flou, train à commande floue, appareil photographique, etc. Cependant la véritable notoriété scientifique fut acquise en 1992 avec la tenue de la première conférence internationale de l'IEEE sur la logique floue.

Le présent chapitre présente les bases de la logique floue, de la théorie des ensembles flous aux opérateurs flous les plus couramment employés en passant par toutes les subtilités des systèmes d'inférence floue, coeur même du présent mémoire. Pour un aperçu plus détaillé de ces thèmes, veuillez vous référer à Asai *et al.* (1991); Dubois et Prade (1980); Folger et Klir (1988); Kaufmann (1973); OFTA (1994); Wang (1997).

## 1.2. THÉORIE DES ENSEMBLES FLOUS

Soit  $U$ , un référentiel absolu qui contient tous les éléments  $\omega$  d'un contexte ou d'une application particulière ainsi que  $A$ , un sous-ensemble de  $U$ . La fonction d'appartenance de cet ensemble se définit comme suit.

**Définition 1.2.1 (Fonction d'appartenance).** *Une fonction  $\mu_A : U \rightarrow [0,1]$  qui, appliquée à un élément  $\omega \in U$ , définit le degré d'appartenance de  $\omega$  à  $A$  est appelée fonction d'appartenance.*

En mathématiques classiques, les bornes de tous les sous-ensembles sont clairement définis. Aucun doute ne subsiste quant à l'appartenance ou non de chaque élément du référentiel  $U$  à l'ensemble  $A$ . Le degré d'appartenance d'un élément  $\omega$  à l'ensemble classique  $A$  se définit donc comme suit :

$$\mu_A(\omega) = \begin{cases} 1, & \omega \in A \\ 0, & \omega \notin A. \end{cases}$$

Par contre, les ensembles flous se distinguent de ce groupe par leur subjectivité. Ils possèdent des bornes plutôt vagues et mal définies tels que représentés à la figure 1.2.1.

Le langage et les mécanismes de la pensée humaine constituent de bons exemples de tels ensembles. Pensons à la vitesse d'un objet mobile quelconque, les termes *faible*, *modérée* et *élevée* revêtent un caractère incertain et imprécis qui diffère d'un contexte à l'autre.

La théorie des ensembles flous s'efforce donc de déterminer à quel degré chaque élément  $\omega$  appartient à l'ensemble flou  $A$ . Désormais, rien n'est plus absolu, le degré d'appartenance, jusqu'ici contraint à  $\{0,1\}$ , s'étend à  $[0,1]$ . De cette façon, contrairement aux théories classiques, un élément  $\omega$  peut appartenir à la fois à un ensemble flou  $A$  et à son complément  $\bar{A}$ .

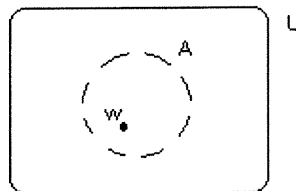


FIG. 1.2.1 -. Un ensemble flou  $A$

### 1.2.1. Concepts de base associés aux ensembles flous

Tout ensemble flou  $A$  se caractérise par deux éléments fondamentaux dénommés support et noyau.

**Définition 1.2.2 (Support).** *Le support regroupe toutes les valeurs possibles, c'est-à-dire tous les éléments  $\omega$  de  $U$  dont le degré d'appartenance à l'ensemble*

flou  $A$  est non nul et se représente comme suit :

$$S(A) = \{ \omega \in U \mid \mu_A(\omega) > 0 \}.$$

**Définition 1.2.3 (Noyau).** *Le noyau désigne, quant à lui, l'ensemble des éléments dont l'appartenance à l'ensemble  $A$  est absolue ou de niveau unitaire. Il est défini par :*

$$\mathring{A} = \{ \omega \in U \mid \mu_A(\omega) = 1 \}.$$

L'ensemble mathématique classique se distingue par un noyau confondu au support (voir Figure 1.2.2) alors que pour l'ensemble flou, il n'en serait être question (voir Figure 1.2.3).

Dans un autre ordre d'idées, les termes *hauteur* et *coupe- $\alpha$*  sont également employés pour caractériser un ensemble flou.

**Définition 1.2.4 (Hauteur).** *La hauteur d'un ensemble flou se définit comme le degré d'appartenance le plus élevé d'un élément  $\omega$  à un ensemble flou  $A$ . Si cette valeur atteint l'unité, l'ensemble flou est alors qualifié de normal.*

**Définition 1.2.5 (Coupe- $\alpha$ ).** *Une coupe- $\alpha$  d'un ensemble flou  $A$  désigne un ensemble flou qui contient tous les éléments d'un référentiel  $U$  dont le degré d'appartenance à l'ensemble  $A$  est supérieur ou égal à  $\alpha$ , soit*

$$A_\alpha = \{ \omega \in U \mid \mu_A(\omega) \geq \alpha \}.$$

Du côté analytique, l'équivalence entre deux ensembles flous  $A$  et  $B$  est vérifiée si chaque élément du référentiel appartient au même degré d'appartenance à  $A$  et à  $B$ , soit

$$A = B \iff \mu_A(\omega) = \mu_B(\omega), \forall \omega \in U,$$

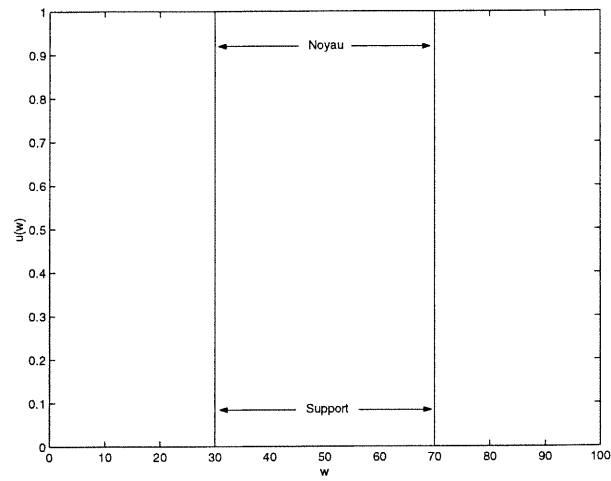


FIG. 1.2.2 -. Représentation mathématique d'un ensemble classique  $A$

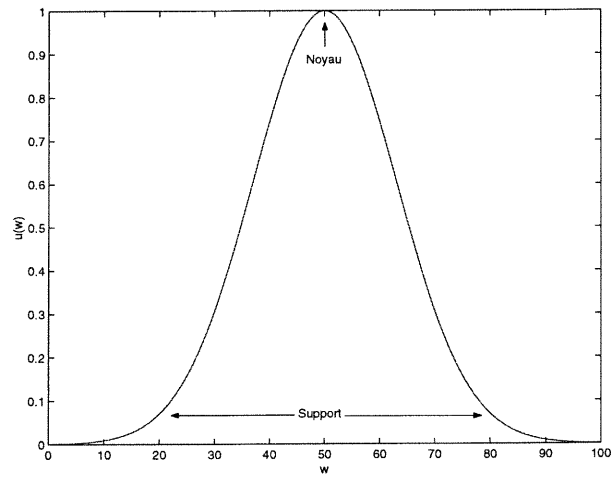


FIG. 1.2.3 -. Représentation mathématique d'un ensemble flou  $A$

et l'inclusion se définit de la manière suivante :

$$A \subset B \iff \mu_A(\omega) \leq \mu_B(\omega), \forall \omega \in U.$$

### 1.3. LOGIQUE FLOUE

Il y a un peu moins de 40 ans, associer le mot flou au mot logique semblait un non-sens. La logique se voulait une conception de la pensée humaine et, croyait-on, ne pouvait porter le qualificatif floue.

Récemment avec l'avancement des mathématiques et de l'étude des mécanismes de la logique, la preuve d'une logique non unique a été démontrée. La pensée humaine, cette superposition de rigueur (nécessairement absolue) et d'intuition (nécessairement floue) se révèle ainsi un pur mécanisme flou.

#### 1.3.1. Propriétés générales

Les grands principes de la logique floue reposent au même titre que les théories classiques sur l'algèbre booléenne. La majeure partie des propriétés et lois des ensembles classiques se voit donc encore vérifiée. Ainsi à l'intérieur du domaine de la logique floue, les lois fondamentales sur les ensembles telles que l'associativité, la commutativité, la distributivité, la loi de la double négation de même que les lois de Morgan restent valides et leur expression inchangée.

Par contre, le fossé entre logique conventionnelle et logique floue se creuse par la perte de deux lois fondamentales dont la validité cesse en logique floue. Désormais, l'union et l'intersection d'un ensemble flou  $A$  et de son complément  $\bar{A}$  sur le référentiel  $U$  respectent les lois suivantes :

$$A \cup \bar{A} \neq U,$$

ainsi que

$$A \cap \bar{A} \neq \emptyset.$$



### 1.3.2. Opérateurs logiques

Une des grandes forces de la logique floue réside dans la variété d'opérateurs disponibles. Particularité intéressante, la définition d'un opérateur n'est plus unique, bien au contraire. Ce phénomène s'explique par des raisons purement pratiques : une définition d'un opérateur pouvant s'avérer plus significative qu'une autre dans une application donnée.

Les règles de construction des opérateurs restent cependant très strictes tel que démontré dans la présente section par les opérateurs de complément, d'union, d'intersection et de moyenne flous.

#### 1.3.2.1. Complément flou

Posons  $c : [0,1] \rightarrow [0,1]$ , une transformation du degré d'appartenance d'un ensemble flou  $A$  en celui de son complément  $\bar{A}$ , c'est-à-dire

$$c[\mu_A(w)] = \mu_{\bar{A}}(w).$$

L'élaboration d'opérateurs de complément flou est régie par deux axiomes :

**Axiome 1.1.**  $c(0) = 1$  et  $c(1) = 0$  (*conditions limites*),

**Axiome 1.2.** Pour tout  $\mu_A(w)$  et  $\mu_B(w) \in [0,1]$ , si  $\mu_A(w) < \mu_B(w)$ , alors  $c[\mu_A(w)] \geq c[\mu_B(w)]$  (*condition de non-accroissement*).

Comme chaque élément  $\omega$  d'un ensemble flou  $A$  peut appartenir à la fois à  $A$  et à son complément  $\bar{A}$ , ces axiomes mènent vers la définition suivante.

**Définition 1.3.1 (Complément flou).** Une transformation  $c : [0,1] \rightarrow [0,1]$  qui satisfait les axiomes 1.1 et 1.2 est appelée complément flou.

Trois principaux types d'opérateurs de complément flou existent dont le plus couramment utilisé selon Wang (1997, section 3), l'opérateur de complément flou

standard (voir équation (1.3.1)), ainsi que les opérateurs de la classe de Sugeno (voir équation (1.3.2)) et de celle de Yager (voir équation (1.3.3)), représentés respectivement par :

$$\mu_{\bar{A}}(\omega) = 1 - \mu_A(\omega), \quad (1.3.1)$$

$$c_\lambda(\mu_A(\omega)) = \frac{(1 - \mu_A(\omega))}{(1 + \lambda\mu_A(\omega))}, \quad (1.3.2)$$

où  $\lambda \in (-1, \infty)$ ,

$$c_z(\mu_A(\omega)) = (1 - \mu_A(\omega)^z)^{1/z}, \quad (1.3.3)$$

où  $z \in (0, \infty)$ .

### 1.3.2.2. Union floue : norme S

Posons  $s : [0,1] \times [0,1] \rightarrow [0,1]$ , une transformation des fonctions d'appartenance de deux ensembles flous  $A$  et  $B$  en celle de leur union, c'est-à-dire

$$s[\mu_A(w), \mu_B(w)] = \mu_{A \cup B}(w).$$

Afin de parvenir à définir adéquatement la norme S ou plus spécifiquement l'union floue (voir définition 1.3.2), quatre axiomes doivent être énoncés, soit :

**Axiome 1.3.**  $s(1,1) = 1$  et  $s(0, \mu_A(w)) = s(\mu_A(w), 0) = \mu_A(w)$  (conditions limites),

**Axiome 1.4.**  $s(\mu_A(w), \mu_B(w)) = s(\mu_B(w), \mu_A(w))$  (condition de commutativité),

**Axiome 1.5.** Si  $\mu_A(w) \leq \mu_A(w')$  et  $\mu_B(w) \leq \mu_B(w')$ , alors  $s(\mu_A(w), \mu_B(w)) \leq s(\mu_A(w'), \mu_B(w'))$  (condition de non-décroissance),

**Axiome 1.6.**  $s(s(\mu_A(w), \mu_B(w)), \mu_C(w)) = s(\mu_A(w), s(\mu_B(w), \mu_C(w)))$  (condition d'associativité).

**Définition 1.3.2 (Union floue).** *Une transformation  $s : [0,1] \times [0,1] \rightarrow [0,1]$  qui satisfait les axiomes 1.3 à 1.6 est appelée opérateur d'union floue.*

Une foule d'opérateurs d'union floue divers existent, mais sont rassemblés ci-dessous trois des opérateurs les plus populaires selon Wang (1997, section 3) dont le maximum tiré des mathématiques classiques (voir équation (1.3.4)), la somme algébrique (voir équation (1.3.5)) et les opérateurs de la classe de Yager (voir équation (1.3.6)). Ils s'expriment respectivement sous la forme :

$$s_{\max}(\mu_A(\omega), \mu_B(\omega)) = \max(\mu_A(\omega), \mu_B(\omega)), \quad (1.3.4)$$

$$s_{sa}(\mu_A(\omega), \mu_B(\omega)) = \mu_A(\omega) + \mu_B(\omega) - \mu_A(\omega) \times \mu_B(\omega), \quad (1.3.5)$$

ainsi que

$$s_z(\mu_A(\omega), \mu_B(\omega)) = \min(1, (\mu_A(\omega)^z + \mu_B(\omega)^z)^{1/z}), \quad (1.3.6)$$

où  $z \in (0, \infty)$ .

### 1.3.2.3. Intersection floue : norme $T$

Posons  $t : [0,1] \times [0,1] \rightarrow [0,1]$ , une transformation des fonctions d'appartenance de deux ensembles flous  $A$  et  $B$  en celle de leur intersection, c'est-à-dire

$$t[\mu_A(w), \mu_B(w)] = \mu_{A \cap B}(w).$$

L'intersection floue ressemble fort à son prédécesseur, car tous les axiomes de l'union floue conservent leur validité à l'exception des conditions limites de l'axiome 1.3 qui se voient remplacées par :

**Axiome 1.7.**  $t(0,0) = 0$  et  $t(1, \mu_A(w)) = t(\mu_A(w), 1) = \mu_A(w)$  (conditions limites).

Tous ces axiomes mènent directement à la définition suivante :

**Définition 1.3.3 (Intersection floue).** *Une transformation*

$t : [0,1] \times [0,1] \rightarrow [0,1]$  *qui satisfait les axiomes 1.4, 1.5, 1.6 ainsi que l'axiome 1.7 est appelée opérateur d'intersection floue.*

La famille des opérateurs d'intersection floue se composent également d'un large éventail d'opérateurs tels que le minimum (voir équation (1.3.7)), le produit algébrique (voir équation (1.3.8)) et les opérateurs de la classe de Yager (voir équation (1.3.9)). Les voici présentés ci-dessous :

$$t_{\min}(\mu_A(\omega), \mu_B(\omega)) = \min(\mu_A(\omega), \mu_B(\omega)), \quad (1.3.7)$$

$$t_{pa}(\mu_A(\omega), \mu_B(\omega)) = \mu_A(\omega) \times \mu_B(\omega), \quad (1.3.8)$$

$$t_z(\mu_A(\omega), \mu_B(\omega)) = 1 - \min[1, ((1 - \mu_A(\omega))^z + (1 - \mu_B(\omega))^z)^{1/z}], \quad (1.3.9)$$

où  $z \in (0, \infty)$ .

#### 1.3.2.4. Moyenne floue

En analysant attentivement la famille des opérateurs d'agrégation floue, on remarque que les opérateurs d'union floue retournent des degrés d'appartenance se situant généralement dans la partie supérieure de l'intervalle  $[0,1]$ . Les valeurs de retour des opérateurs d'intersection floue, quant à elles, se situent davantage dans la partie inférieure du même intervalle.

Pour combler l'espace béant laissé entre ces deux catégories, un dernier type d'opérateurs, nommé moyenne floue, fut élaboré. Celui-ci se retrouve sous diverses formes dont les moyennes max-min (voir équation (1.3.10)) et moyenne généralisée

(voir équation (1.3.11)) représentées respectivement par :

$$v_\lambda(\mu_A(\omega), \mu_B(\omega)) = \lambda \times \max(\mu_A(\omega), \mu_B(\omega)) + (1 - \lambda) \min(\mu_A(\omega), \mu_B(\omega)), \quad (1.3.10)$$

où  $\lambda \in [0,1]$ , ainsi que

$$v_\alpha(\mu_A(\omega), \mu_B(\omega)) = \left( \frac{\mu_A(\omega)^\alpha + \mu_B(\omega)^\alpha}{2} \right)^{1/\alpha}, \quad (1.3.11)$$

où  $\alpha \in \mathbb{R}$  et  $\alpha \neq 0$ .

### 1.3.3. Relation floue

Lorsque le référentiel se compose de plus d'un ensemble flou, la combinaison de toute la masse d'informations vagues s'avère de complexité élevée. L'usage de fonctions d'appartenance jointes, aussi appelées relations floues, permet alors de solutionner ce problème.

Une telle relation constitue, en fait, un ensemble flou contenu dans le produit cartésien de tous les ensembles du référentiel. Elle se formalise de la façon suivante :

**Définition 1.3.4 (Relation floue).** *Soit  $U = A_1 \times \dots \times A_n$ , le produit cartésien des ensembles flous  $A_1, \dots, A_n$ . La relation floue  $Q$  dans  $U$  s'exprime comme*

$$Q = \{\mu_Q(\omega_1, \dots, \omega_n) \mid (\omega_1, \dots, \omega_n) \in U\},$$

où  $\mu_Q : A_1 \times \dots \times A_n \rightarrow [0,1]$ .

À titre d'exemple, les opérateurs flous présentés dans les sous-sections précédentes représentent tous des relations binaires floues.

### 1.3.4. Projection floue et extension cylindrique

Puisque les relations floues se composent de plusieurs ensembles flous, la nécessité d'adapter les concepts classiques de projection et d'extension s'impose. Des concepts jumeaux, appelés projection floue et extension cylindrique, s'avèrent donc essentiels.

**Définition 1.3.5 (Projection floue).** *Soit  $Q$ , une relation floue dans  $A_1 \times \cdots \times A_n$  et  $i_1, \dots, i_k$ , une sous-suite de  $1, 2, \dots, n$ , alors la projection de  $Q$  sur  $A_{i_1} \times \cdots \times A_{i_k}$  devient une relation floue  $Q_p$  dans  $A_{i_1} \times \cdots \times A_{i_k}$  définie par la fonction d'appartenance :*

$$\mu_{Q_p}(\omega_{i_1}, \dots, \omega_{i_k}) = \max_{\omega_{j_1} \in A_{j_1}, \dots, \omega_{j_{(n-k)}} \in A_{j_{(n-k)}}} \mu_Q(\omega_1, \dots, \omega_n),$$

où  $\omega_{j_1}, \dots, \omega_{j_{(n-k)}}$  représente le complément de  $\omega_{i_1}, \dots, \omega_{i_k}$  par rapport à  $\omega_1, \dots, \omega_n$ .

Par exemple, si  $Q$  représente une relation floue binaire définie dans le produit cartésien des référentiels  $U$  et  $V$ , alors la projection  $Q$  sur  $U$ , appelée  $Q_1$ , se définit comme

$$\mu_{Q_1} = \max_{y \in V} \mu_Q(x, y).$$

Contrairement à la projection floue qui contraint la relation floue à un sous-espace, l'extension cylindrique l'étend plutôt à tout l'espace vectoriel.

**Définition 1.3.6 (Extension cylindrique).** *Soit  $Q_P$ , une relation floue dans  $A_{i_1} \times \cdots \times A_{i_k}$  et  $i_1, \dots, i_k$ , une sous-suite de  $1, 2, \dots, n$ , alors l'extension cylindrique de  $Q_P$  à  $A_1 \times \cdots \times A_n$  s'exprime comme une relation floue  $Q_{PE}$  dans  $A_1 \times \cdots \times A_n$  définie comme*

$$\mu_{Q_{PE}}(\omega_1, \dots, \omega_n) = \mu_{Q_P}(\omega_{i_1}, \dots, \omega_{i_k}).$$

Par exemple, si  $Q_1$  représente un ensemble flou défini dans le référentiel  $U$ , alors l'extension cylindrique de  $Q_1$  au produit cartésien  $U \times V$  est une relation binaire floue  $Q_{1E}$  dans  $U \times V$  définie par :

$$\mu_{Q_{1E}}(x,y) = \mu_{Q_1}(x).$$

### 1.3.5. Composition de relations floues

La composition de relations floues généralise la composition de relations mathématiques classiques à la théorie des ensembles flous.

**Définition 1.3.7 (Composition floue).** *Soit  $A, B$  et  $C$ , trois ensembles flous compris dans un référentiel  $U$  et  $P(A, B)$  et  $Q(B, C)$ , deux relations binaires floues qui partagent le même ensemble flou  $B$ , alors  $P \circ Q$  représente la composition floue de  $P(A, B)$  et  $Q(B, C)$  si et seulement si*

$$\mu_{P \circ Q}(x, z) = \max_{y \in B} t[\mu_P(x, y), \mu_Q(y, z)],$$

pour tout  $(x, z) \in A \times C$  et où  $t$  représente une norme  $T$  (intersection floue) quelconque.

Les types de compositions floues les plus fréquemment employés selon Wang (1997, section 7) sont la composition max-min de relations floues ainsi que la composition max-produit dans lesquelles l'opérateur d'intersection floue correspond respectivement au minimum (voir équation (1.3.7)) et au produit algébrique (voir équation (1.3.8)).

### 1.3.6. Principe d'extension

La logique floue permet également de résoudre des équations de la forme  $y = f(x)$  grâce à une technique particulière nommée principe d'extension.

**Définition 1.3.8 (Le principe d'extension).** *Soit  $X$  et  $Y$ , deux référentiels absolus, alors une fonction  $f : X \rightarrow Y$  peut être mathématiquement étendue pour jumeler l'ensemble flou  $A$  de  $X$  à l'ensemble  $f(A)$  de  $Y$  de la manière suivante :*

$$\mu_{f(A)}(y) = \sup_{\{x|y=f(x)\}} \{\mu_A(x)\}.$$

Pour un  $y$  donné, le côté droit de l'équation signifie que l'on recherche la valeur maximale de  $\mu_A(x)$  sur les  $x$  telle que  $y = f(x)$ . La solution peut s'avérer nulle si  $f^{-1}(y)$  correspond à l'ensemble vide.

Voici maintenant à titre de conclusion un exemple traitant de la plupart des notions abordées dans cette section.

**Exemple 1.3.1.** *Soit  $A$  et  $B$ , deux ensembles flous définis sur un référentiel  $U$  représentant des vitesses s'étalant de 0 à 100 km/h. Ces ensembles flous qui qualifient tous les éléments du référentiel sont représentés respectivement par les fonctions d'appartenance suivantes :*

$$\mu_A(\omega) = \begin{cases} 0,034\omega, & \text{si } \omega \in [0,25), \\ -0,034(\omega - 25) + 0,85, & \text{si } \omega \in [25,50], \\ 0, & \text{sinon,} \end{cases}$$



$$\mu_B(\omega) = \begin{cases} 0,0571(\omega - 40), & \text{si } \omega \in [40,57.5), \\ 1, & \text{si } \omega \in [57.5,62.5], \\ -0,0571(\omega - 62,5) + 1, & \text{si } \omega \in [62.5,80), \\ 0, & \text{sinon,} \end{cases}$$

qui sont représentées à la figure 1.3.1.

L'analyse de ces deux ensembles permet de constater les caractéristiques suivantes :

- Tout d'abord, le support de l'ensemble  $A$  s'étend de 0 à 50 km/h et son noyau est nul, aucun élément ne possède un degré unitaire d'appartenance à  $A$ . Pour  $B$ , les choses sont quelque peu différentes. Son support comprend tous les éléments compris entre 40 et 80 km/h et son noyau se compose de ceux allant de 57,5 à 62,5 km/h.
- Le premier ensemble possède une hauteur de 0,85 tandis que le second est qualifié de normal par la présence d'éléments de degré d'appartenance unitaire.
- Il est possible de définir une coupe-0,9 sur l'ensemble  $B$ , par exemple, par la relation suivante  $B_{0,9} = \{\omega \in U \mid \mu_B(\omega) \geq 0,9\}$ . Cette coupe représente un ensemble flou dont tous les éléments sont de degré d'appartenance nul à l'exception de ceux compris entre 55 et 65 km/h. Ces derniers sont inclus à leur degré d'appartenance à l'ensemble  $B$  respectif, c'est-à-dire  $\mu_{B_{0,9}}(\omega) = \mu_B(\omega)\mathbb{I}_{(55,65)}(\omega)$  où  $\mathbb{I}_{(55,65)}(\omega) = 1$  si  $55 \leq \omega \leq 65$  et 0 sinon.
- Ces ensembles ne sont, en aucun point, équivalents ou inclus l'un dans l'autre.

- Le complément de l'ensemble  $A$  (voir équation (1.3.1)), par exemple, se représente simplement par une fonction triangle inversée représentée par :

$$\mu_{\bar{A}}(\omega) = \begin{cases} 1 - 0,034\omega, & \text{si } \omega \in [0,25), \\ 0,034(\omega - 25) + 0,15, & \text{si } \omega \in [25,50], \\ 1, & \text{sinon.} \end{cases}$$

- Finalement, certaines relations floues peuvent être évaluées notamment l'union et l'intersection floues de ces deux ensembles. Prenons l'opérateur du maximum dans le premier cas (voir équation (1.3.4)) et du minimum dans le second (voir équation (1.3.7)), on obtient les fonctions d'appartenance suivantes :

$$\mu_{A \cup B}(\omega) = \begin{cases} 0,034\omega, & \text{si } \omega \in [0,25), \\ -0,034(\omega - 25) + 0,85, & \text{si } \omega \in [25,43.7), \\ 0,0571(\omega - 40), & \text{si } \omega \in [43.7,57.5), \\ 1, & \text{si } \omega \in [57.5,62.5), \\ -0,0571(\omega - 62,5) + 1, & \text{si } \omega \in [62.5,80], \\ 0, & \text{sinon,} \end{cases}$$

$$\mu_{A \cap B}(\omega) = \begin{cases} 0,0571(\omega - 40), & \text{si } \omega \in [40,43.7), \\ -0,034(\omega - 25) + 0,85, & \text{si } \omega \in [43.7,50), \\ 0, & \text{sinon.} \end{cases}$$

Pour de plus amples informations sur l'univers de la logique floue, veuillez consulter les ouvrages de Wang (1997); Asai *et al.* (1991).

#### 1.4. THÉORIE DES POSSIBILITÉS

La théorie des possibilités, une extension de la logique floue, se compare avantageusement à la théorie des probabilités en mathématiques classiques. Elle

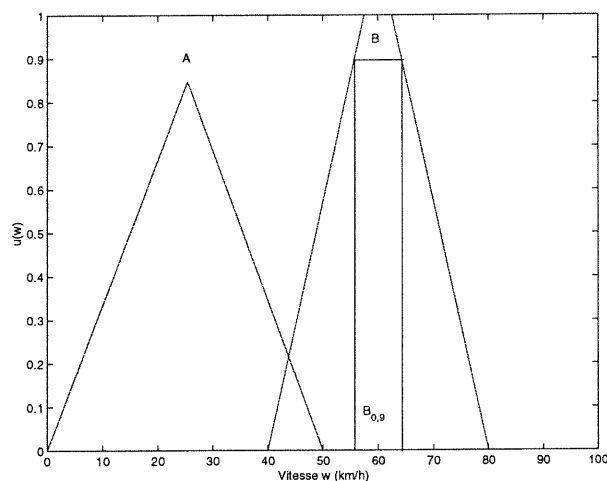


FIG. 1.3.1 — Exemple 1.3.1 : les ensembles flous A et B

offre cependant une vision fort différente du traitement de l'imprécision et de l'incertitude.

On qualifie une information d'imprécise si elle est démontrée insuffisante et incomplète pour répondre clairement à une question spécifique. L'imprécision réfère directement au contenu de cette information. L'incertitude, quant à elle, réfère davantage à la capacité de croire en cette information, en sa validité. Elle se veut une conséquence directe de l'imprécision. La présente section décrit en profondeur les subtilités de cette théorie et démontrent les écarts entre *possibilité* et *probabilité* à la lumière des études de Borne et Delmotte (1998); Dubois et Prade (1988).

### 1.4.1. Principe général

La théorie des possibilités traite d'abord de l'imprécision. Elle suppose que chaque information est graduelle, c'est-à-dire que chaque variable possède des valeurs plus ou moins possibles.

**Définition 1.4.1 (Distribution de possibilités).** *Soit l'ensemble flou  $A$  défini sur un référentiel absolu  $U$ , alors une fonction  $\pi_A(\omega)$  qui assigne un degré de possibilité à chaque élément  $\omega$  du référentiel est appelée distribution de possibilités de l'ensemble flou  $A$ . De plus, tout référentiel étant absolu, il existe toujours au moins un élément  $\omega$  de  $U$  qui possède un degré unitaire de possibilité, c'est-à-dire*

$$\exists \omega \in U \text{ tel que } \pi_A(\omega) = 1.$$

En d'autres termes, la distribution de possibilités  $\pi_A(\omega)$  ne s'avère être autre chose que la fonction d'appartenance  $\mu_A(\omega)$  décrit précédemment, mais à laquelle s'ajoute une condition supplémentaire de normalisation.

### 1.4.2. Traitement de l'incertitude

Puisque la théorie des possibilités modélise l'imprécision, certains degrés d'incertitude peuvent être déduits. Ils sont obtenus à l'aide de deux différentes mesures d'incertitude basées sur les distributions de possibilités, soit  $\Pi$  et  $N$  appelées respectivement mesure de possibilité et de nécessité des ensembles flous.

**Définition 1.4.2 (Mesure de possibilité).** *Soit l'ensemble flou  $A$  défini sur un référentiel absolu  $U$ , la mesure de possibilité de l'ensemble  $A$  se définit comme*

$$\Pi(A) = \sup_{\omega \in A} \{\pi_A(\omega)\}.$$

**Définition 1.4.3 (Mesure de nécessité).** Soit l'ensemble flou  $A$  défini sur un référentiel absolu  $U$ , la mesure de nécessité de l'ensemble  $A$  se définit comme

$$N(A) = \inf_{\omega \notin A} \{1 - \pi_A(\omega)\}.$$

Ces mesures sont liées par une *relation de dualité* qui stipule qu'un événement  $A$  très certain ( $N(A) \approx 1$ ) est possible à coup sûr ( $\Pi(A) \approx 1$ ). Elle s'exprime ainsi :

$$\Pi(A) = 1 - N(\bar{A}).$$

### 1.4.3. Probabilité versus possibilité

La théorie des possibilités et le monde de la logique floue diffèrent de la théorie des probabilités classiques à plusieurs niveaux, en voici les principaux :

- une mesure de probabilité assigne un degré à chaque sous-ensemble non flou d'un référentiel absolu tandis qu'en logique floue, la fonction d'appartenance assigne un degré à chaque élément d'un référentiel absolu dans lequel les ensembles sont mal définis ou flous.
- en logique probabiliste, les ensembles sont mutuellement exclusifs contrairement à la logique floue où ceux-ci représentent davantage des collections non mutuellement exclusives d'éléments.
- la somme des probabilités de tous les éléments du référentiel doit égaler l'unité, mais non celle des possibilités. Dans ce dernier cas cependant, au moins un des éléments présente un degré de possibilité unitaire.
- les mesures de possibilité et de nécessité contrastent également avec la mesure de probabilité : tandis que cette dernière s'applique à des résultats

précis, mais contradictoires, les deux premières s'avèrent des outils essentiels pour synthétiser les connaissances imprécises, mais cohérentes.

- la définition du complément diffère dans chacun des cas. La probabilité d'un événement définit complètement celle de son complément contrairement à la possibilité. Cette situation se définit comme suit :

$$P(A) + P(\bar{A}) = 1.$$

Cependant,

$$N(A) + N(\bar{A}) \leq 1,$$

de même que

$$\Pi(A) + \Pi(\bar{A}) \geq 1.$$

Les mesures de possibilité et de nécessité sont souvent perçues comme des bornes pour la mesure de probabilité, c'est-à-dire

$$N(A) \leq P(A) \leq \Pi(A).$$

En résumé, la théorie de probabilité semble, en quelque sorte, trop riche ou trop stricte pour représenter l'ignorance contrairement à celle des possibilités. Cette dernière s'avérant pourtant inapte à modéliser le hasard.

