

N° d'ordre : 2604

THESE

présentée à

l'Université des Sciences et Technologies de Lille

pour obtenir le titre de Docteur en

Productique : Automatique et Informatique Industrielle

par

Gilles Zunino

Ingénieur H.E.I. (Hautes Etudes Industrielles, Lille)

**Amélioration du Processus de Fusion d'Informations
dans le cadre de la Théorie des Possibilités**

Soutenue le 25 Novembre 1999 devant le jury d'examen composé de :

Président :

S. MAOUCHE

Rapporteurs :



I. BLOCH
L. FOULLOY

Examineurs :

A.M. JOLLY-DESODT
D. JOLLY
F. DELMOTTE
P. VANHEEGHE

Travaux réalisés sous la direction du Professeur A.M. JOLLY-DESODT et la co-direction du Professeur D. JOLLY au Laboratoire d'Automatique I³D, Université des Sciences et Technologies de Lille et à l'école des Hautes Etudes Industrielles.

Remerciements

Le travail présenté ici a été réalisé au sein du Centre d'Automatique de Lille devenu depuis le laboratoire Interaction, Image et Ingénierie de la Décision de Lille et du département Informatique Industrielle de l'Ecole des Hautes Etudes Industrielles. J'aimerais que le Professeur P. VIDAL directeur du Centre d'Automatique de Lille, le Professeur C. VASSEUR directeur du laboratoire I3D et M. VITTU directeur de HEI trouvent ici le témoignage de ma profonde gratitude.

A travers ces années de collaboration avec A.M. JOLLY et D. JOLLY j'ai apprécié les Directeurs de Recherches tout autant que les personnes. Ils m'ont guidé dans ce travail jusqu'à son terme. Leurs critiques et encouragements ont largement contribué à l'aboutissement de ces travaux. Qu'ils soient ici mille fois remerciés.

Je suis honoré que Madame le Professeur I. Bloch de l'Ecole Nationale Supérieure des Télécommunications ait accepté d'être l'un de mes rapporteurs. J'ai eu l'occasion dans le cadre des journées thématiques en fusion d'information du GDR-ISIS puis aux Rencontres francophones sur la Logique Floue et ses Applications d'avoir avec elle des discussions très enrichissantes sur mon travail de thèse.

Je suis très heureux que Monsieur le Professeur L. Foulloy de l'Université de Savoie ait accepté d'être l'un de mes rapporteurs. Peut-être ne le sait-il pas mais c'est à l'occasion de SICICA'97 dont il était l'organisateur que j'ai présenté pour la première fois le résultat de mes travaux. Notre longue discussion en présence d'Anne-Marie à l'aéroport de Lesquin m'a beaucoup apporté et me donne déjà de nouvelles idées de recherche.

Je remercie aussi Monsieur le Professeur P. VanHeege de l'Ecole Centrale de Lille d'avoir accepté de participer à ce jury. Ses travaux sur la fusion multicapteurs ont accompagné mon travail.

Monsieur François Delmotte a toujours été pour moi une source intarissable d'idées et ses précieux conseils m'ont maintes fois évité de m'engager sur une voie sans issue. François est aussi un ami dont j'apprécie beaucoup la compagnie. Je me rappelle de cette promenade à Annecy que nous avons faite ensemble par une belle après-midi d'été. Je le remercie d'avoir accepté de participer à ce jury.

Je tiens également à remercier P. VANCOPPENOLLE, Chef du Département Informatique Industrielle à HEI pour m'avoir accueilli au sein de son laboratoire. C'est lui qui m'a accompagné lors de mes premiers pas en informatique.

Lorsque les maquettes logicielles se bloquent jusqu'à nécessiter le redémarrage de la machine, qu'il est bon de se sentir soutenu ! Merci à T. SOBANSKI, P. BIELA, V. LEFÈVÈRE, J.M. FAYEULLE et H. HADJEM pour leurs précieux conseils et pour leur patience pendant le ralentissement substantiel des serveurs mis à mal par mes tests comportant des algorithmes NP-complets !!

Il y a des rencontres qui comptent et celle avec D. RUCKEBUSH en fait partie. Nos longues discussions m'ont aidé à dissiper mes appréhensions. Qu'il soit ici vivement remercié pour sa patience si souvent sollicitée.

Mes remerciements s'adressent aussi à mes parents pour la relecture de ce manuscrit, le soutien qu'ils m'ont toujours apporté et pour m'avoir donné l'envie d'apprendre. J'espère qu'ils sont aussi fier de moi que je le suis d'eux.

Il est impossible de citer tous ceux qui m'ont aidé soutenu ou simplement rendu ces années de travail plus agréables. Qu'ils soient ici vivement remerciés.

Table des matières

Introduction	1
1 Les opérateurs simples	7
1.1 Les trois comportements de base : conjonction, compromis et disjonction	8
1.1.1 La conjonction	8
1.1.2 Le compromis	8
1.1.3 La disjonction	9
1.2 Hypothèse du monde ouvert ou du monde fermé	9
1.3 Les propriétés des opérateurs de fusion	10
1.4 Les différents types d'opérateurs	14
1.4.1 La normalisation	15
1.4.2 Opérateurs indépendant du contexte à comportement constant (CICB)	16
1.4.2.1 Intersection et union floues ou T-Normes et T-Conormes	16
1.4.2.2 Les moyennes	17
1.4.3 Opérateurs indépendants du contexte à comportement variable (CIVB)	19
1.5 Quelques exemples	20
1.6 L'importance du choix du mode de combinaison	21
1.7 Conclusion	23
2 Les opérateurs adaptatifs	25
2.1 Quantifier l'accord entre les sources	25
2.1.1 Similitude et analyse de données	25
2.1.2 Extension au cas flou	30
2.1.3 Les indices de conflit	32
2.1.4 De nouveaux indices de conflits	33
2.1.4.1 Indices de conflit et distances	33
2.1.5 Un nouvel indice de conflit : le conflit horizontal	35
2.1.6 Une nouvelle vision du conflit : l'indicateur de conflit flou	36
2.2 Exemples	37
2.3 Les opérateurs CD	39
2.3.1 La règle adaptative de DUBOIS et PRADE	39
2.3.2 La règle "saturée" de [Deveughele et Dubuisson, 1996]	42
2.3.3 La règle de fusion linéaire de DUBOIS et PRADE	43
2.4 Nouvelles règles	43
2.4.1 Une nouvelle règle logique	44
2.4.2 Une nouvelle règle arithmétique	45
2.4.3 Une règle union/compromis	46
2.5 Conclusion	47

3	Fusionner un grand nombre de sources	49
3.1	La fusion par “cascade” ou fusion modulaire	50
3.1.1	Les opérateurs associatifs et commutatifs	50
3.1.2	Exemple	51
3.1.3	Opérateur effectuant un classement des sources	52
3.2	La réduction du nombre de sources	52
3.3	Fusionner n sources : la règle “ou-exclusif”	53
3.3.1	F est vide et l’union des sources est choisie	54
3.3.2	F est vide et l’ignorance complète est choisie	54
3.3.3	F est vide et rien n’est imposé au résultat	54
3.3.4	Définition des opérateurs dans le cadre flou et dans le cadre possibiliste	55
3.4	Les méthodes “hybrides”	55
3.5	Formation des classes	55
3.5.1	Conflit pour deux ensembles de sources	56
3.5.2	La stratégie de groupement et le critère d’arrêt	57
3.5.3	La détermination du seuil	59
3.6	La fusion intra classes	59
3.7	La fusion inter classes	60
3.8	Exemple	60
3.8.1	Prise en compte de la fiabilité des sources	60
3.9	Conclusion	61
4	Les opérateurs spéciaux	63
4.1	L’agrégation de critères ou les opérateurs OWA	64
4.1.1	F et le nombre de critères à vérifier	65
4.1.2	Quelques propriétés : la mesure du caractère “et”/”ou” d’un vecteur OWA	66
4.1.3	F et l’importance de chaque critère	67
4.1.4	Exemple	68
4.2	Principe d’extension et opérateurs de fusion	70
4.2.1	Lorsque f est monodimensionnelle	70
4.2.2	Lorsque f est multidimensionnelle	70
4.2.3	Lorsque f est vectorielle	71
4.2.4	Exemple	74
4.2.5	La compatibilité entre les sources	75
4.2.6	L’agrégation partielle	76
4.3	Les intégrales floues	79
4.3.1	Quelques mesures floues	79
4.3.2	L’intégrale de SUGENO	80
4.3.3	L’intégrale de CHOQUET	80
4.3.4	L’intégrale à T-Conormes	81
4.3.5	Quelques exemples	81
4.3.5.1	L’évaluation d’étudiants	82
4.3.5.2	Localisation d’un objet dans un espace à deux dimensions	83
4.4	Conclusion	85
5	La fusion dissymétrique	87
5.1	La fusion dissymétrique d’informations	87
5.1.1	Les règles de conditionnement (révision)	88
5.1.1.1	Révision	88
5.1.1.2	Focalisation	88
5.1.2	Les règles de projection (mise-à-jour)	89
5.2	Opérateurs dissymétriques et théorie des possibilités	89

5.2.1	Les règles de mise-à-jour	89
5.2.2	Les règles de conditionnement	90
5.2.2.1	L'expansion	91
5.2.2.2	La contraction	91
5.2.2.3	La révision "classique"	92
5.2.2.3.1	La nouvelle information est un ensemble classique	92
5.2.2.3.1.1	$T = (x, y)$	93
5.2.2.3.1.2	$T = x.y$	94
5.2.2.3.2	La nouvelle information est un ensemble flou	96
5.3	Un exemple	97
5.4	Conclusion	100
6	DFOL : A Data Fusion Object Library	103
6.1	Le cahier des charges	104
6.1.1	La représentation des distributions	105
6.1.1.1	Réduction du nombre de points	106
6.1.1.2	Elimination des paliers	107
6.1.2	La représentation de segments et d'intervalles	107
6.1.2.1	Les segments	107
6.1.2.2	Les intervalles	108
6.1.2.2.1	L'union.	109
6.1.2.2.2	L'intersection.	109
6.1.3	La fusion de données	109
6.1.3.1	Fusion de deux sources	110
6.1.3.2	Fusion de plus de deux sources	110
6.1.3.3	La fusion dissymétrique	110
6.2	L'architecture de DFOL	111
6.2.1	Introduction à la programmation avec DFOL	112
6.2.2	Les algorithmes de gestion d'intervalles	114
6.2.3	Les indices de conflit	114
6.2.4	Les distributions	116
6.2.5	Les algorithmes de fusion	116
6.3	Exemples	118
6.3.1	Les distributions, segments et intervalles	118
6.3.2	La fusion	119
6.4	Conclusion	120
7	L'évaluation des systèmes de fusion	121
7.1	Quantifier l'amélioration de la connaissance	122
7.1.1	Le gain en fusion symétrique	122
7.1.2	Le gain en fusion dissymétrique	123
7.1.3	Quelques techniques de mesure de la qualité de l'information	123
7.2	Evaluation des systèmes de fusion symétrique	126
7.2.1	Evaluation d'opérateurs conjonctifs	126
7.2.2	Evaluation d'opérateurs disjonctifs	127
7.2.3	Evaluation d'opérateurs de compromis	127
7.2.4	Evaluation d'opérateurs adaptatifs	127
7.2.5	Exemple	128
7.3	Evaluation des systèmes de fusion dissymétrique	129
7.3.1	La contraction	129
7.3.2	L'expansion	130
7.3.3	La révision "classique"	130

7.3.4	Exemple	130
7.4	Quel opérateur choisir?	131
7.4.1	Les applications-type retenues	131
7.4.2	Les applications exploitant la redondance	132
7.4.3	Les applications affinant une information existante	133
7.5	Quelques exemples de fusion adaptative plus complexes	133
7.6	Conclusion	135
	Conclusion	137

Table des figures

1.1	Disjonction.	9
1.2	Compromis.	9
1.3	Conjonction.	9
1.4	Trois exemples de distributions en disjonction, compromis et conjonction.	9
1.5	Les ensembles de test.	21
1.6	Le processus de calcul.	21
1.7	$\top = (\omega_1, \omega_2)$	21
1.8	$\perp = (\omega_1, \omega_2)$	21
1.9	$\top = (0, \omega_1 + \omega_2 - 1)$	22
1.10	$\top = \omega_1 \omega_2$	22
1.11	$m = \sqrt{\omega_1 \omega_2}$	22
1.12	$m = \frac{\omega_1 + \omega_2}{2}$	22
1.13	$\sigma_{min} = \frac{\omega_1 \omega_2}{1 - \omega_1 - \omega_2 + 2\omega_1 \omega_2}$	22
1.14	$\sigma_{max} = \frac{\omega_1 + \omega_2 - \omega_1 \omega_2}{1 + \omega_1 + \omega_2 - 2\omega_1 \omega_2}$	22
1.15	Conjonction	23
1.16	Compromis	23
1.17	Disjonction	23
2.1	Représentation graphique de s, t, u, v pour E classique.	26
2.2	Représentation graphique de s, t, u, v pour E flou.	30
2.3	Calcul de la similarité sur deux nombres flous.	31
2.4	Distributions pour lesquelles h est mal adapté.	33
2.5	Représentation graphique de H_Z	35
2.6	Distributions de test n° 1.	37
2.7	Distributions de test n° 2.	37
2.8	Indices de similarité (n° 1).	38
2.9	Indices de similarité (n° 2).	38
2.10	Nouveaux indicateurs (n° 1).	38
2.11	Nouveaux indicateurs (n° 2).	38
2.12	Indice flou (n° 1).	39
2.13	Indice flou (n° 2).	39
2.14	De la disjonction à la conjonction pour π_R	41
2.15	Fusion saturée ($s = 0.2$).	42
2.16	Fusion saturée ($s = 0.8$).	42
2.17	Fusion linéaire (avec (2.36)).	43
2.18	Fusion linéaire (avec (2.37)).	43
2.19	De la disjonction à la conjonction pour π_Z	45
2.20	De la disjonction à la conjonction pour π_Z^{Lin}	46
2.21	De la disjonction à la conjonction pour π_{UC}	47
3.1	Exemple de fusion en "cascade"	50
3.2	Distributions de test.	51

3.3	Fusion en “cascade” avec (ω_2, ω_1) .	51
3.4	Fusion en “cascade” avec $\frac{\omega_1 + \omega_2}{2}$.	51
3.5	Fusion en “cascade” avec $\frac{\sqrt{\omega_2 + \omega_1}}{2}$.	51
3.6	Calcul du conflit entre deux ensembles de sources.	56
3.7	La partition atomique.	57
3.8	Calcul du critère de classement.	57
3.9	Avant combinaison de C_1 et de C_{max} .	57
3.10	Après combinaison de C_1 et de C_{max} .	57
3.11	Algorithme de classification.	58
3.12	Les sources à fusionner.	61
3.13	Le graphe des conflits.	61
3.14	Les classes formées.	61
3.15	Résultat de la fusion intra-classes.	61
3.16	Résultat final.	62
4.1	L’algorithme du principe d’extension – f monodimensionnelle.	72
4.2	L’algorithme du principe d’extension – f multidimensionnelle et $\pi_{X_1}, \dots, \pi_{X_n}$ interactives.	72
4.3	L’algorithme du principe d’extension – f multidimensionnelle et $\pi_{X_1}, \dots, \pi_{X_n}$ non interactives.	73
4.4	L’algorithme du principe d’extension – f vectorielle.	73
4.5	La cuve et ses capteurs de température.	74
4.6	Moyenne arithmétique “étendue” des trois nombres flous C_1, C_2 et C_3 .	75
4.7	Moyenne arithmétique “étendue” des trois nombres flous C_1, C_2 et C_3 avec compatibilité.	77
4.8	Fusion de C_1, C_2 et C_3 avec compatibilité et agrégation partielle.	78
4.9	La scène expérimentale.	84
4.10	Le résultat en x .	85
4.11	Le résultat en y .	85
4.12	Les résultats en x et y de la fusion.	85
5.1	Les lois de fusion dissymétriques.	89
5.2	Le principe de la mise-à-jour.	90
5.3	Possibilité conditionnelle $\pi(\cdot B)$ pour B classique ($T=\min$).	94
5.4	Un exemple de discontinuité.	94
5.5	Fonctions d’appartenance des ensembles utilisés lors de la simulation.	95
5.6	Possibilité conditionnelle $\pi(\cdot B)$ pour B classique ($T=\min$).	95
5.7	Possibilité conditionnelle $\pi(\cdot B)$ pour B classique ($T=x.y$).	96
5.8	Les distributions de possibilité des trois obstacles.	98
5.9	Etape 1.	99
5.10	Etape 2 (union simple).	99
5.11	Etape 3.	99
5.12	Etape 4 (union simple).	99
5.13	Etape 5.	100
5.14	Etape 6.	100
5.15	Etape 7 (union simple).	100
5.16	Etape 8.	100
5.17	Etape 9.	101
5.18	Etape 10.	101
6.1	Représentation d’une distribution quelconque à l’aide de segments.	106
6.2	Deux possibilités de simplification des distributions quelconques.	106
6.3	Exemples de segments $[0; 3[$ et $]0; 3[$ dans DFOL.	108
6.4	Relation d’ordre entre les segments $]3; 5]$ et $[4; 6]$.	108
6.5	Hiérarchie des classes de gestion d’intervalles dans DFOL.	113