**Exemple 1.4.1.** *Reprenons l'exemple 1.3.1 de la section précédente à la lumière de la théorie des possibilités.*

*L'ensemble flou  $B_{0,9}$  défini par une coupe de niveau 0,9 sur l'ensemble  $B$  possède un degré de possibilité  $\Pi(B_{0,9})$ , défini comme  $\sup_{\omega \in B_{0,9}} \{\pi_{B_{0,9}}(\omega)\}$ , égale à l'unité et celui de nécessité  $N(B_{0,9})$ , c'est-à-dire  $\inf_{\omega \notin B_{0,9}} \{1 - \pi_{B_{0,9}}(\omega)\}$ , de degré 0,1. Pour l'ensemble  $B$ , la possibilité et la nécessité sont unitaires. Les degrés de*

*possibilité et de nécessité de l'ensemble A, quant à eux, ne peuvent être évalués sans une étape antérieure de normalisation.*

## 1.5. INFÉRENCE FLOUE

Les moteurs d'inférence sont nés des récents besoins en ingénierie pour des systèmes de contrôle aptes à renforcer la prise de décision humaine. Ils représentent une fusion presque parfaite des connaissances humaines, des modèles mathématiques et des mesures de capteurs. Parmi les nombreuses applications de la logique floue, ces systèmes constituent le domaine qui suscite la curiosité et la controverse, mais également s'avère être le champ d'applications le plus actif à travers le monde.

Les moteurs d'inférence conventionnels étudient leur environnement immédiat, analysent la masse d'informations recueillies en fonction d'un but précis et statuent sur la meilleure décision à prendre. Les systèmes flous, quant à eux, se démarquent de ce groupe par le choix du domaine d'analyse. L'univers des probabilités classiques est légèrement délaissé pour faire place à un monde où l'assouplissement des règles de calcul est de mise : la logique floue. Moins de certitude et plus de relativité.

Un moteur d'inférence floue, tel que représenté à la figure 1.5.1, ressemble à une véritable boîte noire. Les données du monde réel présentes à l'entrée du système, sont transformées dans le monde de la logique floue, c'est-à-dire *fuzzifiées*, puis analysées à l'aide d'une base de règles mathématiques et d'opérateurs divers. Le moteur d'inférence prend alors la décision la plus éclairée, toujours dans le domaine flou, et la reconvertit par la suite en donnée réelle grâce à une opération de *défuzzification*.

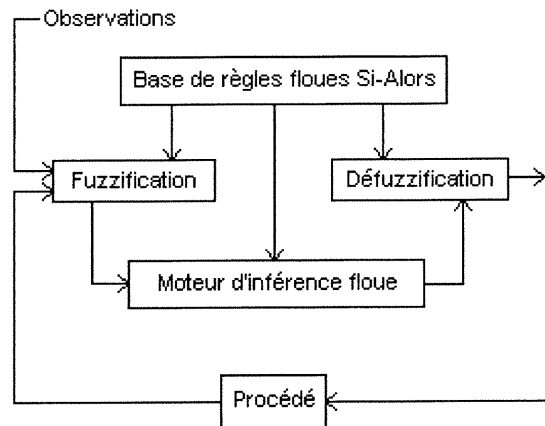


FIG. 1.5.1 —. *Diagramme fonctionnel d'un moteur d'inférence floue*

Les moteurs d'inférence floue représentent, en quelque sorte, l'intégration de tous les différents éléments constituant la logique floue et feront l'objet d'une analyse détaillée dans la présente section.

### 1.5.1. Variables linguistiques

Dans le langage naturel, les mots remplacent souvent les nombres pour qualifier une variable, un élément ou même un état. On emploie les termes *faible*, *modérée* et *élevée*, plutôt qu'une valeur numérique, pour décrire la vitesse d'une automobile.

Les mots, fondement de la représentation de la connaissance humaine, représentent des entités difficilement exprimables en mathématiques classiques. Par contre, la logique floue jouit d'un avantage indéniable : elle constitue une représentation mathématique d'étiquettes linguistiques.

**Définition 1.5.1 (Variable linguistique).** *Une variable qui prend des mots du langage courant comme valeurs est appelée variable linguistique. Chacun des*



mots la qualifiant est alors caractérisé par un ensemble flou défini sur le référentiel dans lequel elle est définie.

**Exemple 1.5.1.** À titre d'exemple, la figure 1.5.2 représente la variable linguistique vitesse qualifiée par les termes faible, modérée et élevée, tous représentés sous forme d'ensembles flous sur un référentiel s'étalant de 0 à 100 km/h.

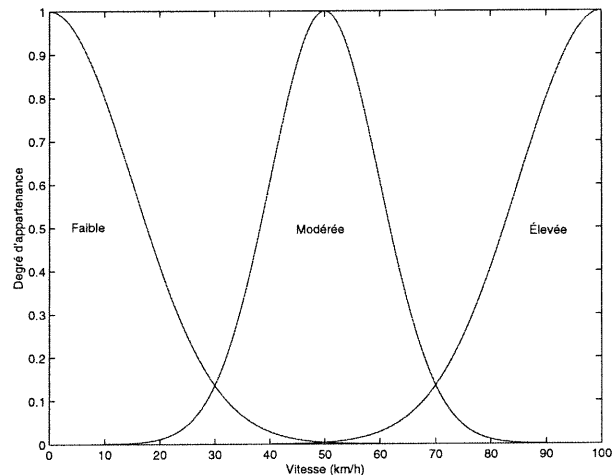


FIG. 1.5.2 — Exemple de variable linguistique

De façon plus détaillée, la valeur d'une variable linguistique est généralement représentée par un terme composite  $x = x_1x_2x_3 \cdots x_n$  constitué d'une série de termes atomiques  $x_1, x_2, \dots, x_n$ .

**Définition 1.5.2 (Termes atomiques).** Les termes atomiques composant une variable linguistique sont classés en trois groupes distincts dont les termes primaires (les mots du langage courant comme faible ou modérée tirés de l'exemple précédent), les connecteurs logiques (les normes floues  $S$  (union) et  $T$  (intersection) ainsi que le complément flou) et les prédicats, des modificateurs linguistiques qui affectent la fonction d'appartenance.

Les prédicats font légion en logique floue, mais les deux plus employés selon Asai *et al.* (1991, section 2), restent *très* (voir équation (1.5.1)) et *plus ou moins* (voir équation (1.5.2)) définis sur l'ensemble flou  $A$ , par exemple, par :

$$\mu_{très A}(x) = [\mu_A(x)]^2, \quad (1.5.1)$$

ainsi que

$$\mu_{plus\ ou\ moins\ A}(x) = [\mu_A(x)]^{1/2}. \quad (1.5.2)$$

#### 1.5.1.1. Conception des ensembles flous

Différents moyens existent pour représenter un mot en terme d'ensembles flous ou de fonctions d'appartenance. Deux approches prévalent cependant selon Dishkant (1981); Wang (1997, section 2). La première approche consiste à construire un ensemble flou à partir de données fournies par un expert. Son expertise est alors jugée sans faille. La seconde consiste plutôt à bâtir la fonction d'appartenance à l'aide de données statistiques généralement fournies par des capteurs aussi appelés senseurs. Les modèles d'ensembles flous les plus employés restent cependant les modèles simples gaussien, triangulaire et trapézoïdal.

#### 1.5.2. Base de règles floues *si-alors*

La base de règles floues *si-alors*, le coeur même des moteurs d'inférence floue, représente l'ensemble des connaissances humaines sur un problème particulier. Une règle floue *si-alors* quelconque, notée  $Ru^{(l)}$ , similaire à un énoncé conditionnel de variables linguistiques, s'exprime ainsi

$$Ru^{(l)} : Si < proposition floue > , alors < proposition floue > .$$

La proposition floue permet de représenter en termes logiques l'état d'une variable linguistique. Les versions atomique et composée constituent les deux

types de propositions floues usuelles. La première, comme son nom l'indique, se compose d'un seul terme et s'exprime simplement de la façon suivante :

$$x \text{ est } A,$$

où  $x$  représente une variable linguistique et  $A$ , la valeur ou l'état de cette variable.

**Exemple 1.5.2.** *La proposition la vitesse est faible tirée de l'exemple précédent 1.5.1 stipule que la variable linguistique vitesse se trouve dans l'état faible et se représente par  $\mu_{\text{faible}}(\omega)$ . Cette fonction d'appartenance (voir Figure 1.5.2) établit à quel degré chacune des vitesses possibles  $\omega$  comprises entre 0 et 100 km/h appartient à l'ensemble flou faible.*

La proposition composée, quant à elle, se veut un amalgame de diverses propositions floues assemblées par des connecteurs logiques du type norme S, T ou complément flou. Cette version de la proposition floue correspond à une relation floue de plusieurs variables linguistiques et prône l'utilisation de fonctions d'appartenance jointes. À titre d'exemple, la proposition composée suivante unit les variables  $x_1$  et  $x_2$  de valeur respective  $A_1$  et  $A_2$  par un opérateur d'intersection floue :

$$(x_1 \text{ est } A_1) \text{ et } (x_2 \text{ est } A_2).$$

#### 1.5.2.1. *Interprétation des règles floues si-alors*

Une règle floue *si-alors* se définit comme une implication logique de table de vérité standard, mais dans laquelle les propositions booléennes sont remplacées par des propositions floues. L'implication de deux propositions floues composées  $PF_1$  et  $PF_2$ ,  $PF_1 \rightarrow PF_2$ , se définit généralement comme

$$\text{Si } PF_1, \text{ alors } PF_2 : (PF_1 \wedge PF_2) \vee \overline{PF_1}.$$

où  $\vee$  représente un opérateur d'union floue,  $\wedge$ , un opérateur d'intersection floue et  $\bar{\phantom{x}}$ , un opérateur de complément flou.

La gamme d'interprétations possibles d'une règle floue *si-alors* se veut très diversifiée. Ces interprétations se présentent sous forme de relations floues,  $\mu_Q$ , intégrant différents types d'opérateurs d'union, d'intersection et de complément flous. Six différentes implications sont jusqu'à présent répertoriées : les implications de Dienes-Rescher (voir équation (1.5.3)), de Lukasiewicz (voir équation (1.5.4)), de Zadeh (voir équation (1.5.5)), de Gödel (voir équation (1.5.6)) et finalement les implications de Mamdani du minimum (voir équation (1.5.7)) et du produit (voir équation (1.5.8)). Elles sont données par :

$$\mu_{Q_{RD}}(x,y) = \max[1 - \mu_{PF_1}(x), \mu_{PF_2}(y)], \quad (1.5.3)$$

$$\mu_{Q_L}(x,y) = \min[1, 1 - \mu_{PF_1}(x) + \mu_{PF_2}(y)], \quad (1.5.4)$$

$$\mu_{Q_Z}(x,y) = \max[\min(\mu_{PF_1}(x), \mu_{PF_2}(y)), 1 - \mu_{PF_1}(x)], \quad (1.5.5)$$

$$\mu_{Q_G}(x,y) = \begin{cases} 1, & \text{si } \mu_{PF_1}(x) \leq \mu_{PF_2}(y) \\ \mu_{PF_2}(y), & \text{ailleurs,} \end{cases} \quad (1.5.6)$$

$$\mu_{Q_{MM}}(x,y) = \min[\mu_{PF_1}(x), \mu_{PF_2}(y)], \quad (1.5.7)$$

et finalement

$$\mu_{Q_{MP}}(x,y) = \mu_{PF_1}(x)\mu_{PF_2}(y). \quad (1.5.8)$$

Les implications de Mamdani,  $\mu_{Q_{MM}}$  et  $\mu_{Q_{MP}}$ , sont, selon Wang (1997, section 5), celles qui jouissent de la plus grande popularité dans la conception de moteurs d'inférence floue. Leur portée plutôt locale ne serait pas étrangère à ce phénomène.

**Exemple 1.5.3.** *Suite à l'exemple précédent 1.5.2, définissons la règle si-alors suivante :*

$A^1$  : si la vitesse est faible, alors l'accélération est élevée,

qui stipule que si la variable linguistique vitesse se trouve dans l'état faible, alors l'état de la variable accélération s'avère être élevée. Cette dernière variable linguistique est présentée à la figure 1.5.3. Si on utilise maintenant une implication de Mamdani du produit, cette expression se représente par :

$$\mu_{Q_{MP}}(x,y) = \mu_{faible}(x) \times \mu_{élevée}(y).$$

où  $x$  et  $y$  représentent respectivement les observations de vitesse reçues et les différentes valeurs d'accélération. En résumé, pour une observation de vitesse donnée, tous les points du référentiel accélération doivent être évalués pour ainsi obtenir le nouvel ensemble flou accélération élevée.

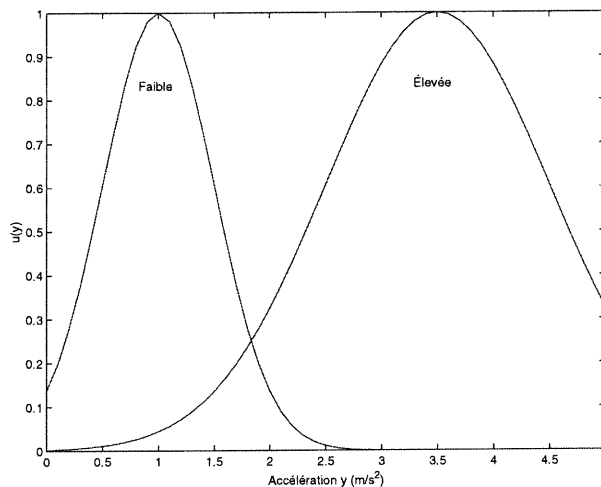


FIG. 1.5.3 -- Variable linguistique de sortie accélération

### 1.5.2.2. Propriétés de la base de règles *si-alors*

La base de règles floues consiste en un ensemble de règles *si-alors*. Afin d'assurer la cohésion des règles et le bon fonctionnement du moteur d'inférence flou, certaines conditions doivent être respectées : les propriétés d'état complet, de

cohérence et de continuité. Voici les définitions de ces propriétés traitant de  $n$  variables linguistiques  $\omega_1, \omega_2, \dots, \omega_n$  différentes dans le référentiel absolu  $U$ .

**Définition 1.5.3 (État complet).** *Une base de règles floues si-alors est qualifiée d'état complet si pour tout  $\omega_i \in U$ , il existe au moins une règle floue  $l$  qui satisfait*

$$\mu_{A_i^l}(\omega_i) \neq 0,$$

*pour tout  $i = 1, 2, \dots, n$  et où  $A_i^l$  représente les différents termes primaires ou les ensembles flous des variables linguistiques d'entrée contenues dans la partie si de chaque règle  $l$  et  $\omega_i$  désigne la donnée réelle d'entrée qui lui est associée.*

**Exemple 1.5.4.** *Dans l'exemple précédent 1.5.3, la règle si-alors non composée  $A^1$  comporte un seul terme atomique faible, représenté par  $A_1^1$ . Ce dernier est associé à la variable vitesse,  $\omega_1$ .*

Intuitivement, cette condition signifie que pour tout point du référentiel, il existe au moins une règle floue si-alors dont la fonction d'appartenance de sa partie si est non nulle.

**Définition 1.5.4 (Cohérence).** *Une base de règles floues si-alors est qualifiée de cohérente si aucune règle floue ne possède de prémisses identiques (partie si) et des conclusions différentes (partie alors).*

**Définition 1.5.5 (Continuité).** *Une base de règles floues si-alors est qualifiée de continue s'il n'existe aucune règle floue telle que l'intersection de sa partie Alors avec celle des autres règles, correspond à l'ensemble vide.*

Le non respect de la condition de cohérence n'est cependant pas critique selon Wang (1997, section 7), en présence de règles contradictoires, le moteur d'inférence floue procède au calcul de la moyenne de leurs résultats.

### 1.5.3. Étape de fuzzification

L'étape de fuzzification consiste essentiellement à convertir une donnée tirée du monde réel  $x^* \in U \subset \mathbb{R}^n$  en sa valeur correspondante en terme d'ensemble flou  $A' \subset U$ . La tâche du système de fuzzification consiste essentiellement à caractériser le degré d'appartenance à un ensemble flou des données reçues à l'entrée du moteur d'inférence.

En résumé, cette étape veille au bon décodage des règles *si-alors* du moteur par l'évaluation des fonctions d'appartenance. Pour qu'elle se réalise efficacement, chacune des valeurs ou états que peut prendre une variable linguistique donnée doit être modélisée par une fonction d'appartenance appropriée, conformément à la sous-section 1.5.1.1.

Des modèles de systèmes de fuzzification plus avancés permettent même de caractériser, de manière numérique, l'imprécision qui peut exister à l'entrée sur la valeur des données.

### 1.5.4. Raisonnement approché

La logique conventionnelle se consacre à l'étude des principes mêmes qui sous-tendent le raisonnement : l'art de créer de nouvelles propositions à l'aide de certaines prémisses déjà existantes. À l'instar de la logique booléenne, les règles du jeu flou s'énoncent ainsi : déduire des conclusions imprécises (propositions floues) à partir d'une collection de prémisses floues. Cette opération est désormais connue sous le vocable de raisonnement approché. Ce but ultime est atteint grâce aux nombreux outils offerts par la logique floue de même que par l'ajout de règles

d'inférence tirées de la logique classique : le modus ponens, le modus tollens et le syllogisme hypothétique généralisés (voir Wang, 1997, section 6).

**Définition 1.5.6 (Modus ponens généralisé).** *Une règle d'inférence qui stipule que sachant deux prémisses  $x$  est  $A'$  et Si  $x$  est  $A$ , alors  $y$  est  $B$ , il est possible d'inférer une nouvelle proposition floue  $y$  est  $B'$  est appelée modus ponens généralisé.*

*Prémisse 1 :  $x$  est  $A'$ ,*

*Prémisse 2 : si  $x$  est  $A$ , alors  $y$  est  $B$ ,*

*Conclusion :  $y$  est  $B'$ .*

Soit l'ensemble flou  $A'$  contenu dans le référentiel  $U$  et l'implication floue  $A \rightarrow B$  définie dans  $U \times V$ , qui représentent respectivement la première et la seconde prémisse, et l'ensemble flou  $B'$  du référentiel  $V$ , le modus ponens généralisé (voir équation (1.5.9)) se définit mathématiquement comme suit :

$$\mu_{B'}(y) = \sup_{x \in U} t[\mu_{A'}(x), \mu_{A \rightarrow B}(x, y)], \quad (1.5.9)$$

où  $\mu_{A \rightarrow B}(x, y)$  est défini à la sous-section 1.5.2.1 (voir équations (1.5.3) à (1.5.8)).

**Définition 1.5.7 (Modus tollens généralisé).** *Une règle d'inférence qui stipule que sachant deux prémisses  $y$  est  $B'$  et Si  $x$  est  $A$ , alors  $y$  est  $B$ , il est possible d'inférer une nouvelle proposition floue  $x$  est  $A'$  est appelée modus tollens généralisé.*

*Prémisse 1 :  $y$  est  $B'$ ,*

*Prémisse 2 : si  $x$  est  $A$ , alors  $y$  est  $B$ ,*

*Conclusion :  $x$  est  $A'$ .*

Soit l'ensemble flou  $B'$  contenu dans le référentiel  $V$  et l'implication floue  $A \rightarrow B$  définie dans  $U \times V$ , qui représentent respectivement la première et la



seconde prémisses, l'ensemble flou  $A'$  du référentiel  $U$  est inféré par le modus tollens généralisé (voir équation (1.5.10)) défini mathématiquement comme suit :

$$\mu_{A'}(x) = \sup_{y \in V} t[\mu_{B'}(y), \mu_{A \rightarrow B}(x, y)]. \quad (1.5.10)$$

**Définition 1.5.8 (Syllogisme hypothétique généralisé).** *Une règle d'inférence qui stipule que sachant deux prémisses Si  $x$  est  $A$ , alors  $y$  est  $B$  et Si  $y$  est  $B'$ , alors  $z$  est  $C$ , il est possible d'inférer une nouvelle proposition floue Si  $x$  est  $A$ , alors  $z$  est  $C'$  est appelée syllogisme hypothétique généralisé.*

*Prémisse 1 : si  $x$  est  $A$ , alors  $y$  est  $B$ ,*

*Prémisse 2 : si  $y$  est  $B'$ , alors  $z$  est  $C$ ,*

*Conclusion : si  $x$  est  $A$ , alors  $z$  est  $C'$ .*

Soit les implications floues  $A \rightarrow B$  définie dans  $U \times V$  et  $B' \rightarrow C$  définie dans  $V \times W$ , qui représentent respectivement la première et la seconde prémisses, la relation floue  $A \rightarrow C'$  définie dans  $U \times W$  est inférée par le syllogisme hypothétique généralisé (voir équation (1.5.11)) défini mathématiquement comme suit :

$$\mu_{A \rightarrow C'}(x, z) = \sup_{y \in V} t[\mu_{A \rightarrow B}(x, y), \mu_{B' \rightarrow C}(y, z)]. \quad (1.5.11)$$

**Exemple 1.5.5.** *Poursuivons l'exemple précédent 1.5.3. L'ensemble de prémisses et conclusion suivant peut-être développé à l'aide du modus ponens généralisé :*

*Prémisse 1 : la vitesse est faible,*

*Prémisse 2 : si la vitesse est faible, alors l'accélération est élevée,*

*Conclusion : l'accélération est élevée.*

*Si on utilise une implication de Mamdani du produit et une norme floue  $T$  du minimum, cette expression se représente par :*

$$\mu_{\text{élevée}}^*(y) = \sup_{v \in \text{vitesse}} \min[\mu_{\text{faible}}(v), \mu_{\text{faible}}(x) \mu_{\text{élevée}}(y)], \quad (1.5.12)$$

où  $x$  et  $y$  représentent respectivement les observations de vitesse reçues et les différentes valeurs d'accélération. Les termes  $\mu_{faible}(x)$  et  $\mu_{élevée}(y)$  correspondent respectivement à l'ensemble flou vitesse faible et accélération élevée tandis que  $\mu_{faible}(x)$  représente le degré d'appartenance de chaque donnée reçue en entrée à l'ensemble flou vitesse faible. Finalement, le terme  $\mu_{élevée}^*(y)$  désigne le nouvel ensemble flou accélération élevée obtenu après inférence.

### 1.5.5. Combinaison des règles si-alors

La section 1.5.2 a établi la procédure pour interpréter adéquatement une règle floue *si-alors* en terme d'implication. Puis, la section 1.5.4 a présenté comment inférer une nouvelle proposition floue à l'aide de règles d'inférence. La présente section démontre maintenant comment inférer une conclusion logique à partir d'une base de données constituée de plus d'une règle floue. Cette tâche délicate est accomplie par deux types distincts d'inférence, soit l'inférence basée sur la composition et celle basée sur les règles individuelles.

#### 1.5.5.1. Inférence basée sur la composition

La méthode d'inférence basée sur la composition possède une vision globale de l'inférence : toutes les règles floues contenues dans la base de règles sont combinées, puis évaluées comme une règle floue *si-alors* unique.

Selon la façon d'envisager la combinaison des règles, les spécialistes de la logique floue préconisent l'emploi de deux différents types de combinaison, soient celle de Mamdani (voir équation (1.5.13)) qui privilégie l'utilisation d'opérateurs

d'union floue et celle de Gödel (voir équation (1.5.14)) privilégiant plutôt l'intersection floue. Elles se représentent respectivement par :

$$Q_M = \bigcup_{l=1}^M Ru^{(l)}, \quad (1.5.13)$$

$$Q_G = \bigcap_{l=1}^M Ru^{(l)}. \quad (1.5.14)$$

**Exemple 1.5.6.** Ajoutons maintenant une règle supplémentaire à celle de l'exemple initial 1.5.5 pour ainsi bâtir une base de règles floues si-alors :

$Ru^{(1)}$  : Si la vitesse est faible, alors l'accélération est élevée,

$Ru^{(2)}$  : Si la vitesse est élevée, alors l'accélération est faible.

Si on utilise une combinaison de Mamdani avec un opérateur d'union floue tel le maximum, on évalue la combinaison de la façon suivante :

$$\mu_{Q_M}(x,y) = \max(\mu_{faible}(x)\mu_{élevée}(y), \mu_{élevée}(x)\mu_{faible}(y)),$$

où  $x$  et  $y$  représentent respectivement les observations de vitesse reçues et les différentes valeurs d'accélération.

Dans ces circonstances, le modus ponens généralisé devient

$$\mu_{élevée}^*(y) = \sup_{v \in \text{vitesse}} \min[\mu_{faible}(v), \mu_{Q_M}(x,y)],$$

où  $x$  et  $y$  représentent respectivement les observations de vitesse reçues et les différentes valeurs d'accélération.

### 1.5.5.2. Inférence basée sur les règles individuelles

Ce type d'inférence opte plutôt pour une vision individualiste : chaque règle floue *si-alors* de la base de règles est évaluée indépendamment des autres à l'aide d'une règle d'inférence et la combinaison de ces résultats intermédiaires correspond à la conclusion inférée par le moteur d'inférence floue.

La combinaison est qualifiée soit de Mamdani soit de Gödel. Elle s'effectue respectivement à l'aide de divers opérateurs d'union floue (voir équation (1.5.15)) ou d'intersection floue (voir équation (1.5.16)). Ces deux types de combinaison se représentent comme suit :

$$Q_M = \bigcup_{l=1}^M Conclusion(Ru^{(l)}), \quad (1.5.15)$$

$$Q_G = \bigcap_{l=1}^M Conclusion(Ru^{(l)}). \quad (1.5.16)$$

Un exemple de *Conclusion* d'une règle floue, le résultat de l'application d'une règle d'inférence, est donnée par l'équation (1.5.12).

**Exemple 1.5.7.** *Poursuivons l'exemple précédent 1.5.6 et évaluons les deux règles  $Ru^{(1)}$  et  $Ru^{(2)}$  à la lumière de la théorie de l'inférence basée sur les règles individuelles. Si on utilise une combinaison de Mamdani avec un opérateur d'union floue tel le maximum, on évalue la combinaison de la façon suivante :*

$$\mu_{Q_M}(x,y) = \max(Conclusion(Ru^{(1)}), Conclusion(Ru^{(2)})),$$

où  $x$  et  $y$  représentent respectivement les observations de vitesse reçues et les différentes valeurs d'accélération.

### 1.5.6. Moteurs d'inférence floue

Le moteur d'inférence floue constitue l'élément rassembleur, il réalise l'unification de la base de règles floues *si-alors* avec les diverses combinaisons des règles floues et le modus ponens généralisé. Ce dernier se veut la règle d'inférence de prédilection dans la conception des moteurs selon Wang (1997, section 7), mais les autres types restent valides. Le moteur d'inférence offre un large éventail d'outils

permettant le calcul complet de l'inférence floue à partir de données réelles fuz-zifiées. La conclusion inférée reste cependant à l'intérieur des limites de l'univers flou.

Une grande variété de moteurs d'inférence floue existe. Les différentes alternatives dans la conception d'un moteur résident dans le choix :

- de l'inférence basée sur la composition ou sur les règles individuelles et dans le premier cas, de la combinaison de Mamdani (voir équation (1.5.13)) ou de Gödel (voir équation (1.5.14)),
- des implications de Dienes-Rescher (voir équation (1.5.3)), de Lukasiewicz (voir équation (1.5.4)), de Zadeh (voir équation (1.5.5)), de Gödel (voir équation (1.5.6)) ou de Mamdani (voir équations (1.5.7) et (1.5.8)),
- et des différents opérateurs d'union et d'intersection floues.

Le choix d'une alternative au dépend d'une autre varie selon le problème posé, les considérations de l'expert et l'efficacité calculatoire. Les principaux modèles de moteurs d'inférence sont au nombre de cinq : les moteurs du produit, du minimum, de Lukasiewicz, de Zadeh et finalement de Dienes-Rescher. Ces derniers sont présentés séquentiellement à l'aide des équations (1.5.17) à (1.5.21).

Tout d'abord, le moteur d'inférence du produit se compose :

- d'inférence basée sur les règles individuelles et de combinaison par opérateur d'union floue,
- d'implication de Mamdani du produit
- du produit algébrique comme opérateur d'intersection floue et du maximum pour l'union floue.

La fonction d'appartenance à l'ensemble flou  $B'$  étant donné l'ensemble flou  $A'$  constitué de diverses parties  $A_i$  se représente comme suit :

$$\mu_{B'}^*(y) = \max_{l \in \{1, \dots, M\}} [\sup_{x \in U} (\mu_{A'}(x) \{ \prod_{i=1}^n \mu_{A_i}(x_i) \} \mu_{B^l}(y))], \quad (1.5.17)$$

où  $l$  indexe chacune des règles floues et  $i$ , les multiples variables linguistiques d'entrée ou de la partie *si* des règles dans le cas de propositions composées.

Il faut noter que les variables  $x_i$  incluses dans les équations du précédent moteur et des suivants réfèrent aux données réelles de différents types présentes à l'entrée du système. Par contre, les variables  $\mu_{A'}(x)$  et  $\mu_{B^l}(y)$  correspondent à des ensembles flous, c'est-à-dire aux termes primaires des variables linguistiques constituant respectivement la partie *si* et *alors* de chacune des règles. Enfin, la variable  $\mu_{B'}^*(y)$  réfère à l'ensemble flou de sortie  $B'$ , transformé suite à l'application du moteur d'inférence floue.

Deuxièmement, le moteur d'inférence du minimum se compose des mêmes éléments que son prédécesseur à l'exception du minimum qui remplace le produit algébrique à titre d'opérateur d'intersection floue :

$$\mu_{B'}^*(y) = \max_{l \in \{1, \dots, M\}} [\sup_{x \in U} \min(\mu_{A'}(x), \mu_{A_1}(x_1), \dots, \mu_{A_n}(x_n), \mu_{B^l}(y))]. \quad (1.5.18)$$

où  $l$  indexe chacune des règles floues.

Le moteur de Lukasiewicz privilégie toujours les éléments suivants :

- l'inférence basée sur les règles individuelles, mais la combinaison est réalisée par un opérateur d'intersection floue,
- l'implication de Lukasiewicz
- le minimum comme opérateur de norme T.

Il prend la forme mathématique suivante :

$$\mu_{B^l}^*(y) = \min_{l \in \{1, \dots, M\}} \left\{ \sup_{x \in U} \min[\mu_{A^l}(x), (1 + \mu_{B^l}(y) - \min_{i \in \{1, \dots, n\}} (\mu_{A_i^l}(x_i)))^+] \right\}. \quad (1.5.19)$$

où  $l$  indexe chacune des règles floues et  $i$ , les multiples variables linguistiques d'entrée ou de la partie *si* des règles dans le cas de propositions composées.

Finalement les moteurs de Zadeh et de Dienes-Rescher possèdent la même structure que le moteur de Lukasiewicz, mais l'implication de Zadeh est utilisée dans le premier cas et celle de Dienes-Rescher dans le second. Leur structure est représentée respectivement par :

$$\mu_{B^l}^*(y) = \min_{l \in \{1, \dots, M\}} \left\{ \sup_{x \in U} \min[\mu_{A^l}(x), \max(\min(\mu_{A_1^l}(x_1), \dots, \mu_{A_n^l}(x_n)), \mu_{B^l}(y)), 1 - \min_{i \in \{1, \dots, n\}} (\mu_{A_i^l}(x_i))] \right\}, \quad (1.5.20)$$

ainsi que

$$\mu_{B^l}^*(y) = \min_{l \in \{1, \dots, M\}} \left\{ \sup_{x \in U} \min[\mu_{A^l}(x), \max(\mu_{B^l}(y), 1 - \min_{i \in \{1, \dots, n\}} (\mu_{A_i^l}(x_i))) \right\}. \quad (1.5.21)$$

où  $l$  indexe chacune des règles floues et  $i$ , les multiples variables linguistiques d'entrée ou de la partie *si* des règles dans le cas de propositions composées.

Les moteurs du produit et du minimum jouissent de la plus grande popularité dans le domaine de l'ingénierie selon Wang (1997, section 7). Ce fait s'explique par leur simplicité calculatoire et leur facilité d'implémentation sur une foule de problèmes pratiques.

**Exemple 1.5.8.** *Construisons maintenant un moteur d'inférence du produit à l'aide de la base de règles floues de l'exemple précédent 1.5.7. Pour chaque observation concernant la vitesse reçue à l'entrée du moteur, chacune des règles*

si-alors est évaluée, le *modus ponens* généralisé est lancé pour chacune d'elle et finalement les résultats intermédiaires d'inférence sont combinés. Les ensembles flous représentant la variable linguistique de sortie, accélération,  $B'$ , sont ainsi modifiés à chaque nouvelle observation reçue. Le moteur du produit se représente comme suit :

$$\mu_{B'}^*(y) = \max_{l \in \{1,2\}} [ \sup_{x \in \text{vitesse}} (\mu_{A^l}(x) \mu_{A^l}(x_1) \mu_{B^l}(y)) ],$$

où  $x_1$  et  $y$  représentent respectivement les observations de vitesse reçues et les différentes valeurs d'accélération,  $l$  indexe les règles d'inférence et  $A^l$  correspond aux termes primaires de la variable linguistique vitesse, soit l'ensemble flou  $\mu_{faible}(x)$  dans le cas de la première règle et  $\mu_{élevée}(x)$  dans l'autre. Quant au terme  $\mu_{B^l}(y)$ , il correspond respectivement aux ensembles flous  $\mu_{élevée}(y)$  et  $\mu_{faible}(y)$ .

### 1.5.7. Étape de défuzzification

En queue de liste figure l'étape de défuzzification qui, comme son nom l'indique, réalise la tâche inverse du procédé de fuzzification : le retour au monde réel. Ce dernier chaînon du système d'inférence floue reconvertit l'ensemble flou obtenu après inférence en une donnée réelle unique. Il spécifie, en fait, la donnée réelle qui représente le mieux cet ensemble flou parmi tout le référentiel.

Encore ici, il existe plusieurs types de défuzzification : la méthode du centre de gravité, celle du centre moyen et celle du maximum. Ces derniers sont présentés plus loin à l'aide des équations (1.5.22) à (1.5.24). Un choix judicieux de la méthode de défuzzification s'effectue selon certains critères précis tels la plausibilité (le choix de l'élément représentant le mieux l'ensemble flou doit être presque intuitif), la simplicité calculatoire et la continuité (l'influence sur la solution réelle d'un petit changement dans l'ensemble flou résultant doit être minimale).



La méthode du centre de gravité (voir équation (1.5.22)), évaluée sur le référentiel  $V$ , spécifie le centre de l'aire comprise sous la fonction d'appartenance de l'ensemble flou  $B'$  et s'exprime ainsi

$$y^* = \frac{\int_V y \mu_{B'}(y) dy}{\int_V \mu_{B'}(y) dy}, \quad (1.5.22)$$

où  $\int_V$  représente une intégrale conventionnelle.

Cette méthode s'impose par sa plausibilité : déterminer le centre de gravité de l'ensemble flou résultant semble, en effet, presque intuitif. Par contre, la complexité calculatoire peut être assez élevée.

Puisque l'ensemble flou résultant  $B'$  représente l'union ou l'intersection de  $M$  ensembles flous, une bonne approximation du centre de gravité s'avère être la moyenne pondérée du centre de chacun (voir équation (1.5.23)) où la hauteur de chaque ensemble flou représente son poids. L'équation de la méthode du centre moyen s'exprime ainsi

$$y^* = \frac{\sum_{l=1}^M \bar{y}^l \omega_l}{\sum_{l=1}^M \omega_l}, \quad (1.5.23)$$

où  $\bar{y}$  représente le centre d'un ensemble flou intermédiaire et  $\omega_l$ , son poids, soit  $\omega_l = \sup_{y \in V} \mu_{B^l}(y)$ .

Côté analyse, ce modèle jouit des mêmes avantages et inconvénients que la précédente méthode du centre de gravité.

Finalement, la méthode du maximum (voir équation (1.5.24)) choisit comme valeur de retour, le point de référentiel  $V$  où la fonction d'appartenance atteint son maximum :

$$y^* = \{y \in V \mid \mu_{B'}(y) = \sup_{y \in V} \mu_{B'}(y)\}. \quad (1.5.24)$$

Si plusieurs de ces valeurs existent, les méthodes du plus petit ou du plus large maximum ou encore du maximum moyen peuvent être choisies pour parfaire la défuzzification.

Cette méthode plausible, de complexité minimale, peut dans certaines occasions souffrir d'une lacune majeure: de petits changements dans l'ensemble flou  $B'$  peuvent résulter en d'importantes modifications de  $y^*$ , la solution réelle.

**Exemple 1.5.9.** *Finalemment à la suite de l'analyse d'une observation par le moteur comme dans l'exemple initial 1.5.8, l'étape de défuzzification permet l'obtention d'une valeur d'accélération non plus sous forme d'ensembles flous, mais réelle.*

*Si l'on utilise une méthode de défuzzification par centre de gravité, on obtient la valeur optimale d'accélération par :*

$$y^* = \frac{\int y \mu_{\text{accélération}}(y) dy}{\int \mu_{\text{accélération}}(y) dy},$$

*où  $y$  représente les différentes valeurs d'accélération.*

Ce chapitre a présenté les multiples facettes de la logique floue tant au niveau de ses propriétés intrinsèques, de la théorie des possibilités ou, encore, de l'inférence floue. Pour un aperçu plus détaillé de ces thèmes, veuillez vous référer à Gacogne (1997); Sangalli (1998); Sugeno (1977); Zadeh (1969); Zimmermann (1985). Les moteurs d'inférence floue, coeur même de ce mémoire, feront l'objet d'une analyse plus approfondie dans le prochain chapitre. Ils seront alors appliqués au sujet même de cette maîtrise, l'identification de cibles.

## Chapitre 2

---

# MODÉLISATION ET CONCEPTION DE SYSTÈMES EXPERTS FLOUS

### 2.1. INTRODUCTION

Le présent chapitre constitue la véritable entrée en matière de ce mémoire, soit l'application de la logique floue et des moteurs d'inférence floue à l'identification de cibles.

À l'heure actuelle, les étapes de combinaison et d'identification de l'information sont habituellement accomplies de façon manuelle dans les centres de contrôle. Par conséquent, ces processus sont sujets à l'interprétation erronée et à l'erreur humaine. Cette recherche vise principalement à automatiser le traitement de l'information provenant de sources diverses. Ces résultats faciliteront ainsi les prises de décision et assureront l'efficacité des systèmes de contrôle.

Le présent chapitre, en plus de préciser la nature des travaux de recherche, aborde différentes facettes de l'identification de plates-formes dont les attributs et leur modélisation, les techniques de conception de systèmes experts flous ainsi que les fondements du raisonnement approché. Pour un aperçu plus détaillé de ces applications de la logique floue, veuillez vous référer à Abdulghafour *et al.* (1994); Bailey et Singh (1997); Chen et Yang (1998); Cumani (1982); de Oliviera et Pedrycz (1996).

Tout au long de ce document certains thèmes techniques seront abordés fréquemment. Un léger éclaircissement s'impose avant de débiter. Veuillez noter que les mots *plate-forme* et *cible* désignent des engins de transport d'usage commercial ou militaire et que les termes *senseur* et *capteur* concernent des appareils électroniques utilisés pour la cueillette d'information.

## 2.2. DÉFINITION DU PROBLÈME

Le besoin de traiter en temps réel une importante quantité d'information incertaine et incomplète se fait sentir dans plusieurs contextes industriels et militaires contemporains. La littérature a déjà suggéré quelques approches pour corriger le problème notamment les théories bayésienne (voir Shafer, 1976; Smith, 1988; West, 1988) et évidentielle (voir Shafer, 1976). La première impose que les probabilités *a priori* soient complètement spécifiées, ce qui restreint parfois son application. La seconde théorie, par contre, semble moins restrictive, mais souffre d'un problème structurel : l'omniprésence d'une génération excessive de propositions. Son application s'en trouve ainsi également limitée. La théorie des ensembles flous offre ici une nouvelle vision pour aborder ce problème et semble même posséder des avantages indéniables permettant d'accomplir un traitement de l'information efficace.

L'étude de la théorie des ensembles flous, le but principal de ce projet, se subdivise en deux sous-objectifs généraux dont l'un est théorique et l'autre, pratique. Le premier énonce l'idéal académique poursuivi par ce projet tandis que le second concerne davantage les attentes réelles ou concrètes en termes de simulations et d'applications.

### 2.2.1. Sous-objectif théorique

Le sous-objectif théorique de ce projet réside dans l'exploration des avenues permettant l'application de la théorie des ensembles flous à la combinaison de l'information incertaine et possiblement contradictoire provenant de différentes sources. Celui-ci sera atteint par le développement d'algorithmes capables de combiner l'information à l'aide de la logique floue et des théories connexes et ainsi d'assurer la fiabilité de la prise de décision. Une étape d'évaluation à la fois qualitative et quantitative des algorithmes figure également en tout premier plan.

### 2.2.2. Sous-objectif pratique

Dans un scénario militaire hypothétique d'identification de cibles, une plateforme inconnue effectue diverses manoeuvres et mouvements. Différents senseurs épient constamment la scène et recueillent de l'information sur cet inconnu et ses caractéristiques. Les observations sont ensuite transmises à un centre de contrôle dont la tâche principale consiste à identifier la nature précise de l'objet.

Le sous-objectif pratique poursuivi consiste ainsi en l'identification de cibles militaires et civiles à l'aide de divers algorithmes intégrant logique floue et moteurs d'inférence floue développés au chapitre précédent. De façon plus précise, il consiste à *fuzzifier* le fonctionnement du centre de contrôle, c'est-à-dire remplacer les systèmes experts bayésiens et évidentiels par de nouveaux basés sur la logique floue. Une étape subséquente d'analyse, à la fois quantitative et qualitative, est jointe permettant ainsi d'étudier les performances de ces nouveaux systèmes experts.

## 2.3. CARACTÉRISTIQUES DES PLATES-FORMES

Les plates-formes possèdent certaines caractéristiques particulières, appelées attributs, permettant de les discriminer. Toutes ces informations sont conservées dans une base de données de plates-formes connue de tout système expert. La tâche de ce dernier consiste essentiellement à déterminer l'identité de toute cible inconnue en se basant sur l'observation de ces attributs.

Tous de seconde génération, les systèmes experts développés dans ce projet présentent un réalisme accentué. L'identification pratiquée ici dépasse le simple seuil de l'étude des observations et s'étend désormais également à l'étude de la hiérarchie entre les plates-formes.

La présente section traite plus en profondeur de chacun de ces thèmes, soit la base de données de plates-formes, les attributs et leur modélisation mathématique ainsi que l'identification dite hiérarchique.

### 2.3.1. Base de données de plates-formes

Toute l'information concernant les plates-formes et leurs attributs est emmagasinée dans une base de données. Les plates-formes, au nombre total de cent onze, possèdent toutes différents attributs allant du nom, aux dimensions physiques, en passant par la vitesse ou l'alléance. Dans cette étude, quatre de ces attributs ont été privilégiés pour ainsi assurer la discrimination des plates-formes : le *type de la plate-forme*, les *vitesse minimale* et *maximale*, le *nombre d'émetteurs* présents sur chaque plate-forme de même que leur *type*.

Le premier attribut, le *type de plate-forme*, réfère directement à l'environnement dans lequel évolue les plates-formes. Certaines se déplacent à la surface de la Terre, que ce soit sur mer ou sur terre, et sont qualifiées de type *surface* tandis

que d'autres, de type *air*, évoluent plutôt dans les airs. La confusion de type est généralement impossible, l'évaluation de cet attribut se veut exempte d'erreur.

Le second attribut concerne la plage de vitesses possibles pour chaque plate-forme, c'est-à-dire les vitesses limites. Les bornes de celles-ci vont de 0 à 50 noeuds pour les plates-formes de type *surface* et de 0 à 2000 noeuds pour celles de type *air*. Cet attribut se subdivise en deux sous-catégories pour chacune des plates-formes : les *vitesses minimale* et *maximale*.

Les derniers attributs paraissent fortement liés. Ils concernent, en effet, tous deux les émetteurs ou les appareils de communication présents sur les plates-formes. Ceux-ci sont très variés et leur nature dépend généralement du type de plate-forme concernée. Tous les capteurs employés dans ce projet sont aptes à fournir simultanément ces deux attributs : le *nombre d'émetteurs* présents sur la cible et leur *type*.

Dans tout scénario, la cible inconnue figure dans la base de données de plates-formes et l'objectif principal du système expert consiste à la démasquer. Les scénarios se limitent dans ce mémoire à la détection d'une seule plate-forme, mais ceux-ci pourraient, sans peine, être étendus à la détection de plusieurs.

**Exemple 2.3.1.** *Voici un échantillon (voir tableau 2.3.1 ) de la base de données de plates-formes globale. Il regroupe quinze plates-formes de type surface choisies pour la variété de leurs attributs.*

### 2.3.2. Attributs

Tel que mentionné précédemment, l'identification pratiquée ici dépasse le simple seuil de l'étude des observations sur les attributs. La hiérarchie entre les plates-formes au niveau de chacun des attributs est également sujet à analyse.

TAB. 2.3.1 –. *Base de données de plates-formes simplifiée*

Plate-forme	Vitesse min (noeuds)	Vitesse max (noeuds)	Nombre d'émetteurs	Type d'émetteurs
1	0	11	1	79
2	0	16	0	-
3	0	17	5	44 47 83 103 109
4	0	21	5	8 42 43 75 76
5	0	24	13	7 13 16 17 31 32 54 113 114 115 116 117 118
6	0	25	3	33 39 40
7	0	25	10	45 46 64 65 68 89 93 101 103 105
8	0	26	6	7 18 33 92 127 130
9	0	28	4	121 135 136 137
10	0	31	8	47 62 77 83 84 85 103 104
11	0	32	15	46 65 71 88 91 94 95 96 97 98 99 100 101 102 106
12	0	33	12	8 14 18 31 32 43 53 57 114 115 119 121
13	0	35	9	7 13 15 16 17 18 31 32 57
14	0	35	9	7 8 13 16 18 33 34 35 57
15	0	38	9	7 8 13 16 17 18 19 20 57

Pour ce faire, une classification interne des attributs s'avère essentielle afin d'obtenir un classement entre les plates-formes. Cette étape se nomme *fuzzification*. Elle divise chacun des attributs en sous-catégories pour ainsi permettre l'analyse



et la représentation de la hiérarchie sous forme d'arbres de détection des plates-formes. Ces derniers seront présentés sous tous les angles dans la sous-section suivante.

Dans un même ordre d'idées, l'identification de plates-formes est réalisée par l'analyse des attributs. Certaines hypothèses ont ainsi dû être posées quant aux lois de probabilité à employer pour modéliser adéquatement chacun de ses attributs et ainsi assurer l'identification.

La présente sous-section s'attarde donc aux grandes lignes directrices de la fuzzification de même qu'aux différentes hypothèses utilisées pour modéliser les attributs.

#### 2.3.2.1. *Fuzzification*

Catégoriser chacun des attributs, voilà l'objectif de la fuzzification. Les deux principaux attributs nécessitant l'emploi de cette technique sont l'attribut *vitesse* et l'attribut *nombre d'émetteurs*. Le *type de plate-forme*, directement intégré au premier attribut, ne nécessite aucune fuzzification. Tout comme l'attribut *type des émetteurs* dont l'emploi de cette technique s'avère non réalisable.

L'attribut *vitesse* sépare tout d'abord les plates-formes en deux groupes bien distincts, soit celles de type *surface* et les autres de type *air*. Une fuzzification spécifique est appliquée dans chaque cas. À l'intérieur d'un de ces groupes, la vitesse moyenne est classifiée ou fuzzifiée en cinq différentes catégories : *la vitesse très lente, lente, modérée, élevée et très élevée*. Chacune représente une plage de vitesse précise et caractérise toute plate-forme se déplaçant à l'intérieur de ces limites. Les tableaux 2.3.2 et 2.3.3 présentent respectivement la fuzzification de l'attribut vitesse pour les plates-formes de type surface et air.

TAB. 2.3.2 – Fuzzification de l'attribut vitesse pour les plates-formes de type surface

Numéro	Catégorie	Étendue de la plage de vitesses (noeuds)
1	Très lente	[0, 20)
2	Lente	[20, 25)
3	Modérée	[25, 30)
4	Élevée	[30, 35)
5	Très élevée	[35, 50]

TAB. 2.3.3 – Fuzzification de l'attribut vitesse pour les plates-formes de type air

Numéro	Catégorie	Étendue de la plage de vitesses (noeuds)
1	Très lente	[0, 200)
2	Lente	[200, 400)
3	Modérée	[400, 600)
4	Élevée	[600, 800)
5	Très élevée	[800, 2000]

La consultation d'un spécialiste en pilotage maritime a permis de constater que prédire une vitesse de croisière pour les navires était chose possible. Les navires se déplacent en moyenne à une vitesse se situant à quelques noeuds en-dessous de leur vitesse maximale. En se basant sur une différence d'un noeud, la classe respective de chaque plate-forme a pu être ainsi établie avec précision. La même stratégie a été employée pour les plates-formes de type air, mais avec respect d'une marge plus importante de cent noeuds.

L'attribut *nombre d'émetteurs* se voit également fuzzifié en cinq catégories, mais celles-ci sont déterminées par une plage de nombres d'émetteurs plutôt que

de vitesse. Elles se nomment respectivement *quantité très minime (QTMIn)*, *minime (QMin)*, *modérée (QM)*, *importante (QI)* et *très importante (QTI)*. Dans cette optique, chacune des plates-formes est caractérisée en fonction du nombre d'émetteurs qu'elles possèdent et ce, peu importe son type. Le tableau 2.3.4 résume la fuzzification de l'attribut nombre d'émetteurs pour tout type de plate-forme.

TAB. 2.3.4 – Fuzzification de l'attribut nombre d'émetteurs pour tout type de plate-forme

Numéro	Catégorie	Étendue de la plage de nombres d'émetteurs
1	Quantité très minime	0, 1 et 2
2	Quantité minime	3, 4 et 5
3	Quantité modérée	6, 7 et 8
4	Quantité importante	9, 10 et 11
5	Quantité très importante	12, 13, 14 et 15

**Exemple 2.3.2.** Suite à l'exemple précédent 2.3.1, la fuzzification des attributs vitesse et nombre d'émetteurs pour chacune des quinze plates-formes de type surface est donnée au tableau 2.3.5.

#### 2.3.2.2. Hypothèses et modélisation

Pour parvenir à résoudre le problème posé par l'identification de cibles, certaines hypothèses ont dû être posées quant aux lois de probabilité à employer pour modéliser sous forme d'ensembles flous chacun des attributs et ainsi assurer une identification adéquate. En voici une description détaillée.

TAB. 2.3.5 –. *Fuzzification des attributs vitesse et nombre d'émetteurs pour les plates-formes de la base de données simplifiée*

<i>Plate-forme</i>	<i>Catégorie vitesse</i>	<i>Catégorie nombre d'émetteurs</i>
1	1	1
2	1	1
3	1	2
4	2	2
5	2	5
6	2	2
7	2	4
8	3	3
9	3	2
10	4	3
11	4	5
12	4	5
13	4	4
14	4	4
15	5	4

Tout d'abord, la modélisation de l'attribut *vitesse* requiert une fonction d'appartenance continue. La base de données de plates-formes renferme toute l'information pertinente concernant les vitesses minimale et maximale, mais là s'arrête l'étendue des connaissances. Même si la vitesse de croisière est donnée, nous connaissons peu de chose sur la véritable distribution de la vitesse entre ces deux bornes. Une plate-forme peut ainsi se déplacer à n'importe quelle vitesse à l'intérieur de ces limites. Une distribution continue uniforme s'étalant de la vitesse

minimale à la vitesse maximale de la plate-forme constitue donc la modélisation privilégiée pour cet attribut. Cette fonction d'appartenance se définit comme suit :

$$\mu_{plate-forme_{vitesse}}(x|v_{min},v_{max}) = \frac{1}{(v_{max} - v_{min})} \mathbb{I}_{[v_{min},v_{max})}(x),$$

où  $x$  représente une vitesse donnée et  $\mathbb{I}_{[v_{min},v_{max})}(x) = 1$  si  $v_{min} \leq x < v_{max}$  et 0 sinon.

La modélisation de l'attribut *nombre d'émetteurs* requiert également une loi de probabilité peu informative. Une insuffisance de renseignements sur le comportement réel des senseurs ne permet pas de déterminer aisément des modes réalistes. Par contre, les lois de probabilité continues sont à proscrire, le nombre d'émetteurs demeure une donnée purement discrète. L'hypothèse suivante a ainsi été avancée : si un senseur détecte un certain nombre  $x$  d'émetteurs, plus de masse est accordée aux plates-formes comptant au moins  $x$  émetteurs qu'à celles en possédant moins. Celle-ci signifie que nous tenons compte qu'il peut exister certains subterfuges électroniques pouvant tromper le système expert et laissant croire à un nombre supérieur d'émetteurs sur la plate-forme observée. Cette hypothèse permet ainsi de couvrir toute l'étendue des possibilités.

La modélisation de cet attribut pour une plate-forme donnée nécessite l'emploi d'une distribution discrète non uniforme dont le maximum se situe au nombre exact d'émetteurs possédés par celle-ci. La courbe est ensuite décroissante de part et d'autre de ce sommet, mais avec un taux supérieur vers la gauche en respect avec la dernière hypothèse. La fonction d'appartenance s'étend ainsi sur toute l'étendue de valeurs possibles pour le nombre d'émetteurs, soit de zéro à quinze.

Elle se représente comme suit :

$$\mu_{plate-forme_{nombre\ d'\acute{e}metteurs}}(x|z) = \begin{cases} \frac{1}{(z-1)^{(z-x+1)}}, & \text{si } x = 0, 1, \dots, z-1, \\ \frac{1}{(x+1)}, & \text{si } x = z, z+1, \dots, 15, \end{cases}$$

où  $x$  représente un nombre d'émetteurs observé et  $z$ , le nombre exact d'émetteurs de la plate-forme étudiée (voir la figure 2.3.2 pour un exemple de cette loi de probabilité).

Finalement, l'attribut *type des émetteurs* se veut intimement lié au précédent. Concrètement, la tâche d'évaluation du nombre d'émetteurs et de leur type est effectuée par le même senseur. Si ce dernier détecte une plate-forme, il est en mesure de fournir simultanément la liste des émetteurs potentiels et leur nombre. Cependant, cet attribut ne nécessite pas, à proprement parler, de modélisation par une loi de probabilité.

**Exemple 2.3.3.** *Poursuivons l'exemple précédent 2.3.2 à la lumière des techniques de modélisation employées pour les différents attributs des plates-formes. Les figures suivantes 2.3.1 et 2.3.2 présentent respectivement les distributions des attributs (ensembles flous) vitesse et nombre d'émetteurs pour la plate-forme numéro trois de la base de données simplifiée. Cette dernière se déplace à une vitesse variant entre zéro et dix-sept noeuds et possède cinq émetteurs.*

### 2.3.3. Identification hiérarchique

Dans un scénario standard, l'identification est orientée principalement vers l'analyse des données réelles. Les senseurs rapportent directement des informations sur les attributs de la plate-forme observée et les transmettent par la suite au centre de commande. À ce stade, le présent projet modifie quelque peu les règles du jeu et l'orientation générale de l'identification. Elle devient maintenant

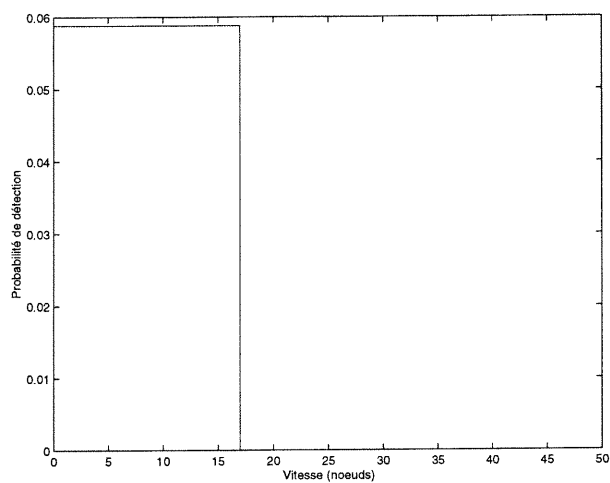


FIG. 2.3.1 – *Distribution de l'attribut vitesse pour la plate-forme numéro trois*

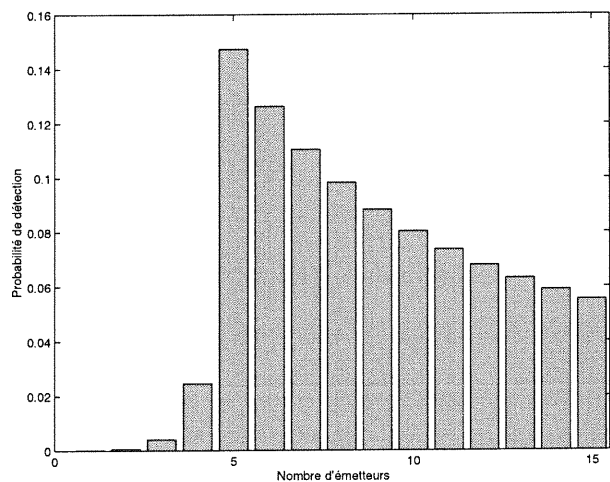


FIG. 2.3.2 – *Distribution de l'attribut nombre d'émetteurs pour la plate-forme numéro trois*

hiérarchique. Désormais un capteur ne détecte plus directement des données sur les attributs, mais identifie plutôt un point ou un niveau dans la hiérarchie de plates-formes. Celui-ci peut revêtir différentes formes que ce soit une plate-forme

unique ou encore un groupe de plates-formes qui partagent alors toutes une caractéristique commune, une même catégorie d'un attribut. Cette forme de détection reflète davantage l'état de l'information reçue, de détaillée, dans le cas de l'identification d'une seule plate-forme précise, à partielle, dans le cas de la détection d'un groupe de plates-formes. Mais néanmoins, elle ne certifie nullement que l'information est exempte d'erreurs.

Les attributs concernés par cette technique de détection sont les mêmes qui ont subi préalablement l'étape de fuzzification. Il s'agit des attributs *vitesse* et *nombre d'émetteurs*. En ce qui concerne l'attribut *vitesse*, un senseur peut rapporter seulement un groupe de plates-formes de même type (air ou surface), un groupe de plates-formes représentant un type et une catégorie (surface et vitesse très lente par exemple) ou encore une plate-forme spécifique. Pour l'attribut *nombre d'émetteurs* par contre, le rapport d'un senseur se situe uniquement à deux niveaux, soit à celui de la catégorie (quantité minimale par exemple) ou à celui d'une plate-forme spécifique. Une fois un de ces niveaux déterminé par un senseur, le système expert flou est alors en mesure de retracer la nature des attributs de la ou des plates-formes en question. La véritable tâche d'identification débute : déduire l'identité de la plate-forme inconnue.

Pour représenter la hiérarchie entre les plates-formes, l'arbre de détection des plates-formes constitue un moyen privilégié. Il illustre de façon non équivoque les multiples liens unissant les plates-formes pour les attributs *vitesse* et *nombre d'émetteurs*. Le prochain exemple (voir exemple 2.3.4) en présente quelques uns.

**Exemple 2.3.4.** *Suite à l'exemple précédent 2.3.3, voici les arbres de détection de plates-formes de la base de données simplifiée pour les attributs vitesse et nombre d'émetteurs. Les figures 2.3.3 et 2.3.4 représentent directement la hiérarchie entre les quinze plates-formes au niveau de ces deux attributs.*



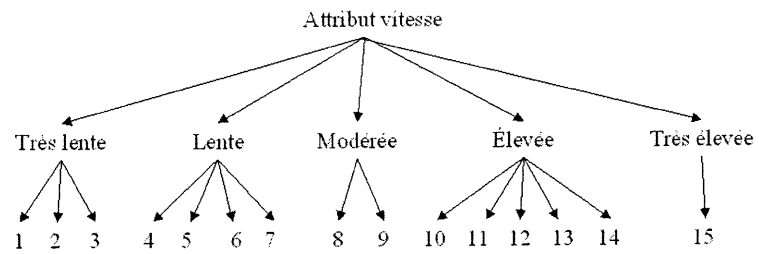


FIG. 2.3.3 – Arbre de détection pour l'attribut vitesse pour les plates-formes de la base de données simplifiée

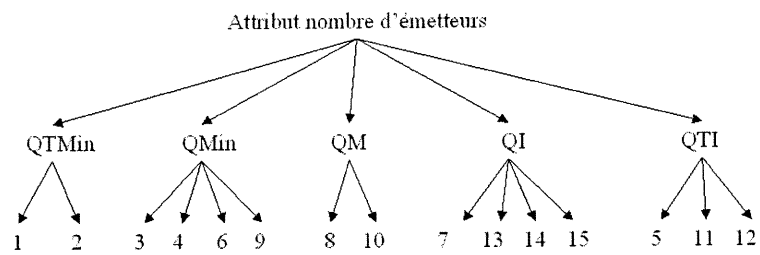


FIG. 2.3.4 – Arbre de détection pour l'attribut nombre d'émetteurs pour les plates-formes de la base de données simplifiée

Tel que discuté plus haut, un senseur peut détecter une plate-forme spécifique, un groupe de plates-formes déterminées par une catégorie d'un attribut ou encore, dans le cas de la vitesse, un groupe de plates-formes du même type. Le tableau 2.3.6 démontre quelques exemples de ces rapports de senseurs.

TAB. 2.3.6 – Exemples de rapport de senseurs concernant la base de données simplifiée

<i>Attribut étudié par le senseur</i>	<i>Niveau de la hiérarchie détecté</i>	<i>Plate(s)-forme(s) concernée(s)</i>
<i>Vitesse</i>	<i>Type surface</i>	<i>toutes</i>
<i>Vitesse</i>	<i>Catégorie vitesse lente</i>	<i>4, 5, 6 et 7</i>
<i>Vitesse</i>	<i>Plate-forme unique</i>	<i>3</i>
<i>Nombre d'émetteurs</i>	<i>Catégorie quantité minimale</i>	<i>3, 4, 6 et 9</i>
<i>Nombre d'émetteurs</i>	<i>Plate-forme spécifique</i>	<i>5</i>

## 2.4. CONCEPTION DE SYSTÈMES EXPERTS FLOUS

Concrètement, une procédure d'identification hiérarchique de cibles se déroule de façon bien précise. Tout d'abord, un nombre variable de senseurs épient constamment la scène. Chacun est dédié à la détection d'un attribut particulier de la plate-forme observée, soit la *vitesse* ou le *nombre d'émetteurs* et leur *type*.

Ils effectuent ainsi une première identification d'un niveau dans la hiérarchie de l'attribut étudié. Ces observations, parfois incertaines, sont ensuite transmises à intervalle régulier au centre de commande central. La tâche principale de ce dernier consiste alors essentiellement à déterminer l'identité de cette plate-forme inconnue. Pour ce faire, les informations reçues sont tour à tour étudiées et combinées au niveau de chacun des attributs. Puis, ces résultats intermédiaires sont fusionnés à nouveau pour obtenir le résultat global représenté par la plate-forme

dont les chances d'être observées sont les plus élevées. Ce scénario se poursuit jusqu'à l'atteinte d'un équilibre, l'identité de la plate-forme la plus probable est alors déduite. Notons qu'en tout temps, le centre de contrôle est en mesure de référer à la base de données de plates-formes pour obtenir des informations complémentaires.

Le présent projet traite ainsi en profondeur de la conception du centre de contrôle. Une toute nouvelle gamme de ces systèmes experts est ainsi développée, basée sur l'univers de la logique floue et de ses dérivés. Ils consistent principalement en une modernisation et une application des moteurs d'inférence floue, abordés au chapitre précédent, à l'identification de cibles.

#### **2.4.1. Moteurs d'inférence floue privilégiés**

Des limites et contraintes strictes encadrent la présente recherche. Un choix judicieux a donc dû être établi quant aux différents systèmes experts flous à privilégier et aux avenues à explorer. Globalement, les efforts ont été concentrés autour des moteurs d'inférence floue jugés les plus populaires dans le domaine du génie selon Wang (1997, section 7), soit les moteurs du produit et du minimum. Cependant une tangente statistique a également été ajoutée au projet. En effet, un tout nouveau moteur a été conçu, qui s'inspire grandement de la statistique bayésienne. Malgré une structure non conventionnelle, son fonctionnement satisfait toujours les standards des systèmes experts flous. On le qualifie de moteur d'inférence floue bayésien.