6.6	Hierarchie des classes CIntervalle et CSegment.	114
6.7	Hierarchie (simplifiée) des classes de conflit dans DFOL.	116
7.1	Un système de fusion atomique.	122
7.2	Interprétation de la spécificité (définition n° 2).	124
7.3	Calcul de π_{opt} et π_{pess}	125
7.4	La cuve et ses capteurs.	128
7.5	Distributions de test	128
7.6	Conjonction.	129
7.7	Compromis.	129
7.8	Disjonction.	129
7.9	Fusion des nombres flous.	129
7.10	Evaluation (min).	129
7.11	Evaluation (moy).	129
7.12	Evaluation (max).	129
7.13	Evaluation des trois modes.	129
7.14	L'évaluation.	131
7.15	Distributions de test et résultat de la fusion (ex 1).	134
7.16	Distributions de test et résultat de la fusion (ex 2).	135
7.17	Distributions de test et résultat de la fusion (ex 3).	135
7.18	Distributions de test et résultat de la fusion (ex 4).	135

Liste des tableaux

1.1	Propriétés des opérateurs de fusion.	14
1.2	Quelques T-Normes/T-Conormes classiques.	18
1.3	Quelques exemples de moyennes.	19
1.4	Quelques sommes symétriques.	20
1.5	Le comportement des opérateurs de fusion simples d'après [Bloch, 1996].	24
2.2	Quelques similarités dans le cadre de l'analyse de données.	29
2.3	Les opérateurs dans chacun des trois modes de fusion.	44
2.4	Les poids pour chacun des trois modes de fusion.	44
3.1	Les conditions aux limites	54
4.1	Trois vecteurs OWA remarquables.	65
4.2	Les alternatives retenues.	68
4.3	Fonction d'évaluation de chaque critère.	68
4.4	Les alternatives et leurs notes.	69
4.5	Les alternatives retenues.	69
4.6	Les notes des trois étudiants.	82
4.7	Les résultats des trois étudiants.	82
4.8	La mesure floue retenue.	82
4.9	Les évaluations des trois étudiants par l'intégrale floue.	83
4.10	Les mesures fournies par les trois sources.	83
5.1	Les postulats de la mise-à-jour.	90
5.2	Les postulats de l'expansion.	91
5.3	Les postulats de la contraction.	92
6.1	Les trois manières de fusion (cas multisources) vues par DFOL.	110
6.2	Les trois manières de fusion vues par DFOL.	111
6.3	Les paramètres des modèles dans le cas multisources.	111
6.4	Fichiers d'en-tête pour les segments et les intervalles.	112
6.5	Fichiers d'en-tête pour les distributions.	112
6.6	Fichiers d'en-tête pour les algorithmes de fusion entre deux sources.	114
6.7	Quelques méthodes de la classe CIntervalle.	115
6.8	Méthodes de la classe CSimilarite.	116
6.9	Quelques méthodes de la classe CIntervalle.	117
6.10	Méthodes pour le modèle CFusionTNorme.	118
7.1	Les opérateurs recommandés dans les situations de fusion symétrique classiques.	133
7.2	Les opérateurs recommandés dans les situations de fusion dissymétrique classiques.	134



Introduction

LE contrôle et la commande de tout processus nécessitent la connaissance précise de son état. Cette connaissance est acquise en partie à l'aide de capteurs qui sont imprécis (ils fournissent une vision biaisée d'une mesure) et incertains (ils peuvent tomber en panne à chaque instant). Pour limiter l'effet d'une éventuelle panne, une solution éprouvée consiste à doubler les capteurs et à comparer les mesures obtenues. Lorsque celles-ci sont jugées trop divergentes, il y a erreur sur la mesure. La partie commande doit donc être capable de gérer de multiples sources d'informations hétérogènes (provenant de capteurs de température, de télémètres laser, de caméras . . .), éventuellement redondantes de manière à prendre les décisions les plus adaptées possibles et ce, le plus rapidement possible.

Associer plusieurs données hétérogènes dans le but d'obtenir un jugement plus fondé que celui obtenu en considérant chaque donnée isolément est un processus de "fusion de données". Les données en question proviennent soient de sources multiples et ont été acquises au même instant soit d'une même source mais ont été acquises au cours du temps. En utilisant au mieux les apports de chaque donnée, il est possible de pallier leurs faiblesses. Par ailleurs, il faut toujours garder à l'esprit que pour réaliser une fusion de données avec succès, il faut au moins deux sources, l'une d'entre elles au moins étant proche de "la vérité terrain" (c'est-à-dire que toutes les sources ne sont pas erratiques).

La fusion de données est née dans les années 1970 et a connu un large développement dans les années 1980, surtout aux États-Unis. De nos jours, elle est utilisée partout dans le monde. A l'origine, la fusion de données était surtout appliquée aux problèmes militaires (reconstruction de la position d'unités sur le champ de bataille . . .). De nos jours, elle s'est étendue à de nombreux domaines comme la robotique, le traitement d'images, les transports . . . Des applications classiques consistent à estimer un paramètre ou à mettre à jour une information *a priori* à la lumière d'une nouvelle donnée.

Les données susceptibles d'être combinées se divisent en quatre familles :

- Les mesures. Ce sont les valeurs fournies par les capteurs comme une température, une pression, une image . . . Il s'agit du niveau le plus bas car les données sont brutes,
- Les attributs. Ce sont des propriétés ou caractéristiques extraites d'une ou plusieurs mesures. La couleur d'un objet dans une image est un attribut.
- Les règles. Ce sont des relations entre les attributs, souvent exprimées sous forme d'équations ou en langage naturel. Par exemple "si l'objet est rouge et chaud, il s'agit d'un métal en fusion" ou encore, "si le métal est en fusion alors réduire la température".
- Les décisions. Ce sont les résultats de l'application des règles (ou d'un ensemble de règles) sur les attributs. Par exemple, "réduire la température". Il s'agit du niveau d'information le plus haut.

La fusion peut avoir lieu à chacun de ces niveaux. Lorsque les informations à combiner appartiennent à la même famille la fusion est dite homogène et dans le cas contraire, elle est dite hétérogène. Dans ce document et sauf mention contraire, les "sources" ou "informations" désigneront les éléments à fusionner d'une manière générale, sachant qu'ils peuvent appartenir à chacun des niveaux présentés ci-dessus.

Dans la littérature, les verbes "mélanger" ou "combiner" désignent "toute opération mathématique réalisée sur au moins deux ensembles d'informations". Par la suite, nous utiliserons indifféremment "fusionner" ou "combiner" pour désigner une opération de fusion d'informations.

Pour utiliser conjointement plusieurs informations provenant de sources hétérogènes, il est nécessaire d'exprimer toutes les données dans le même formalisme. Les techniques de fusion dépendent du formalisme choisi pour modéliser les données (probabilités, ensembles flous/distributions de possibilités, bases des connaissances . . .) mais aussi du type d'information manipulé (des mesures, des attributs . . .) et du domaine d'application. Les formalismes les plus connus sont :

- Les méthodes probabilistes comme la règle de BAYES, le maximum de vraisemblance . . . ,
- Le filtrage de KALMAN,
- Les réseaux de neurones,
- Les bases de connaissances,
- Les méthodes fondées sur les mesures de confiance comme la théorie des possibilités, les fonctions de croyance, les logiques non monotones, les mesures floues . . .

Les méthodes probabilistes (voir notamment [Waltz et Llinas, 1990] et [Abidi et Gonzalez, 1992]) sont par exemples utilisées pour estimer un ou plusieurs paramètres lorsque les sources sont redondantes (on dispose par exemple de trois mesures de la même température). Les sources supposées incorrectes sont tout d'abord écartées puis l'estimation est réalisée avec les sources restantes. Les mesures de probabilité vérifient toujours la propriété d'additivité qui – dans l'optique d'une application en intelligence artificielle – est jugée trop restrictive pour approcher fidèlement tous les aspects du raisonnement humain.

Le filtrage de KALMAN est un outil de prédiction de paramètres à partir d'observations bruitées. Les paramètres sont obligatoirement des mesures, donc des informations de bas niveau. Cette technique consiste à estimer la valeur des paramètres à l'instant $t + 1$ en utilisant les connaissances acquises jusqu'à l'instant t . Cette approche nécessite la connaissance d'un modèle mathématique, et la fiabilité des résultats dépend beaucoup de l'exactitude du modèle.

Les réseaux de neurones [Chin, 1994] sont utilisés dans des applications de classification. Il faut d'abord "entraîner" le réseau à la reconnaissance en lui présentant des éléments de référence qu'il sera susceptible de reconnaître plus tard. Une fois le réseau entraîné, la présentation sur ses entrées de données implique la reconnaissance d'une classe. La limitation de cette technique est évidente : le réseau ne peut reconnaître que des classes qu'il a apprises et risque d'être perturbé par la présentation d'un élément qu'il ne connaît pas.

Les bases de connaissances manipulent des données de type règle ou décision. Il s'agit d'un système expert utilisant des techniques d'inférence empruntées à l'intelligence artificielle.

Les méthodes fondées sur les mesures de confiance représentent les notions de croyance, d'incertitude et de vague. Elles permettent de modéliser explicitement "l'ignorance" au lieu de lui donner une valeur *a priori* totalement arbitraire comme cela est fait lorsque l'on adopte l'hypothèse de l'équiprobabilité. Par ailleurs, elles permettent d'intégrer des informations de types différents.

Il est impossible d'étudier dans le cadre d'une thèse tous les formalismes précédents. La théorie des possibilités permet une modélisation conjointe de divers aspects des imperfections qui entachent les informations : imprécision, incertitude et incomplétude. Ainsi, la théorie des possibilités est apte à modéliser des informations hétérogènes sémantiquement et qualitativement. C'est pour cela que ce document s'intéresse uniquement aux techniques de fusion disponibles dans le cadre de la théorie des possibilités. Les données que nous combinerons (qu'il s'agisse de mesures, d'attributs, de règles ou de décisions) seront supposées être représentées par des distributions de possibilités. L'obtention de ces distributions de possibilités est une étape indispensable avant la fusion de données mais n'est pas une technique de fusion à proprement parler. Cette étape ne sera pas abordée et les données seront supposées être déjà exprimées par des distributions de possibilités.

La fusion de données dans le cadre de la théorie des possibilités sert deux causes bien distinctes. Le but de la première est d'utiliser la redondance de sources différentes pour obtenir une information plus fiable. Dans la seconde, une information *a priori* existe et doit être affinée à la lumière d'une nouvelle évidence. Il n'existe donc qu'une source d'information au travers de laquelle les nouvelles informations sont acquises

séquentiellement. La dernière acquisition joue un rôle particulier¹. C'est pourquoi ce document est divisé en sept chapitres. Les quatre premiers s'intéressent à l'exploitation de la redondance dans les sources. Puis, les techniques de révision sont décrites. Le composant logiciel DFOL² "a Data Fusion Object Library" développé pendant toute la durée de ce travail est alors abordé avant de présenter comment il est possible d'évaluer le "gain" introduit par le processus de fusion.

Le premier chapitre est consacré à la présentation des opérateurs de fusion "simples". Leur comportement est constant quelles que soient les données qu'ils doivent combiner. Il existe trois grands types de comportements : la conjonction, la disjonction et le compromis. La conjonction est le mode de combinaison à appliquer lorsque les sources sont en accord, c'est-à-dire qu'elles expriment des informations similaires. Typiquement, ce type de comportement consiste à calculer une intersection entre les informations à combiner. La zone commune où les données s'accordent est donc conservée et les zones où les données divergent sont oubliées. La disjonction doit être utilisée lorsque les sources expriment des points de vues très différents. Techniquement, le résultat est obtenu par union des sources. De ce fait, l'opérateur a un comportement prudent : "si les sources sont en désaccord, il est possible que la bonne information soit dans l'une ou dans l'autre". Enfin, le compromis doit être utilisé pour combiner des sources exprimant des points de vues partiellement contradictoires. Un opérateur de compromis est par exemple la moyenne arithmétique. L'application du mauvais mode de fusion entraîne immédiatement un résultat erroné ou inexploitable car trop imprécis. Ce chapitre présente les propriétés élémentaires d'un opérateur de fusion de manière générale, donne une classification de ces opérateurs simples et décrit précisément le comportement de chacun et les éventuelles propriétés intéressantes. La fin du chapitre attire l'attention sur les limitations de ces opérateurs et notamment sur la difficulté de choisir le bon mode de combinaison automatiquement.

Le deuxième chapitre aborde les opérateurs dit "adaptatifs". Introduits par DUBOIS et PRADE, ils sont plus complexes que les précédents. Ils tentent tout d'abord d'évaluer le degré d'accord entre les sources. Ce degré d'accord, nommé "indice de conflit" peut-être comparé à l'indice de similarité bien connu dans le domaine de la classification. Ce degré de conflit est un réel de $[0, 1]$ tel que, plus les sources s'accordent (elles expriment la même information) plus le degré est proche de 1. Inversement, plus les sources divergent, plus le degré est proche de 0. Une fois ce degré déterminé, l'opérateur est à même de combiner en utilisant le mode adapté. En fait, les opérateurs adaptatifs peuvent passer continuellement du comportement disjonctif au comportement conjonctif en fonction de l'indice de conflit. En première approche, de tels opérateurs semblent plus intéressants. Ils disposent cependant d'inconvénients notamment dans la complexité des calculs permettant d'arriver au résultat et dans le choix de l'opérateur représentant le conflit. Ce chapitre présente les concepts et propriétés des opérateurs adaptatifs, puis il s'intéresse aux techniques de calcul du conflit. Après avoir détaillé de nouvelles méthodes de détermination du conflit, le comportement des opérateurs adaptatifs existants est décrit. Après avoir expliqué pourquoi les opérateurs existants ne peuvent combiner plus de deux sources d'une manière sûre, le chapitre propose alors de nouveaux opérateurs adaptatifs et présente leurs avantages et leurs inconvénients.

Le troisième chapitre s'intéresse à la fusion de plus de deux sources. En général, la littérature décrit les opérateurs de fusion sans préciser leur comportement lorsque le nombre de sources croît. Si le nombre de sources qu'un opérateur peut combiner dans une seule et même opération est pris comme critère de classement, il existe deux grandes familles d'opérateurs. Les opérateurs de la première famille permettent, par leur conception, de fusionner $n > 2$ distributions. C'est par exemple le cas des opérateurs de type "T-Normes" comme le min mais aussi des opérateurs dit "Ordered Weighted Average" (pour lesquels n doit être fixé ce qui n'est pas le cas des T-Normes). Ces opérateurs sont souvent commutatifs (l'ordre dans lequel les distributions sont présentées à l'opérateur n'a pas d'importance) et associatifs, mais ce n'est pas une règle générale. Lorsque les opérateurs ne sont pas associatifs et/ou non commutatifs, l'ordre dans lequel les sources sont présentées devient capital. La seconde famille groupe les opérateurs limités, par conception, à deux sources. Les opérateurs adaptatifs de la famille "DUBOIS et PRADE" en sont un excellent exemple. Pour pouvoir combiner

1. Par exemple, si un robot mobile s'approche d'un obstacle, la dernière acquisition de son capteur frontal est supposée plus importante dans le processus de décision car elle a été obtenue plus près de l'obstacle.

2. Data Fusion Object Library.

plus de deux sources à l'aide de ces opérateurs, il existe grossièrement deux méthodes. La première consiste à définir un ordre de combinaison où les sources et les résultats intermédiaires des fusions sont combinés deux à deux jusqu'à obtenir le résultat. Ce type de fusion est nommé "fusion en cascade" et nécessite le choix automatique de l'ordre. Une autre technique consiste à se ramener à deux sources en formant des groupements de sources cohérentes. Le chapitre présente les deux familles d'opérateurs avant de s'intéresser aux techniques permettant d'ordonner les sources lorsque cela est nécessaire. La réduction à deux sources est alors présentée. Enfin, les procédés de fusion utilisant une combinaison d'opérateurs des deux familles sont étudiés.

Le quatrième chapitre décrit les opérateurs de fusion de type "Ordered Weighted Average" ou OWA. Ces opérateurs utilisent le principe d'extension pour appliquer un opérateur de fusion arithmétique à des sous-ensembles flous et donc des distributions de possibilités. Ils permettent aussi de prendre en compte la fiabilité des sources ce qui est difficile avec les opérateurs décrits jusqu'à présent. Enfin, par leur conception, ces opérateurs peuvent combiner plus de deux sources en une seule et même étape. Le chapitre présente les propriétés de ces opérateurs et analyse leur comportement dans le cas particulier où il faut combiner deux sources. Puis, l'utilisation avec plus de deux sources est suggérée (elle fera l'objet d'une étude approfondie dans le chapitre suivant). De nouveaux opérateurs dérivés des OWA sont alors présentés : il s'agit des intégrales floues avec mesures additives.

Le cinquième chapitre présente les opérateurs de fusion dissymétriques spécialement conçus pour combiner des sources non complémentaires. Dans ce cadre, les opérateurs décrits dans les chapitres précédents ne sont plus utilisables. La fusion dissymétrique doit être employée lorsque les informations à fusionner sont acquises séquentiellement par la même source ou lorsqu'une connaissance *a priori* doit être ajustée à la lumière d'une nouvelle information. Le comportement de ces opérateurs est fortement lié au formalisme utilisé pour représenter les informations mais, en première approche, il est possible de distinguer deux grandes familles de comportement : le conditionnement et la mise-à-jour (ou projection). Le conditionnement s'applique lorsque le monde est statique ; la nouvelle information traduisant alors un nouveau point de vue sur le monde plutôt qu'une remise en question de la connaissance courante. Quant à la projection, elle s'applique aux situations où le monde change. Il s'agit alors d'intégrer dans une connaissance existante une information qui contredit partiellement les croyances précédemment admises. Ce chapitre détaille tout d'abord les concepts associés au conditionnement et à la projection de manière indépendante d'un quelconque formalisme puis s'intéresse plus précisément au cas particulier de la théorie des possibilités avant de présenter la notion de possibilité conditionnelle entre deux distributions de possibilité.

Le sixième chapitre aborde les aspects pratiques de l'implantation d'un système de fusion de distributions de possibilités sur un ordinateur. Les moyens logiciels existant peuvent être grossièrement classés en deux grandes familles : les logiciels intégrés et les bibliothèques logicielles. Les logiciels intégrés comme MATLAB ne disposent pas encore de boîte à outil "Fusion de Données" et la boîte à outil "Logique Floue" ne permet pas la représentation de distributions de possibilités continues³. Les logiciels intégrés sont souvent coûteux, gourmands en mémoire et disposent d'une interface graphique qui ne s'intègre pas toujours facilement sur le ordinateur. Les bibliothèques logicielles sont souvent incomplètes, nécessitent un environnement de développement (compilateur, outils . . .) particulier, sont peu ou pas maintenues et rarement livrées avec le code source. Elles ne sont par conséquent pas facilement personnalisables et n'utilisent pas les techniques de programmation par objets qui facilitent l'usage du composant logiciel et contribuent à augmenter la portabilité. Toutes ces raisons ont conduit à l'écriture – à l'aide d'un langage à objets – d'une bibliothèque de représentation et de fusion de distributions de possibilités portable, compacte et dont le code source est disponible. Ce chapitre présente le cahier des charges de DFOL, décrit les algorithmes cruciaux, commente des exemples concrets et détaille l'extension des fonctionnalités de la bibliothèque.

Le septième chapitre s'intéresse aux techniques de mesure du "gain" obtenu à l'issue d'une opération de fusion. Le but de ce chapitre est de vérifier la rentabilité de l'investissement dans le système de fusion. Tout d'abord, le terme "gain" très général est précisé en fonction du but recherché lors de la fusion : augmenter la spécificité, écarter les sources incohérentes, fournir une valeur estimée d'un paramètre . . . Puis, les techniques

3. Il est cependant possible d'y arriver au prix d'une lourde programmation.

de mesures correspondantes à chacun de ces critères sont abordées. Les résultats conduisent à l'ébauche d'un tableau de "prescription" des opérateurs en fonction de l'application.

La conclusion résume les opérateurs présentés, détaille quelques applications réelles et propose de nouvelles directions de recherche dans le cadre de la théorie des possibilités mais aussi dans d'autres formalismes.

Chapitre 1

Les opérateurs simples

LA vie courante est faite de combinaison de données. Par exemple, un automobiliste va regarder le prix des carburants dans les stations-service proches de son domicile et choisira le revendeur le moins cher. Il s'agit d'un processus de fusion sur des mesures (les prix) dont le but est de déterminer le prix du carburant le plus avantageux parmi toutes les sources d'informations (les revendeurs). La première étape du choix de l'automobiliste consistera à acquérir les données (en regardant les enseignes des stations-service) puis il présentera ces informations dans un même formalisme (la même monnaie). Dans une seconde étape, il faudra choisir l'opérateur de combinaison en fonction de l'application (payer le moins cher par exemple, donc calculer le minimum des prix) et utiliser l'opérateur choisi pour effectuer la fusion. Notre automobiliste a – en quelque sorte – utilisé la redondance des sources d'informations (les revendeurs de carburant) pour prendre une décision.

La combinaison d'informations issues de sources différentes est présentée dans [Dubois et Prade, 1992]. Cet article décrit succinctement certains opérateurs capables d'utiliser la redondance d'information pour fournir une nouvelle information plus fiable et plus robuste. L'exploitation de la redondance entre plusieurs sources d'informations n'est pas toujours possible. Que dire par exemple de la fusion d'informations toutes erronées ? Intuitivement, il semble difficile d'arriver à une information plus fiable uniquement à partir d'informations farfelues. Nous supposons donc dans tout ce document qu'au moins une des sources d'information est correcte sans qu'il soit cependant nécessaire de connaître quelle est la source cohérente parmi toutes les sources à combiner.

Dans un processus de fusion nous recherchons à renforcer la confiance ou la fiabilité en une information présente "presque identiquement" dans au moins deux sources. Supposons maintenant que certaines sources soient liées : le processus de fusion va donc augmenter la confiance dans l'information fournie par les sources liées puisqu'une information similaire est présente dans plusieurs sources. La présence de sources dépendantes va donc entraîner une dérive du résultat alors que des sources liées ne fournissent pas plus d'information. Il n'existe actuellement pas de méthode rapide permettant de déterminer si deux sources sont liées et la suppression des sources liées avant fusion est donc difficilement envisageable. Dans tout ce document, nous supposons que la fusion est réalisée sur des sources indépendantes.

De manière générale, la combinaison d'informations peut se faire dans trois grands modes différents : la conjonction, la disjonction ou le compromis. Le mode conjonctif doit être utilisé lorsque les sources sont majoritairement en accord. L'opérateur conjonctif va alors privilégier les degrés de possibilités communs à toutes les sources. Le mode disjonctif doit être utilisé lorsque les sources sont en désaccord. L'opérateur va adopter une attitude prudente en fournissant l'union de toutes les sources. Le compromis est un mode de combinaison intermédiaire où les sources ne sont pas réellement en désaccord mais pas non plus réellement en accord. Cet état intermédiaire est géré par un opérateur de moyenne, un peu comme lorsqu'un professeur désire évaluer le niveau de ses élèves sur plusieurs épreuves¹.

Les opérateurs auxquels ce chapitre s'intéresse sont dits "simples" car leur comportement est très primitif.

1. Cet exemple permettra aussi au chapitre 4 de montrer que les intégrales floues ne sont pas toujours des opérateurs de moyenne.

Ils réalisent la fusion indépendamment de toute information extérieure autre que les degrés de possibilité à combiner. Ils sont donc incapables de prendre en compte la concordance ou la fiabilité des sources. Par ailleurs, un tel opérateur agit toujours de la même manière sur les informations : un opérateur prévu pour dégager le consensus entre les sources tentera toujours de dégager un consensus même si toutes les sources sont en désaccord. Utilisés dans des applications simples (calculer la moyenne d'un étudiant, déterminer le prix le plus bas d'un carburant, . . .), ces opérateurs servent aussi de briques de constructions que l'on assemble pour créer d'autres opérateurs plus performants mais plus complexes dont l'étude sera faite au chapitre 2.

Ce chapitre se propose de formaliser et d'étudier en détail les opérateurs de fusion de données dits "simples". Nous nous limitons volontairement à la combinaison de deux sources et reportons l'étude de la combinaison de plus de deux sources au chapitre 3. Dans un premier paragraphe, les trois modes de fusion : conjonctif, disjonctif et compromis sont décrits. Le second paragraphe aborde la notion de monde ouvert et de monde fermé. Le troisième paragraphe s'intéresse aux propriétés des opérateurs de fusion dans le cadre de la théorie des possibilités. Les opérateurs simples sont alors divisés en deux catégories : CICB (Context Independent Constant Behavior ou "opérateurs indépendants du contexte à comportement constant") et CIVB (Context Independent Variable Behavior ou "opérateurs indépendants du contexte à comportement variable"). De nombreux opérateurs classiques de ces deux catégories sont alors présentés en détaillant leurs propriétés, avantages, inconvénients et comportement de la disjonction à la conjonction. Enfin, la conclusion précise quelques applications des opérateurs "simples" et aborde les problèmes existants lorsque le mode de combinaison est mal choisi.

Par la suite, Ω désignera un univers dont $\omega, \omega_1, \omega_2$ et ω_3 sont quatre éléments. π, π_1, π_2, π_3 désignent quatre distributions de possibilité sur Ω . \top et \perp désignent respectivement une T-Norme et une T-conorme. Enfin, $\Pi(A)$ et $N(A)$ désignent respectivement les mesures de possibilité et de nécessité de la partie A de Ω . La notation $\bar{\omega}$ désignera le complémentaire de ω .

1.1 Les trois comportements de base : conjonction, compromis et disjonction

L'envie d'exploiter la redondance entre diverses sources non liées suggère qu'il ne peut exister qu'un seul mode de combinaison qui satisferait tous les cas de figure, toutes les applications. Il est possible de différencier plusieurs modes d'agrégation [Dubois et Prade, 1992] : conjonctif, disjonctif et compromis. Les deux premiers modes d'agrégation sont dits "logiques" car leur action est une opération d'intersection ou d'union au sens flou entre plusieurs ensembles. Nous détaillons chacun de ces modes dans les paragraphes suivants.

1.1.1 La conjonction

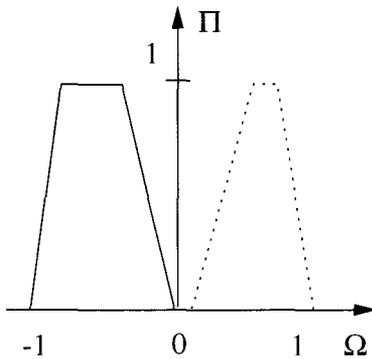
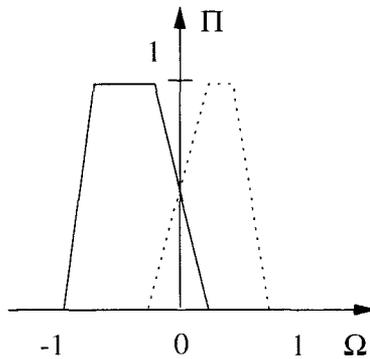
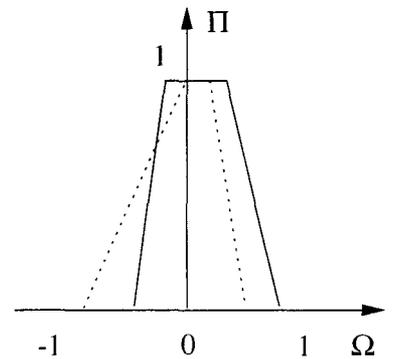
Ce mode d'agrégation logique est réservé à la combinaison de sources fiables ou concordantes qui expriment donc des informations presque identiques. Un exemple de distributions de possibilités en conjonction est présenté figure 1.3 page ci-contre. Un opérateur conjonctif va donc renforcer la confiance en un événement commun à toutes les sources et fortement réduire (voire annuler) la confiance en d'autres événements. Il s'agit de T-Normes (voir paragraphe 1.4.2.1 page 16) qui sont des opérateurs d'intersection d'ensembles flous. Le plus connu de ces opérateurs est le minimum : $\top = \min(\pi_1(\omega), \pi_2(\omega))$. Le résultat de la fusion des distributions présentées figure 1.3 page ci-contre serait l'intersection des deux distributions de possibilités.

1.1.2 Le compromis

Le compromis est utile lorsque les sources ne sont ni en discordance forte ni en concordance. Typiquement, chacune des sources est sensée et il faut extraire "la moyenne" de toutes ces informations et faire un compromis. C'est exactement ce qui est fait par un professeur désireux d'évaluer ses élèves. Ce procédé de combinaison est plus arithmétique que logique. Il s'agit d'utiliser un opérateur de moyenne (voir le paragraphe 1.4.2.2 page 17). L'exemple type d'un tel opérateur est $\frac{\pi_1(\omega) + \pi_2(\omega)}{2}$.

1.1.3 La disjonction

Supposons que quelques sources fiables apparaissent dans un groupe de sources moins fiables. En l'absence de plus de précision, une solution valide (et prudente) consiste à supposer que l'information réelle est située dans l'une des sources, sans préciser laquelle. Ce comportement est typique d'un opérateur d'union. L'opérateur typique de la disjonction est la T-Conorme (voir paragraphe 1.4.2.1) $\perp = \max(\pi_1(\omega), \pi_2(\omega))$ qui est un opérateur d'union au sens flou. Avec un tel procédé, il existe un risque de saturation : si le nombre de sources en désaccord augmente, le résultat de l'union finira par être proche de tout l'univers Ω (c'est-à-dire une distribution pour laquelle $\pi(\omega) = 1$ pour tout ω) et devenir inutilisable. La figure 1.1 présente un exemple de sources en disjonction.

FIG. 1.1 – *Disjonction.*FIG. 1.2 – *Compromis.*FIG. 1.3 – *Conjonction.*FIG. 1.4 – *Trois exemples de distributions en disjonction, compromis et conjonction.*

D'autres modes de combinaison sont envisageables. Il est possible d'effectuer une combinaison pondérée entre les sources, d'effectuer une combinaison contenant un pourcentage de conjonction, de disjonction, de compromis, ... Ces autres modes de combinaison plus complexes seront abordés dans les chapitres suivants. Cependant, face à un ensemble d'informations à fusionner, il sera nécessaire de déterminer le mode approprié en fonction de la concordance des sources avant de choisir l'opérateur. Toute erreur dans le choix du mode conduira à un résultat incohérent².

Il est intéressant de noter que le résultat fourni par les opérateurs min et max cités précédemment ne sont pas des distributions de possibilités et par conséquent l'opération de fusion qu'ils réalisent n'est pas une opération interne. Le paragraphe suivant s'intéresse à ce problème connu sous le nom de "hypothèse du monde fermé".

1.2 Hypothèse du monde ouvert ou du monde fermé

Soit π une distribution de possibilité [Dubois et Prade, 1988]. Par définition, il existe un élément ω de Ω tel que $\pi(\omega) = 1$. Cette contrainte connue sous le nom de "hypothèse du monde fermé" indique que Ω contient tous les événements possibles. Par définition $\Pi(\Omega) = N(\Omega) = 1$. Ainsi, nous sommes sûrs qu'un et un seul des événements de Ω va se produire, sans pouvoir préciser lequel à priori.