La présente recherche s'attarde donc à la transformation de ces trois moteurs d'inférence floue de la théorie vers la pratique, c'est-à-dire l'application à l'identification hiérarchique de cibles.

## 2.5. TRANSFORMATION ET APPLICATION DES MOTEURS D'INFÉRENCE FLOUE À L'IDENTIFICATION DE PLATES-FORMES

La présente section s'attarde principalement aux différentes étapes de transformation subies par les moteurs d'inférence floue au cours de leur adaptation à l'identification de cibles. Les différents thèmes abordés au précédent chapitre concernant les constituants d'un moteur seront systématiquement repris et décortiqués. Leur assemblage formera la structure globale des moteurs d'inférence floue du minimum, du produit et de Bayes, présentés en toute fin de section.

### 2.5.1. Base de règles floues si-alors

Tout d'abord, le statut de la base de règles floues si-alors, appliquée à l'identification de cibles, diffère légèrement de celui d'un moteur d'inférence floue théorique. Les différentes règles floues, notées  $Ru^{(l)}$ , ne possèdent maintenant plus systématiquement la forme prédéterminée suivante :

$$Ru^{(l)} : Si < proposition floue > , alors < proposition floue > .$$

Au contraire, elles se voient plutôt remplacées par les observations des senseurs. En effet, l'identification étant hiérarchique, les senseurs transmettent directement au centre de contrôle l'identité d'une plate-forme ou d'un groupe de celles-ci. Elle devient ainsi la nouvelle étape d'évaluation des règles floues si-alors où le niveau identifié dans l'arbre de détection d'un attribut désigne la partie *si* d'une règle et les plates-formes correspondantes, la partie *alors*. On représente désormais sa nouvelle structure de la façon suivante :

$$Ru^{(l)} : Si < niveau de détection observé > , alors < plates-formes identifiées > .$$

Les résultats de cette identification constituent l'étape préliminaire dans la transformation des règles floues. Plusieurs autres la caractérisent.

Premièrement, la ou les plates-formes identifiées permettent également de retracer les caractéristiques de l'attribut concerné. Sa plage de validité peut ainsi être complètement déterminée pour un rapport de capteur donné. Cette étape de l'évaluation des règles si-alors équivaut, en partie, à une recherche d'attributs à l'intérieur de la base de données de plates-formes.

Une fois cette étape terminée, la véritable identification débute. Les probabilités de détection de chacune des plates-formes de la base de données sont évaluées. Pour un capteur ou un attribut donné, l'intégrale ou la sommation de la fonction d'appartenance de chaque plate-forme sur la plage de validité de l'attribut fournit ce résultat. Dans le cas de l'attribut *vitesse*, une variable continue, la probabilité de détection d'une plate-forme donnée  $y_i$  suite au rapport d'un capteur  $l$  s'exprime

$$\mu_{\text{capteur}_l \text{ vitesse}}(y_i) = \int_{x \in U_o} \mu_{\text{plate-forme vitesse}}(x | v_{\min}(y_i), v_{\max}(y_i)) dx,$$

où  $l$  correspond à un capteur donné,  $y_i$ , à une des plates-formes de la base de données,  $v_{\min}(y_i)$  et  $v_{\max}(y_i)$  représentent respectivement ses vitesses minimale et maximale et  $U_o$ , la plage de validité de l'attribut, c'est-à-dire les vitesses valides.

L'expression de l'attribut *nombre d'émetteurs*, plutôt de type discret, diffère légèrement de la précédente et se représente par :

$$\mu_{\text{capteur}_l \text{ nombre d'émetteurs}}(y_i) = \sum_{x \in U_o} \mu_{\text{plate-forme nombre d'émetteurs}}(x | z(y_i)),$$

où  $l$  représente un capteur donné,  $y_i$  correspond à une des plates-formes de la base de données,  $U_o$  représente la plage de validité de l'attribut, c'est-à-dire les nombres d'émetteurs valides et  $z(y_i)$ , le nombre d'émetteurs de la plate-forme  $y_i$ . Toutefois, si la détection correspond à une plate-forme spécifique, les règles

du jeu sont strictes. L'ensemble  $U_o$  correspond alors au singleton représentant uniquement le nombre d'émetteurs de cette plate-forme.

Une légère variante est appliquée au système dans le cas de l'analyse de l'attribut *type des émetteurs*. Fortement dépendant du précédent attribut, il ne nécessite aucune fuzzification ni modélisation sous forme de fonction d'appartenance. Ainsi, suite à la réception d'un rapport de senseur concernant les émetteurs, la probabilité de détection de chaque plate-forme est déduite de la possession ou non des types d'émetteurs identifiés. L'évaluation débute par l'assignation d'un poids de départ constant à chaque plate-forme. Cette précaution prévient l'assignation de probabilités nulles qui pourraient perturber le processus d'identification de cibles. Par la suite, les divers types d'émetteurs détectés sont analysés un à un. Si une plate-forme possède un émetteur, elle se voit ajoutée un poids constant additionnel. Ce processus se poursuit ainsi pour tous les émetteurs et mène à l'assignation de l'ensemble des probabilités. L'évaluation de celles-ci est cependant restreinte à l'identification sommaire d'une plate-forme unique. Dans le cas de l'identification d'un groupe de plates-formes (une catégorie), la détection n'est pas possible, aucune étape préalable de fuzzification n'étant réalisable.

Finalement, une règle floue si-alors,  $Ru^{(l)}$ , appliquée à l'identification de cibles hiérarchique se représente sous la forme suivante :

$$Ru^{(l)} : \mu_{\text{senseur}_{i_k}}(y),$$

où  $y$  correspond à chacune des plates-formes de la base de données et où  $l$  représente un senseur donné, dédié à un attribut précis  $k$ .

**Exemple 2.5.1.** *Poursuivons l'exemple précédent 2.3.4, à la lumière de l'application des règles floues à l'identification de cibles. La figure 2.5.1 présente les probabilités de détection des plates-formes de la base de données simplifiée suite à*

*la réception au centre de contrôle de divers rapports de senseurs confirmant tous la présence de la plate-forme numéro trois. Le graphique de gauche (a) illustre tout d'abord les probabilités de détection suite à la réception d'un rapport unique sur la vitesse tandis que les suivants présentent des résultats similaires, mais concernant plutôt les attributs nombre d'émetteurs (b) et type des émetteurs (c).*

*Dans le cas de l'attribut vitesse, toutes les plates-formes possédant une vitesse maximale inférieure à celle de la plate-forme numéro trois sont détectées avec une probabilité supérieure aux autres. Il s'agit des plates-formes un, deux et trois, membres de la catégorie vitesse très lente (voir tableau 2.3.1). Elles sont toutes aptes à se déplacer à une vitesse se situant à l'intérieur de la plage de vitesses de la plate-forme trois. En ce qui concerne l'attribut nombre d'émetteurs, les plates-formes possédant le même nombre d'émetteurs que la plate-forme trois se voient accordées une probabilité de détection supérieure, soit les plates-formes trois et quatre. Finalement, plus une plate-forme détient un grand nombre d'émetteurs en commun avec la plate-forme trois, plus sa probabilité de détection au niveau du type des émetteurs se trouve élevée.*

### **2.5.2. Pondération des capteurs**

Dans un scénario d'identification de cibles, les informations transmises d'un senseur au centre de contrôle ne sont pas toujours exactes. Les senseurs peuvent, par exemple, subir des dommages matériels ou encore être volontairement piégés. Pour refléter la confiance du centre de contrôle en chacun d'eux, une pondération leur a été ajoutée. Elle assure que les données reçues de chaque senseur possèdent un degré de confiance qui rélète équitablement sa précision et sa fiabilité. La pondération affecte, en fait, l'impact des probabilités primaires de détection déduites des observations d'un senseur dans le processus global d'identification de cibles.

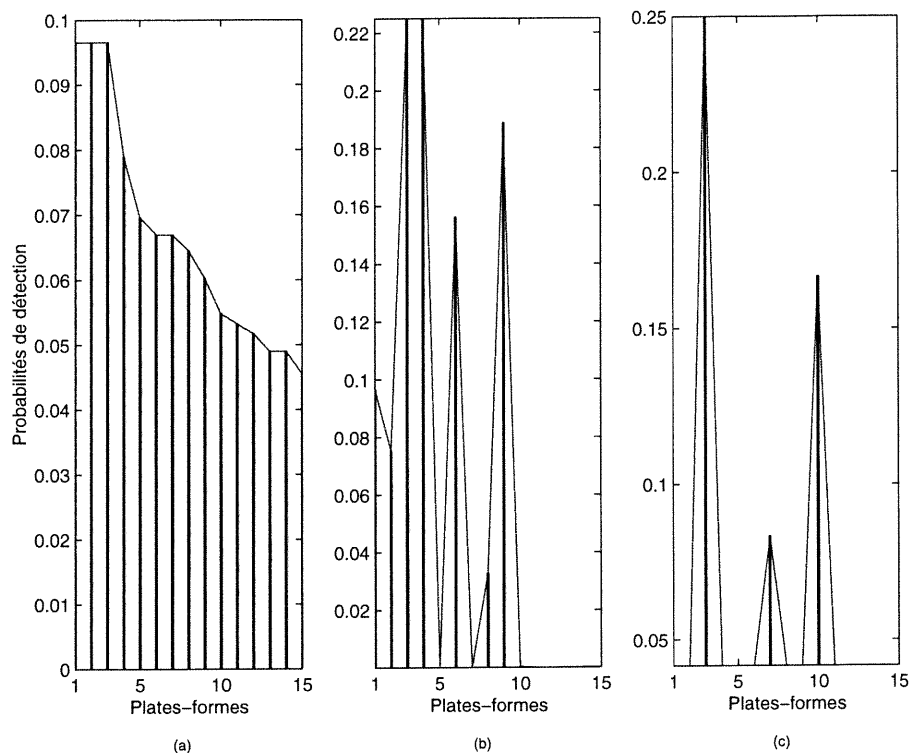


FIG. 2.5.1 — Probabilités de détection primaires des plates-formes suite à des rapports sur la vitesse (a), le nombre d'émetteurs (b) et leur type (c) favorisant la plate-forme numéro trois

Ainsi plus la pondération du capteur est faible, plus l'importance accordée à ses observations diminue.

La pondération ajoutée au capteur, tel que confirmée par Genest et Zidek (1986), se représente donc de la façon suivante :

$$(\mu_{\text{capteur}_{l_k}}(y))^{w_{l_k}},$$

où  $y$  correspond à une des plates-formes de la base de données,  $l$  représente un capteur donné dédié à un attribut précis  $k$  et  $w_{l_k}$ , sa pondération.



Généralement la pondération d'un senseur est déterminée par des experts aptes à évaluer efficacement toute situation. Par défaut cependant, elle est fixée à l'unité pour tout senseur.

### 2.5.3. Modus ponens généralisé

L'essence même du modus ponens généralisé, la règle d'inférence privilégiée par l'ensemble des systèmes experts flous, diffère une fois appliquée à l'identification de cibles. Sa structure, permettant l'inférence d'une conclusion suite à l'étude d'une série de prémises (voir équation (1.5.9)), sert désormais d'opérateur de combinaison. Il effectue, en effet, la fusion des rapports des senseurs à des niveaux multiples dans le but ultime de déterminer la meilleure plate-forme, soit celle dont les probabilités de détection sont les plus élevées.

Dans un premier temps, le modus ponens fusionne les probabilités de détection de chacune des plates-formes au niveau d'un même attribut. Des probabilités de détection intermédiaires en sont ainsi déduites. Puis, il termine par une combinaison globale de ces probabilités intermédiaires pour obtenir la distribution finale de probabilité. Sa structure appliquée complète se présente ainsi :

$$\mu_{CSU}(y) = \sup_{k \in K} t_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}],$$

où  $CSU$  signifie combinaison de senseurs avec opérateur d'union floue ( $\sup$ ),  $y$  correspond à une plate-forme de la base de données,  $t$ , à un opérateur d'intersection floue (voir section 1.3.2.3 pour de plus amples explications) et finalement  $w_l$ , désigne la pondération de chaque senseur  $l$ , membre de la catégorie  $L_k$ . Cette dernière représente les senseurs dédiés à  $k$ , un attribut donné contenu dans  $K$ , l'ensemble de tous les attributs possibles. L'opérateur d'intersection floue  $t$  représente la combinaison intermédiaire au niveau de chaque attribut tandis que la fonction  $\sup$  se charge de la combinaison globale.

Une seconde version plutôt radicale et remodelée du modus ponens flou a été également développée. La transformation vise principalement l'étape finale, soit la combinaison globale des résultats au niveau de chaque attribut. Celle-ci, jusque là pratiquée par un opérateur d'union floue de type maximum (sup), opte maintenant pour l'utilisation d'un opérateur d'intersection floue de type minimum (inf). Ce changement vise principalement à accentuer la rigueur de l'inférence finale. La seconde image non unioniste du modus ponens se présente ainsi :

$$\mu_{CSI}(y) = \inf_{k \in K} t_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}],$$

où *CSI* signifie combinaison de senseurs avec opérateur d'intersection floue (inf), *y* correspond à une plate-forme de la base de données, *t*, à un opérateur d'intersection floue (voir section 1.3.2.3 pour de plus amples explications), et finalement  $w_l$ , désigne la pondération de chaque senseur *l*, membre de la catégorie  $L_k$ . Cette dernière représente les senseurs dédiés à *k*, un attribut donné contenu dans *K*, l'ensemble de tous les attributs possibles.

L'opérateur d'intersection floue *t* représente la combinaison intermédiaire au niveau de chaque attribut tandis que la fonction inf représente la combinaison globale.

**Exemple 2.5.2.** *Illustrons maintenant le travail intermédiaire et global exécuté par le modus ponens généralisé avec opérateur d'union et d'intersection floues. La figure 2.5.2 présente d'abord pour le modus ponens avec union floue les probabilités intermédiaires de détection des plates-formes aux rapports de senseurs de l'exemple précédent 2.5.1 concernant l'attribut vitesse (a), l'attribut nombre d'émetteurs (b) et le type des émetteurs (c). Le résultat de la combinaison globale complète l'illustration (d). La figure 2.5.3 présente les mêmes informations, mais*

obtenues du *modus ponens* généralisé avec intersection floue. Dans chacun des cas, la plate-forme numéro trois est la véritable plate-forme observée.

Comme prévu, les *modus ponens* avec opérateur d'union et d'intersection floues présentent un comportement identique au niveau intermédiaire, soit la combinaison au niveau de chaque attribut. En effet, le rapport de senseurs à ce niveau est unique, la fusion intermédiaire ne transforme donc pas l'évaluation primaire. Il en est cependant tout autrement pour la situation globale ( $d$ ) où les résultats diffèrent d'un modèle à l'autre. Dans les deux cas néanmoins, le maximum de la fonction d'appartenance est atteint pour la plate-forme numéro trois, celle à identifier.

#### 2.5.4. Combinaison des règles si-alors

La combinaison des règles si-alors constitue l'opération terminale dans la schématisation d'un moteur d'inférence floue. Elle confirme les résultats obtenus du *modus ponens* généralisé et complète l'identification de cibles. L'opérateur de combinaison de Mamdani (voir équation (1.5.13)) exécute désormais une tâche unique de recherche du maximum de vraisemblance ou d'identification des plates-formes les plus probables.

À l'aide des résultats globaux obtenus du *modus ponens* généralisé, il détecte les plates-formes parmi toute la base de données de plates-formes qui possèdent les trois probabilités de détection les plus élevées. En cas d'égalité, plus d'une plate-forme peut occuper chacune de ces positions.

Pour ainsi dire, seul l'opérateur de Mamdani trouve vraiment une application directe en identification de cibles. Sa nouvelle structure se définit dans le cas d'un *modus ponens* avec union (voir équation (2.5.1)) et intersection floues (voir

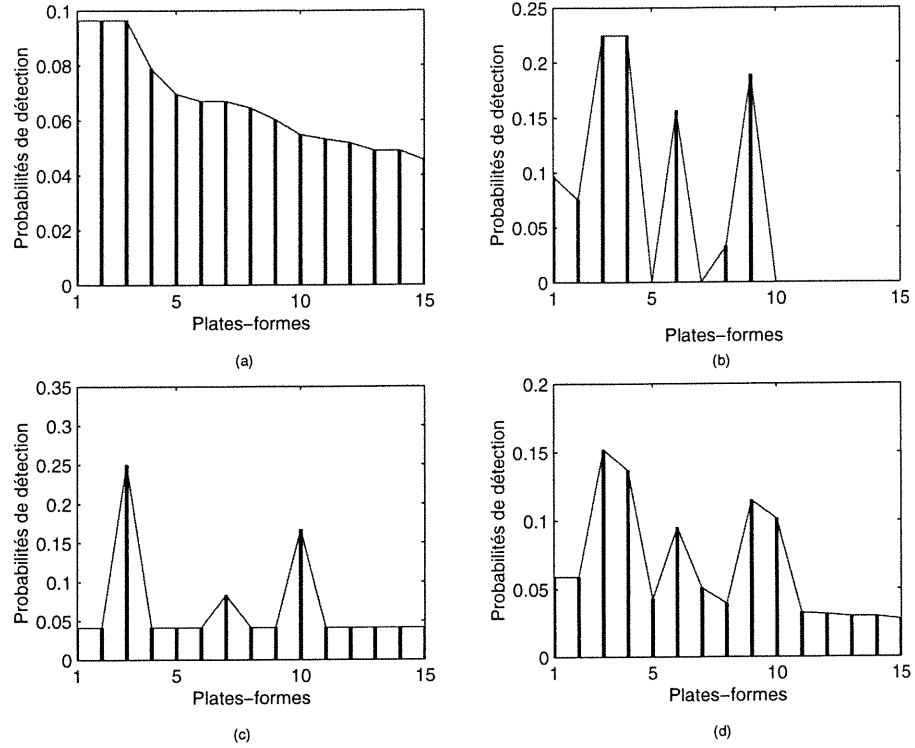


FIG. 2.5.2 –. Probabilités de détection intermédiaires au niveau de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et finalement globales de plates-formes (d) déduites de l'application du modus ponens généralisé avec opérateur d'union floue

équation (2.5.2)) comme suit :

$$\mu_{global_U} = \sup_{y \in V} \left\{ \sup_{k \in K} t_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.1)$$

$$\mu_{global_I} = \inf_{y \in V} \left\{ \inf_{k \in K} t_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.2)$$

où  $global_U$  et  $global_I$  signifie respectivement identification globale et combinaison de senseurs avec opérateur d'union (sup) et d'intersection (inf) floues,  $V$  correspond à l'ensemble des plates-formes de la base de données,  $t$ , à un opérateur d'intersection floue (voir section 1.3.2.3 pour de plus amples explications), et finalement  $w_l$ , désigne la pondération de chaque senseur  $l$ , membre de la catégorie

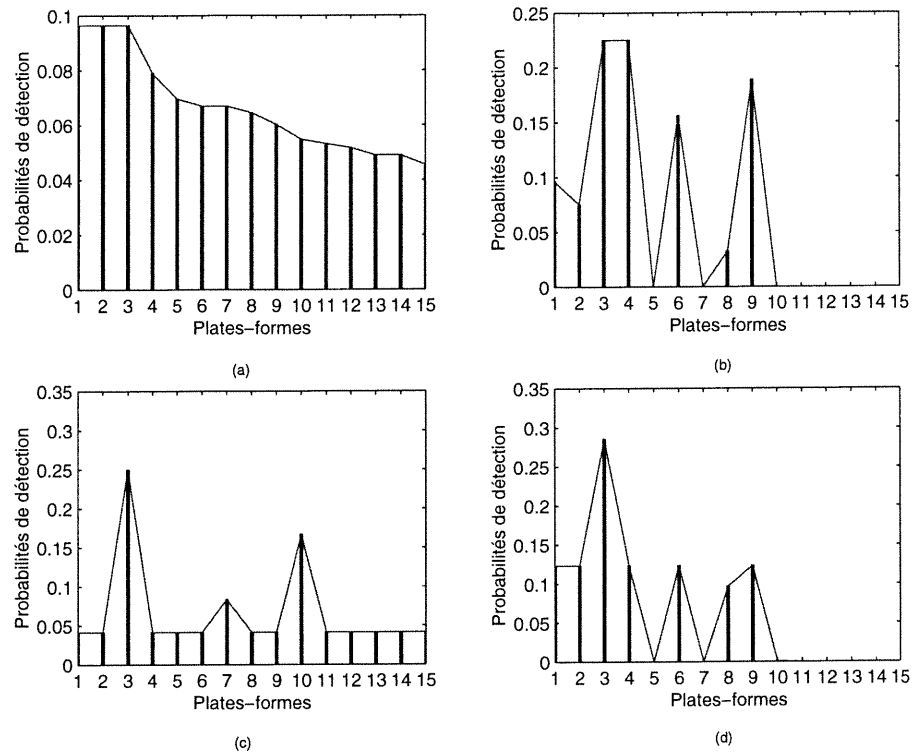


FIG. 2.5.3 — Probabilités de détection intermédiaires au niveau de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et finalement globales de plates-formes (d) déduites de l'application du modus ponens généralisé avec opérateur d'intersection floue

$L_k$ . Cette dernière représente les senseurs dédiés à  $k$ , un attribut donné contenu dans  $K$ , l'ensemble de tous les attributs possibles.

**Exemple 2.5.3.** Le tableau suivant (voir tableau 2.5.1) présente les plates-formes les plus probables après analyse des différentes probabilités globales de détection tirées de l'exemple précédent 2.5.2. Il traite des résultats obtenus dans le cas d'un moteur d'inférence floue avec modus ponens généralisé utilisant des opérateurs d'union et d'intersection floues. La plate-forme numéro trois représente la véritable plate-forme inconnue.

*On remarque que les résultats diffèrent d'un modèle à l'autre tant en termes de plates-formes détectées que de probabilités obtenues. Dans les deux cas néanmoins, la plate-forme numéro trois, celle à identifier, possède la probabilité de détection la plus importante. Le moteur avec opérateur d'intersection floue se veut plus généreux avec un plus grand nombre de plates-formes détectées parmi les plus probables.*

TAB. 2.5.1 —. Résultats de l'identification de la cible numéro trois

Modus Ponens généralisé		Cible(s) plus probable(s)	Seconde(s) cibles plus probable(s)	Troisième(s) cibles plus probable(s)
Union floue	Plates-formes	3	4	9
	Probabilités(%)	15,19	13,67	11,47
Intersection floue	Plates-formes	3	1, 2, 4, 6 et 9	8
	Probabilités(%)	28,56	12,33	9,66

### 2.5.5. Moteurs d'inférence floue résultant

La présente sous-section introduit la structure complète des systèmes experts flous responsables de l'identification de cibles. Leur nouvelle charpente sera abondamment illustrée et analysée. Tout d'abord, choisis pour leur vaste emploi dans l'industrie, le moteur d'inférence floue du produit et celui du minimum seront synthésés à la lumière des dernières modifications apportées à leurs composantes. Puis un tout nouveau moteur d'inférence floue, inspiré cette fois de la statistique moderne, sera développé. La différence entre ce moteur de type bayésien et les précédents sera également abordé.

### 2.5.5.1. Moteur d'inférence floue du produit

La nouvelle structure du moteur d'inférence floue du produit imite sensiblement celle du moteur théorique illustrée par l'équation (1.5.17), mais se complexifie par son application à l'identification de cibles. Cette complexité réside dans l'intégration des notions d'identification hiérarchique, d'observations de capteurs et de base de données de plates-formes au concept plus théorique de moteur d'inférence floue.

Tel que démontré précédemment, les règles si-alors, anciennement réservées uniquement à l'analyse logique, viennent désormais remplacer directement les observations des capteurs. Différenciation encore plus grande, les capteurs, des considérations purement pratiques, forcent l'emploi d'une pondération pour accentuer le réalisme. D'un autre côté, le modus ponens généralisé assure toujours l'inférence en effectuant séquentiellement une combinaison partielle des résultats au niveau d'un même attribut suivie d'une combinaison plus générale. D'ailleurs, le moteur d'inférence du produit se démarque principalement par une combinaison partielle privilégiant le produit à titre d'opérateur d'intersection floue (voir équation (1.3.8)). La combinaison globale, quant à elle, permet de classifier le moteur du produit en deux gammes distinctes : un premier type composé du modus ponens avec opérateur d'union floue (voir équation (2.5.3)) et un second intégrant davantage l'intersection floue (voir équation (2.5.4)) au processus d'inférence. Ce scénario se termine par une étape de maximisation des probabilités de détection, autrefois nommée combinaison des règles si-alors. Finalement, l'identité formelle de la plate-forme inconnue se trouve déterminée.

Les deux versions du moteur d'inférence floue du produit se présentent ainsi :

$$\mu_{prod_U} = \sup_{y \in V} \left\{ \sup_{k \in K} \prod_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.3)$$

$$\mu_{prod_I} = \sup_{y \in V} \left\{ \inf_{k \in K} \prod_{l \in L_K} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.4)$$

où  $prod_U$  et  $prod_I$  désignent respectivement le moteur d'inférence floue du produit avec opérateur d'union (sup) et d'intersection (inf) floues,  $V$  correspond à l'ensemble des plates-formes de la base de données, et  $w_l$ , désigne la pondération de chaque senseur  $l$ , membre de la catégorie  $L_k$ . Cette dernière représente les senseurs dédiés à  $k$ , un attribut donné contenu dans  $K$ , l'ensemble de tous les attributs possibles.

**Exemple 2.5.4.** *De la même manière que l'exemple précédent 2.5.3, les figures suivantes illustrent le résultat d'un scénario d'identification de cibles effectué par le moteur d'inférence du produit avec union et intersection floue. Dans tous les cas, la plate-forme numéro trois s'avère être la véritable plate-forme à identifier.*

*La première figure, Figure 2.5.4, présente pour le moteur avec union floue les probabilités intermédiaires de détection des plates-formes suite à la réception de quelques rapports de senseurs concernant l'attribut vitesse (a), l'attribut nombre d'émetteurs (b) et le type des émetteurs (c). Le résultat de la combinaison globale complète l'illustration (d). La figure 2.5.5 présente les mêmes informations, mais obtenues du modus ponens avec intersection floue. Les graphiques (a), tous les modèles confondus, combinent quatre rapports de senseurs identifiant le type surface, la catégorie de vitesse numéro un et deux fois la plate-forme numéro trois. Les graphiques (b) combinent également quatre rapports de senseurs qui identifient la catégorie de nombre d'émetteurs numéro deux et trois fois la plate-forme numéro trois. L'évaluation du type des émetteurs (c) entre en opération seulement dans ces trois derniers cas, lors de l'identification d'une plate-forme spécifique.*

*On remarque que les résultats diffèrent pour chacun des modèles seulement au niveau de la combinaison globale. Ceci est causé principalement par l'utilisation*



de mêmes observations et surtout d'un opérateur de combinaison intermédiaire identique, le produit flou. Les opérateurs utilisés au niveau de la combinaison globale diffèrent, ainsi va le résultat final. Par contre, la plate-forme numéro trois est bien celle dont les probabilités de détection sont les plus importantes dans toutes les situations. Le moteur avec union floue reste plus optimiste dans sa décision finale, plusieurs plates-formes possèdent, en effet, une probabilité finale élevée. Ces dernières sont d'ailleurs résumées au tableau 2.5.2.

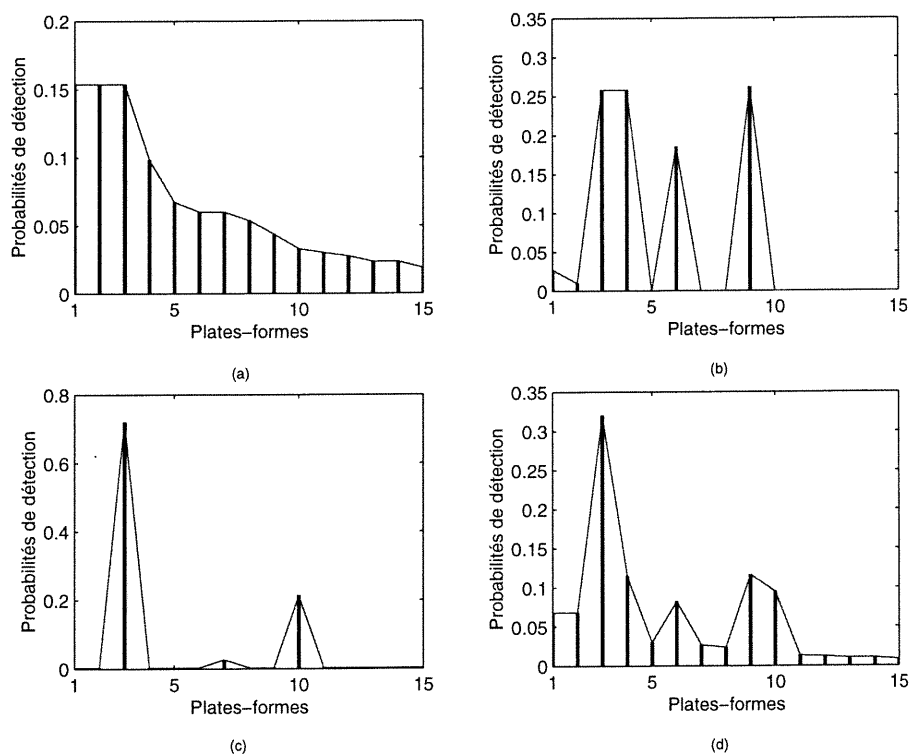


FIG. 2.5.4 -. Détections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du produit avec union floue

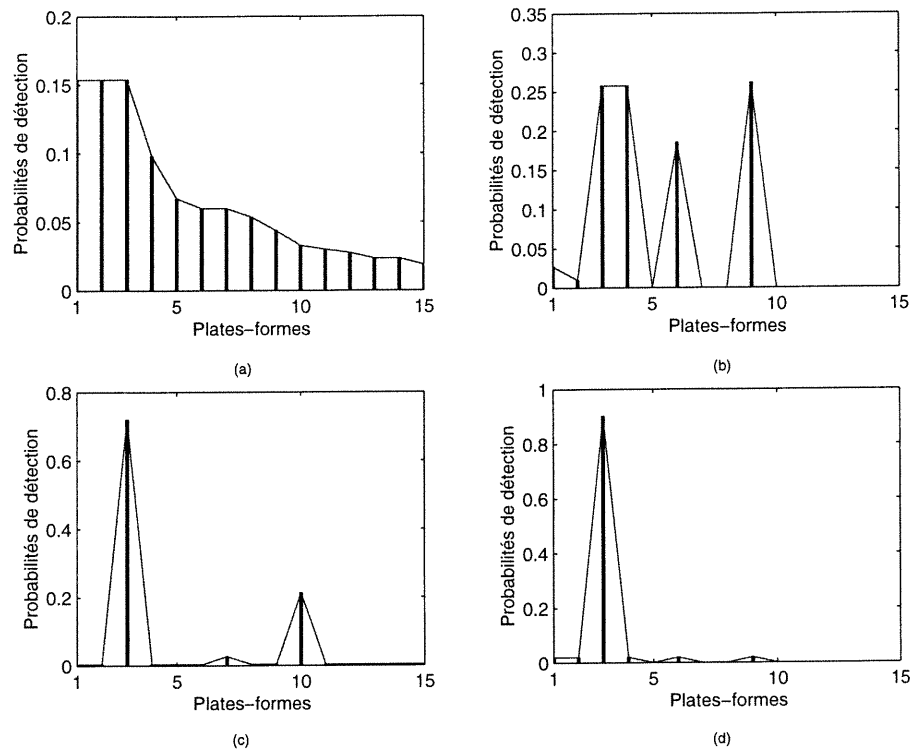


FIG. 2.5.5 –. *Détections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du produit avec intersection floue*

TAB. 2.5.2 –. *Résultats de l'identification de la cible numéro trois avec moteur d'inférence floue du produit*

Modus Ponens généralisé		Cible(s) plus probable(s)	Seconde(s) cibles plus probable(s)	Troisième(s) cibles plus probable(s)
Union floue	Plates-formes	3	9	4
	Probabilités(%)	32,00	11,62	11,47
Intersection floue	Plates-formes	3	1, 2, 4, 6 et 9	8
	Probabilités(%)	90,16	1,95	0,07

### 2.5.5.2. Moteur d'inférence du minimum

Le moteur d'inférence floue du minimum ressemble grandement au précédent, mais avec une légère variante apportée à la combinaison partielle lors du processus

d'inférence. Il privilégie l'emploi du minimum à titre d'opérateur d'intersection floue (voir équation (1.3.7)), mais conserve une charpente tout-à-fait similaire. Il se présente également en deux modèles : un premier constitué du modus ponens avec opérateur d'union floue (voir équation (2.5.5)) et un second intégrant plutôt l'intersection floue (voir équation (2.5.6)).