Supposons que Ω ne contienne pas tous les événements possibles. π serait donc telle que :

$$\max_{\omega \in \Omega} \pi(\omega) < 1 \Rightarrow \Pi(\Omega) < 1 \quad (1.1)$$

Un événement hors de Ω (représenté par l'ensemble E_I) pourrait se produire. Cet événement est inconnu car sinon il serait possible de l'inclure dans Ω . Puisque E_I n'est pas dans Ω , il est impossible de calculer distinctement sa possibilité et sa nécessité. Cependant, nous pouvons calculer la possibilité qu'un événement hors de Ω se produise.

2. Agréger des informations disjonctives avec un opérateur conjonctif conduira à π_N telle que $\forall \omega \in \Omega, \pi_N(\omega) = 0 \dots$

Le seul événement qui ne se produit jamais par rapport à Ω est l'ensemble vide. Dans ce contexte, calculer la possibilité de l'ensemble vide revient donc à calculer la possibilité qu'un événement hors de Ω se produise. Par rapport à Ω , l'ensemble vide est un événement qui va se produire, mais qui n'existe pas. Il y a contradiction. L'hypothèse du monde ouvert permet donc de détecter (et de quantifier) les inconsistances dans la définition de Ω ou dans les données.

Un exemple permet de mieux comprendre l'hypothèse du monde ouvert. Une accident de la circulation est arrivé et la voiture fautive a pris la fuite. Les deux seuls témoins sont invités à choisir la couleur de la voiture fautive parmi un ensemble de couleurs $C = \{\text{rouge, bleu, vert, blanc, noir}\}$. L'un affirme "bleu" tandis que l'autre soutient "rouge". Les sources sont en complète contradiction et dans le cas du monde fermé, la fusion de ces deux informations conduirait à penser que la voiture est "rouge ou bleue". Cependant, les deux témoins sont peut-être daltoniens et la voiture tout simplement violette. Dans ce cas et en faisant l'hypothèse du monde ouvert, la fusion dans l'ensemble $C_2 = C \cup \{\text{violet}\}$ est alors naturelle, les sources étant alors en concordance totale.

Dans les paragraphes précédents, nous avons remarqué que la fusion du couple (π_1, π_2) peut conduire à une nouvelle distribution π_R ne vérifiant pas la contrainte imposée par l'hypothèse du monde fermé. Il apparaît donc nécessaire de mettre en œuvre un processus permettant de transformer π_R en une distribution de possibilité vérifiant l'hypothèse du monde fermé. Ce processus est appelé "normalisation".

Une autre approche consiste à abandonner l'hypothèse du monde fermé et supposer que l'univers Ω ne contient pas tous les événements qui peuvent se produire. Cette hypothèse s'appelle "l'hypothèse du monde ouvert". Dans ce contexte, la fusion de sources concordantes est faite à l'aide de procédés décrits ici mais la combinaison de sources en total désaccord est effectuée différemment. Deux sources en fort désaccord peuvent indiquer une incomplétude dans le référentiel Ω . Il semble alors intéressant d'essayer de compléter le référentiel avant de réaliser la fusion. Dans l'hypothèse du monde ouvert, le référentiel est potentiellement en changement constant.

L'hypothèse du monde ouvert est une autre manière d'interpréter le conflit. Dans ce cadre, fusionner deux sources en fort désaccord signifie que le référentiel Ω est incomplet et qu'il faut essayer de le compléter avant de fusionner. Peu d'auteurs se placent dans l'hypothèse du monde ouvert ce qui nous amène à supposer dans ce document que l'hypothèse du monde fermé toujours vérifiée.

1.3 Les propriétés des opérateurs de fusion

Les opérateurs de fusion sont des applications de $\Omega \times \Omega$ vers un ensemble K . Ces applications peuvent posséder certaines propriétés décrites dans ce paragraphe. Pour référence, les propriétés sont résumées succinctement dans le tableau 1.1 page 14. Nous supposons que $\pi_1, \pi_2, \pi_3, \pi_4$ sont quatre distributions de possibilités quelconques d'un univers Ω et Φ un opérateur de combinaison.

Propriété 1. *Fermeture (ou loi de composition interne).*

Un opérateur Φ est un opérateur de composition interne si combiner deux éléments de Ω donne un nouvel élément de Ω . En d'autres termes, la combinaison de deux distributions de possibilités par Φ donne une nouvelle distribution de possibilité.

Dans le cadre des distributions de possibilités, la plupart des opérateurs ne sont pas des lois de composition interne. Une étape de normalisation doit être ajoutée après l'étape de fusion, sauf bien entendu si l'on autorise les distributions non normalisées en faisant l'hypothèse du monde ouvert.

Propriété 2. *Commutativité.*

Il y a commutativité si quels que soient ω_1, ω_2 , $\Phi(\omega_1, \omega_2) = \Phi(\omega_2, \omega_1)$. Un opérateur commutatif est insensible à l'ordre dans lequel les sources lui sont fournies. Il s'agit d'une propriété très importante car il est en général difficile de choisir automatiquement le bon ordre d'agrégation. L'opérateur $\top = \min$ (voir tableau 1.4.2.2 page 18) est un exemple d'opérateur commutatif.

Propriété 3. Associativité.

Il y a associativité si quels que soient $\omega_1, \omega_2, \omega_3$, $\Phi(\Phi(\omega_1, \omega_2), \omega_3) = \Phi(\omega_1, \Phi(\omega_2, \omega_3))$. Cette propriété prend tout son intérêt lors de la combinaison de plus de deux sources³ à l'aide d'un opérateur prévu uniquement pour fusionner deux sources. Pour un opérateur associatif, le choix des couples intermédiaires formés n'a aucune importance tant que l'ordre des sources est inchangé. Lorsqu'un opérateur est à la fois commutatif et associatif, l'ordre dans lequel les sources sont combinées n'a aucune importance. L'opérateur $\top = \min$ (voir tableau 1.4.2.2 page 18) est un exemple d'opérateur associatif.

Propriété 4. Quasi-associativité.

Il y a quasi-associativité si quels que soient $(\omega_1, \omega_2, \dots, \omega_n)$ des éléments de Ω^n , il existe une fonction f inversible et une loi de composition associative \otimes :

$$f : \Omega \rightarrow \Omega \qquad \otimes : \Omega \times \Omega \rightarrow \Omega \qquad (1.2)$$

$$\omega \mapsto f(\omega) \qquad (\omega_1, \omega_2) \mapsto \omega_1 \otimes \omega_2 \qquad (1.3)$$

$$\text{telles que } \Phi(\omega_1, \omega_2, \dots, \omega_n) = f(\omega_1 \otimes \omega_2 \otimes \dots \otimes \omega_n) = \omega_R$$

La quasi-associativité permet – tout comme l'associativité – d'étendre à plus de deux sources un opérateur prévu uniquement pour combiner deux sources. Avec l'associativité, le choix des paires intermédiaires était possible ; avec la quasi-associativité, ce choix est perdu.

Supposons maintenant qu'il faille combiner une distribution supplémentaire π_{n+1} . La loi \otimes étant associative et la fonction f inversible, nous avons :

$$\begin{aligned} \Phi(\omega_1, \omega_2, \dots, \omega_n, \omega_{n+1}) &= f(\omega_1 \otimes \omega_2 \otimes \dots \otimes \omega_n \otimes \omega_{n+1}) \\ &= f((\omega_1 \otimes \omega_2 \otimes \dots \otimes \omega_n) \otimes \omega_{n+1}) \\ &= f(f^{-1}(\omega_R) \otimes \omega_{n+1}) \end{aligned} \qquad (1.4)$$

Il est possible d'utiliser le résultat ω_R obtenu avec les n éléments pour fusionner un élément supplémentaire ω_{n+1} . Cette propriété de la quasi-associativité est particulièrement intéressante pour réduire le temps de calcul.

Si Φ est un opérateur associatif, il est aussi quasi-associatif. Pour le montrer, il suffit de poser $\otimes = \Phi$ et $f(\omega) = \omega$. L'opérateur $\top = \min$ (voir tableau 1.4.2.2 page 18) est un exemple d'opérateur quasi-associatif.

Voici deux exemples d'opérateurs quasi-associatif mais pas associatifs. Dans cet exemple, $\Omega = \mathbb{R}$:

$$\begin{aligned} f(u) &= u^3 \quad \text{et} \quad \otimes = . & f(u) &= 2u \quad \text{et} \quad \otimes = + \\ \Phi(\omega_1, \omega_2) &= (\omega_1 \omega_2)^3 & \Phi(\omega_1, \omega_2) &= 2(\omega_1 + \omega_2) \\ \Phi(\omega_1, \Phi(\omega_2, \omega_3)) &= \omega_1^3 \omega_2^9 \omega_3^9 & \Phi(\omega_1, \Phi(\omega_2, \omega_3)) &= 2\omega_1 + 4\omega_2 + 4\omega_3 \\ \Phi(\Phi(\omega_1, \omega_2), \omega_3) &= \omega_1^9 \omega_2^9 \omega_3^3 & \Phi(\Phi(\omega_1, \omega_2), \omega_3) &= 4\omega_1 + 4\omega_2 + 2\omega_3 \end{aligned} \qquad \text{ou} \qquad (1.5)$$

Propriété 5. Auto-dualité.

Il y a auto-dualité si quels que soient ω_1, ω_2 , $\Phi(\omega_1, \omega_2) = \overline{\Phi(\overline{\omega_1}, \overline{\omega_2})}$. Le résultat de la combinaison reste inchangé lorsque l'on fusionne deux informations ou lorsque l'on considère le complémentaire de la fusion des complémentaires. Le résultat est inchangé lorsque l'on raisonne sous l'angle de la préférence ou de la désapprobation.

Propriété 6. Idempotence.

Il y a idempotence si quel que soit ω , $\Phi(\omega, \omega) = \omega$. La combinaison de deux éléments identiques conduit au même élément. Les opérateurs $\top = \min$ (voir 1.4.2.2 page 18) et $\perp = \max$ (voir 1.4.2.2 page 18) sont idempotents.

3. Combinaison que nous étudierons en détail dans le chapitre 3 page 49.

Propriété 7. Archimédien.

Un opérateur est archimédien si quel que soit ω , $\Phi(\omega, \omega) > \omega$ ou $\Phi(\omega, \omega) < \omega$. La combinaison de deux éléments identiques conduit à un résultat différent. La confiance sur le résultat est alors renforcée ou affaiblie. L'opérateur $\top = \cdot$ (voir 1.4.2.2 page 18) est archimédien et affaiblit la confiance tandis que la T-Conorme $\perp = \omega_1 + \omega_2 - \omega_1\omega_2y$ (voir tableau 1.4.2.2 page 18) renforce la confiance.

Propriété 8. Nilpotence.

Φ est dit nilpotent lorsqu'il existe un entier k et un élément e neutre pour Φ tels que $\Phi(\omega_1, \omega_2, \dots, \omega_k) = e$. L'accumulation d'informations fait tendre le résultat de la fusion vers l'élément neutre, c'est-à-dire l'ignorance totale. L'opérateur muni de cette propriété est victime de saturation : au-delà d'une certaine limite, l'ajout de nouvelles évidences ne change plus le résultat. En général, cette propriété est plutôt gênante. L'opérateur $\top = \max(0, \omega_1 + \omega_2 - 1)$ (voir 1.4.2.2 page 18) est nilpotent.

Propriété 9. Continuité.

Il existe de nombreuses définitions de la continuité pour les fonctions de plusieurs variables. Nous reproduisons ici la plus simple d'entre elles. Soit $M = (\omega_A, \omega_B)$, l'opérateur Φ sera dit continu au point $M_0 = (\omega_1, \omega_2)$ lorsque :

$$\lim_{M \rightarrow M_0} \Phi(M) = \Phi(M_0) \quad (1.6)$$

Par ailleurs, Φ tend vers $L = (\omega_A, \omega_B)$ en $M_0 = (\omega_1, \omega_2)$ si, quel que soit ϵ réel fixe et d une distance entre deux distributions de possibilités, il est possible de trouver un réel fixe l tel que :

$$d(M, M_0) < l \Rightarrow d(\Phi(M), L) < \epsilon \quad (1.7)$$

Notons qu'une fonction continue en M_0 par rapport à chacune des variables n'est pas nécessairement continue.

Cette propriété, encore appelée robustesse permet d'assurer qu'une faible variation des entrées ne conduit pas à un résultat foncièrement différent (le choix de la distance influe sur le résultat). La continuité est une propriété souhaitable pour un opérateur de fusion. En effet, il devient alors possible d'appliquer bon nombre de théorèmes sur les fonctions continues comme le théorème des valeurs intermédiaires.

Propriété 10. Monotonie et stricte monotonie.

Un opérateur vérifiant l'une au moins des conditions suivantes sera dit monotone :

$$\omega_1 \geq \omega_2 \text{ et } \omega_3 \geq \omega_4 \Rightarrow \Phi(\omega_1, \omega_3) \geq \Phi(\omega_2, \omega_4) \quad (1.8)$$

$$\omega_1 \leq \omega_2 \text{ et } \omega_3 \leq \omega_4 \Rightarrow \Phi(\omega_1, \omega_3) \leq \Phi(\omega_2, \omega_4) \quad (1.9)$$

$$\omega_1 > \omega_2 \text{ et } \omega_3 > \omega_4 \Rightarrow \Phi(\omega_1, \omega_3) > \Phi(\omega_2, \omega_4) \quad (1.10)$$

$$\omega_1 < \omega_2 \text{ et } \omega_3 < \omega_4 \Rightarrow \Phi(\omega_1, \omega_3) < \Phi(\omega_2, \omega_4) \quad (1.11)$$

Cette définition fait appel à la notion d'ordre sur les distributions de possibilités. En raison du principe possibiliste, nous avons⁴ :

$$\omega_1 < \omega_2 \iff \forall \omega \in \Omega, \omega_1(\omega) < \omega_2(\omega) \quad (1.12)$$

Les deux premières conditions dénotent une monotonie croissante (condition (1.8)) et strictement croissante (condition (1.9)). Les deux dernières dénotent quant à elles une monotonie décroissante (condition (1.10)) et strictement décroissante (condition (1.11)). Cette propriété est parfois souhaitable lorsque la fusion opère sur des mesures issues de capteurs.

Propriété 11. Homogénéité.

Un opérateur est homogène si Φ est une fonction homogène, continue et dérivable au premier ordre. Cette propriété est utile lors de la conception d'opérateurs.

4. Les autres relations s'obtiennent en remplaçant $<$ par \geq , $>$ ou \leq suivant la relation désirée.

Propriété 12. *Non contradiction.*

Il y a non contradiction lorsque quel que soit ω , $\Phi(\omega, \bar{\omega}) = 0$. La combinaison de deux parties contraires avec un opérateur non contradictoire conduit à l'événement impossible. Cela peut être le cas pour les opérateurs traduisant l'intersection floue comme l'opérateur $\top = \max(0, \omega_1 + \omega_2 - 1)$ (voir 1.4.2.2 page 18) qui est non contradictoire. La norme $\top = \min$ (voir 1.4.2.2 page 18) ne satisfait pas la non-contradiction.

Propriété 13. *Tiers exclu.*

Il s'agit du dual de la propriété (12): la combinaison de deux événements contradictoires conduit à l'événement complètement possible. Quel que soit ω , l'opérateur vérifie la loi du tiers exclu si $\Phi(\omega, \bar{\omega}) = 1$. Les opérateurs d'union floue vérifient souvent la loi du tiers exclu. La T-conorme $\perp = \min(1, \omega_1 + \omega_2)$ (voir 1.4.2.2 page 18) satisfait la loi du tiers exclu.

La non-contradiction (12) et la loi du tiers exclu sont mutuellement exclusives avec l'idempotence (6). Tous les opérateurs nilpotents satisfont soit la non-contradiction (12), soit la loi du tiers exclu.

Propriété 14. *Préservation du zéro.*

Il y a préservation du zéro lorsque l'opérateur Φ vérifie $\Phi(0, 0) = 0$. Si un événement de Ω est considéré comme impossible par toutes les sources, alors, il est aussi considéré comme impossible dans le résultat de la combinaison par un opérateur préservant le zéro.

Une condition plus forte consiste à décréter l'événement impossible s'il est considéré comme impossible par au moins une source.

Propriété 15. *Préservation de la plausibilité.*

L'opérateur Φ préserve la plausibilité si $\Phi(1, 1) = 1$. Cette propriété est duale de la propriété de préservation du zéro (14). Elle exprime que si un événement de Ω est considéré comme possible par toutes les sources, alors il est aussi considéré comme possible dans le résultat.

Une condition plus forte consiste à décréter l'événement possible si et seulement s'il est considéré comme possible par au moins une source.

Propriété 16. *Conjonctivité.*

Quels que soient ω_1, ω_2 , l'opérateur est conjonctif si $\Phi(\omega_1, \omega_2) \leq \min(\omega_1, \omega_2)$. L'opérateur sera dit conjonctif s'il fournit des résultats toujours inférieurs ou égaux à l'intersection des parties au sens flou. Un opérateur de ce type favorise donc la satisfaction simultanée d'un ou plusieurs événements. Elle réduit l'information la moins certaine et une plus grande importance est accordée aux valeurs les plus faibles.

Propriété 17. *Disjonctivité.*

Quels que soient ω_1, ω_2 , l'opérateur est disjonctif si $\Phi(\omega_1, \omega_2) \geq \max(\omega_1, \omega_2)$. L'opérateur sera dit disjonctif s'il fournit des résultats toujours plus grands ou égaux à l'union des parties au sens flou. Un opérateur de ce type augmente la certitude que l'on a sur l'information et accorde toujours une plus grande importance à l'entité la plus certaine, c'est-à-dire aux valeurs les plus élevées.

Propriété 18. *Compromis.*

Quels que soient ω_1, ω_2 , Φ est un opérateur de compromis si $\min(\omega_1, \omega_2) \leq \Phi(\omega_1, \omega_2) \leq \max(\omega_1, \omega_2)$. Un opérateur de compromis est intermédiaire à l'intersection et l'union des parties. Il a un comportement de moyenne.

Nous avons décrit la plupart des propriétés des opérateurs de fusion hormis quelques unes particulièrement spécifiques que seuls un ou deux opérateurs vérifient. Nous aborderons ces curiosités lors de la description des opérateurs concernés.

Il est possible de grouper les opérateurs en familles d'éléments ayant le même comportement. En pratique, les éléments d'une même famille disposent, en général, des mêmes propriétés. C'est cette décomposition que nous abordons dans le paragraphe suivant.

Propriété	Définition
Commutativité	$\forall (\omega_1, \omega_2) \in \Omega^2, \Phi(\omega_1, \omega_2) = \Phi(\omega_2, \omega_1)$
Associativité	$\forall (\omega_1, \omega_2, \omega_3) \in \Omega^3, \Phi(\Phi(\omega_1, \omega_2), \omega_3) = \Phi(\omega_1, \Phi(\omega_2, \omega_3))$
Quasi-associativité	Il existe \otimes loi associative, f fonction inversible telles que $\forall (\omega_1, \omega_2) \in \Omega^2, \phi(\omega_1, \omega_2) = f(\omega_1 \otimes \omega_2)$
Auto-dualité	$\forall (\omega_1, \omega_2) \in \Omega^2, \Phi(\omega_1, \omega_2) = \Phi(\overline{\omega_1}, \overline{\omega_2})$
Idempotence	$\forall \omega_1 \in \Omega, \omega_1 = \omega_2 = \omega \Rightarrow \Phi(\omega_1, \omega_2) = \omega$
Archimédien	$\forall \omega_1 \in \Omega, \omega_1 = \omega_2 = \omega \Rightarrow \Phi(\omega_1, \omega_2) > \omega$ ou ou $\forall \omega_1 \in \Omega, \omega_1 = \omega_2 = \omega \Rightarrow \Phi(\omega_1, \omega_2) < \omega$
Nilpotence	Il existe un entier k tel que $\Phi(\omega_1, \dots, \omega_k) = e$ (élément neutre de Φ)
Continuité (robustesse) en $M_0 = (\omega_1, \omega_2)$	$M = (\omega_A, \omega_B), \lim_{M \rightarrow M_0} \Phi(M) = \Phi(M_0)$
Monotonie	$\forall (\omega_1, \omega_2, \omega_3, \omega_4) \in \Omega^4, \omega_1 \geq \omega_2$ et $\omega_3 \geq \omega_4$ entraîne $\Phi(\omega_1, \omega_3) \geq \Phi(\omega_2, \omega_4)$
Stricte monotonie	$\forall (\omega_1, \omega_2, \omega_3, \omega_4) \in \Omega^4, \omega_1 > \omega_2$ et $\omega_3 > \omega_4$ entraîne $\Phi(\omega_1, \omega_3) > \Phi(\omega_2, \omega_4)$
Homogénéité	Φ est continue et différentiable (au moins à l'ordre 1)
Non contradiction	$\forall \omega_1 \in \Omega, \Phi(\omega_1, \overline{\omega_1}) = 0$
Loi du tiers-exclu	$\forall \omega_1 \in \Omega, \Phi(\omega_1, \overline{\omega_1}) = 1$
Préservation de zéro	$\Phi(0, 0) = 0$
Préservation de la plausibilité maximale	$\Phi(1, 1) = 1$
Conjonctivité	$\forall (\omega_1, \omega_2) \in \Omega^2, \Phi(\omega_1, \omega_2) \leq \min(\omega_1, \omega_2)$
Disjonctivité	$\forall (\omega_1, \omega_2) \in \Omega^2, \Phi(\omega_1, \omega_2) \geq \max(\omega_1, \omega_2)$
Compromis	$\forall (\omega_1, \omega_2) \in \Omega^2, \min(\omega_1, \omega_2) \leq \Phi(\omega_1, \omega_2) \leq \max(\omega_1, \omega_2)$

TAB. 1.1 – Propriétés des opérateurs de fusion.

1.4 Les différents types d'opérateurs

Les opérateurs disponibles peuvent être classés en trois grandes catégories : CICB (Context Independent Constant Behavior ou “opérateurs indépendants du contexte à comportement constant”), CIVB (Context Independent Variable Behavior ou “opérateurs indépendants du contexte à comportement variable”) et CD (Context Dependand ou “opérateurs dépendants du contexte”). Cette division est fondée sur les informations nécessaires à l'opérateur Φ pour réaliser la combinaison. Soient ω_1 et ω_2 les éléments à combiner, $(\alpha_1, \dots, \alpha_k)$ k éléments pris dans k ensembles T_k et K l'ensemble d'arrivée de Φ . Les deux applications Φ_{CICB}, Φ_{CIVB} représentant respectivement les opérateurs CICB et CIVB sont définies par :

$$\begin{aligned} \Phi_{CICB} : \Omega^2 &\rightarrow K & \Phi_{CIVB} : \Omega^2 &\rightarrow K \\ (\omega_1, \omega_2) &\mapsto \Phi_{CICB}(\omega_1, \omega_2) & (\omega_1, \omega_2) &\mapsto \Phi_{CIVB}(\omega_1, \omega_2) \end{aligned} \quad (1.13)$$

Un opérateur CICB n'a donc besoin d'aucun autre paramètre que les deux éléments ω_1 et ω_2 pour réaliser la combinaison. Cet opérateur a le même comportement quelles que soient les valeurs à combiner. Aucune information contextuelle ou externe (comme la fiabilité d'une source par exemple) ne peut intervenir. Une opérateur CICB est soit conjonctif, soit disjonctif, soit opérateur de compromis. L'opérateur $\top = \min(\omega_1, \omega_2)$ appartient à cette famille.

Un opérateur CIVB utilise lui aussi uniquement les éléments ω_1 et ω_2 mais son comportement dépend des valeurs ω_1 et ω_2 . Bien souvent, ce changement de comportement est réalisé à l'aide de paramètres $(\alpha_1, \dots, \alpha_k)$. Par exemple, l'opérateur suivant est de type CIVB :

$$\Phi_{med_{\alpha_1}} : \begin{cases} \omega_1 & \text{si } \omega_2 < \omega_1 < \alpha_1 \text{ ou si } \alpha_1 < \omega_1 < \omega_2 \\ \omega_2 & \text{si } \omega_1 < \omega_2 < \alpha_1 \text{ ou si } \alpha_1 < \omega_2 < \omega_1 \\ \alpha_1 & \text{sinon} \end{cases} \quad (1.14)$$

Un opérateur CIVB peut donc être disjonctif si les possibilités à combiner sont faibles et conjonctif sur les possibilités à combiner sont fortes, par exemple pour combiner les notes d'un étudiant Aucune information contextuelle ou externe (comme la fiabilité d'une source par exemple) ne peut intervenir.

Les opérateurs CD (qui nous étudierons plus en détail au chapitre 2.3 page 39) sont définis par :

$$\begin{aligned} \Phi_{CD} : \Omega^2 \times T_1 \times T_2 \times \dots \times T_k &\rightarrow K \\ (\omega_1, \omega_2, \alpha_1, \dots, \alpha_k) &\mapsto \Phi_{CD}(\omega_1, \omega_2, \alpha_1, \dots, \alpha_k) \end{aligned} \quad (1.15)$$

Ces opérateurs sont dépendants du contexte. Le résultat ne provient pas uniquement des valeurs de ω_1 et ω_2 mais aussi d'informations supplémentaires extérieures aux degrés à combiner. La fiabilité des sources et le degré de conflit entre les sources (voir 2.1 page 25) en sont quelques exemples.

Il est donc possible de définir des opérateurs conjonctifs lorsque les sources sont concordantes, disjonctifs dans le cas contraire et qui ont le comportement d'une moyenne lorsque les sources s'accordent faiblement. De tels opérateurs dits "adaptatifs" peuvent donc évoluer automatiquement de la conjonction à la disjonction de manière continue.

Etant donnée l'hypothèse du monde fermé faite au paragraphe 1.2, nous poserons $K = \Omega$ ce qui transforme tous les opérateurs en lois de composition internes ce qui implique que le résultat π_R de la fusion doit être une distribution de possibilité, c'est-à-dire qu'elle vérifie $\sup_{\omega \in \Omega} \pi_R(\omega) = 1$. Pour respecter cette contrainte, il sera souvent nécessaire de faire suivre l'étape de fusion par une étape de normalisation que nous allons étudier dans le prochain paragraphe. Les deux paragraphes suivants décriront les opérateurs CICB et les opérateurs CIVB les plus courants. Nous nous limitons volontairement au cas où seulement deux sources doivent être combinées.

1.4.1 La normalisation

Un opérateur de normalisation est une application de $[0, 1] \rightarrow [0, 1]$ qui à tout degré de possibilité $\pi(\omega)$ associe un autre degré de possibilité $No(\pi(\omega))$:

$$\begin{aligned} No : [0, 1] &\rightarrow [0, 1] \\ \pi(\omega) &\mapsto \pi_R(\omega) = No(\pi(\omega)) \end{aligned} \quad \text{telle que} \quad \max_{\omega \in \Omega} \pi_R(\omega) = 1 \quad (1.16)$$

Il existe de nombreux opérateurs de normalisation. Nous présentons ici les deux plus usités. L'opérateur le plus connu est certainement h [Dubois et Prade, 1994a] qui utilise la valeur maximale de la distribution comme facteur d'échelle :

$$\begin{aligned} h : [0, 1] &\rightarrow [0, 1] \\ \pi(\omega) &\mapsto \pi_R(\omega) = \frac{\pi(\omega)}{\max_{x \in \Omega} \pi(x)} \end{aligned} \quad (1.17)$$

Lorsque les sources sont en fort désaccord, $\max_{x \in \Omega} \pi(x) \simeq 0$ et le résultat de la division tend alors vers l'infini ce qui perturbe la distribution de possibilité π_R . En général, effectuer une normalisation à l'aide de cet opérateur conduira à une règle de fusion non continue pour des sources en fort conflit. Un autre opérateur de normalisation dû à YAGER est défini par :

$$\begin{aligned} Y : [0, 1] &\rightarrow [0, 1] \\ \pi(\omega) &\mapsto \pi_R(\omega) = \pi(\omega) + 1 - \max_{x \in \Omega} \pi(x) \end{aligned} \quad (1.18)$$

Parfaitement continu, cet opérateur est beaucoup moins employé que le premier car il est moins informatif. Le support du résultat est la totalité de l'espace Ω ce qui fait perdre la spécificité des distributions normalisées.

1.4.2 Opérateurs indépendant du contexte à comportement constant (CICB)

Dans le formalisme des ensembles flous et dans la théorie des possibilités, les opérateurs CICB combinent les données de manière ensembliste, par exemple en calculant l'union entre les sources. En théorie des ensembles classiques, il n'existe au moins trois manières de combiner des ensembles de façon symétrique : l'intersection, l'union et la différence symétrique. La théorie des sous-ensembles flous offre au moins trois manières de combiner deux ensembles : l'union, l'intersection et la moyenne. Les opérateurs CICB se divisent donc en deux sous-familles : les opérateurs d'intersection/union (dits T-Norme et T-Conorme) et les opérateurs de moyenne. Nous nous intéressons tout d'abord aux opérateurs d'union/intersection avant d'aborder les moyennes.

1.4.2.1 Intersection et union floues ou T-Normes et T-Conormes

Les T-Normes (respectivement T-Conormes) sont des applications de $[0, 1]^2$ vers $[0, 1]$ construites pour définir l'intersection (l'union) entre ensembles flous en conservant au maximum les propriétés de l'intersection et de l'union entre ensembles "classiques". Cependant, les propriétés de l'algèbre de ne peuvent pas toutes être conservées. Les opérations d'intersection et d'union devront vérifier certains axiomes. Dans ce paragraphe \top désigne une T-Norme, \perp une T-Conorme et $\omega_1, \omega_2, \omega_3$ trois éléments de l'ensemble Ω .

Axiome 1. *Commutativité.*

$$\top(\omega_1, \omega_2) = \top(\omega_2, \omega_1) \quad \text{et} \quad \perp(\omega_1, \omega_2) = \perp(\omega_2, \omega_1) \quad (1.19)$$

Axiome 2. *Associativité.*

$$\top(\omega_1, \top(\omega_2, \omega_3)) = \top(\top(\omega_1, \omega_2), \omega_3) \quad \text{et} \quad \perp(\omega_1, \perp(\omega_2, \omega_3)) = \perp(\perp(\omega_1, \omega_2), \omega_3) \quad (1.20)$$

Axiome 3. *Élément neutre.*

$$\top(\omega_1, 1) = \omega_1 \quad \text{et} \quad \perp(\omega_1, 0) = \omega_1 \quad (1.21)$$

Propriété 19. *Coïncidence avec l'intersection et l'union classiques.*

$$\begin{aligned} \top(0, 1) = \top(0, 0) = \top(1, 0) = 0 & \quad \text{et} \quad \top(1, 1) = 1 \\ \perp(0, 1) = \perp(1, 1) = \perp(1, 0) = 1 & \end{aligned} \quad (1.22)$$

Les T-Normes et T-Conormes disposent d'une propriété intéressante : la dualité relativement à une fonction de complémentation. Nous devons d'abord introduire la notion de fonction de complémentation qui étend au cas flou la complémentation d'un ensemble classique. Pour cela, il est nécessaire de définir une fonction C de $[0, 1]$ vers $[0, 1]$ qui est telle que :

$$\begin{aligned} C(0) = 1 \text{ et } C(1) = 0 \\ \omega_1 > \omega_2 \implies C(\omega_1) < C(\omega_2) \\ C \text{ est continue} \\ C \circ C = Id \end{aligned} \quad \text{le complémenté de } \mu_A \text{ sera } \mu_{A^c}(\omega) = C(\mu_A(\omega)) \quad (1.23)$$

Etant donnée une fonction de complémentation C , il est possible de mettre en évidence la dualité relative à la complémentation C entre une T-Norme et une T-Conorme à l'aide de la loi de DE MORGAN.

Propriété 20. *Loi de DE MORGAN.*

Pour certaines T-Normes/T-conormes, il existe une complémentation C telle que :

$$C(\top(\omega_1, \omega_2)) = \perp(C(\omega_1), C(\omega_2)) \quad \text{et} \quad C(\perp(\omega_1, \omega_2)) = \top(C(\omega_1), C(\omega_2)) \quad (1.24)$$

Ainsi, étant donnée C et une T-Norme, il est possible de déterminer la T-Conorme duale. Le plus souvent, la fonction C est définie par $C(x) = 1 - x$. Le tableau 1.2 (en haut) présente quelques T-Normes et T-Conormes duales. La fonction de complémentation utilisée est $C(x) = 1 - x$.