Les deux versions de ce moteur d'inférence floue du minimum s'expriment ainsi :

$$\mu_{min_U} = \sup_{y \in V} \left\{ \sup_{k \in K} \min_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.5)$$

$$\mu_{min_I} = \sup_{y \in V} \left\{ \inf_{k \in K} \min_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.6)$$

où  $min_U$  et  $min_I$  désignent respectivement moteur d'inférence floue du minimum avec opérateur d'union (sup) et d'intersection (inf) floues,  $V$  correspond à l'ensemble des plates-formes de la base de données, et  $w_l$ , désigne la pondération de chaque senseur  $l$ , membre de la catégorie  $L_k$ . Cette dernière représente les senseurs dédiés à  $k$ , un attribut donné contenu dans  $K$ , l'ensemble de tous les attributs possibles.

**Exemple 2.5.5.** *Les figures suivantes présentent respectivement les mêmes résultats que dans l'exercice précédent 2.5.4 avec opérateur d'union tout d'abord (voir Figure 2.5.6) et intersection floue par la suite (voir Figure 2.5.7). Cependant, le moteur d'inférence floue du minimum remplace maintenant celui du produit. Dans chacun des cas, la plate-forme numéro trois s'avère être la véritable plate-forme observée.*

*Chacune des figures présente les probabilités intermédiaires de détection des plates-formes suite à la réception de quelques rapports de senseurs concernant l'attribut vitesse (a), l'attribut nombre d'émetteurs (b) et le type des émetteurs (c). Puis termine avec le résultat de la combinaison globale (d). Les graphiques*

(a), tous les modèles confondus, combinent quatre rapports de senseurs identifiant le type surface, la catégorie de vitesse numéro un, et deux fois la plate-forme numéro trois. Les graphiques (b) combinent également quatre rapports de senseurs qui identifient la catégorie de nombre d'émetteurs numéro deux et trois fois la plate-forme numéro trois. L'évaluation du type des émetteurs (c) entre en opération seulement dans ces trois derniers cas, lors de l'identification d'une plate-forme spécifique.

On remarque que les résultats diffèrent pour chacun des modèles seulement au niveau de la combinaison globale. Ceci est causé principalement par l'utilisation de mêmes observations et surtout d'un opérateur de combinaison intermédiaire identique, le minimum flou. Les opérateurs utilisés au niveau de la combinaison globale diffèrent et influencent ainsi le résultat final même si la plate-forme numéro trois est bien celle dont les probabilités de détection sont les plus importantes dans toutes les situations bien que la combinaison de l'attribut nombre d'émetteurs lui soit défavorable dans chacun des cas. Le moteur avec union floue reste, encore ici, plus optimiste dans sa décision finale, plusieurs plates-formes possèdent, en effet, une probabilité finale élevée. Cependant, ce moteur se veut globalement beaucoup plus optimiste que celui du produit. Les probabilités globales sont d'ailleurs résumées au tableau 2.5.3.

### 2.5.5.3. Moteur d'inférence bayésien

Le présent système expert se distingue des précédents par son audace. En effet, la ligne dure des moteurs d'inférence est quelque peu délaissée pour faire place aux statistiques modernes. À l'instar de la statistique bayésienne, cette nouvelle avenue privilégie l'emploi du produit flou (voir équation (1.3.8)) dans toutes les

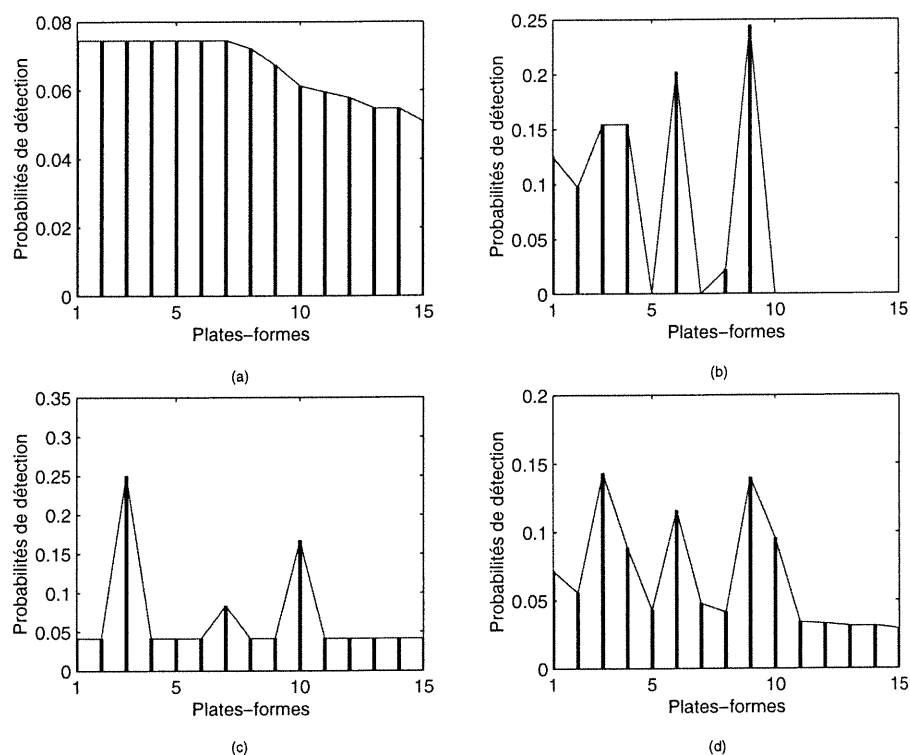


FIG. 2.5.6 – Detections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du minimum avec union floue

TAB. 2.5.3 – Résultats de l'identification de la cible numéro trois avec moteur d'inférence floue du minimum

Modus Ponens généralisé		Cible(s) plus probable(s)	Seconde(s) cibles plus probable(s)	Troisième(s) cibles plus probable(s)
Union floue	Plates-formes	3	9	6
	Probabilités(%)	14,30	13,98	11,56
Intersection floue	Plates-formes	3	1, 2, 4, 6, et 9	8
	Probabilités(%)	24,43	13,66	7,16

étapes du processus d'inférence. Il ressemble ainsi fortement à un moteur d'inférence floue du produit, mais avec une combinaison globale des probabilités de

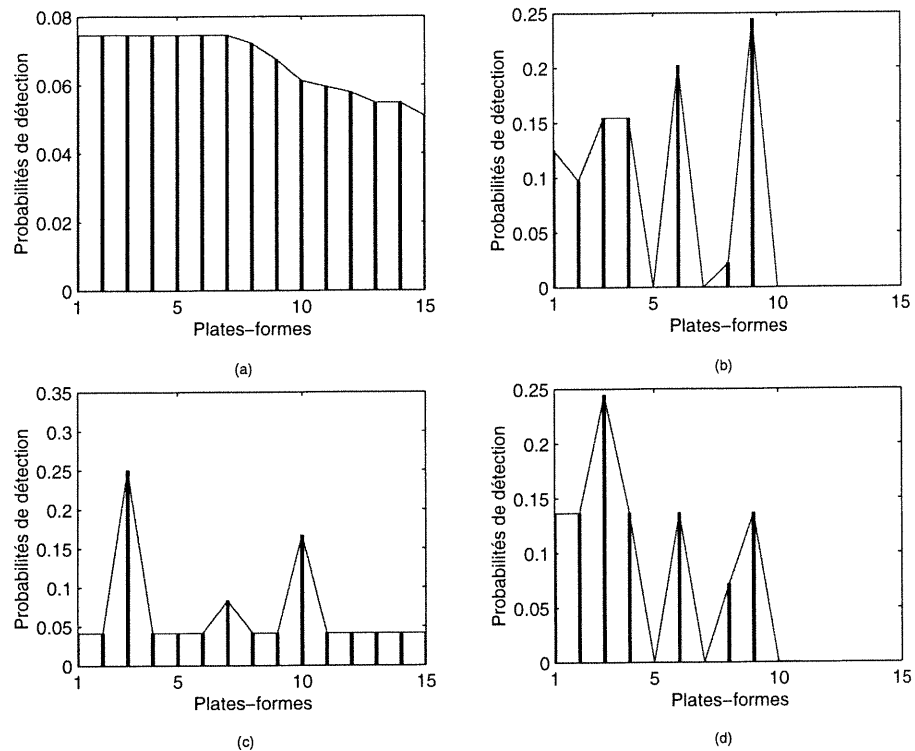


FIG. 2.5.7 –. *Détections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du minimum avec intersection floue*

détection assurée également par un opérateur du produit. Il se nomme d'ailleurs moteur d'inférence flou bayésien pour le différencier de ce précédent moteur et accentuer la ressemblance entre sa forme mathématique et celle du numérateur de la densité *a posteriori* de Bayes.

La charpente du moteur d'inférence de Bayes, désormais uniquement spécifiée par l'équation (2.5.7), se présente de la façon suivante :

$$\mu_{Bayes} = \sup_{y \in V} \left\{ \prod_{k \in K} \prod_{l \in L_k} [(\mu_{senseur_l}(y))^{w_l}] \right\}, \quad (2.5.7)$$

où  $V$  correspond à l'ensemble des plates-formes de la base de données, et  $w_l$ , désigne la pondération de chaque senseur  $l$ , membre de la catégorie  $L_k$ . Cette

dernière représente les senseurs dédiés à  $k$ , un attribut donné contenu dans  $K$ , l'ensemble de tous les attributs possibles.

**Exemple 2.5.6.** *La figure suivante (voir figure 2.5.8) présente sensiblement les mêmes résultats que dans les exercices précédents (voir exemples 2.5.4 et 2.5.5). Cependant, le moteur d'inférence floue bayésien figure maintenant à titre de système expert. La figure comprend quatre graphiques distincts dont l'un illustre les probabilités intermédiaires de détection des plates-formes suite à la réception de quelques rapports de senseurs concernant l'attribut vitesse (a), un second, l'attribut nombre d'émetteurs (b) et un troisième, l'attribut type des émetteurs (c). Le résultat de la combinaison globale complète l'illustration (d). Le graphique (a) combine quatre rapports de senseurs identifiant le type surface, la catégorie de vitesse numéro un et deux fois la plate-forme numéro trois. Le graphique (b) combine également quatre rapports de senseurs qui identifient la catégorie de nombre d'émetteurs numéro deux et trois fois la plate-forme numéro trois. L'évaluation du type des émetteurs (c) entre en opération seulement dans ces trois derniers cas, lors de l'identification d'une plate-forme spécifique.*

*Dans ce scénario, la plate-forme numéro trois est la véritable plate-forme observée et par conséquent celle dont les probabilités de détection sont les plus importantes. Du côté de la combinaison intermédiaire, les résultats (a, b et c) sont identiques à ceux obtenus dans la cas du moteur du produit puisque le même opérateur est utilisé, le produit flou. Ce moteur se distingue notamment des précédents par sa conclusion sévère, la plate-forme numéro trois se veut de très loin la plus plausible. Les probabilités de détections des plates-formes les plus probables sont présentées au tableau 2.5.4.*

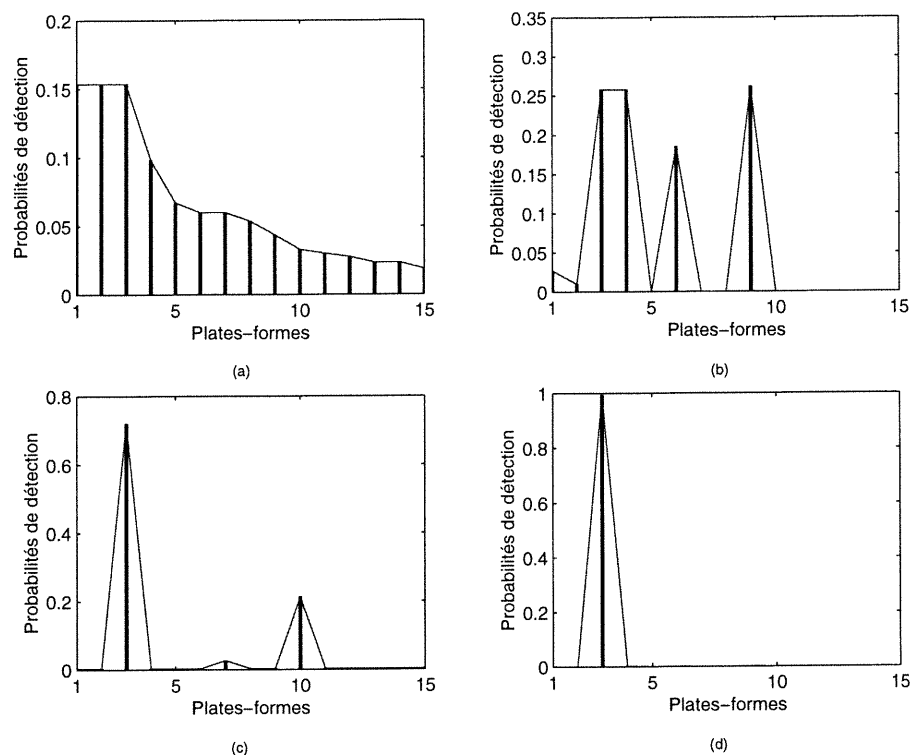


FIG. 2.5.8 – Detections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence bayésien

TAB. 2.5.4 – Résultats de l'identification de la cible numéro trois avec moteur d'inférence floue bayésien

Modus Ponens généralisé		Cible(s) plus probable(s)	Seconde(s) cibles plus probable(s)	Troisième(s) cibles plus probable(s)
Produit flou	Plates-formes	3	4	9
	Probabilités(%)	99,38	0,29	0,13

## 2.6. RAISONNEMENT APPROCHÉ

Parvenir à effectuer efficacement l'identification de cibles, voilà l'objectif majeur poursuivi par les moteurs d'inférence floue. Mais quels sont véritablement les



fondements de ce raisonnement approché? Là figure le sujet même de la présente section.

Le fonctionnement des systèmes experts flous est régi par deux idéologies fort différentes. La première, plus répandue, traite généralement des ensembles mathématiques classiques, mais peut également être étendue à l'univers flou. On la qualifie de loi de probabilité conventionnelle. L'autre alternative, la théorie des possibilités, traitée en profondeur dans le précédent chapitre, se veut inapte à modéliser le hasard, mais parfaitement adaptée aux réalités de la logique floue. Les moteurs d'inférence floue présentés jusqu'ici, tous développés du point de vue probabiliste, peuvent tout aussi bien faire l'objet de ce second type d'analyse. La présente section s'attarde donc à ces deux modes de raisonnement approché et à leurs effets sur les systèmes experts flous.

#### 2.6.0.4. *Logique probabiliste*

La logique probabiliste stipule sommairement que l'intégrale d'une fonction continue sur tout son univers possible doit forcément égaler l'unité. Cette condition de normalisation essentielle des fonctions d'appartenance se voit également vérifiée pour toute fonction discrète.

L'un des fondements des systèmes experts flous repose d'ailleurs sur cette technique. Tous les précédents moteurs, que ce soit celui du produit, du minimum ou encore de Bayes, possèdent un modèle probabiliste. Dans chacun des cas, un soin minutieux a été apporté à toute fonction d'appartenance afin d'assurer un respect total de la condition de normalisation. Les probabilités de détection des plates-formes au niveau d'un attribut donné de même que tout résultat découlant d'une combinaison visent systématiquement ce standard.

Les nombreux graphiques présentés dans les exemples 2.5.4, 2.5.5 et 2.5.6 sont basés sur cette logique et illustrent des fonctions d'appartenance où les probabilités de détection des plates-formes somment à un.

#### 2.6.0.5. *Logique possibiliste*

L'idéologie possibiliste se démarque quelque peu de la précédente au niveau de la normalisation. Puisque tout référentiel est absolu, il doit toujours exister au moins un élément qui possède un degré de possibilité ou d'appartenance unitaire (voir définition 1.4.1).

L'autre fondement des systèmes experts flous s'appuie sur cette optique. Les moteurs d'inférence flous ainsi développés possèdent toujours au moins une plate-forme dont la possibilité de détection frôle l'unité au niveau de toute fonction d'appartenance et spécialement à celui des possibilités de détection primaires des plates-formes. Désormais, la sommation ou l'intégrale à un n'est plus une nécessité.

**Exemple 2.6.1.** *Dans le même ordre d'idées que les exemples précédents 2.5.4, 2.5.5 et 2.5.6, les différentes figures suivantes illustrent les résultats des simulations obtenues avec moteurs d'inférence du produit (voir Figure 2.6.1), du minimum (voir Figure 2.6.2) et de Bayes (voir Figure 2.6.3) avec opérateur d'union floue dans les deux premiers cas. Cependant, le fondement possibiliste détrône ici le probabiliste. Chacune des figures présente les possibilités intermédiaires de détection des plates-formes suite à la réception de quelques rapports de senseurs concernant l'attribut vitesse (a), l'attribut nombre d'émetteurs (b) et le type des émetteurs (c). Le résultat de la combinaison globale complète l'illustration (d). La plate-forme numéro trois correspond, sans exception, à la véritable plate-forme observée. Les graphiques (a), tous les modèles confondus, combinent toujours quatre*

*rapports de senseurs identifiant le type surface, la catégorie de vitesse numéro un et deux fois la plate-forme numéro trois. Les graphiques (b) combinent également quatre rapports de senseurs qui identifient la catégorie de nombre d'émetteurs numéro deux et trois fois la plate-forme numéro trois. L'évaluation du type des émetteurs (c) entre en opération seulement dans ces trois derniers cas, lors de l'identification d'une plate-forme spécifique.*

*On remarque que dans tous les graphiques, qui représentent une combinaison intermédiaire ou globale, au moins une des plates-formes possède une possibilité de détection unitaire. De plus, le moteur de Bayes conserve toujours un comportement sévère et sans équivoque. Par contre, la version possibiliste des moteurs d'inférence floue du produit et du minimum est de loin plus optimiste que celle probabiliste avec un apogée atteint au niveau de celui du minimum. Les moteurs restent tous aptes à reconnaître la plate-forme numéro trois, mais seul Bayes la détecte seule au premier rang. Il discrimine, en fait, entre les trois plates-formes plus probables présentant ainsi un jugement à la fois sévère et précis. Finalement, les possibilités de détections des plates-formes les plus possibles sont présentées au tableau 2.6.1.*

*La modélisation de l'attribut vitesse perd son efficacité dans l'univers possibiliste. L'impact de cet attribut est fort limité contrairement à sa version probabiliste. Il serait préférable d'utiliser une loi plus informative pour modéliser le phénomène dans ce cas précis. Dans le présent projet toutefois, cette transformation irait à l'encontre des conditions initiales du problème qui imposaient des lois peu informatives.*

Le précédent chapitre a présenté les multiples facettes de l'application des systèmes experts flous à l'identification de cibles. L'identification hiérarchique, la base de données de plates-formes, les diverses hypothèses de modélisation ainsi

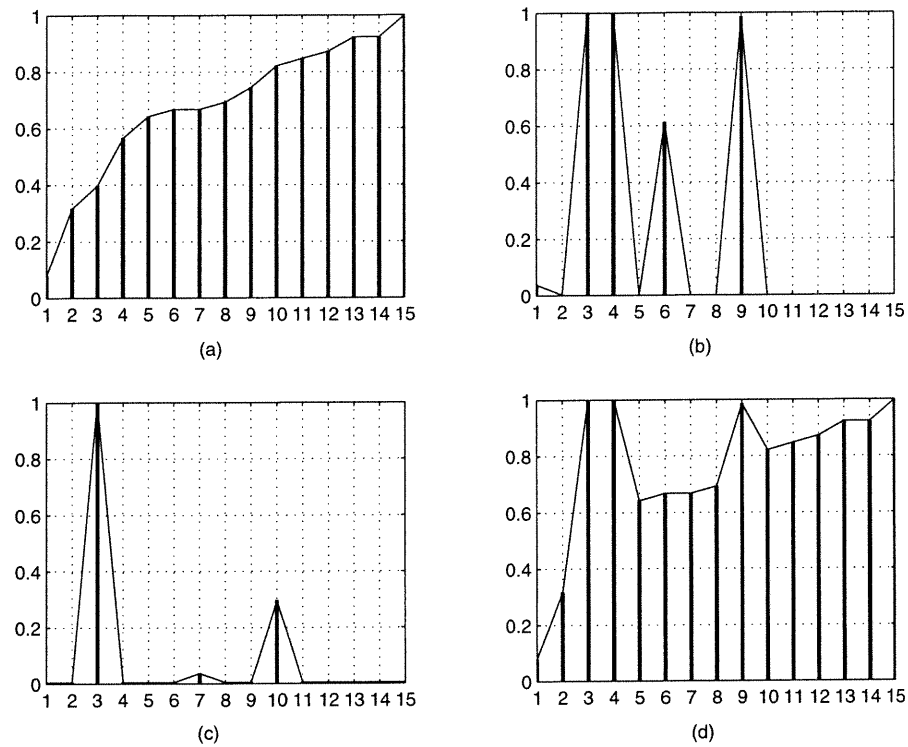


FIG. 2.6.1 –. *Détections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du produit avec union floue et fondement possibiliste*

TAB. 2.6.1 –. *Résultats de l'identification de la cible numéro trois avec moteurs d'inférence floue du produit, du minimum et de Bayes avec opérateur d'union floue (deux premiers cas) fondée sur une approche possibiliste*

Moteur d'inférence floue		Cible(s) plus probable(s)	Seconde(s) cibles plus probable(s)	Troisième(s) cibles plus probable(s)
Produit	Plates-formes	3,4 et 15	9	13 et 14
	Possibilités(%)	100	98,51	92,33
Minimum	Plates-formes	3,9 et 15	13 et 14	12
	Possibilités(%)	100	92,33	87,21
Bayes	Plates-formes	3	9	4
	Possibilités(%)	100	0,0085	0,0066

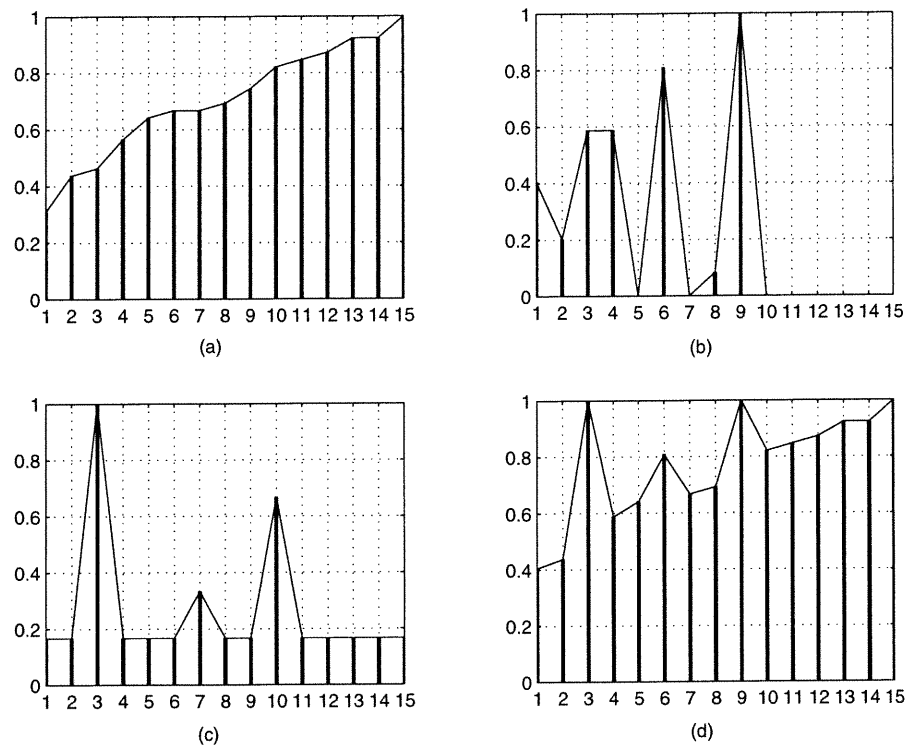


FIG. 2.6.2 –. *Détections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence du minimum avec union floue et fondement possibiliste*

que les moteurs d'inférence flous ont constitué l'essentiel de ce sujet d'études. Pour un aperçu plus détaillé de ces thèmes, veuillez vous référer à Bobrowicz *et al.* (1990); Kummert (1993); Langari *et al.* (1995); Samarasooriya et Varshney (1996). Le chapitre suivant poursuit en présentant les performances des trois moteurs d'inférence floue principaux à la suite d'une série de simulations théoriques. Celles-ci seront par la suite largement analysées et commentées.

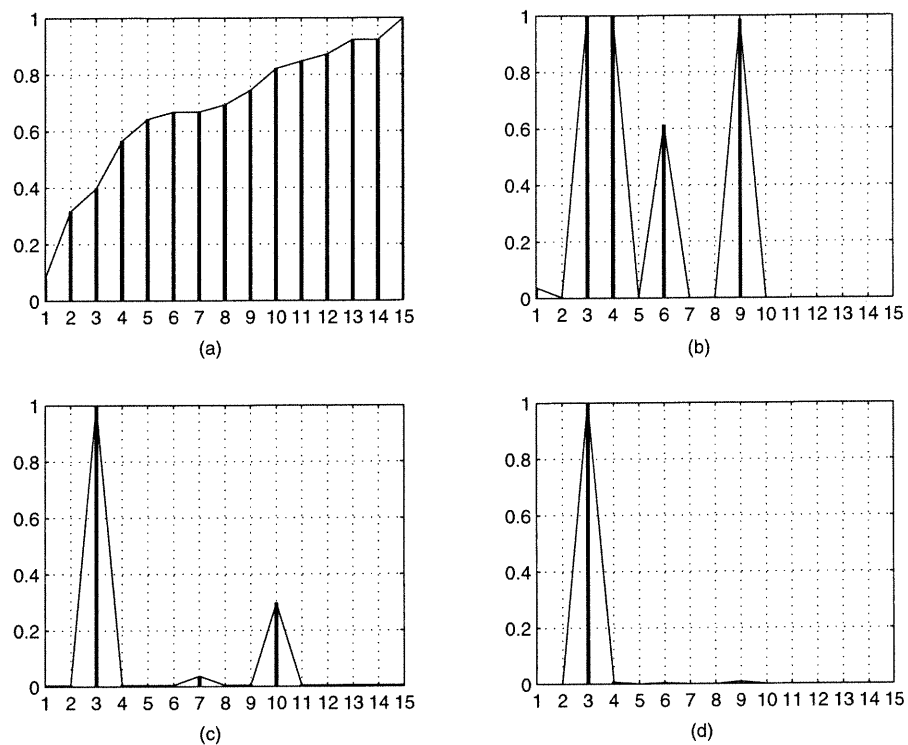


FIG. 2.6.3 — Detections intermédiaires de la vitesse (a), du nombre d'émetteurs (b) et de leur type (c) et inférence globale (d) suite à l'emploi du moteur d'inférence bayésien avec fondement possibiliste

# Chapitre 3

---

## SIMULATION ET ANALYSE

### 3.1. INTRODUCTION

L'orientation de ce dernier chapitre diffère grandement de celle des précédents. Les systèmes experts flous, d'abord aperçus du côté théorique, puis appliqués à leur tâche spécifique, l'identification de cibles, seront maintenant scrutés sous différents angles. En effet, une dizaine de simulations variées permettront d'analyser les trois moteurs d'inférence floue développés précédemment selon leur approche probabiliste et possibiliste respective. Les moteurs du produit, du minimum et bayésien seront ainsi comparés pour parvenir à établir tout d'abord les forces et faiblesses de chacun, mais également pour statuer sur le bien-fondé de l'utilisation de la logique floue en technologie de l'information. Cette procédure permettra donc de valider toute la démarche entreprise depuis le tout début de ce projet.

Le présent chapitre débute par la présentation des multiples modes de fonctionnement des moteurs d'inférence floue suivi d'un survol du déroulement d'une simulation type. Puis il enchaîne avec une panoplie de résultats numériques, obtenus au terme de ces simulations. L'étude du comportement des moteurs d'inférence floue sous diverses conditions théoriques et l'impact du nombre de capteurs utilisés sur ce dernier constituent les points majeurs privilégiés. Une analyse détaillée de tous ces résultats clôture cette discussion.

## 3.2. FONCTIONNEMENT DES MOTEURS D'INFÉRENCE FLOUE

Les moteurs d'inférence floue développés jusqu'ici, qu'ils soient de fondement probabiliste ou possibiliste, possèdent deux modes de fonctionnement précis. Le premier, qualifié de mode séquentiel, calque le fonctionnement d'un système de contrôle réel. Les données ou les observations sont alors reçues des capteurs de façon sérielle, c'est-à-dire une à la fois et traitées individuellement. La seconde alternative offre, par contre, un comportement diamétralement opposé, soit le traitement parallèle de plus d'une observation à la fois. On la qualifie plutôt de mode non séquentiel.

La présente section aborde la question des modes de fonctionnement de moteurs et s'attarde aux différentes subtilités des simulations. Toutes les hypothèses avancées et les limites imposées aux simulations dans le cadre de cette recherche sont ensuite décrites en long et en large. Le tout se termine par la description complète d'une procédure de simulation.

### 3.2.1. Traitement non séquentiel

Un moteur d'inférence floue non séquentiel traite de multiples observations en parallèle et la procédure est bien stricte. À chaque itération de l'algorithme ou, l'équivalent, à chaque réception de rapports de senseurs, le système flou reçoit simultanément des rapports concernant tous les attributs, soit la *vitesse*, le *nombre d'émetteurs* et leur *type*. Pour chacun, le nombre de rapports est variable et dépend de la quantité de senseurs en état de rapporter. Par la suite le moteur réalise, tel que décrit au précédent chapitre, différentes étapes complémentaires d'évaluation et de combinaison des observations et se trouve alors en mesure de fournir les probabilités de détection globales des plates-formes propres à cette



itération. En tout temps, les itérations sont traitées indépendamment et aucune corrélation n'existe entre les résultats découlant de deux itérations successives.

#### 3.2.1.1. *Détermination du nombre de senseurs*

Afin d'accentuer le réalisme, une hypothèse a été avancée concernant le nombre de senseurs rapportant des observations sur un attribut donné. Il a été proposé qu'à chaque itération, le nombre de senseurs rapporteurs soit déterminé aléatoirement. Ainsi pour un attribut précis, il a été fixé que :

- 5% du temps, cette étape de la simulation ne comporte que *deux senseurs*,
- 20% du temps, *trois senseurs* rapportent,
- 75% du temps, *quatre senseurs* sont en mesure de fournir un rapport.

Ce choix aléatoire reflète davantage une situation tactique réelle d'identification de cibles dans laquelle un nombre fixe de senseurs est employé, mais où les rapports ne sont pas tous transmis simultanément. Toutefois, les différentes options proposées ici sont purement hypothétiques et servent uniquement à des fins de simulations.

#### 3.2.1.2. *Type de détection*

L'identification hiérarchique, présentée en profondeur à la section 2.3.3, nécessite qu'un senseur détecte un niveau dans l'arbre de détection d'un attribut plutôt que de fournir directement des observations réelles. Ce dernier peut représenter une plate-forme spécifique ou encore un groupe de plates-formes et reflète directement l'état de l'information transmise, de partielle à détaillée.

À ce stade, diverses hypothèses ont été avancées concernant le type de détection des senseurs. Il a été proposé que pour chaque rapport de senseurs, le

niveau de détection soit déterminé aléatoirement avec une prédominance pour l'identification précise d'une plate-forme unique.

Pour un capteur dédié à l'attribut *vitesse*, il a ainsi été fixé que :

- 20% du temps, le capteur rapporte peu d'information et est alors seulement en mesure d'identifier le *type de plate-forme*, soit *surface* ou *air*,
- 30% du temps, il identifie une catégorie donnée de plates-formes,
- 50% du temps, il précise son jugement et n'identifie qu'une seule plate-forme.

Par contre, pour un capteur dédié aux attributs concernant les émetteurs, il a été établi que :

- 40% du temps, le capteur identifie une catégorie donnée de plates-formes pour l'attribut *nombre d'émetteurs*, mais rend toutefois impossible la procédure d'identification pour l'attribut *type des émetteurs*,
- 60% du temps, il précise son jugement et identifie une plate-forme unique pour l'attribut *nombre d'émetteurs* et se voit du même coup apte à procéder à une étape d'identification du *type des émetteurs*.