Parmi les T-Normes (T-Conormes), le couple (\min, \max) est particulièrement intéressant. Ce sont les seuls opérateurs d'intersection (d'union) idempotents et mutuellement distributifs. Par ailleurs, ils agissent comme des bornes pour les autres opérateurs d'intersection (d'union) :

$$\forall (x, y) \in \mathbb{R}^2, \top(x, y) \leq \min(x, y) \quad \text{et} \quad \forall (x, y) \in \mathbb{R}^2, \perp(x, y) \geq \max(x, y) \quad (1.25)$$

Il existe une relation d'ordre entre les T-Normes et T-Conormes les plus usités dont \min et \max sont les bornes. Nous sommes alors à même de classer les T-Normes/T-Conormes de la plus petite à la plus grande. Etant données une T-Norme \cap et une T-Conorme \cup , la relation d'encadrement suivante est toujours vérifiée (voir tableau 1.2 page suivante (en haut)) :

$$\top_{Weber} \leq \cap \leq \top_{Zadeh} \leq \perp_{Zadeh} \leq \cup \leq \perp_{Weber} \quad (1.26)$$

où \top_{Weber} , \perp_{Weber} , \top_{Zadeh} , \perp_{Zadeh} désignent respectivement les T-Norme, T-Conorme de WEBER et les T-Norme, T-Conorme de ZADEH. La tableau récapitulatif 1.5 rappelle l'ordre existant entre quelques T-Normes/T-Conormes.

Le tableau 1.2 (en bas) présente quelques familles de T-Normes/T-Conormes dites "paramétrées". Il s'agit de T-Normes/T-Conormes dont l'expression dépend d'un réel qui permet de faire varier le comportement de la T-Norme/T-Conorme entre deux limites. Par exemple, la famille de DUBOIS et PRADE permet d'obtenir toute T-Norme/T-Conorme entre le minimum \min pour $\alpha = 0$ et le produit xy pour $\alpha = 1$. Dans ce tableau, les T-Conormes ont été obtenues à l'aide de la fonction de complémentation $C(x) = 1 - x$ sauf, la T-Conorme de WEBER (dernière ligne du tableau) qui a été obtenue avec $C(x) = \frac{1-x}{1+\lambda x}$.

Le paragraphe suivant présente la deuxième famille d'opérateurs CICB : les moyennes.

1.4.2.2 Les moyennes

Les opérations d'intersection et d'union floues étant inférieure ou supérieure aux fonctions minimum (respectivement maximum), une partie des opérations d'agrégation n'est pas couverte par l'union et l'intersection. Ce vide est comblé par les opérations de moyennes ensemblistes. Une moyenne est une application m de $[0, 1]^2$ vers $[0, 1]$ vérifiant :

$$\begin{aligned} \triangleright & \forall (x, y) \in \mathbb{R}^2, \min(x, y) \leq m(x, y) \leq \max(x, y), m \notin \{\min, \max\} \\ \triangleright & \forall (x, y) \in \mathbb{R}^2, m(x, y) = m(y, x) \\ \triangleright & m \text{ est non décroissante en fonction de chaque argument} \end{aligned} \quad (1.27)$$

Ces conditions étant vérifiées, m est idempotent. Si m est strictement croissant, il ne peut pas être associatif de sorte que les seules moyennes croissantes sont les médianes définies pour un seuil $\alpha \in]0, 1[$ par :

$$\text{med}_\alpha(x, y, \alpha) = \begin{cases} y & \text{si } x \leq y \leq \alpha \\ \alpha & \text{si } x \leq \alpha \leq y \\ x & \text{si } \alpha \leq x \leq y \end{cases} \quad (1.28)$$

L'associativité étant difficilement compatible avec la notion de moyenne, elle est parfois remplacée par la bisymétrie :

$$\forall (x, y, z, t) \in \mathbb{R}^4, m(m(x, y), m(z, t)) = m(m(x, z), m(y, t)) \quad (1.29)$$

Les moyennes continues et croissantes strictement vérifiant (1.29) sont de la forme [Dubois et Prade, 1988] :

$$m(x, y) = k^{-1} \left(\frac{k(x) + k(y)}{2} \right) \quad (1.30)$$

T-Normes/T-Conormes			
Numéro	Nom	T-Norme	T-Conorme
1	LUKASIEWICZ	$\max(0, x + y - 1)$	$\min(x + y, 1)$
2	Probabiliste	$x \cdot y$	$x + y - x \cdot y$
3	ZADEH	$\min(x, y)$	$\max(x, y)$
4	WEBER (non continue)	$\begin{cases} x & \text{si } y = 1 \\ y & \text{si } x = 1 \\ 0 & \text{sinon} \end{cases}$	$\begin{cases} x & \text{si } y = 0 \\ y & \text{si } x = 0 \\ 1 & \text{sinon} \end{cases}$
5		$\frac{xy}{x + y - xy}$	$\frac{x + y - 2xy}{1 - xy}$

T-Normes/T-Conormes paramétrées				
Nom	T-Norme	T-Conorme	Limites de la T-Norme	
			Minimum	Maximum
HAMACHER $\gamma \geq 0$	$\frac{xy}{\gamma + (1 - \gamma)(x + y - xy)}$	$\frac{x + y - xy - (1 - \gamma)xy}{1 - (1 - \gamma)xy}$	(5) pour $\gamma = 0$	(4) pour $\gamma \rightarrow \infty$
DUBOIS-PRADE	$\frac{xy}{\max(x, y, \alpha)}$	$1 - \frac{(1 - x)(1 - y)}{\max(1 - x, 1 - y, \alpha)}$	(3) pour $\alpha = 0$	(2) pour $\alpha = 1$
FRANCK	$\log_s \left(1 + \frac{(s^x - 1)(s^y - 1)}{s - 1} \right)$	$1 - \log_s \left(1 + \frac{(s^{1-x} - 1)(s^{1-y} - 1)}{s - 1} \right)$	(3) pour $s = 0$	(1) pour $s \rightarrow +\infty$
YAGER	$\max(0, ((1 - x)^p + (1 - y)^p)^{\frac{1}{p}})$	$\min(1, (x^p + y^p)^{\frac{1}{p}})$	(1) pour $p = 1$	(4) pour $p \rightarrow +\infty$
YU-YANDONG	$\max((1 + \lambda)(x + y - 1) - \lambda xy, 0)$	$\min(x + y + \lambda xy, 1)$	(2) pour $\lambda = -1$	(4) pour $\lambda \rightarrow +\infty$
DOMBI	$\frac{1}{1 + ((\frac{1}{x} - 1)^\lambda + (\frac{1}{y} - 1)^\lambda)^{\frac{1}{\lambda}}}$	$\frac{1}{1 + ((\frac{1}{x} - 1)^{-\lambda} + (\frac{1}{y} - 1)^{-\lambda})^{\frac{1}{-\lambda}}}$	(4) pour $\lambda = 0$	(2) pour $\lambda \rightarrow +\infty$
WEBER	$\max(0, \frac{x+y-1+\lambda xy}{1+\lambda})$	$\min(x + y + \lambda xy, 1)$	(4) pour $\lambda = -1$	(4) pour $\lambda \rightarrow +\infty$

TAB. 1.2 – Quelques T-Normes/T-Conormes classiques.

où k est une fonction continue strictement monotone. Lorsque $k(x) = x^\alpha, \alpha \in \mathbb{R}$, nous obtenons la famille m_α . Quelques moyennes classiques et quelques cas particuliers de m_α sont rappelés dans le tableau 1.3. Les opérateurs OWA et les intégrales floues seront présentés au paragraphe 4.3.

Famille	Expression	Remarques												
Médiane	$\text{med}_\alpha(x, y, \alpha) = \begin{cases} y & \text{si } x \leq y \leq \alpha \\ \alpha & \text{si } x \leq \alpha \leq y \\ x & \text{si } \alpha \leq x \leq y \end{cases}$	Famille des moyennes associatives												
Famille " m_α "	<table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>α</th> <th>m_α</th> </tr> </thead> <tbody> <tr> <td>$-\infty$</td> <td>$\min(x, y)$</td> </tr> <tr> <td>-1</td> <td>$\frac{2xy}{x+y}$</td> </tr> <tr> <td>0</td> <td>\sqrt{xy}</td> </tr> <tr> <td>$+1$</td> <td>$\frac{x+y}{2}$</td> </tr> <tr> <td>$+\infty$</td> <td>$\max(x, y)$</td> </tr> </tbody> </table>	α	m_α	$-\infty$	$\min(x, y)$	-1	$\frac{2xy}{x+y}$	0	\sqrt{xy}	$+1$	$\frac{x+y}{2}$	$+\infty$	$\max(x, y)$	(moyenne harmonique) (moyenne géométrique) (moyenne arithmétique)
α	m_α													
$-\infty$	$\min(x, y)$													
-1	$\frac{2xy}{x+y}$													
0	\sqrt{xy}													
$+1$	$\frac{x+y}{2}$													
$+\infty$	$\max(x, y)$													
OWA	Voir paragraphe 4.1													
Intégrales floues	Voir paragraphe 4.3													

TAB. 1.3 – Quelques exemples de moyennes.

1.4.3 Opérateurs indépendants du contexte à comportement variable (CIVB)

Les opérateurs de cette famille sont les sommes symétriques. Une somme symétrique est une fonction σ définie [Dubois et Prade, 1988] par :

$$\sigma : [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (x, y) \mapsto \sigma(x, y) \quad \text{vérifiant} \quad \begin{cases} \sigma \text{ non décroissante en fonction de chaque argument,} \\ \sigma \text{ continue et commutative,} \\ \sigma \text{ auto-duale,} \\ \sigma(0, 0) = 0 \text{ et } \sigma(1, 1) = 1 \end{cases} \quad (1.31)$$

Pour vérifier l'auto-dualité (propriété 5) il est nécessaire de choisir un opérateur de complémentation. Souvent, le choix se porte sur l'opérateur de ZADEH $\forall x, C(x) = 1 - x$. Dans ce cas, la fonction σ peut être mise sous la forme :

$$\sigma : [0, 1] \times [0, 1] \rightarrow [0, 1] \quad (x, y) \mapsto \sigma(x, y) = \frac{g(x, y)}{g(x, y) + g(1 - x, 1 - y)}, \quad g \text{ vérifiant} \quad \begin{cases} g(0, 0) = 0, \\ g \text{ continue,} \\ g \text{ non négative,} \\ g \text{ non décroissante.} \end{cases} \quad (1.32)$$

Il est possible de distinguer deux grands types de sommes symétriques : les sommes associatives et les sommes non associatives. Ces dernières – qui sont des moyennes – ont été traitées au paragraphe 1.4.2.2 car il s'agit d'opérateurs CICB. Il existe cependant une exception : la seule somme symétrique qui soit associative et qui soit un opérateur de moyenne est $\text{med}(x, y, \frac{1}{2})$. Nous nous intéressons plus particulièrement aux sommes symétriques associatives.

Pour comprendre le comportement des sommes associatives, considérons le cas de $\text{med}(x, y, \frac{1}{2})$. Si x et y sont faibles, c'est-à-dire plus petits que $\frac{1}{2}$, le résultat est disjonctif et indulgent : $\max(x, y)$. Si x et y sont grands, c'est-à-dire plus grands que $\frac{1}{2}$, le résultat est conjonctif et sévère : $\min(x, y)$. Enfin, si l'une des

deux valeurs x ou y est plus petite que $\frac{1}{2}$ et que l'autre est plus grande que $\frac{1}{2}$, l'opérateur a un comportement de compromis en fournissant $\frac{1}{2}$. Plus généralement, toute somme associative symétrique (sauf la médiane) possède le comportement décrit ci-dessous :

Comportement	Lorsque	Ce qui conduit à
Conjonctif	$\max(x, y) < \frac{1}{2}$	$\sigma(x, y) \leq \min(x, y)$
Disjonctif	$\min(x, y) > \frac{1}{2}$	$\sigma(x, y) \geq \max(x, y)$
Compromis	$x \geq \frac{1}{2} \geq y$	$x \leq \sigma(x, y) \leq y$
	$y \geq \frac{1}{2} \geq x$	$y \leq \sigma(x, y) \leq x$

L'opérateur $\text{med}(x, y, \frac{1}{2})$ apparaît donc comme la limite entre les opérateurs CICB et les opérateurs CIVB. Nous constatons une modification du comportement en fonction des valeurs devant être combinées. Le tableau 1.4 rappelle quelques sommes symétriques classiques.

Somme	$g(x, y)$	Remarques
$\sigma_0(x, y) = \frac{xy}{1 - x - y + 2xy}$	xy	Associative
$\sigma_+(x, y) = \frac{x + y - xy}{1 + x + y - 2xy}$	$x + y - xy$	Non associative
$\sigma_{\min}(x, y) = \frac{\min(x, y)}{1 - x - y }$	$\min(x, y)$	Moyenne
$\sigma_{\max}(x, y) = \frac{\max(x, y)}{1 + x - y }$	$\max(x, y)$	Moyenne

TAB. 1.4 – Quelques sommes symétriques.

Il existe aussi des opérateurs CIVB paramétrés. Ils permettent en général d'aller de la conjonction à la disjonction simplement en modifiant la valeur d'un paramètre. C'est le cas de la fonction $h(\omega_1 + \omega_2 - \alpha)$ définie par :

$$h(t) = \begin{cases} 0 & \text{si } t \leq 0 \\ t & \text{si } 0 \leq t \leq 1 \\ 1 & \text{si } t \geq 1 \end{cases} \tag{1.33}$$

Cet opérateur évolue de $\max(0, x + y - 1)$ pour $\alpha = 1$ à $\min(1, x + y)$ pour $\alpha = 0$. L'opérateur est donc CICB aux limites et CIVB ailleurs. Il est possible de définir un opérateur CIVB en utilisant une T-Norme et une T-Conorme. Soient \top et \perp une T-Norme et une T-Conorme l'opérateur défini par :

$$\top(x, y)^\gamma \perp(x, y)^{1-\gamma} \tag{1.34}$$

varie entre \top pour $\gamma = 1$ et \perp pour $\gamma = 0$. Aux limites, cet opérateur est donc CICB alors que partout ailleurs, il est CIVB. Le tableau 1.5 présente un récapitulatif ordonné de nombreux opérateurs CICB et CIVB et le comportement de quelques-uns d'entre eux est étudié dans le paragraphe suivant.

1.5 Quelques exemples

Nous présentons ici graphiquement le comportement de quelques opérateurs détaillés dans les paragraphes précédents. Pour obtenir ces résultats, nous considérons les ensembles flous de la figure 1.5 page suivante. Nous partons de la configuration de la figure 1.5 page ci-contre, calculons le résultat de la fusion par un opérateur (T-Norme/T-Conorme/Moyenne ou somme symétrique). Puis, l'ensemble π_1 est décalé vers l'ensemble π_2 d'un réel δ (voir figure 1.6 page suivante), le résultat de la fusion est calculé et le processus recommence. L'arrêt intervient lorsque π_1 a "dépassé" π_2 . Durant ce processus, les informations représentées par π_1 et π_2 sont tout d'abord en désaccord, puis passent en accord total et deviennent de nouveau en désaccord. En groupant les différents résultats pour δ variant entre 0 et 8 sur un même graphique, nous obtenons une idée du

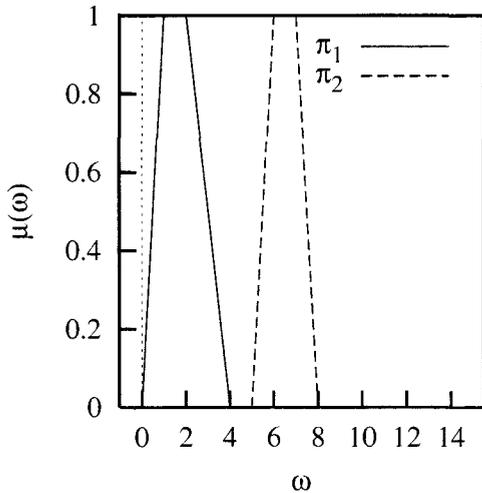


FIG. 1.5 – Les ensembles de test.

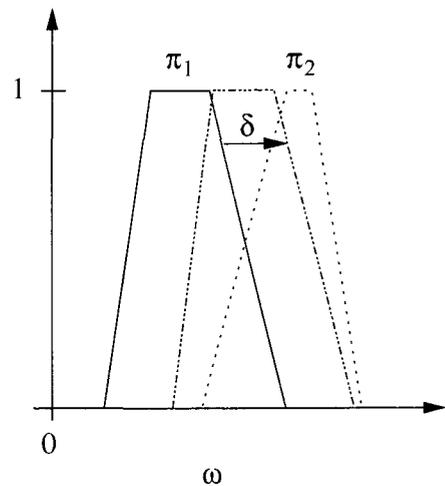


FIG. 1.6 – Le processus de calcul.

comportement de l'opérateur lorsque les informations passent de la disjonction à la conjonction. Ainsi, pour un δ donné il est possible de retrouver la position des distributions π_1 et π_2 et de visualiser le résultat de la fusion.

C'est en utilisant ce procédé que les figures 1.7 à 1.12 ont été obtenues. Il est important de noter que le résultat de la fusion n'a pas été normalisé pour montrer l'effet de l'opérateur. Nous pouvons vérifier graphi-

$\min(\omega_1, \omega_2)$

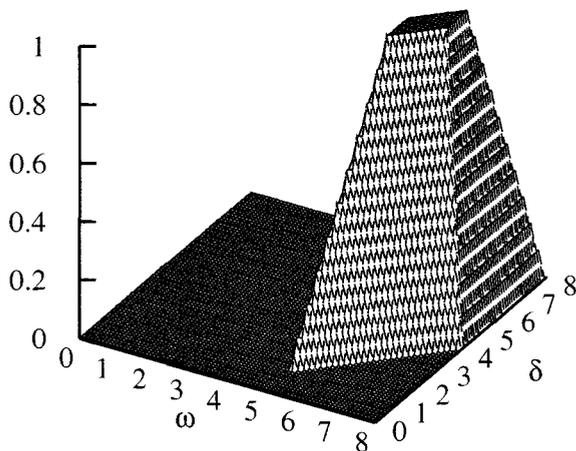


FIG. 1.7 – $\top = \min(\omega_1, \omega_2)$.

$\max(\omega_1, \omega_2)$

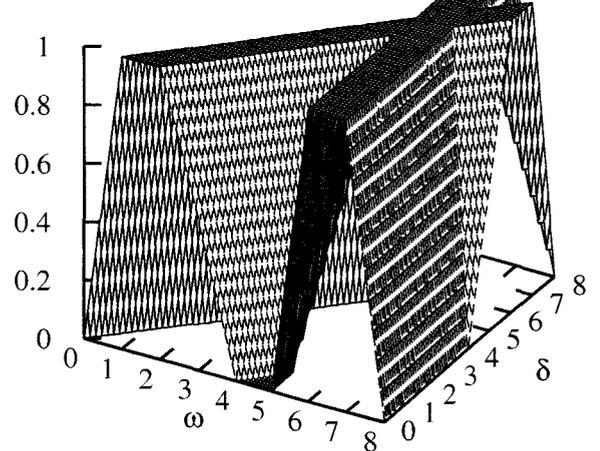


FIG. 1.8 – $\perp = \max(\omega_1, \omega_2)$.

quement que les opérateurs \min et \max ci-dessus conduisent bien à l'intersection et l'union des ensembles.

Sur la figure 1.13 page suivante, les zones pour lesquelles le dénominateur n'est pas défini ont été placées à la valeur arbitraire de 2. Il est ainsi facile de visualiser les zones où l'opérateur n'est pas défini. Cet opérateur n'est pas défini lorsque les sources sont en désaccord. Il est possible de fixer à 1 les zones où le dénominateur n'est pas défini pour obtenir un prolongement par continuité de l'opérateur.

1.6 L'importance du choix du mode de combinaison

Dans ce paragraphe, nous utilisons les opérateurs présentés ci-dessus pour la résolution d'un problème de fusion simple. Supposons qu'il se produise un accident. Deux passants sont présents sur les lieux et sont interrogés par un policier. Le premier témoin affirme que la voiture qui a provoqué l'accident roulait environ à

$\max(0, \omega_1 + \omega_2 - 1)$

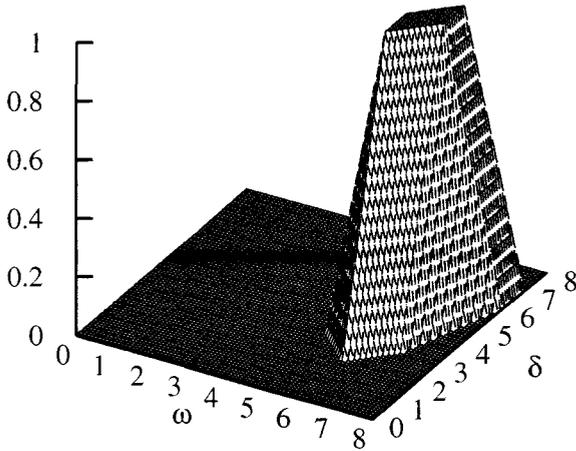


FIG. 1.9 - $T = \max(0, \omega_1 + \omega_2 - 1)$.

$\omega_1 \omega_2$

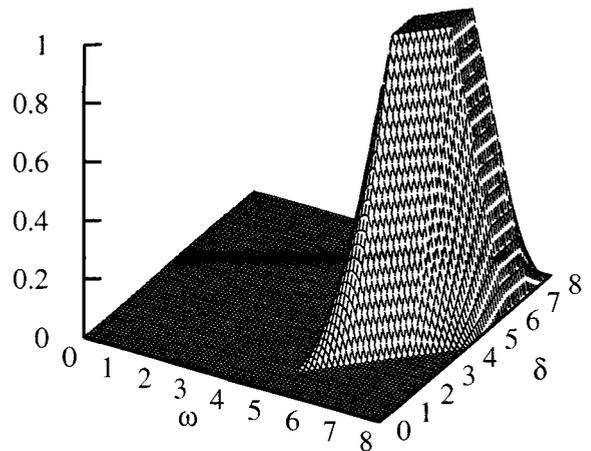


FIG. 1.10 - $T = \omega_1 \omega_2$.

$\sqrt{\omega_1 \omega_2}$

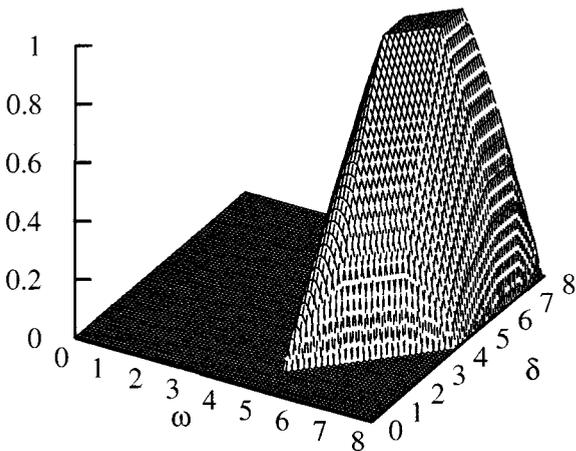


FIG. 1.11 - $m = \sqrt{\omega_1 \omega_2}$.

$(\omega_1 + \omega_2)/2$

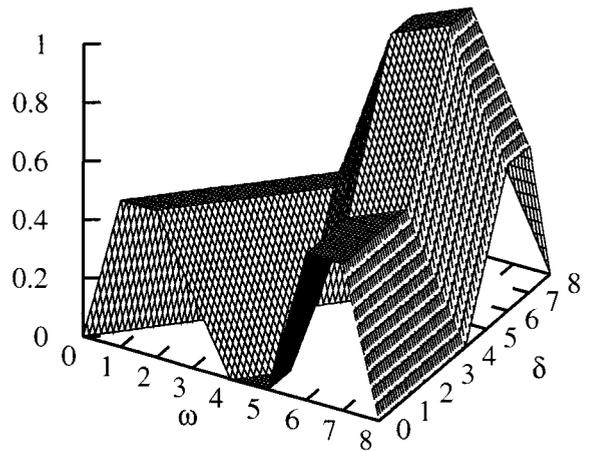


FIG. 1.12 - $m = \frac{\omega_1 + \omega_2}{2}$.

$g = \omega_1 \omega_2$

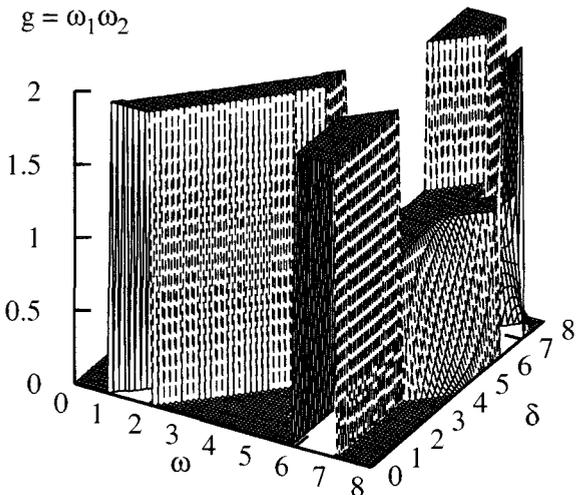


FIG. 1.13 - $\sigma_{min} = \frac{\omega_1 \omega_2}{1 - \omega_1 - \omega_2 + 2\omega_1 \omega_2}$.

$g = \omega_1 + \omega_2 - \omega_1 \omega_2$

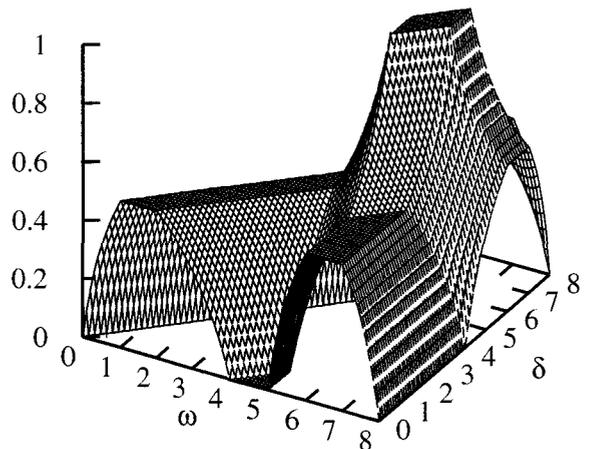


FIG. 1.14 - $\sigma_{max} = \frac{\omega_1 + \omega_2 - \omega_1 \omega_2}{1 + \omega_1 + \omega_2 - 2\omega_1 \omega_2}$.

90 kmh^{-1} . L'autre témoin affirme que cette même voiture roulait entre 65 kmh^{-1} et 85 kmh^{-1} . La première étape consiste à représenter ces informations dans un même formalisme, par exemple les sous-ensembles flous (voir figure 1.15 page suivante). Le policier chargé de l'enquête doit alors choisir parmi tous les opérateurs

de fusion celui qui est adapté à la situation. Etant donné le nombre d'opérateurs disponibles, le choix est difficile. Nous écartons tous les opérateurs non symétriques, c'est-à-dire ceux pour lesquels l'ordre dans lequel les sources sont combinées a une grande importance. En effet, dans notre problème, l'ordre dans lequel les témoignages sont recueillis ne doit pas avoir d'influence sur le résultat. Pour simplifier l'exemple, nous retenons trois opérateurs : le min pour l'opérateur conjonctif, le max pour l'opérateur disjonctif et la moyenne arithmétique pour l'opérateur de moyenne. Les résultats sont présentés sur les figures 1.15 à 1.17.

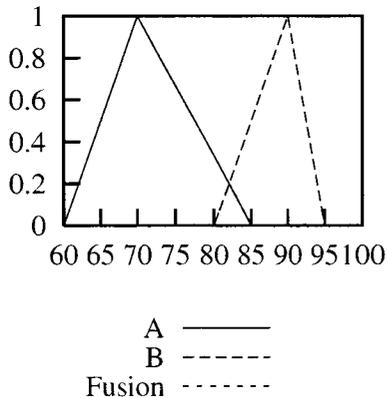


FIG. 1.15 – Conjonction

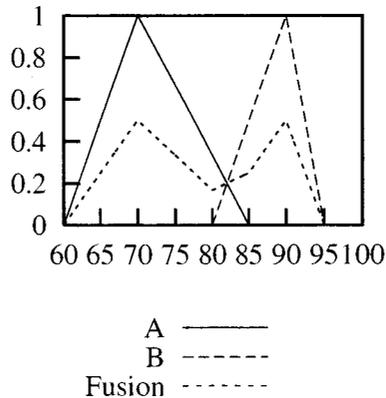


FIG. 1.16 – Compromis

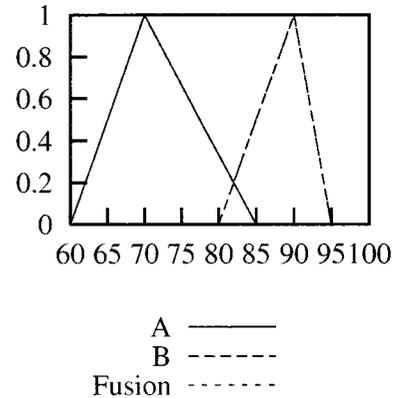
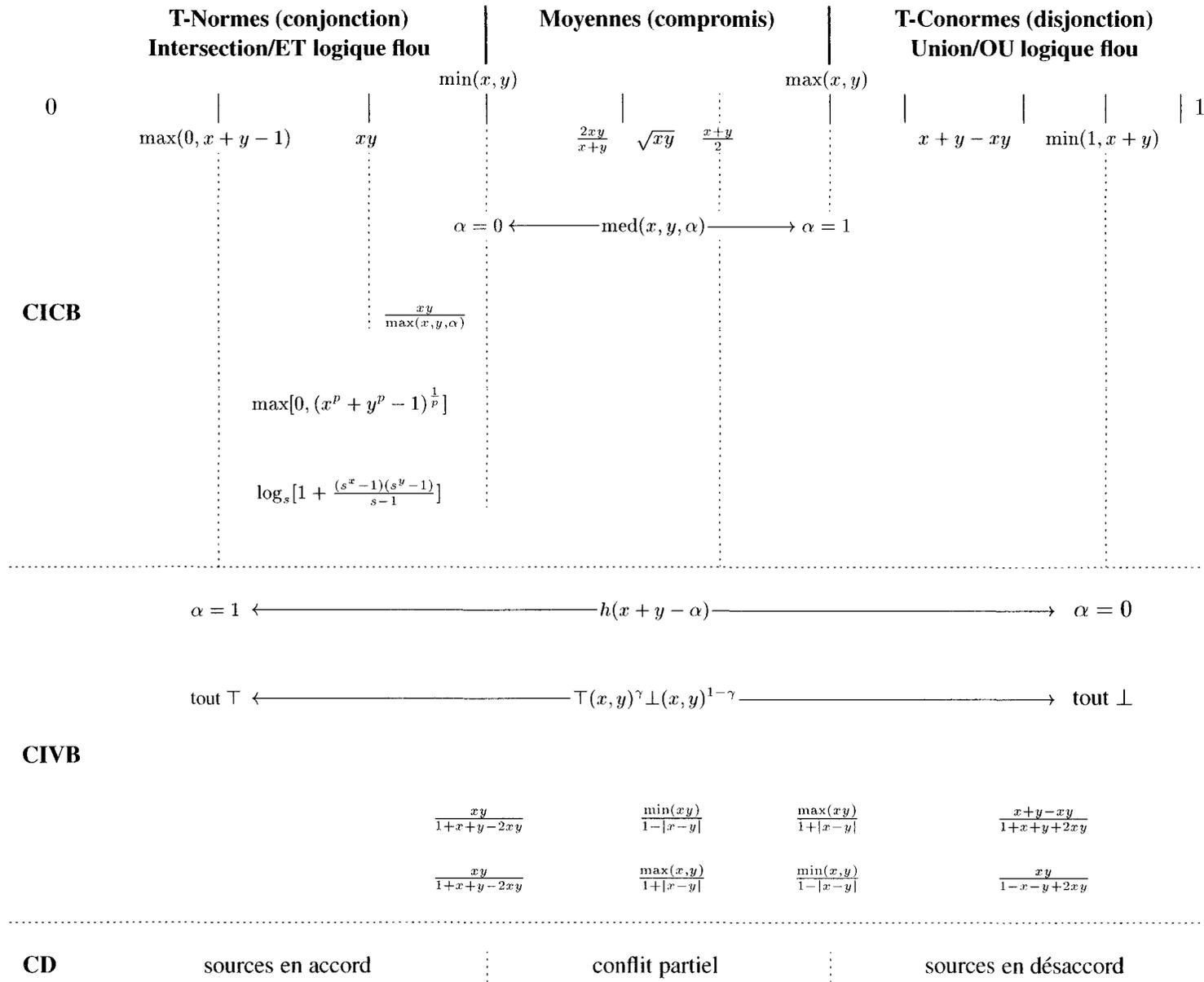


FIG. 1.17 – Disjonction

Les informations A et B fournies par les deux témoins sont en désaccord. La fusion par le minimum (figure 1.15) conduit à un nombre flou qui est l'intersection des entrées. La vitesse de la voiture appartient très faiblement à un ensemble flou "entre 80 kmh^{-1} et 85 kmh^{-1} ". La moyenne (figure 1.15) accorde crédit aux deux témoins mais faiblement tandis que le maximum a pour effet d'affirmer que l'un des deux témoins à forcément raison. Le choix de l'opérateur est donc fortement conditionné par les entrées. Dans ce cas précis, le policier aura tendance à privilégier le mode de combinaison disjonctif car en l'absence d'autres informations, la réalité est peut-être fournie par l'un ou par l'autre (et éventuellement par aucun, mais nous avons supposé que, dans le processus de fusion, une source au moins disait la vérité).