Rappelons que, tel que décrit à la section 2.5.1, l'évaluation de l'attribut *type des émetteurs* se voit restreinte à l'identification sommaire d'une plate-forme unique. Dans le cas de l'identification d'un groupe de plates-formes (une catégorie), la détection n'est pas rendue possible, car aucune étape préalable de fuzzification n'est jugée réalisable.

### 3.2.1.3. *Type de conflit*

Lors de simulations, les moteurs d'inférence floue sont soumis à diverses conditions théoriques afin d'analyser leur performance. Une de ces conditions réside dans l'insertion volontaire d'erreurs afin de mieux juger de la qualité globale de

l'identification. Celles-ci, insérées au niveau des observations primaires des senseurs, introduisent des conflits ou des contradictions que le moteur se doit de résoudre. Ainsi un senseur, au lieu de rapporter la bonne cible ou la bonne catégorie de plates-formes transmet respectivement une plate-forme ou une catégorie erronée.

Ainsi trois environnements de simulation différents ont été élaborés et se définissent par ordre de difficulté :

- un modèle *sans conflit* dans lequel tout conflit est à proscrire,
- un second comportant des *conflits limités*,
- une dernière catégorie extrême avec *conflits non limités*.

Dans un environnement sans conflit, l'identification sommaire est irréprochable. Les senseurs déterminent en toute circonstance l'identité exacte de la plate-forme observée. La tâche du système global ne consiste alors qu'à confirmer leurs affirmations.

Dans un environnement avec conflits limités, le senseur peut possiblement émettre un jugement défavorable à la plate-forme réellement observée. Cependant l'erreur est circonscrite. Il peut survenir qu'un senseur identifie incorrectement une plate-forme, mais l'erreur se situe dans la même catégorie que la véritable plate-forme.

Finalement, le cas extrême survient lors de simulations avec conflits non limités. L'erreur n'est alors plus restreinte à la même catégorie que la véritable plate-forme, mais s'étend plutôt aux catégories directement adjacentes à celle de la plate-forme à détecter.

Pour une simulation donnée, la probabilité de conflits est déterminée avant l'exécution de l'algorithme et demeure constante tout au long de la simulation. De

plus, elle se répartit en proportion égale aux attributs *vitesse* et à ceux étudiant les émetteurs.

Ainsi pour un taux d'erreur global donné, le taux d'erreur au niveau de chaque attribut se calcule comme suit :

$$taux_{attribut} = (1 - \sqrt{(1 - taux_{global})}),$$

où  $taux_{attribut}$  représente le taux d'erreur d'un attribut donné et  $taux_{global}$ , le taux global d'erreur prédéterminé.

**Exemple 3.2.1.** *À titre d'exemple, si les probabilités de conflits ou le taux d'erreur global sont fixés à 50%, les probabilités de conflits pour les senseurs dédiés à l'attribut vitesse se situent alors à 30% de même que pour ceux étudiant les émetteurs. En effet, les attributs nombre et type des émetteurs, rapportés par le même senseur, sont fortement dépendants. Ils possèdent ainsi la même probabilité d'erreur.*

#### 3.2.1.4. Déroulement d'une simulation non séquentielle type

Une simulation de type non séquentiel observe une procédure prédéfinie. Voici à ce sujet une présentation détaillée du déroulement d'une simulation sans conflit, avec conflits limités ou non (voir section précédente 3.2.1.3).

La procédure débute à chaque itération par l'analyse de l'attribut *vitesse*. Dans un premier temps, le nombre de senseurs rapporteurs est déterminé aléatoirement et la pondération de chacun établie. Puis à tour de rôle, les détections des senseurs sont simulées. Le système détermine d'abord si l'identification sera exacte ou erronée dans le cas d'environnement avec conflits limités ou non et identifie par la suite, toujours de façon aléatoire, un niveau dans l'arbre de détection de la vitesse. La probabilité de détection primaire de chaque plate-forme est

ensuite déduite de ces informations (voir section 2.5.1). Ce processus se poursuit ainsi pour tous les senseurs rapporteurs et finalement, ces résultats préliminaires sont combinés afin d'obtenir les probabilités intermédiaires de détection pour l'attribut vitesse. À cette étape, les moteurs du produit et bayésien privilégient une combinaison avec opérateur du produit flou tandis que celui du minimum opte pour le minimum flou (voir section 2.5.5).

L'algorithme se poursuit par l'étude des attributs *nombre* et *type des émetteurs*. Dans un premier temps, le nombre de senseurs rapporteurs est déterminé aléatoirement et la pondération de chacun fixée. Puis la procédure d'analyse de l'attribut *nombre d'émetteurs*, identique à celle du précédent attribut *vitesse*, est lancée. Des probabilités de détection intermédiaires sont ainsi calculées. Puis vient l'analyse de l'attribut *type des émetteurs*. Le déroulement diffère largement des précédents. En effet, pour chacun des senseurs, une évaluation primaire des probabilités de détection n'est effectuée que si celui-ci a identifié préalablement une plate-forme unique au niveau de l'attribut *nombre d'émetteurs*. Dans le cas contraire, aucune évaluation n'est réalisée. Rappelons que celle-ci est effectuée par analyse séquentielle des types d'émetteurs possédés par la plate-forme identifiée (voir la section 2.5.1 pour de plus amples informations). Une étape de combinaison, identique à celle des précédents attributs, est finalement exécutée et des probabilités de détection intermédiaires en découlent également.

L'itération se termine par la combinaison des résultats intermédiaires provenant de chacun de trois attributs (voir section 2.5.3) et des probabilités globales de détection des plates-formes en sont déduites. Les opérateurs du minimum et du maximum flous sont préconisés dans le cas des moteurs d'inférence floue du produit et du minimum tandis que celui de Bayes privilégie plutôt une combinaison avec produit flou. Finalement, l'algorithme conserve en mémoire l'identité

des plates-formes qui possèdent les trois probabilités globales de détection les plus élevées (voir section 2.5.4).

La simulation se déroule ainsi à chacune des itérations et à la suite d'une série de simulations équivalentes, certaines observations peuvent être déduites dont le taux d'erreur de l'algorithme et le nombre moyen de plates-formes identifiées comme plus probables à chaque itération. Celles-ci permettent par la suite de juger de la performance des moteurs d'inférence floue.

### 3.2.2. Traitement séquentiel

Un moteur d'inférence floue séquentiel présente un comportement différent de celui de son prédécesseur. Les données ou les observations ne sont plus traitées en parallèle, mais plutôt reçues séquentiellement des capteurs, une à la suite de l'autre.

À chaque itération de l'algorithme, le système flou reçoit un rapport unique de capteur concernant exclusivement l'attribut *vitesse* ou le *nombre d'émetteurs* et leur *type*. Par la suite, il effectue différentes étapes complémentaires d'évaluation et de combinaison et se trouve alors en mesure de fournir les probabilités de détection globales des plates-formes à chaque itération. L'historique de la détection joue désormais un rôle de premier plan. Les données tirées de l'observation d'un capteur sont combinées avec celles obtenues précédemment pour fournir le meilleur estimé à cet instant précis. Un tel type de système accentue le réalisme des simulations et rapproche davantage les moteurs d'inférence floue des systèmes experts concrets fonctionnant en temps réel.

### 3.2.2.1. *Types d'informations analysées*

Les observations reçues des senseurs concernent des attributs spécifiques. Une hypothèse de simulation a ainsi été avancée au sujet du choix de l'attribut concerné par une observation.

Par conséquent, lors de la réception du rapport d'un senseur, il a été établi que :

- 50% du temps, la détection concerne l'attribut *vitesse*,
- le reste du temps, les émetteurs sont au premier plan. La détection concerne plutôt les attributs *nombre d'émetteurs* et leur *type*.

Cette démarche suppose bien sûr, qu'en moyenne, un nombre égal de senseurs dédiés à la vitesse et aux émetteurs soient employés.

### 3.2.2.2. *Type de détections*

L'identification exécutée par les systèmes experts flous séquentiels conserve son statut hiérarchique. Ainsi à chaque réception du rapport d'un senseur, l'identification d'un niveau dans l'arbre de détection de l'attribut concerné s'impose. Le choix du niveau pour les attributs *vitesse* et *nombre d'émetteurs* respecte toujours les mêmes hypothèse avancées à la section 3.2.1.2.

### 3.2.2.3. *Type de conflit*

Les types de conflits présentés à la section 3.2.1.3 concernant le moteurs d'inférence floue non séquentiels s'appliquent toujours ici. Les simulations séquentielles peuvent également ne comporter aucun conflit, des conflits limités ou non limités. Les mêmes techniques de calcul du taux d'erreur s'appliquent.

#### 3.2.2.4. Critère d'arrêt

Les simulations de type séquentiel se poursuivent jusqu'à ce qu'un critère d'arrêt soit satisfait. Voici le modèle conçu ici pour les moteurs d'inférence floue.

Tout d'abord, au lancement de l'algorithme, certaines limites doivent être imposées à la simulation, soit des nombres minimaux et maximaux d'itérations ainsi qu'un niveau de précision. La première condition assure que l'algorithme reçoit suffisamment d'observations pour que son jugement soit jugé crédible tandis que la seconde prévient l'emballement. Finalement, le niveau de précision représente le critère d'arrêt complet de l'algorithme.

Lors d'une simulation, une identification globale est effectuée à chaque itération. Des statistiques portant sur les plates-formes jugées les plus probables sont alors calculées et conservées en mémoire. En effet, la ou les plates-formes ayant été jugées les plus favorables lors d'un plus grand nombre d'itérations sont identifiées et ce nombre d'itérations, noté *seuil primaire*, repertorié. La même stratégie s'applique également pour les secondes plates-formes plus favorables et un *seuil secondaire* est déduit. Finalement, pour que l'algorithme cesse avant l'exécution de l'itération suivante, les limites mentionnées précédemment doivent être atteintes, c'est-à-dire que le nombre minimal d'itérations doit être dépassé et la différence entre les *seuils primaire* et *secondaire* doit excéder le niveau de précision demandé. Cette procédure assure qu'à travers toute l'historique de la détection, certaines plates-formes s'illustrent davantage. Dans le cas de non-atteinte de ces limites, la simulation se poursuit et, dans le pire des cas, cesse *in extremis* lorsque le nombre maximal d'itérations est atteint.



### 3.2.2.5. Déroulement d'une simulation séquentielle type

Une simulation de type séquentiel observe également une procédure prédéfinie dont voici une description générale sans conflit, avec conflits limités et non limités (voir section 3.2.2.3). Remarquons d'abord que ces types de simulations se fondent grandement sur l'historique de l'identification de plates-formes. En effet, à la fin de chaque itération, les probabilités intermédiaires au niveau de chaque attribut sont stockées en mémoire et utilisées pour fins de calculs lors de la suivante. L'indépendance entre les itérations n'est ainsi plus un trait caractéristique.

La procédure ne débute plus systématiquement pour chaque itération par l'analyse de l'attribut *vitesse*. L'attribut analysé est déterminé aléatoirement et chaque fois, le rapport de capteur associé, unique. La moitié des chances sont allouées à l'attribut *vitesse* tandis que l'autre va aux attributs *nombre* et *type des émetteurs* (voir section 3.2.2.1).

Dans le cas de l'attribut *vitesse*, la pondération du capteur est tout d'abord établie. Puis, sa détection est simulée. Le système détermine d'abord si l'identification est exacte ou erronée dans le cas d'environnement avec conflits limités ou non et identifie par la suite, toujours de façon aléatoire, un niveau dans l'arbre de détection de la vitesse. La probabilité de détection primaire de chaque plate-forme est ensuite déduite de ces informations (voir section 2.5.1). Finalement, vient l'étape de la combinaison intermédiaire des résultats pour cet attribut. Les probabilités primaires de détection sont combinées avec celles intermédiaires obtenues à l'itération précédente afin d'obtenir les nouvelles probabilités intermédiaires de détection. Celles-ci sont à nouveau conservées en mémoire pour utilisation à l'itération suivante. Rappelons que les moteurs du produit et bayésien privilégient une combinaison avec opérateur du produit flou tandis que celui du minimum opte plutôt pour le minimum flou (voir section 2.5.5).

L'algorithme peut également privilégier les attributs *nombre* et *type des émetteurs*. Dans ce cas, la pondération du senseur est tout d'abord fixée et ses détections simulées. Puis la procédure d'analyse de l'attribut *nombre d'émetteurs*, identique à celle du précédent attribut *vitesse*, est lancée. Des probabilités de détection intermédiaires sont ainsi calculées à l'aide des probabilités primaires actuelles et de celles intermédiaires tirées de l'itération précédente. Celles-ci sont à nouveau conservées en mémoire pour une utilisation ultérieure. Puis vient l'analyse de l'attribut *type des émetteurs*. Le déroulement diffère des précédents. Une évaluation primaire des probabilités de détection est effectuée seulement si celui-ci a identifié une plate-forme unique au niveau de l'attribut *nombre d'émetteurs*. Dans le cas contraire, aucune évaluation n'est effectuée. Rappelons que celle-ci est effectuée par analyse des types d'émetteurs possédés par la plate-forme identifiée (voir la section 2.5.1 pour de plus amples informations). Une étape de combinaison, identique à celle des précédents attributs, est finalement exécutée, toujours à l'aide de probabilités de détection intermédiaires tirées de l'itération précédente, et de nouvelles probabilités de détection intermédiaires en découlent.

L'itération se termine toujours par la combinaison des résultats intermédiaires provenant de chacun de trois attributs (voir section 2.5.3) qu'ils aient fait l'objet d'une analyse ou non. Dans ce dernier cas, les résultats intermédiaires emmagasinés en mémoire sont utilisés. Les opérateurs du minimum et du maximum flous sont préconisés dans le cas des moteurs d'inférence floue du produit et du minimum tandis que celui de Bayes privilégie plutôt une combinaison avec produit flou. Puis, l'algorithme conserve en mémoire l'identité des plates-formes qui possèdent les trois probabilités de détection globales les plus élevées (voir section 2.5.4). Finalement, le critère d'arrêt est évalué pour déterminer si la simulation se poursuit ou non.

À la suite de plusieurs simulations équivalentes, certaines observations peuvent être déduites dont le taux d'erreur de l'algorithme, le nombre moyen de plateformes identifiées comme plus probables à chaque itération et le nombre moyen d'itérations par simulation. Celles-ci permettent par la suite de juger de la performance des moteurs d'inférence floue.

### 3.3. RÉSULTATS NUMÉRIQUES

La présente section met à dure épreuve les systèmes experts flous développés jusqu'ici, soit les moteurs d'inférence floue du produit, du minimum et bayésien. Une batterie de simulations sévères leur est imposée afin d'analyser leur comportement sous différentes conditions allant du fondement, probabiliste ou possibiliste, au type de conflit : sans conflit, conflits limités ou non. Tous ces moteurs seront ainsi comparés pour parvenir à établir tout d'abord les forces et faiblesses de chacun, mais également statuer sur le bien-fondé de l'utilisation de la logique floue en technologie de l'information.

L'étude du comportement des moteurs d'inférence floue se concentre principalement au niveau séquentiel et non séquentiel sous divers fondements et types de conflits et sur l'impact du nombre de capteurs utilisés lors d'une simulation non séquentielle. Chacune de ces sous-sections enchaîne d'abord avec la panoplie de résultats numériques obtenus aux termes des simulations, puis termine avec son analyse. Une analyse globale de tous ces résultats vient clore ce chapitre.

#### **3.3.1. Comportement non séquentiel des moteurs d'inférence floue**

Les simulations effectuées ici concernent le comportement non séquentiel des moteurs d'inférence floue du produit, du minimum et bayésien, c'est-à-dire le traitement de multiples observations en parallèle. Les conditions de simulation y

sont diversifiées et la variété des types de conflits figure au tout premier plan. En effet, la performance de chacun des moteurs est évaluée dans un environnement sans conflit, avec conflits limités et non limités avec taux d'erreur théoriques de 10%, 25% et 40%. Celles-ci déterminent en quelle proportion la détection primaire d'un senseur peut être entachée d'erreur (voir section 3.2.1.4 pour de plus amples informations). De plus sous tout type de conflit, les simulations sont lancées pour une véritable plate-forme observée de type *surface* et *air* afin de mieux juger de la polyvalence des algorithmes. Enfin, chacune de ces batteries de tests est appliquée tant à des moteurs de fondement probabiliste que possibiliste et la pondération de chaque senseur est conservée de niveau unitaire.

Les résultats, tous recueillis à la suite d'une série de 100 simulations de 100 itérations chacune, concernent le taux d'erreur réel et le nombre moyen de plates-formes détectées. Le premier critère illustre l'efficacité de la détection, c'est-à-dire le pourcentage d'itérations où l'identification est défavorable à la véritable plate-forme présente. Et comme plus d'une plate-forme peut être identifiée plus probable à la suite d'une même itération, le second critère s'attarde à ce problème. Il évalue, en effet, le nombre moyen de plates-formes détectées par itération et permet de confirmer la précision des résultats. Dans tous les cas, ceux-ci sont suivis d'une étape d'analyse locale par fondement, probabiliste et possibiliste, complétée par une dernière plus globale en toute fin de sous-section. Les moteurs possédant à la fois une efficacité et une précision élevées seront alors jugés les plus performants.

#### 3.3.1.1. *Évaluation probabiliste*

Le tableau 3.3.1 présente le comportement non séquentiel des moteurs d'inférence floue bayésien, du produit et du minimum de fondement probabiliste. Les

termes  $Tet$  et  $Tpf$  réfèrent respectivement au taux d'erreur théorique (probabilité d'identification primaire erronée) et au type de plates-formes tandis que ceux  $Ter$  et  $Nmpf$  concernent plutôt le taux d'erreur réel et le nombre moyen de plates-formes détectées par itération, c'est-à-dire l'efficacité et la précision du moteur.

Tout d'abord, du côté du taux d'erreur réel ( $Ter$ ), plusieurs constatations peuvent être faites sur ces résultats. On observe, en général, une croissance du taux d'erreur avec l'augmentation du degré de conflit, de sans conflit à non limités, et à l'intérieur d'un même niveau, on relève ce même phénomène lors de l'augmentation du taux d'erreur théorique de 10% à 40% ( $Tet$ ). Il existe ainsi une corrélation sans équivoque entre le niveau de difficulté de la simulation, c'est-à-dire l'étendue de l'erreur potentielle avant simulation, et celle effectivement observée après simulation, mais le rapport n'est pas direct. Ainsi un taux d'erreur théorique de  $x\%$ , ne mène pas forcément à un taux réel de  $x\%$  à en juger par les résultats obtenus.

Pour un même type de plates-formes, par exemple *surface*, on observe que le moteur d'inférence du produit avec opérateur d'union floue présente souvent les taux d'erreur réels les plus faibles suivi de près par le moteur du minimum du même type. Ces résultats découlent de la similitude de leur structure qui ne diffère qu'au niveau de la combinaison intermédiaire. Le moteur bayésien avec opérateur d'union floue présente dans l'ensemble des taux d'erreur légèrement supérieurs à ceux des précédents modèles. De par sa structure et son jugement fort précis, ses chances de commettre une erreur sont, en moyenne, accentuées. Les résultats sont de loin moins satisfaisants avec l'utilisation d'un opérateur d'intersection floue au niveau de la combinaison intermédiaire des moteurs du produit et du minimum.

TAB. 3.3.1 – Résultats obtenus aux simulations pour les moteurs non séquentiels d'inférence floue du produit, du minimum et de Bayes de fondement probabiliste

Caractéristiques des simulations			Moteurs d'inférence floue											
			Bayésien			Produit				Minimum				
			Ter	Nmpf	Tpf	Union		Intersection		Union		Intersection		
Sans conflits	0,00	Surface	0,1	1,11	0,03	1,24	0,19	2,37	0,02	1,16	0,03	4,12		
		Air	0,03	1,03	0,04	1,12	0,00	1,03	0,00	1,28	0,00	1,05		
Limités	0,10	Surface	0,14	1,03	0,06	1,4	0,23	2,79	0,18	2,09	0,11	3,16		
		Air	0,02	1,08	0,10	1,02	0,07	1,17	0,01	1,85	0,07	1,57		
	0,25	Surface	0,17	1,12	0,14	1,61	0,45	3,12	0,27	2,50	0,30	4,14		
		Air	0,09	1,03	0,13	1,10	0,08	1,27	0,10	4,21	0,19	2,21		
	0,40	Surface	0,35	1,28	0,22	2,30	0,56	3,63	0,41	3,41	0,48	4,86		
		Air	0,09	1,02	0,19	1,21	0,19	1,66	0,12	4,59	0,34	3,58		
Non limités	0,10	Surface	0,18	1,08	0,08	1,36	0,28	3,02	0,12	1,94	0,19	4,88		
		Air	0,11	1,02	0,09	1,00	0,15	2,79	0,07	3,33	0,17	3,34		
	0,25	Surface	0,23	1,15	0,17	1,47	0,45	2,91	0,29	2,39	0,46	4,68		
		Air	0,31	1,27	0,15	1,32	0,42	4,65	0,15	4,43	0,36	5,69		
	0,40	Surface	0,44	1,21	0,29	1,94	0,59	3,30	0,42	3,52	0,61	4,78		
		Air	0,31	1,13	0,32	1,61	0,63	5,40	0,25	5,73	0,71	10,07		

Le taux d'erreur est alors, pour un moteur donné et dans le pire des cas, deux à trois fois plus élevé que dans leur version avec opérateur d'union floue.

Au niveau du nombre moyen de plates-formes détectées par itération (Nmpf), les résultats privilégient le moteur bayésien. En effet, lorsqu'il effectue une identification, ses conclusions sont sans équivoque et centrées autour d'une seule plate-forme par itération. On observe souvent une croissance du nombre moyen de plates-formes avec l'augmentation du degré de conflit, de sans conflit à non limités, et à l'intérieur d'un même niveau, on relève ce même phénomène lors de l'augmentation du taux d'erreur théorique (Tet). Dans les cas extrêmes, les erreurs trompent les moteurs qui donnent ainsi une masse élevée à plus d'une candidate. L'identification finale est ainsi moins nette.

Comme dans le cas du taux d'erreur réel, l'utilisation d'un opérateur d'intersection floue au lieu d'un d'union floue, au niveau de la combinaison intermédiaire des moteurs du produit et du minimum, offre des performances moins reluisantes. Le même constat peut être fait entre les moteurs du produit et du minimum en général. Le second se veut largement plus optimiste, car un plus grand nombre de plates-formes sont détectées avec une probabilité supérieure à chaque itération.

On remarque qu'en général et pour tous les critères confondus, les résultats sont meilleurs dans le cas d'une plate-forme de type *air* que de type *surface* pour des raisons purement extérieures aux moteurs eux-mêmes. La base de données contient d'abord moins de plates-formes de ce type et elles ont des attributs moins variés que dans l'autre groupe. La plate-forme choisie se distinguent d'ailleurs des autres par des attributs presque uniques, par conséquent son identité est plus facile à déterminer. Dans les cas extrêmes cependant (conflits non limités et taux d'erreur théorique élevé et des moteurs très optimiste) où de mauvaises détections brouillent les pistes, les résultats peuvent se détériorer grandement tant au niveau du taux d'erreur réel que du nombre moyen de plates-formes.

En conclusion, au niveau du fondement probabiliste, le moteur d'inférence floue bayésien s'avère le plus efficace tant par ses jugements sans équivoque qu'efficaces. Il est cependant suivi de très près par celui du produit.

### 3.3.1.2. *Évaluation possibiliste*

Le tableau 3.3.2 présente le comportement non séquentiel des moteurs d'inférence floue bayésien, du produit et du minimum de fondement possibiliste. Les termes *Tet* et *Tpf* réfèrent respectivement au taux d'erreur théorique (possibilité d'identification primaire erronée) et au type de plates-formes tandis que ceux *Ter* et *Nmpf* concernent plutôt le taux d'erreur réel et le nombre moyen de plates-formes, c'est-à-dire l'efficacité et la précision du moteur.

Une analyse similaire à la précédente peut être effectuée ici. On peut remarquer tout d'abord les résultats impressionnants obtenus par les moteurs du produit et du minimum au niveau du taux d'erreur réel. Ceux-ci cachent cependant un optimisme exagéré au niveau du nombre moyen de plates-formes détectées par itération. Les moteurs ne se trompent pas, mais identifient tellement de plates-formes comme plus probables que la bonne se trouve forcément à l'intérieur de ce groupe. Ces comportements sont donc à rejeter. Contrairement au précédent fondement probabiliste, ces moteurs possibilistes avec opérateur d'intersection floue offrent de meilleurs rendements qu'avec l'union floue et des comportements somme toute meilleurs. Toutefois, le moteur de Bayes se veut de loin le plus performant, avec des taux d'erreur réels comparables et souvent meilleurs aux moteurs du produit et du minimum avec intersection floue et une précision inégalable. Ses conclusions sont très sévères et centrées.

De façon plus générale, on observe une croissance des deux critères avec l'augmentation du degré de conflit, de sans conflit à non limités, et à l'intérieur d'un



TAB. 3.3.2 – Résultats obtenus aux simulations pour les moteurs non séquentiels d'inférence floue du produit, du minimum et de Bayes de fondement possibiliste

Caractéristiques des simulations			Moteurs d'inférence floue															
			Bayésien				Produit				Minimum							
			Ter		Nmpf		Ter		Nmpf		Ter		Nmpf		Ter		Nmpf	
			Tet	Tpf	Ter	Nmpf	Ter	Nmpf	Ter	Nmpf	Ter	Nmpf	Ter	Nmpf	Ter	Nmpf	Ter	Nmpf
Sans conflits	0,00	Surface	0,10	1,06	0,01	15,01	0,17	1,33	0,03	16,77	0,04	1,00	0,00	6,82	0,02	1,00		
		Air	0,03	1,00	0,00	6,91	0,08	1,13	0,00	6,82	0,02	1,00	0,00	6,82	0,02	1,00		
Limités	0,10	Surface	0,18	1,06	0,05	15,87	0,22	1,68	0,07	18,10	0,13	1,42	0,00	10,61	0,10	1,00		
		Air	0,02	1,00	0,00	6,93	0,07	1,11	0,00	10,61	0,10	1,00	0,00	10,61	0,10	1,00		
	0,25	Surface	0,30	1,00	0,07	16,24	0,39	1,99	0,16	16,86	0,34	1,72	0,00	12,22	0,33	1,01		
		Air	0,04	1,00	0,00	6,94	0,13	1,56	0,02	12,22	0,33	1,01	0,00	12,22	0,33	1,01		
	0,40	Surface	0,33	1,25	0,05	15,84	0,47	2,03	0,07	20,74	0,43	2,10	0,00	18,07	0,35	1,06		
		Air	0,07	1,00	0,00	6,98	0,10	1,21	0,01	18,07	0,35	1,06	0,00	18,07	0,35	1,06		
Non limités	0,10	Surface	0,11	1,05	0,02	15,56	0,15	1,37	0,03	16,88	0,14	1,19	0,00	9,21	0,23	2,97		
		Air	0,08	1,00	0,04	9,16	0,06	1,19	0,03	9,21	0,23	2,97	0,00	9,21	0,23	2,97		
	0,25	Surface	0,29	1,11	0,05	16,05	0,38	1,82	0,13	17,18	0,35	1,79	0,00	12,11	0,35	4,31		
		Air	0,16	1,02	0,02	12,00	0,20	1,52	0,02	12,11	0,35	4,31	0,00	12,11	0,35	4,31		
	0,40	Surface	0,45	1,20	0,12	15,68	0,48	1,68	0,22	16,34	0,54	2,05	0,00	14,81	0,63	7,06		
		Air	0,39	1,05	0,04	14,95	0,29	2,28	0,04	14,81	0,63	7,06	0,00	14,81	0,63	7,06		

même niveau, on relève ce même phénomène lors de l'augmentation du taux d'erreur théorique (Tet). On remarque également que, tous les critères confondus, les

résultats sont habituellement supérieurs dans le cas d'une plate-forme de type *air* que de type *surface*, toujours pour des raisons purement extérieures aux moteurs eux-mêmes.

En conclusion, au niveau du fondement possibiliste, le moteur d'inférence floue bayésien s'avère le plus efficace tant par ses jugements sans équivoque qu'efficaces. Il est suivi des moteurs du produit et du minimum avec opérateur d'intersection floue.

### 3.3.1.3. *Analyse des résultats*

Globalement, les moteurs d'inférence floue, qu'ils soient de fondement probabiliste ou possibiliste, présentent un comportement général similaire face à la croissance du taux d'erreur théorique : l'augmentation de l'imprécision des résultats. Mais l'impact de ce phénomène varie grandement d'un modèle à l'autre et certains moteurs se distinguent tout particulièrement par une précision et une efficacité supérieures.

Tout d'abord, notons que certains moteurs s'illustrent par un trop grand optimisme et une imprécision élevée. En effet, les moteurs du produit et du minimum avec opérateur d'intersection floue de tout fondement présentent ainsi des comportements peu reluisants. Un optimisme exagéré s'avère également néfaste aux moteurs du produit et du minimum avec opérateur d'union floue et fondement possibiliste même si celui-ci est jumelé à une efficacité inégalable. Tous ces moteurs restent cependant dignes d'intérêt, mais malheureusement concrètement inapplicables.

D'autres moteurs présentent toutefois un comportement fort acceptable et parfois même étonnant. En effet, les moteurs du produit et du minimum avec

opérateur d'union floue et fondement probabiliste s'illustrent. Leur comportement est fort similaire, mais une prédominance est accordée aux modèles du produit avec opérateur d'union floue et fondement probabiliste qui présente un taux d'erreur réel et un nombre moyen de plates-formes détectées inférieurs. Il est cependant légèrement déclassé par le moteur bayésien au niveau du nombre moyen de plates-formes détectées par itération. En effet, ce pur modèle statistique, présente le rendement le plus étonnant : un taux d'erreur réel toujours limité et un nombre moyen de plates-formes détectées frisant l'unité. Son jugement se veut ainsi à la fois sans équivoque et sévère. Le fondement possibiliste s'avère toutefois ici plus intéressant. La plate-forme la plus probable se voit dans ce cas accordée une probabilité de détection unitaire au niveau de la normalisation (voir section 2.6.0.5) et ses chances d'être déclarée plus probable se voient ainsi largement favorisées tout au cours du processus d'identification.

En conclusion, à la suite de l'étude du comportement non séquentiel des moteurs d'inférence floue, le moteur bayésien avec fondement possibiliste présente les performances les plus satisfaisantes. Il est suivi de près de son homologue probabiliste et du moteur du produit avec opérateur d'union floue et fondement probabiliste.

### **3.3.2. Comportement séquentiel des moteurs d'inférence floue**

Les simulations effectuées ici concernent le comportement séquentiel des moteurs d'inférence floue du produit, du minimum et bayésien où les données sont reçues en série des capteurs, une à la suite de l'autre. Les conditions de simulation y sont fort variées. En effet, la performance de chacun des moteurs est évaluée dans un environnement sans conflit, avec conflits limités et non limités avec taux

d'erreur théoriques de 10%, 25% et 40%. Ces derniers déterminent à quelle probabilité la détection primaire d'un senseur peut être entachée d'erreur (voir section 3.2.1.4 pour de plus amples informations). De plus sous tout type de conflit, les simulations sont lancées pour une véritable plate-forme observée de type *surface* et *air* afin de mieux juger de la polyvalence des algorithmes. Enfin, chacune de ces batteries de tests est appliquée tant à des moteurs de fondement probabiliste que possibiliste et la pondération de chaque senseur est conservé en tout temps de niveau unitaire.

Les résultats ont tous été recueillis à la suite d'une série de 100 simulations d'au minimum 20 itérations et d'au plus 100, chacune avec une précision de 10, 25 et 40 comme critère d'arrêt de l'algorithme (voir section 3.2.2.4). Ils concernent notamment le taux d'erreur réel et le nombre moyen d'itérations par simulation. Le premier critère illustre l'efficacité de la détection, c'est-à-dire le pourcentage d'itérations où l'identification fut défavorable à la véritable plate-forme effectivement présente. Le second critère s'attarde, quant à lui, à l'évaluation du nombre moyen d'itérations avant arrêt complet d'une simulation et permet de confirmer le rendement du moteur employé. Dans tous les cas, ceux-ci sont suivis d'une étape d'analyse locale par fondement (probabiliste et possibiliste), complétée par une dernière plus globale en toute fin de sous-section. Les moteurs possédant à la fois une efficacité et un rendement élevés seront alors jugés les plus performants. Notons en terminant que seul le comportement des plates-formes de type *surface* sera étudié au cours de cette sous-section, les mêmes conclusions étant applicables aux plates-formes de type *air*.

### 3.3.2.1. Évaluation probabiliste

La présente sous-section illustre le comportement séquentiel des moteurs d'inférence floue du produit, du minimum et de Bayes de fondement probabiliste à l'aide des graphiques 3.3.1 à 3.3.3. Chacun d'eux présente tout d'abord le nombre moyen d'itérations par simulation (a), soit le rendement du moteur et termine par le taux d'erreur réel (b), son efficacité, en fonction du type de conflit : sans conflit (SC), conflits limités et taux d'erreur théoriques de 10% (CL10), 25% (CL25) et 40% (CL40) ainsi qu'avec conflits non limités et taux d'erreur théoriques de 10% (CNL10), 25% (CNL25) et 40% (CNL40). Tous ces résultats sont présentés pour des moteurs avec opérateur d'union et d'intersection floues (respectivement max et min) et un critère d'arrêt de 10, 20 ou 40.

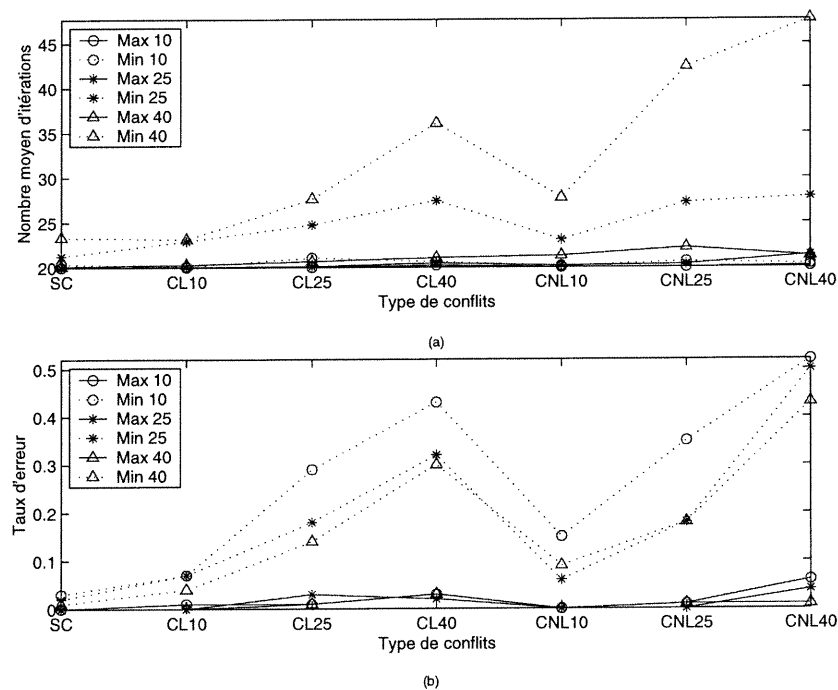


FIG. 3.3.1 – Effets du moteur séquentiel d'inférence floue du produit de fondement probabiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b)

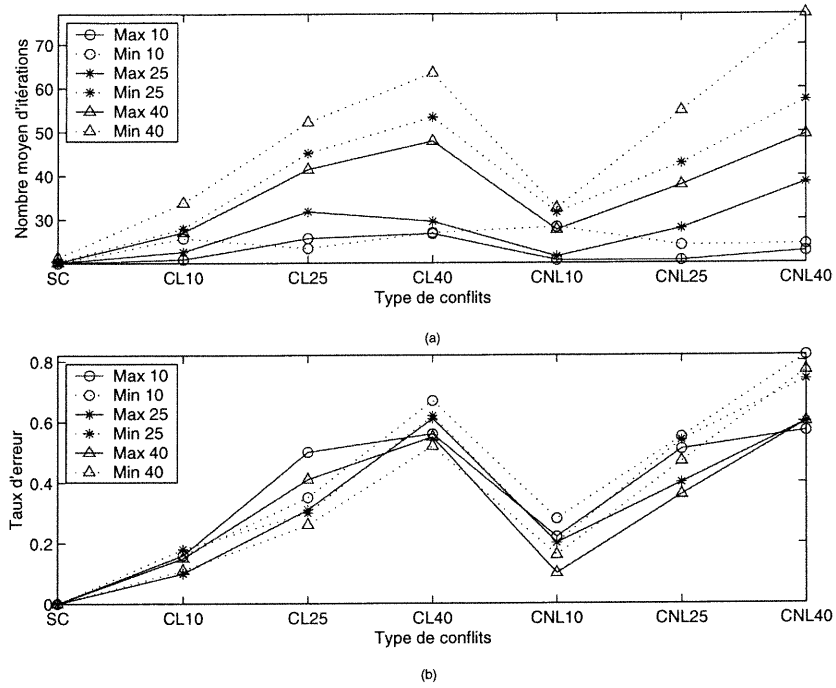


FIG. 3.3.2 – Effets du moteur séquentiel d'inférence floue du minimum de fondement probabiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b)

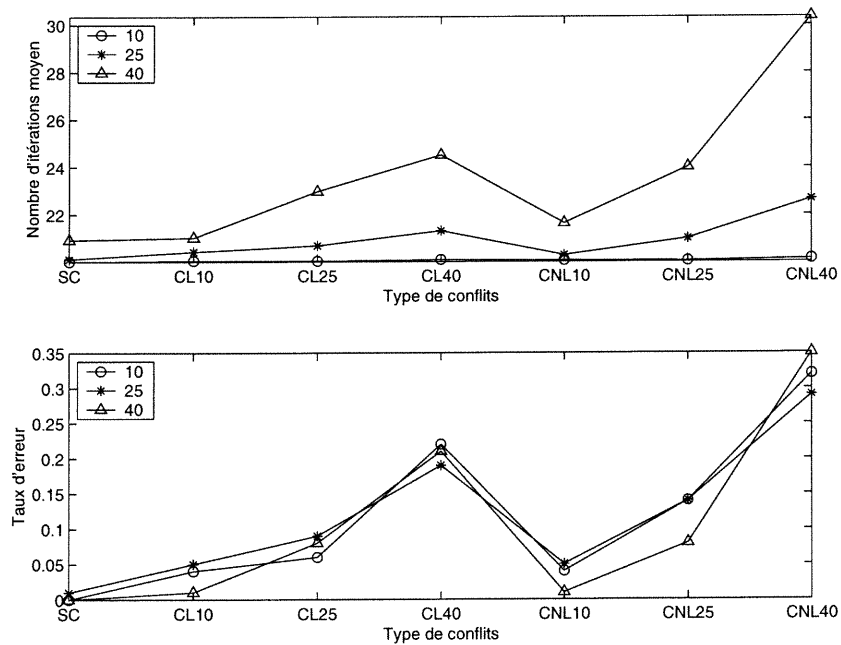


FIG. 3.3.3 – Effets du moteur séquentiel d'inférence floue bayésien de fondement probabiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b)

Suite à l'analyse des résultats, plusieurs constatations peuvent être tirées sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b). On observe tout d'abord, dans les deux cas, une croissance du taux d'erreur avec l'augmentation du degré de conflit, de sans conflit à non limités, et à l'intérieur d'un même niveau. De plus, on remarque une augmentation du nombre moyen d'itérations avec la croissance du critère d'arrêt, car le nombre de rapports de senseurs nécessaires pour le satisfaire augmente sensiblement. Le taux d'erreur pour un type de conflit donné, quant à lui, semble relativement stable tout particulièrement pour le moteur bayésien où les variations ne sont que légères. On le considère ainsi, en moyenne, constant en fonction de l'augmentation du critère d'arrêt. De façon plus détaillée, il ressort que les moteurs d'inférence floue bayésien et du produit avec opérateur d'union floue sont les plus performants. Mais le second s'illustre davantage cette fois avec un nombre moyen de plates-formes détectées par itération se situant près du nombre minimal d'itérations possible, soit 20 et un taux d'erreur réel, dans le pire des cas, sous la barre du 10%. D'un autre côté, on remarque que l'utilisation d'un opérateur d'intersection floue au profit d'un d'union floue pour les moteurs du minimum et du produit est nettement défavorable : le rendement et l'efficacité sont inférieurs. Le moteur du minimum se veut toujours trop optimiste et ses performances figurent encore ici en queue de liste.

Dans un autre ordre d'idées, des résultats ont également été recueillis sur le nombre moyen de plates-formes plus probables par itération et sur le nombre de simulations au cours d'une série de 100 où la limite supérieure de 100 itérations est atteinte. Ils ne sont toutefois pas reproduits explicitement dans cette sous-section, mais servent tout de même d'objet d'analyse. Les mêmes comportements observés dans le cas non séquentiel sont sensiblement observés ici (voir sous-section 3.3.1). Globalement, il a été constaté que peu importe le critère d'arrêt, en moyenne,

pour un même type de conflit, la précision est stable pour tout type de moteur. Les moteurs d'inférence floue bayésien et du produit avec opérateur d'union floue se caractérisent par une précision étonnante : un nombre moyen de plates-formes détectées peu sensible aux variations du type de conflit, du taux d'erreur et du critère d'arrêt et, de plus, quasi unitaire. Le moteur bayésien, par contre, atteint quelques rares fois le nombre maximal d'itérations permis contrairement au second. Le moteur du produit se veut ainsi toujours en mesure de fournir ses conclusions dans un délai raisonnable. En général, il a été observé que plus le critère d'arrêt est important, plus l'atteinte du maximum d'itérations permises survient lors d'un grand nombre de simulations. Le nombre d'itérations requis pour le satisfaire croît alors augmentant du même coup les risques de conclusions ambiguës ne satisfaisant pas complètement le critère d'arrêt imposé. Du côté du moteur du produit avec intersection floue, les résultats obtenus restent acceptables, mais beaucoup moins impressionnants que pour les précédents systèmes. Finalement, le moteur du minimum qu'il soit composé d'un opérateur d'union ou d'intersection floue ne s'avoue guère performant et beaucoup trop optimiste dans ses conclusions. Le nombre moyen de plates-formes frise, dans le pire des cas, la dizaine tandis que le plafond en terme d'itérations est atteint parfois dans plus de 80% des simulations, un constat bien peu reluisant.

En conclusion, à la suite de l'étude du comportement séquentiel des moteurs d'inférence floue avec fondement probabiliste, le moteur du produit avec opérateur d'union floue présente les performances les plus satisfaisantes par son taux d'erreur réel faible, par son nombre moyen minimal d'itérations ou de plates-formes détectées et par un dépassement du nombre d'itérations maximal fort peu probable. Il est suivi du modèle bayésien.



### 3.3.2.2. *Évaluation possibiliste*

La présente sous-section illustre le comportement séquentiel des moteurs d'inférence floue du produit, du minimum et de Bayes de fondement possibiliste, à l'aide des graphiques 3.3.4 à 3.3.6. Chacun d'eux présente tout d'abord le nombre moyen d'itérations par simulation (a), soit le rendement du moteur, et termine par le taux d'erreur réel (b), son efficacité, en fonction du type de conflit : sans conflit (SC), conflits limités et taux d'erreur théoriques de 10% (CL10), 25% (CL25) et 40% (CL40) ainsi qu'avec conflits non limités et taux d'erreur théoriques de 10% (CNL10), 25% (CNL25) et 40% (CNL40). Tous ces résultats sont présentés pour des moteurs avec opérateur d'union et d'intersection floues (respectivement max et min) et un critère d'arrêt de 10, 20 ou 40.

Tout d'abord, les mêmes constatations générales soulevées à la section précédente 3.3.2.1 concernant l'évolution du taux d'erreur réel et du nombre moyen d'itérations en fonction de l'importance des conflits et du critère d'arrêt s'appliquent ici. Le moteur bayésien s'avère de loin le plus performant et, somme toute, assez similaire à sa version probabiliste. Au niveau des moteurs du produit et du minimum avec opérateur d'union floue, on peut être tout d'abord berné par les résultats impressionnants obtenus au niveau du taux d'erreur réel. Ceux-ci cachent cependant un optimisme exagéré au niveau du nombre moyen d'itérations et du nombre moyen de plates-formes détectées par itération (résultats non présentés). Ils atteignent également le plafond de 100 itérations lors d'une simulation à un nombre anormalement élevé d'occasions. Ils ne se trompent pas, mais identifient tellement de plates-formes comme plus probables que la bonne se trouve forcément à l'intérieur de cet ensemble. Ces comportements sont donc à rejeter. Contrairement au précédent fondement probabiliste, ces moteurs possibilistes avec opérateur d'intersection floue offrent de meilleurs rendements qu'avec

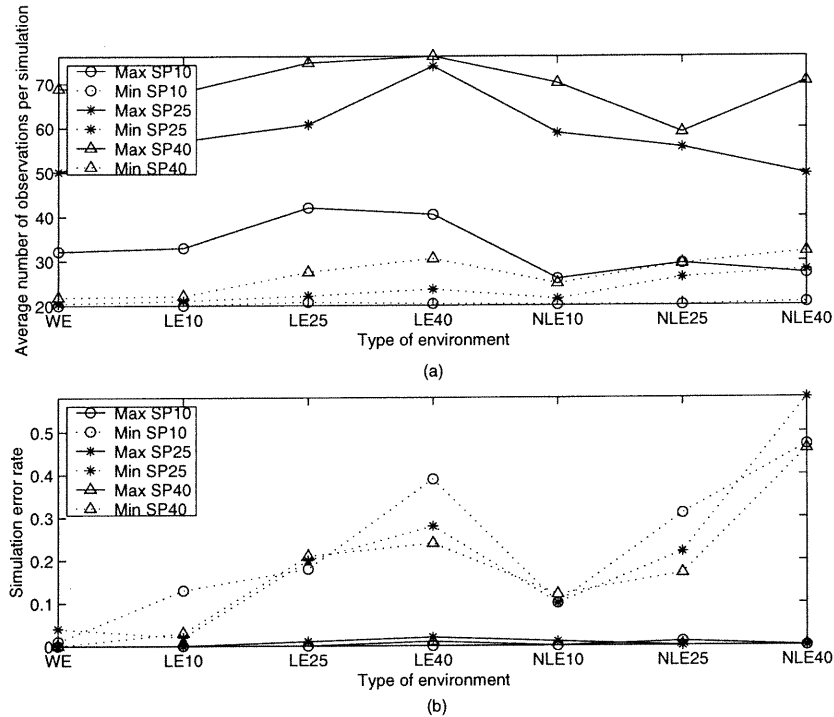


FIG. 3.3.4 – Effets du moteur séquentiel d'inférence floue du produit de fondement possibiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b)

l'union floue. Le moteur du minimum reste toujours cependant déclassé par celui du produit. L'atteinte du plafond en termes d'itérations survient relativement souvent (7% pour le produit et tout près de 45% pour le minimum dans le pire des cas). L'importance de ces résultats est minimisée par l'ampleur des performances du moteur bayésien. L'atteinte du plafond en termes d'itérations se situe alors à aussi peu que 5% dans le pire des cas et le nombre moyen de plates-formes détectées frisent l'unité (résultats non représentés).

Les résultats obtenus présentent plus d'instabilité que dans le cas probabiliste. Ce phénomène s'observe en majeure partie au niveau des moteurs du produit et du minimum avec opérateur d'union floue. Le nombre d'itérations, limité à 100 pour une simulation donnée, peut en être la cause. Toutefois, les résultats présentés

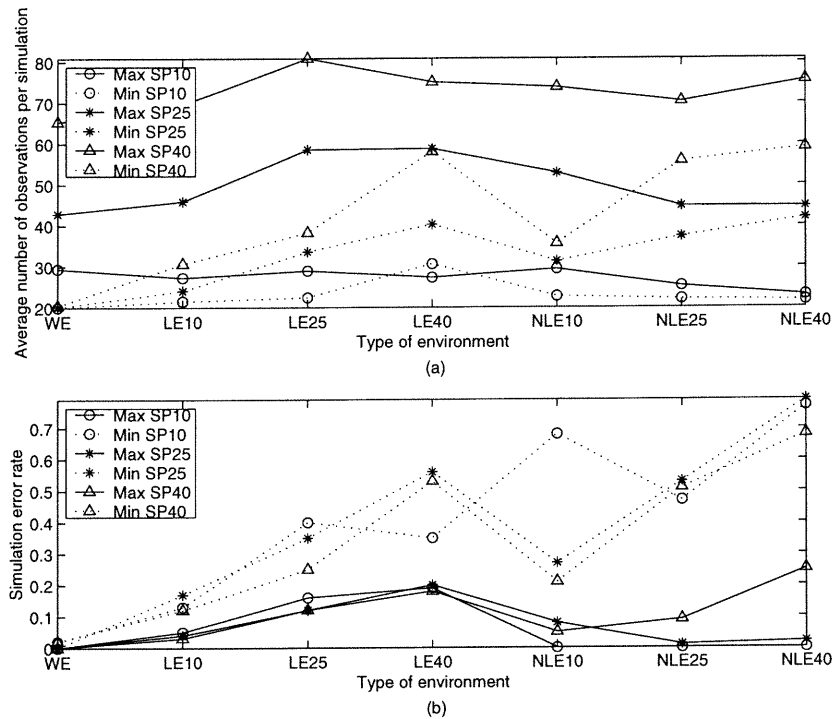


FIG. 3.3.5 — Effets du moteur séquentiel d'inférence floue du minimum de fondement possibiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b)

dans ce document ont été validés par une série de simulations équivalentes. Ils présentent ainsi sensiblement le comportement général des différents moteurs.

En conclusion, à la suite de l'étude du comportement séquentiel des moteurs d'inférence floue avec fondement possibiliste, le moteur bayésien se révèle le plus performant tant par son taux d'erreur réel faible, son nombre moyen minimal d'itérations ou de plates-formes détectées que par un dépassement du nombre d'itérations maximal fort peu probable.

### 3.3.2.3. Analyse des résultats

Globalement, les moteurs d'inférence floue, qu'ils soient de fondement probabiliste ou possibiliste, présentent un comportement général similaire face à

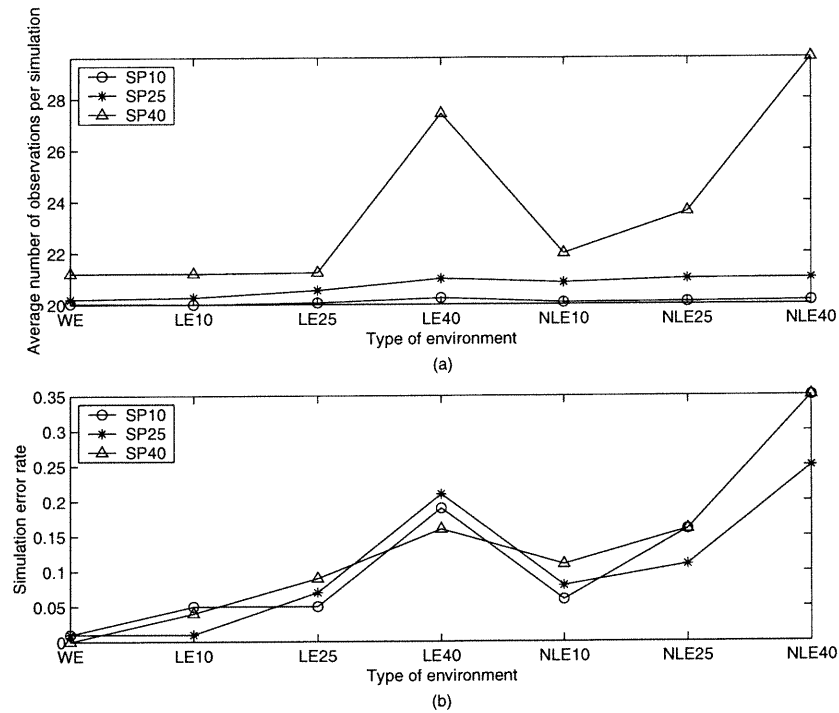


FIG. 3.3.6 — Effets du moteur séquentiel d'inférence floue bayésien de fondement possibiliste sur le nombre moyen d'itérations (a) et le taux d'erreur réel (b)

la croissance du taux d'erreur théorique et du critère d'arrêt de l'algorithme. L'impact de ces phénomènes varie grandement d'un modèle à l'autre, mais sensiblement les mêmes conclusions tirées à la section 3.3.1.3 pour la version non séquentielle des moteurs sont applicables ici. Les mêmes moteurs ont été étudiés, mais sous un angle à peine différent.

Tout d'abord, notons que certains moteurs par un trop grand optimisme et une imprécision élevée s'illustrent, mais négativement. En effet, les moteurs du produit et du minimum avec opérateur d'intersection floue et tout type de fondement présentent ainsi des comportements peu reluisants. Un optimisme exagéré s'avère également néfaste aux moteurs du produit et du minimum avec opérateur

d'union floue et fondement possibiliste même si celui-ci est jumelé à une efficacité inégalable. En résumé, le moteur du minimum reste concrètement inapplicable.

D'autres moteurs présentent toutefois un comportement fort acceptable et parfois même étonnant. En effet, le moteur du produit avec opérateur d'union floue et fondement probabiliste s'illustre notamment. Son comportement est comparable au moteur bayésien, mais son rendement et sa précision sont supérieurs. Il présente un taux d'erreur réel toujours limité, n'atteint jamais le plafond en termes d'itérations, possède un nombre moyen de plates-formes détectées par itération frisant l'unité et un nombre d'itérations moyen par simulation tout près de la limite inférieure permise. Son jugement se veut ainsi à la fois précis, sévère. Du côté du modèle de Bayes, les fondements possibiliste et probabilistes semblent équivalents et fournissent les mêmes performances. Sa grande sévérité, démontrée largement lors d'une évaluation non séquentielle, semble jouer cette fois légèrement en sa défaveur. Son jugement est centré autour d'une seule plate-forme et lors de l'insertion d'erreurs, les dommages causés plus importants.

En conclusion, à la suite de l'étude du comportement séquentiel des moteurs d'inférence floue, il a été révélé que le moteur du produit avec opérateur d'union floue et fondement probabiliste présente les performances les plus satisfaisantes suivi du moteur bayésien de tout type de fondement.

### 3.3.3. Impact du nombre de capteurs utilisés

Les simulations effectuées ici ne concernent plus spécifiquement le comportement des moteurs d'inférence floue, mais plutôt l'impact du nombre de senseurs utilisés lors d'une simulation. En effet, le nombre de capteurs fournissant des informations sur les attributs *vitesse*, *nombre d'émetteurs* et leur *type* n'est désormais plus aléatoire (voir section 3.2.1.1), mais davantage contrôlé afin de mesurer ses

effets sur le processus d'identification d'une plate-forme. Ainsi à l'instar des moteurs non séquentiels, de multiples données sont reçues à chaque itération, mais leur nombre est fixé arbitrairement à un, deux, trois ou quatre par attribut et demeure constant tout au cours d'une simulation.

Les conditions de simulation y sont toujours variées et la performance de chacun des moteurs est évaluée dans un environnement sans conflit, avec conflits limités et non limités avec taux d'erreur théoriques de 10%, 25% et 40% pour déterminer à quelle probabilité la détection primaire d'un senseur peut être entachée d'erreur (voir section 3.2.1.4 pour de plus amples informations). Désormais cependant, sous tout type de conflit, les simulations sont lancées pour une véritable plate-forme observée de type *surface* uniquement. Enfin, chacune de ces batteries de tests est appliquée tant à des moteurs de fondement probabiliste que possibiliste et la pondération de chaque senseur est conservé au niveau unitaire.

Les résultats, tous recueillis à la suite d'une série de 100 simulations de 100 itérations chacune, concernent principalement les effets sur le taux d'erreur réel et le nombre moyen de plates-formes détectées par itération. Dans tous les cas, ceux-ci sont suivis d'une étape d'analyse locale par critère (fondements confondus), complétée par une dernière plus globale en toute fin de sous-section. Les variations dans l'efficacité et la précision des moteurs d'inférence floue du produit, du minimum et bayésien seront ainsi évaluées.

### 3.3.3.1. *Effets sur le taux d'erreur*

La présente sous-section illustre les effets du nombre de capteurs utilisés dans les simulations avec moteurs d'inférence floue du produit, du minimum et bayésien de fondement probabiliste et possibiliste. Elle met davantage l'accent sur l'étude des effets sur le taux d'erreur réel à en juger par les graphiques 3.3.7, 3.3.8 et

3.3.9 dans le cas d'un fondement probabiliste et 3.3.10, 3.3.11 et 3.3.12 dans le cas possibiliste. Pour les moteurs du produit et du minimum, ces derniers présentent tout d'abord les résultats pour un moteur avec opérateur d'union floue (a) et terminent avec l'intersection floue (b).

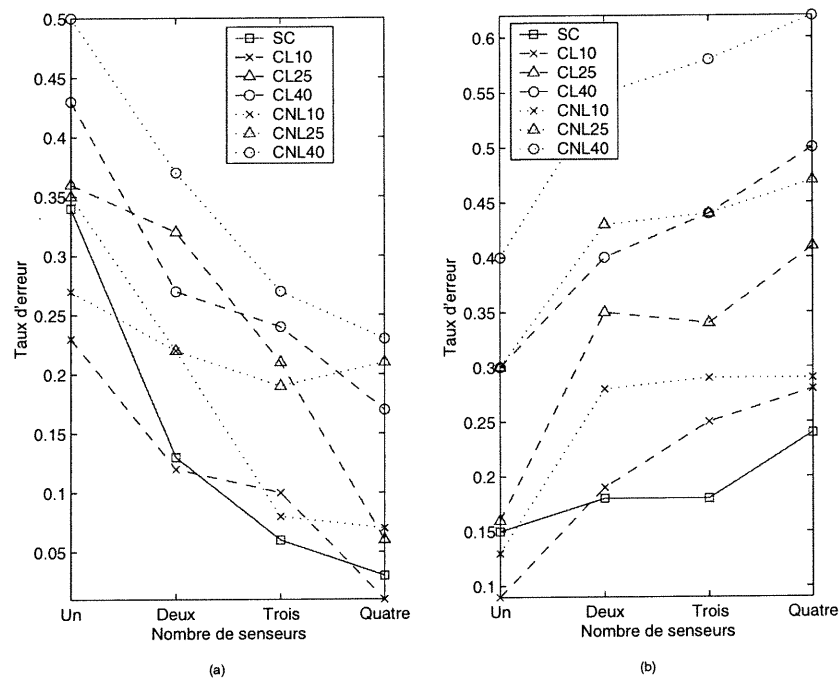


FIG. 3.3.7 -. Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du produit de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b)

Les graphiques semblent tous légèrement différents, mais en observant attentivement, le comportement général d'un moteur d'inférence floue face à la variation du nombre de capteurs utilisés peut être déduit. Les effets sur le taux d'erreur réel peuvent être obtenus notamment par l'étude détaillée du modèle bayésien. On constate tout d'abord une diminution sans équivoque du taux d'erreur avec l'augmentation du nombre de capteurs, mais celle-ci plafonne rapidement et tente même à croître, dans certain cas, à l'approche d'un nombre élevé de capteurs

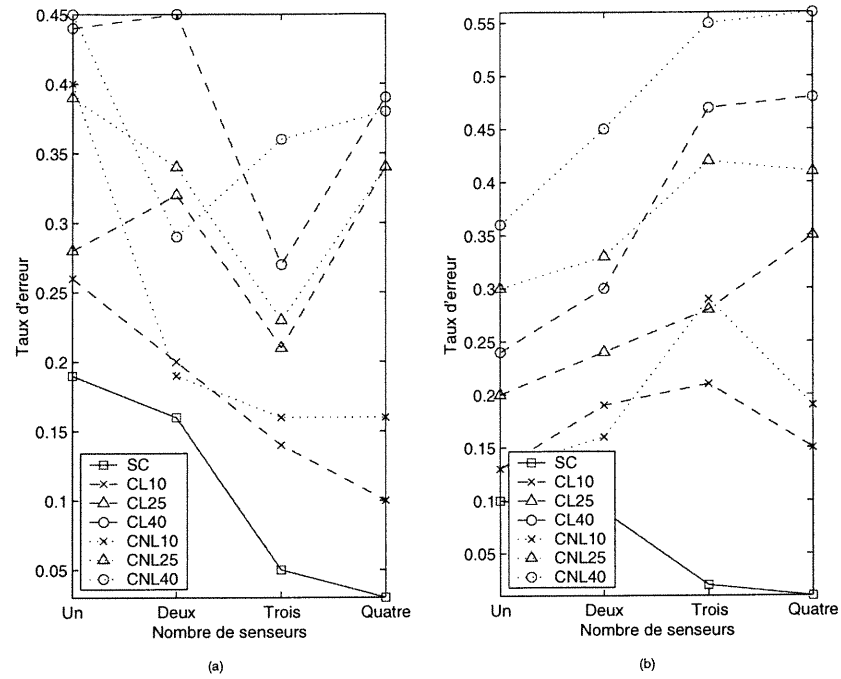


FIG. 3.3.8 – Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du minimum de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b)

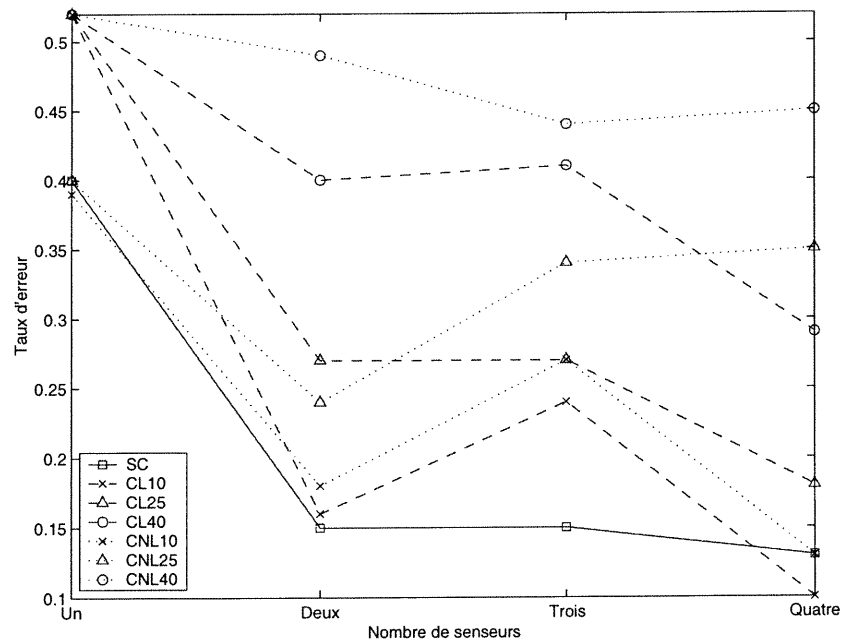


FIG. 3.3.9 – Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue bayésien de fondement probabiliste



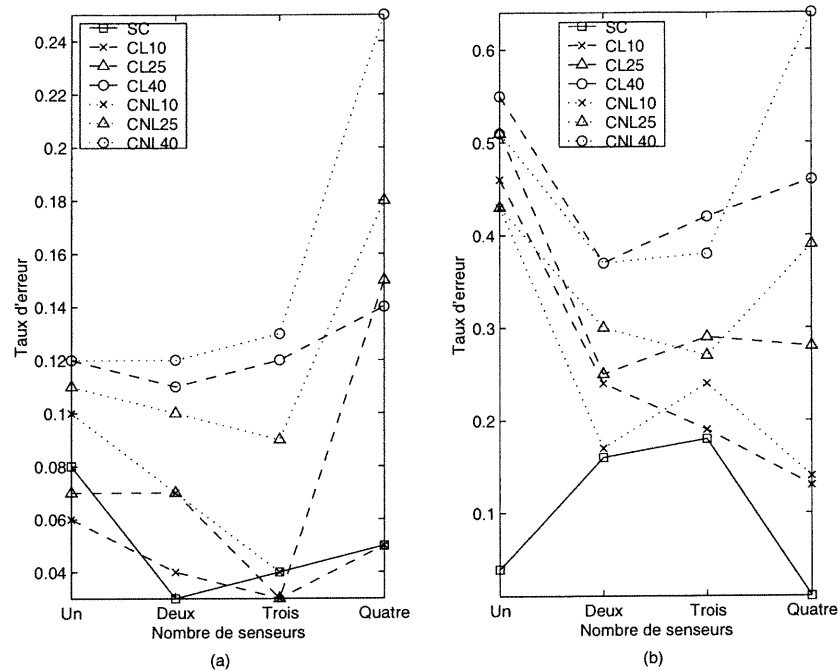


FIG. 3.3.10 – Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du produit de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b)

(quatre). En augmentant le nombre d'observations, la précision croît, mais du même coup les chances de recevoir des observations contradictoires aussi. Le taux d'erreur réel n'est pas influencé par le fondement du moteur étudié, qu'il soit probabiliste ou possibiliste, ces conclusions restent sensiblement les mêmes.

De façon plus détaillée, on remarque que le moteur du produit probabiliste avec opérateur d'union floue présente le comportement le plus radical face à l'augmentation du nombre de capteurs. Plus ce dernier augmente, plus son identification est digne de confiance. La courbe décroît cependant de façon non continue et, en extrapolant, finit par plafonnée. Dans sa version avec opérateur d'intersection floue de même que le minimum probabiliste, les résultats semblent presque tout-à-fait contraires. La pente de la courbe, tantôt négative, devient maintenant

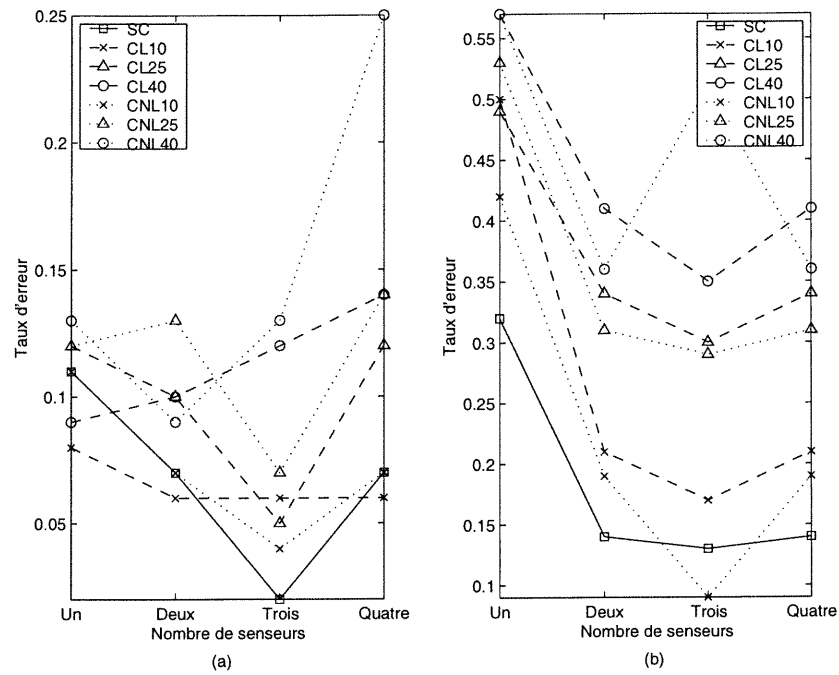


FIG. 3.3.11 – Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue du minimum de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b)

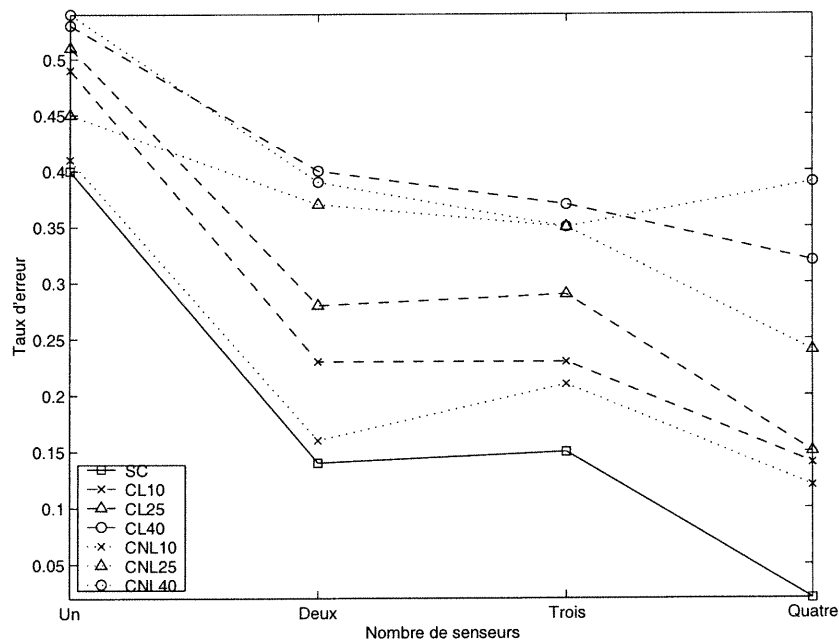


FIG. 3.3.12 – Effets du nombre de capteurs sur le taux d'erreur réel pour le moteur non séquentiel d'inférence floue bayésien de fondement possibiliste

positive surtout dans le cas de conflits élevés. En effet, ces moteurs, d'efficacité faible, reçoivent de plus en plus d'informations avec l'augmentation du nombre de capteurs et leur confusion augmente. Ce comportement prouve de nouveau leur incapacité à effectuer adéquatement l'identification de cibles. Dans le cas présent, à l'optimalité, l'emploi d'un nombre de deux ou trois senseurs serait ainsi à privilégier.

Les effets sur le taux d'erreur d'un nombre de senseurs variable semblent d'importance égale peu importe le type de conflit et le taux d'erreur théorique appliqués à la simulation. Il est, en effet, plutôt ardu de conclure qu'un type de conflit subit plus de variations qu'un autre suite à la seule analyse des graphiques 3.3.8 à 3.3.12. Mais, en général, plus le niveau de difficulté d'une simulation est élevé, plus les taux d'erreur réels correspondants le sont. Ainsi une courbe décrivant le comportement d'un moteur suite à une simulation sans conflit sera généralement inférieure en ordonnée à une autre d'une même simulation, mais avec conflits limités et taux d'erreur de 40%.

En conclusion, le taux d'erreur diminue avec l'augmentation du nombre de capteurs, mais de façon non régulière et présente même des effets négatifs s'il est trop élevé. Dans ces conditions un nombre de capteurs limités à deux ou trois est à privilégier.

### 3.3.3.2. *Effets sur le nombre moyen de plateformes détectées*

La présente sous-section illustre toujours les effets du nombre de capteurs utilisés dans les simulations avec moteurs d'inférence floue du produit, du minimum et de Bayes de fondement probabiliste et possibiliste. Mais elle met davantage l'accent sur l'étude des effets sur le nombre moyen de plates-formes détectées tel que démontré par les graphiques 3.3.13, 3.3.14 et 3.3.15 dans le cas d'un fondement

probabiliste et 3.3.16, 3.3.17 et 3.3.18 dans le cas possibiliste. Pour les moteurs du produit et du minimum, ces derniers présentent tout d'abord les résultats pour un moteur avec opérateur d'union floue (a) et terminent avec l'intersection floue (b).

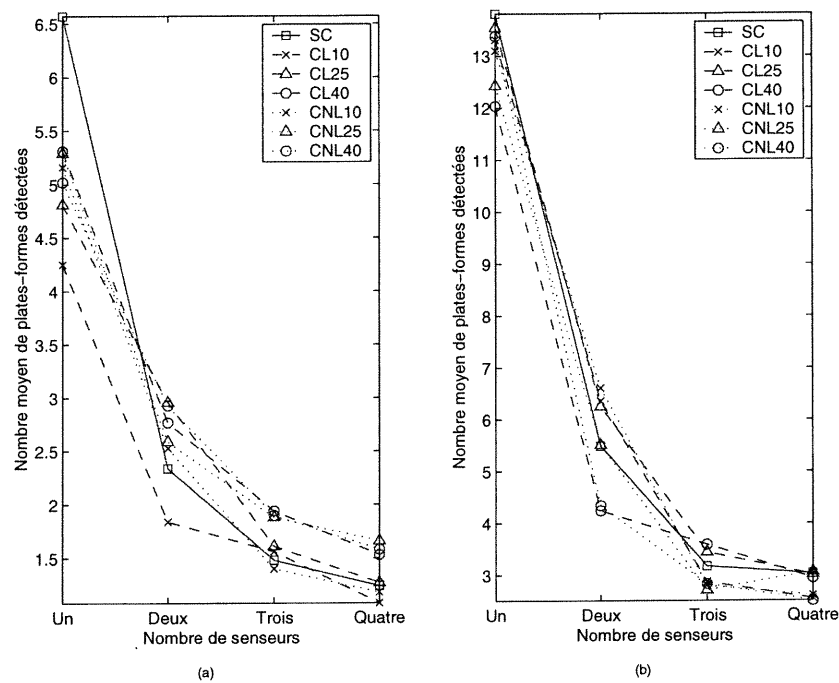


FIG. 3.3.13 – Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du produit de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b)

Les graphiques 3.3.13 à 3.3.18 illustrent de façon éloquentes les conséquences de la variation du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération. Les effets sont également beaucoup plus évidents qu'au niveau du taux d'erreur réel (voir section 3.3.3.1). Un même comportement général s'applique toujours aux différents moteurs de tout fondement. On constate tout d'abord une diminution importante du nombre moyen de plates-formes avec

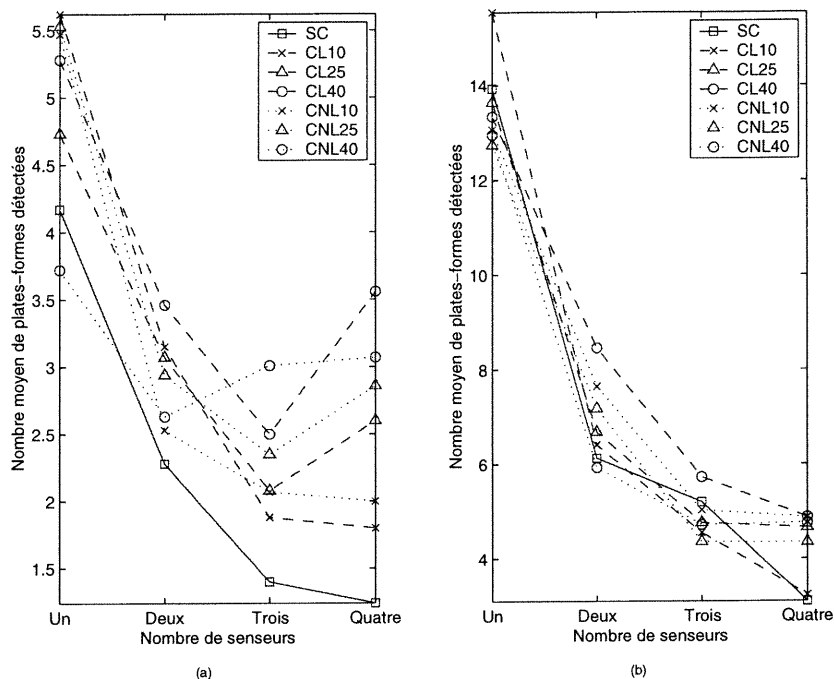


FIG. 3.3.14 – Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du minimum de fondement probabiliste avec opérateur d'union (a) et d'intersection floues (b)

l'augmentation du nombre de capteurs. La courbe finit par plafonner, mais lentement. La pente est nettement plus brusque que dans la cas de l'analyse du taux d'erreur réel. Elle tente même à croître, dans certain cas à l'approche d'un nombre élevé de capteurs (quatre), mais moins fréquemment. Ce phénomène se produit surtout lors d'une simulation avec probabilité élevée de conflits, soit des taux réels théoriques de 25% et 40%. En augmentant le nombre d'observations, la précision croît de façon importante, mais du même coup les chances de recevoir des observations contradictoires croissent aussi. Le nombre moyen de plates-formes détectées n'est toujours pas influencé par le fondement du moteur étudié, qu'il soit probabiliste ou possibiliste, ces conclusions restent sensiblement les mêmes.

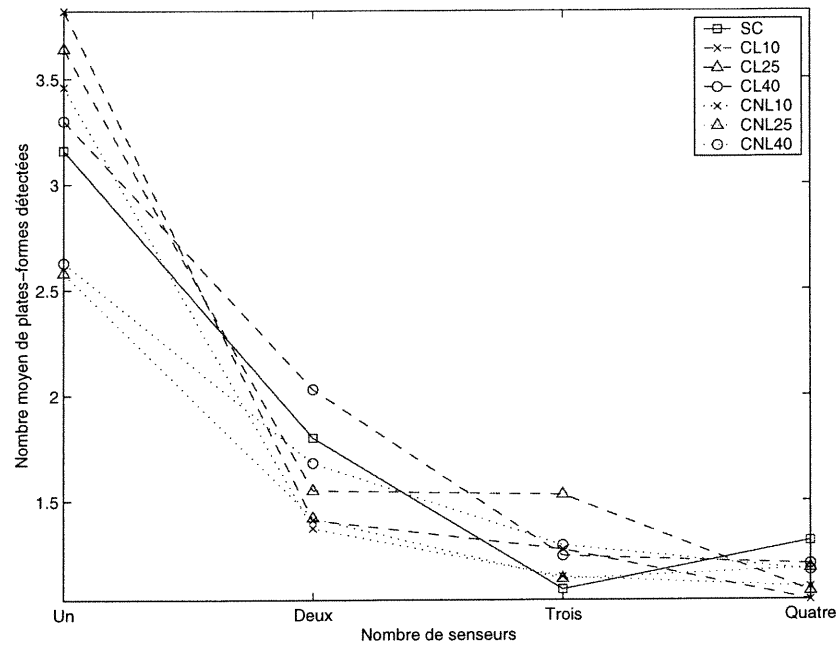


FIG. 3.3.15 — Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur d'inférence floue bayésien de fondement probabiliste

De façon plus détaillée, on remarque que les moteurs les plus performants (produit probabiliste avec opérateur d'union floue et bayésien de tout fondement) possèdent une courbe dont le pente est la plus élevée. Les effets observés sur le nombre moyen de plates-formes détectées par itération sont vraiment drastiques. Dans leur cas, le nombre de capteurs le plus important possible serait à privilégier, soit quatre dans le cas présent. Pour les autres moteurs de moindre performance, la courbe subit quelque fois une légère remontée au niveau d'un nombre élevé de capteurs. L'optimalité dans ce cas est atteint au minimum de cette courbe, soit pour trois capteurs.

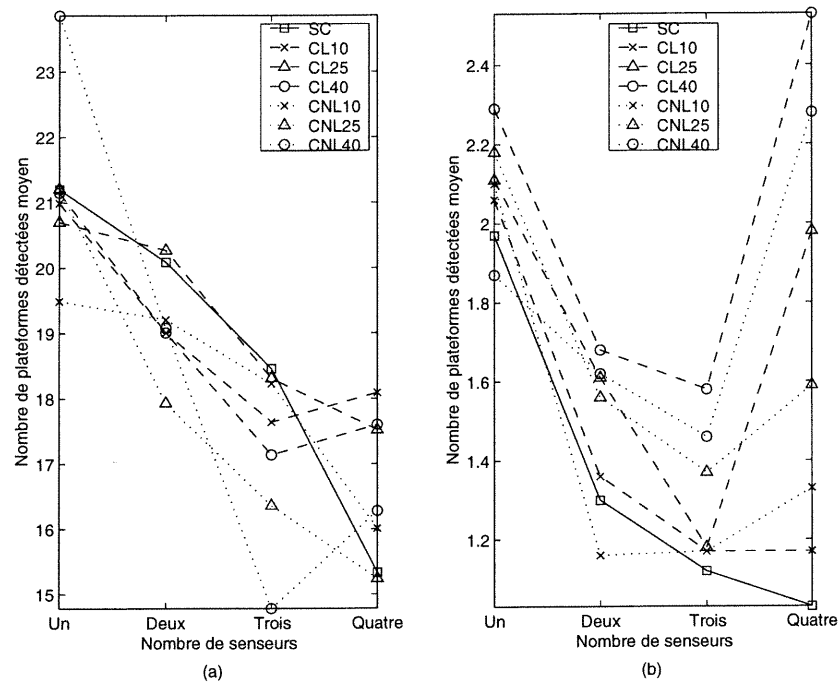


FIG. 3.3.16 – Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du produit de fondement possibiliste avec opérateur d'union (a) et d'intersection floues (b)

### 3.3.3.3. Analyse des résultats

En résumé, les effets subis par les différents moteurs sont assez similaires face à la variation du nombre de senseurs. Mais ils deviennent beaucoup plus révélateurs au niveau du nombre moyen de plates-formes détectées par itération qu'au niveau du taux d'erreur réel. En général, les courbes observées décroissent rapidement et finissent par plafonner avec l'augmentation du nombre de capteurs. Parfois même une légère remontée est observable lorsque le taux d'erreur théorique est important. Il apparaît ainsi clairement que l'augmentation du nombre de capteurs peut être bénéfique aux moteurs, mais de façon limitée. En effet, un

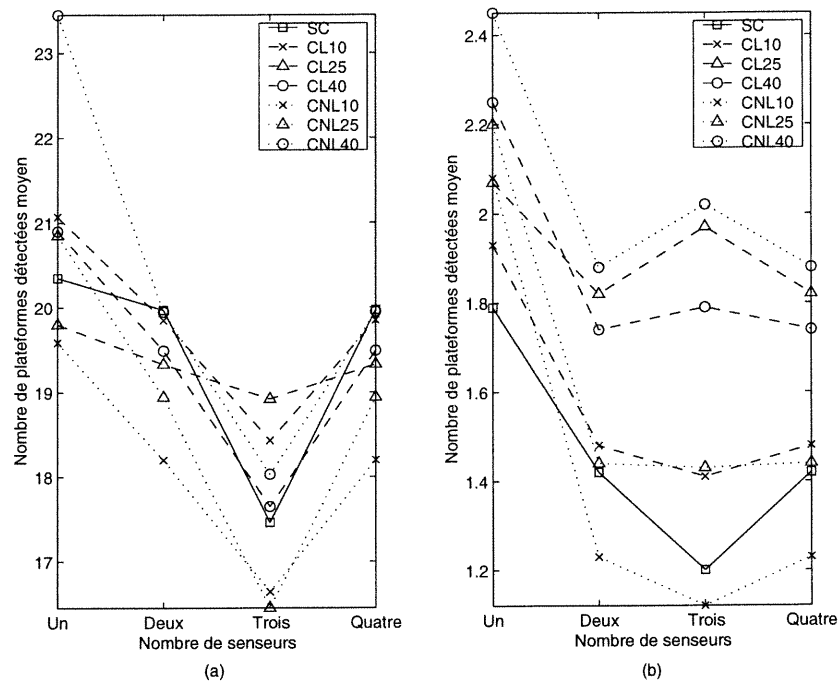


FIG. 3.3.17 — Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur du minimum de fondement possibiliste avec opérateur d'union (a) et d'intersection flous (b)

nombre trop important de capteurs fournit un apport d'information négligeable et il est même parfois néfaste.

Cette analyse aura également permis indirectement de confirmer la grande performance des moteurs jugés fort intéressants dans les dernières analyses du comportement séquentiel (voir section 3.3.2) et non séquentiel (voir section 3.3.1) des moteurs. En effet, les moteurs du produit avec opérateur d'union floue et fondement probabiliste et ceux bayésiens de tout fondement présentent des taux de décroissance plus élevés donc une réponse plus rapide à l'augmentation du nombre de capteurs. Ils sont rarement noyés dans la confusion apportée par une quantité importante d'information.



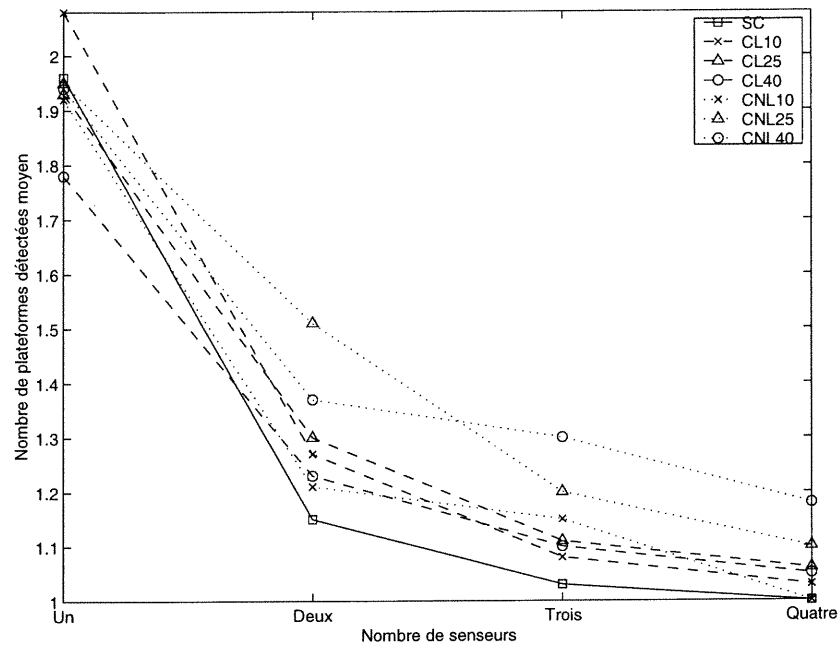


FIG. 3.3.18 — Effets du nombre de capteurs sur le nombre moyen de plates-formes détectées par itération pour le moteur d'inférence floue bayésien de fondement possibiliste

Ainsi à la suite de l'analyse de toutes ces données, il est aisé de conclure que pour des résultats optimaux le nombre de senseurs doit être limité. Pour les moteurs jugés supérieurs comme pour les autres, un nombre de capteurs de trois et non de quatre serait à privilégier. Le maximum de capteurs ne permet donc pas nécessairement d'atteindre un rendement optimum.

### 3.3.4. Analyse globale des résultats

Ce dernier chapitre a présenté le fruit de plusieurs mois de recherche dans un nouveau domaine des mathématiques, la logique floue. Différents moteurs d'inférence floue, d'abord définis au premier chapitre, puis appliqués à l'identification de cibles, ont ainsi été scrutés sous tous les angles afin de démontrer ce dont ils

étaient vraiment capables. C'est ainsi que les moteurs du produit, du minimum et bayésien ont fait l'objet d'une batterie de tests concernant à la fois leur comportement non séquentiel (traitement parallèle des données) et séquentiel (traitement sériel des données) pour parvenir à déterminer leur efficacité, leur précision et leur rendement respectifs. Une analyse indirecte sur les moteurs a également été abordée portant sur l'impact du nombre de capteurs utilisés lors d'une simulation.

Toute cette série d'analyse a permis de faire ressortir les performances de trois moteurs d'inférence floue. En effet, les moteurs du produit avec opérateur d'union floue et fondement probabiliste et bayésien de tout fondement se sont illustrés tout spécialement dans de multiples circonstances. Au niveau de l'analyse non séquentielle, le moteur bayésien avec fondement possibiliste a su démontrer sa supériorité par un taux d'erreur réel toujours limité et un nombre moyen de plates-formes détectées frisant l'unité. Son jugement se veut ainsi à la fois précis et sévère. Il est suivi de près de son homologue probabiliste et du moteur du produit avec opérateur d'union floue et fondement probabiliste. Du côté de l'analyse séquentielle, le moteur du produit avec opérateur d'union floue et fondement probabiliste présente les performances les plus satisfaisantes suivi, par une marge importante, par le moteur bayésien de tout type de fondement. Ainsi, globalement, il s'avère que le moteur d'inférence floue du produit avec opérateur d'union floue et fondement probabiliste est le plus performant suivi du moteur bayésien possibiliste qui surpasse légèrement celui probabiliste. Ces moteurs peuvent être appliqués sans gêne à tout type de simulation et les autres modèles, de moindre performance, restent somme tout dignes d'intérêt.

Du côté de l'influence du nombre de capteurs utilisés pour recueillir de l'information sur le comportement des moteurs, il a été observé que le taux d'erreur réel et le nombre moyen de plates-formes détectées par itération diminuent avec

l'augmentation du nombre de capteurs. Ce phénomène se produit toutefois de façon non régulière et présente même des effets négatifs si le nombre de capteurs est trop élevé. Dans ces conditions il est jugé pertinent de restreindre le nombre de senseurs à trois pour tout type de moteur. L'optimalité n'est pas forcément une conséquence de l'emploi du nombre maximum de capteurs.

Le présent chapitre a finalement démontré sans équivoque le bien-fondé de l'utilisation de la logique floue en identification de cibles : un rendement et une efficacité remarquables jumelés à une précision souvent fort étonnante. Cette nouvelle venue du monde mathématique présente, en effet, un point de vue digne d'intérêt qui risque de se comparer avantageusement aux techniques plus conventionnelles. L'identification y est efficace même en présence d'un taux élevé de contradictions ou d'incertitude et par dessus tout, cette technique ne possède aucun désavantage majeur gênant son utilisation. À titre de projet futur, il semblerait intéressant de considérer une comparaison globale des principales théories de pointe en identification de cibles, soit celles bayésienne (réseaux bayésiens), évidentielle et floue (voir la section 2.2 pour un rappel de ces méthodes).

## CONCLUSION

---

Ce mémoire de maîtrise a présenté une des applications modernes de la logique floue au cours des dernières années, soit la conception de systèmes experts flous destinés à l'identification de cibles. Les diverses étapes de ce projet ont été fort ordonnées et détaillées. Le premier chapitre a tout d'abord présenté un aperçu global des bases de la logique floue, de la théorie des ensembles flous aux opérateurs flous les plus couramment employés en passant par toutes les subtilités des systèmes d'inférence floue, coeur même du présent mémoire. Le tout a été largement agrémenté d'exemples et d'illustrations diverses.

Le second chapitre, quant à lui, a marqué la véritable entrée en matière de ce mémoire. En plus de préciser la nature des travaux de recherche, il a abordé différentes facettes de l'identification de plates-formes dont les attributs et leur modélisation, les techniques de conception de systèmes experts flous ainsi que les fondements du raisonnement approché. Globalement, les efforts ont été concentrés autour des moteurs d'inférence floue jugés les plus populaires dans le domaine du génie : les moteurs du produit et du minimum. Cependant une tangente statistique a également été ajoutée au projet. En effet, un tout nouveau moteur bayésien a été conçu. Il s'inspire grandement du monde de la statistique.

Finalement, les systèmes experts flous ont été scrutés sous tous les angles au troisième et dernier chapitre. Une dizaine de simulations variées ont permis d'analyser les moteurs d'inférence floue développés selon leur approche probabiliste et possibiliste respective. Les résultats obtenus ont révélé, hors de tout doute

possible, que le moteur d'inférence floue du produit avec opérateur d'union floue et fondement probabiliste s'avère le plus performant suivi du moteur bayésien possibiliste qui surpasse légèrement celui probabiliste. Ces moteurs peuvent être appliqués sans gêne à tout type de simulation. Les autres modèles, de moindre performance, restent dignes d'intérêt. L'influence du nombre de capteurs utilisés sur le comportement des moteurs d'inférence a également été sujet d'étude. Il a été observé que le taux d'erreur réel et le nombre moyen de plates-formes détectées par itération diminuent avec l'augmentation du nombre de capteurs. Ce phénomène se produit toutefois de façon non régulière et présente même des effets négatifs dans le cas de conflits extrêmes si le nombre de capteurs est trop élevé. Dans ces conditions il est jugé pertinent de restreindre le nombre de senseurs à trois pour tout type de moteur. L'optimalité n'est pas forcément une conséquence de l'emploi du nombre maximum de capteurs.

Le présent mémoire a démontré le bien-fondé de l'utilisation de la logique floue en identification de cibles : un rendement et une efficacité remarquables jumelés à une précision souvent fort étonnante. Cette nouvelle venue du monde mathématique s'illustre principalement par une diversité inégalable. Elle est constituée d'un éventail fort complet d'opérateurs récupérant même ceux de son proche parent, les mathématiques classiques. Il semble d'ailleurs pertinent d'entrevoir, à titre de projet futur, la conception de nouveaux moteurs d'inférence floue possédant toujours la structure standard, mais tirant davantage profit de cette diversité. Du côté de leur structure, des hypothèses pointues et une modélisation fort rigoureuse ont été employées. Il est donc pertinent d'envisager l'étude des effets de la modélisation des lois de probabilité et de possibilité sur leurs performances globales. Enfin, les différents systèmes experts flous développés ont confirmé leur efficacité même en présence d'un taux élevé de contradictions ou d'incertitude et

ne possèdent aucun désavantage majeur gênant leur utilisation. Cette technique s'avère ainsi digne d'un intérêt certain et une comparaison globale avec les autres théories dans le domaine de l'identification de cibles, soient les réseaux bayésiens et la théorie évidentielle reste à considérer dans le cadre de prochains travaux.

## Bibliographie

---

- Abdulghafour, M., M. A. Abidi, et A. Fellah (1994). Fuzzy logic - based data integration: Theory and applications. *Proceedings of the 1994 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 151–160.
- Asai, K., M. Sugeno, et T. Terano (1991). *Fuzzy Systems Theory and Its Applications*. San Diego, États-Unis: Academic Press.
- Bailey, W. H. et R.-N. P. Singh (1997). Fuzzy logic applications to multisensor-multitarget correlation. *IEEE Transactions on Aerospace and Electronic Systems* **33**(3), 752–768.
- Bobrowicz, O., C. Choulet, A. Haurat, F. Sandoz, et M. Tebaa (1990). A method to build membership functions, application to numerical / symbolic interface building. *Proceedings of the Third International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 136–142.
- Borne, P. et F. Delmotte (1998). Modeling of reliability with possibility theory. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans* **28**(1), 78–88.
- Chen, C.-H. et M.-S. Yang (1998). On the edited fuzzy k-nearest neighbor rule. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* **28**(3), 461–466.

- Cumani, A. (1982). On a possibilistic approach to the analysis of fuzzy feedback systems. *IEEE Transactions on Systems, Man, and Cybernetics* **12**(3), 417–422.
- de Oliviera, J. V. et W. Pedrycz (1996). Optimization of fuzzy models. *IEEE Transactions on Systems, Man, and Cybernetics - Part B: Cybernetics* **26**(4), 627–636.
- Dishkant, H. (1981). About membership functions estimation. *Fuzzy Sets and Systems* **5**, 141–147.
- Dubois, D. et H. Prade (1980). *Fuzzy Sets and Systems*. New York, États-Unis: Academic Press.
- Dubois, D. et H. Prade (1988). *Théorie des possibilités*. Paris, France: Masson.
- Folger, T. A. et G. J. Klir (1988). *Fuzzy Sets, Uncertainty, and Information*. New York, États-Unis: Prentice-Hall.
- Gacôgne, L. (1997). *Éléments de logique floue*. Paris, France: Hermes.
- Genest, C. et J. V. Zidek (1986). Combining probability distributions: A critique and an annotated bibliography. *Statistical Science* **1**(1), 114–148.
- Kaufmann, A. (1973). *Introduction à la théorie des sous-ensembles flous*. Paris, France: Masson et Cie.
- Kummert, A. (1993). Fuzzy technology implemented in sonar system. *IEEE Journal of Oceanic Engineering* **18**(4), 483–490.
- Langari, R., J. Yen, et L. A. Zadeh (1995). *Industrial Applications of Fuzzy Logic and Intelligent Systems*. Piscataway, États-Unis: IEEE Press.
- OFTA (1994). *Logique floue*. Paris, France: Masson.
- Samarasooriya, V. N. S. et P. K. Varshney (1996). A fuzzy modeling approach to decision fusion under uncertainty. *Proceedings of the 1996 IEEE/SICE/RSJ*



- International Conference on Multisensor Fusion and Integration for Intelligent Systems*, 788–795.
- Sangalli, A. (1998). *The Importance of Being Fuzzy and Other Insights from the Border between Math and Computer*. Princeton, États-Unis: Princeton University Press.
- Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton, États-Unis: Princeton University Press.
- Smith, J. Q. (1988). *Decision Analysis A Bayesian Approach*. Londre, Grande-Bretagne: Chapman and Hall.
- Sugeno, M. (1977). Fuzzy measures and fuzzy integrals - a survey. *Fuzzy Automata and Decisions Processes*, 89–102.
- Wang, L.-X. (1997). *A Course in Fuzzy Systems and Control*. Upper Saddle River, États-Unis: Prentice Hall.
- West, M. (1988). Modelling expert opinion. *Bayesian Statistics* **3**, 493–508.
- Zadeh, L. A. (1965). Fuzzy sets. *Information and Control* **8**, 338–353.
- Zadeh, L. A. (1969). Toward a theory of fuzzy systems. Technical report, Université de Berkeley, Berkeley, États-Unis.
- Zimmermann, H. J. (1985). *Fuzzy Set Theory and Its Applications*. Norwell, États-Unis: Kluwer-Nijhoff Publishing.