1.7 Conclusion

Ce chapitre a décrit les trois modes de combinaisons de base : la conjonction adaptée à des données concordantes, la disjonction utile pour des sources en fort désaccord et le compromis intéressant lorsque les sources ne sont ni concordantes ni discordantes et dans les applications courantes où le concept courant de moyenne est traditionnellement utilisé. Nous avons proposé un tour d'horizon des opérateurs "simples" ou indépendant du contexte ce qui signifie que l'opérateur a le même comportement que les sources soient en accord ou en désaccord. Nous avons suggéré que l'utilisation pratique de ces opérateurs est un peu freinée par deux problèmes : la difficulté du choix parmi tous les expressions mathématiques disponibles et les dangers de l'agrégation dans un mode incohérent avec les informations à combiner. Le chapitre suivant s'intéresse à ce second problème en décrivant les opérateurs CD (Context Dependiant ou "dépendants du contexte") qui choisissent automatiquement le mode de combinaison adapté.



TAB. 1.5 – Le comportement des opérateurs de fusion simples d’après [Bloch, 1996].

Chapitre 2

Les opérateurs adaptatifs

LE chapitre précédent a montré que la présence de trois modes de combinaison pose un problème de choix. Etant données les informations à combiner, est-il possible de déterminer le mode de combinaison le plus adapté? Intuitivement, la conjonction est plutôt réservée au cas où les sources sont considérées comme fiables toutes les deux tandis que la disjonction est plutôt adaptée à deux sources dont seule l'une apparaît comme fiable. Le compromis est un mode intermédiaire, utilisé lorsque les deux sources ne sont pas considérées comme entièrement fiables ou lorsque l'on suppose des problèmes d'imprécision. Le choix du mode passe donc par l'évaluation de la discordance entre les deux sources ou encore l'indice de conflit. Une fois l'intensité de l'accord entre les sources déterminée, l'opérateur doit alors pouvoir passer du mode conjonctif au mode disjonctif de manière continue. Il existe actuellement deux opérateurs disposant de cette capacité : les deux règles de DUBOIS et PRADE [Dubois et Prade, 1994a]. Ce chapitre est dédié à l'étude de cette famille très restreinte d'opérateurs.

Dans un premier paragraphe, nous décrivons la notion d'indice de conflit permettant d'évaluer le niveau de consensus entre les sources. Les différents indices de conflits sont présentés et leurs faiblesses analysées. De nouveaux indices de conflits sont alors proposés. Le second paragraphe décrit les opérateurs adaptatifs de DUBOIS et PRADE et présente leurs avantages, propriétés et défauts. De nouveaux opérateurs adaptatifs sont alors présentés.

2.1 Quantifier l'accord entre les sources

La prise en compte de la concordance des sources dans le processus de fusion nécessite la détection et la quantification du désaccord entre les sources. D'une manière intuitive, deux sources s'accordent lorsqu'elles fournissent des informations proches et sont en désaccord dans le cas contraire. Il est intéressant de disposer d'un outil permettant de mesurer la ressemblance entre deux sources d'information. L'idée d'évaluer la ressemblance entre deux objets est très ancienne. Dans la vie courante, il existe de nombreuses situations dans lesquelles nous utilisons la ressemblance entre deux objets sans même nous en rendre compte. A qui n'a-t-on jamais dit : "Votre fils vous ressemble tellement"? Dans cette situation, votre interlocuteur a mentalement estimé le degré de similitude entre le visage de votre fils et le votre et ce degré était assez haut pour susciter une remarque.

Après avoir rappelé les principaux indices de similarité utilisés en analyse de données [Lerman, 1970], nous proposons une extension possible de ces indices pour des ensembles flous. Nous faisons alors le point sur les indices spécifiques à la théorie des possibilités avant de présenter de nouvelles approches pour la mesure du conflit entre deux distributions de possibilités.

2.1.1 Similitude et analyse de données

Soit E un ensemble fini (au sens classique) d'objets ou de sujets dont le cardinal sera noté p . L'analyse de données consiste à rassembler en classes les objets de E jugés similaires. Comme dans tout classement fondé

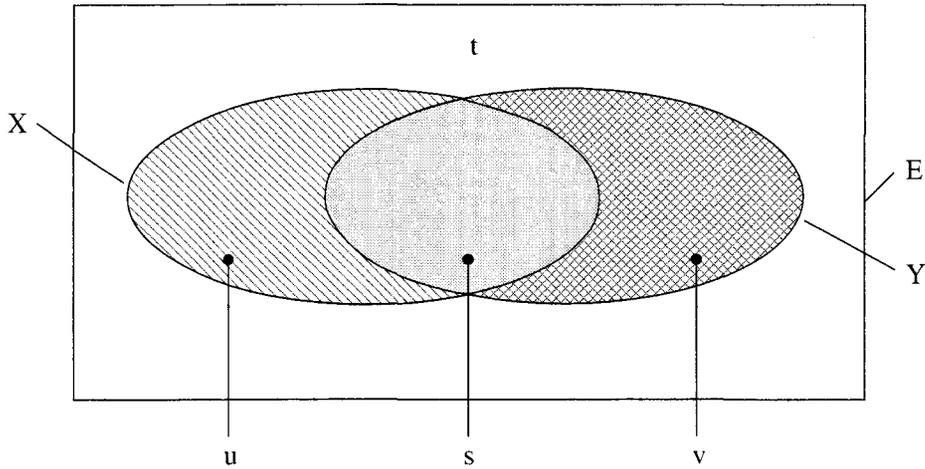


FIG. 2.1 – Représentation graphique de s, t, u, v pour E classique.

sur la similitude, il faut déterminer un seuil à partir duquel deux objets seront rangés dans la même classe. Le choix de cette limite est critique : lorsqu'elle est trop élevée, tous les objets appartiendront à une même classe et lorsqu'elle est trop faible, il y aura autant de classes que d'éléments dans E . La détermination du seuil est d'autant plus critique qu'elle dépend des données, de la mesure de similarité choisie et du résultat que désire obtenir l'analyste. Avant de définir la similarité entre deux parties X et Y de $\mathbb{P}(E)$, il est nécessaire de définir la notion d'indicateur du couple (X, Y) (voir figure 2.1).

Définition 1. *Indicateur du couple (X, Y) .*

L'indicateur du couple (X, Y) est un triplet d'entiers (s, u, v) défini à l'aide du cardinal d'ensembles :

$$\begin{aligned} s &= |X \cap Y| \\ v &= |X^C \cap Y| \\ u &= |X \cap Y^C| \end{aligned} \quad \text{On définit aussi} \quad \begin{aligned} t &= |X^C \cap Y^C| \\ p &= |E| = s + t + u + v \end{aligned}$$

Définition 2. *Similarité.*

La similarité S entre deux parties X et Y de $\mathbb{P}(E)$ est définie à l'aide de l'indicateur du couple (X, Y) :

$$S : (X, Y) \in \mathbb{P}(E)^2 \longrightarrow \mathbb{R}^+ \\ X, Y \longmapsto S(X, Y) = S'(s, u, v) \quad \text{avec} \quad \begin{cases} S' : \{(s, u, v) \in \mathbb{N}^3 / s + u + v \leq p\} \longrightarrow \mathbb{N} \\ S'(s, u, v) = S'(s, v, u) \text{ (symétrie en } u \text{ et } v) \\ \text{strictement croissante par rapport à } s \\ \text{strictement décroissante par rapport à } u \end{cases}$$

Souvent, la fonction $S(s, u, v)$ prend la forme d'une fraction rationnelle dans laquelle apparaissent les entiers caractéristiques s, t, u, v pondérés par des coefficients entiers α_i :

$$S(X, Y) = \frac{\alpha_1 s + \alpha_2 t + \alpha_3 u + \alpha_4 v}{\alpha_5 s + \alpha_6 t + \alpha_7 u + \alpha_8 v} \tag{2.1}$$

Le choix de ces entiers est dépendant de l'application. Cependant, de nombreuses combinaisons des α_i ont été étudiées et les indices classiques qu'elles induisent sont rappelés avec quelques unes de leurs propriétés dans le tableau 2.2. Dans ce résumé, E est supposé être un ensemble non vide au sens classique et S représente la mesure de similarité.

Nom (suite)	Expression (suite)	Remarques (suite)
HARMAN	$1 - \frac{2(u+v)}{p}$	<ul style="list-style-type: none"> ▷ La similarité de HARMAN varie entre -1 et 1. ▷ Quelles que soient X et Y, $-1 \leq S(X, Y) \leq 1$. ▷ $X = Y \implies S(X, Y) = 1$ et $S(X, Y) = -1$. ▷ X et Y forment une partition de $E \implies S(X, Y) = -1$. ▷ $u + v = \frac{p}{2} \implies S(X, Y) = 0$.
ROGER et TAMINO	$\frac{p-(u+v)}{p+(u+v)}$	<ul style="list-style-type: none"> ▷ Cette similarité varie entre 0 et 1. ▷ $S(X, Y) = 0$ dès que X et Y forment une partition de E. ▷ $X = Y \implies S(X, Y) = 1$.
KULEZYNSKI	$\frac{s}{u+v}$	<ul style="list-style-type: none"> ▷ $X = Y \implies u + v = 0 \implies S(X, X) = +\infty$. ▷ $X \cap Y = \emptyset \implies S(X, Y) = 0$. ▷ Plus X se rapproche de Y, plus la mesure grandit.
JACCARD	$\frac{s}{s+u+v}$	<ul style="list-style-type: none"> ▷ Cette similarité varie entre 0 et 1. ▷ Lorsque les deux parties sont disjointes $S(X, Y) = 0$. ▷ $X = Y \implies S(X, Y) = 1$. ▷ Évalue le nombre d'objets communs aux deux parties par rapport au nombre d'objets considérés dans les deux parties.
DICE	$\frac{2s}{2s+u+v}$	<ul style="list-style-type: none"> ▷ Similarité de JACCARD avec coefficient 2 au cardinal de l'intersection s
KULEZYNSKI modifié	$\frac{1}{2} \left(\frac{s}{s+u} + \frac{s}{s+v} \right)$	<ul style="list-style-type: none"> ▷ $X = Y \implies S(X, Y) = 1$.

Nom (suite)	Expression (suite)	Remarques (suite)
SOKAL-SNEATH	$\frac{s}{s+2(u+v)}$	<p>▷ Similarité de JACCARD avec coefficient 2 à la somme uv.</p> <p>▷ $\forall (X, Y) \in \mathbb{P}(E)^2, (X, Y) \neq \emptyset^2 \implies \text{Jaccard}(X, Y) \leq S(X, Y)$</p>
OCHIAI	$\frac{s}{\sqrt{(s+u)(s+v)}}$	▷ $X = Y \implies S(X, Y) = 1$.
YULE modifié	$1 + \frac{st-uv}{st+uv}$	▷ $X = Y \implies S(X, Y) = 2$.
BRAVAIS-PAARSON	$\frac{ st-uv }{\sqrt{(s+u)(s+v)(t+u)(t+v)}}$	▷ $X = Y \implies S(X, Y) = 1$.
BRAVAIS-PAARSON modifié	$1 + \frac{ st-uv }{\sqrt{(s+u)(s+v)(t+u)(t+v)}}$	▷ $X = Y \implies S(X, Y) = 2$.

TAB. 2.2 – Quelques similarités dans le cadre de l'analyse de données.

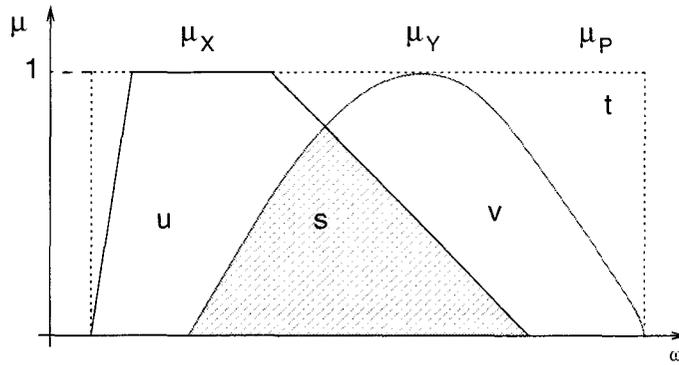


FIG. 2.2 – Représentation graphique de s , t , u , v pour E flou.

2.1.2 Extension au cas flou

Les mesures de similarité présentées précédemment utilisent l'indice du couple (X, Y) pour évaluer la similarité entre deux parties X et Y . Cet indice évalue le cardinal d'ensembles particuliers comme l'intersection de X et $Y \dots$ Etendre les mesures de similarité au cas flou (voir [Zwisch et al., 1987], [Chen et al., 1995], [Wang et al., 1995], [Bouchon-Meunier et al., 1996] et [Wang, 1997]) peut donc être fait en étendant la notion d'indice de couple à deux sous-ensembles flous. Cette extension que nous proposons maintenant tente de conserver le sens intuitif des paramètres s , u et v .

Dans ce paragraphe, E est maintenant un ensemble flou supposé fini et non vide et μ_X, μ_Y les fonctions d'appartenance¹ des deux sous-ensembles X et Y . Pour conserver le sens des paramètres s , t , u , v et p , considérons la figure 2.2 présentant les deux fonctions d'appartenance μ_X et μ_Y . Le paramètre s représentant le cardinal de l'intersection de X et de Y peut être vu comme la surface sous la fonction d'appartenance de l'intersection de X et Y . De même, le paramètre u représente la portion de la surface sous la courbe de μ_X qui n'est pas commune à μ_Y (et inversement pour v). D'une manière plus formelle, nous devons calculer l'intersection de deux sous-ensembles flous et le cardinal d'un sous-ensemble flou.

L'intersection/union de sous-ensembles flous est réalisé à l'aide de T-Normes/T-Conormes (voir paragraphe 1.4.2.1 page 16 et suivants). (\min, \max) étant le seul couple d'opérateurs d'intersection/union idempotents et mutuellement distributifs, nous les utiliserons pour calculer l'intersection et l'union des sous-ensembles flous :

$$\mu_{X \cap Y}(\omega) = \min(\mu_X(\omega), \mu_Y(\omega)) \quad \mu_{X \cup Y}(\omega) = \max(\mu_X(\omega), \mu_Y(\omega)) \quad (2.2)$$

Le cardinal d'un ensemble flou possède deux définitions : l'une pour un ensemble discret et l'autre pour un ensemble non discret. Nous rappelons ci-dessous les deux expressions :

$$\text{Pour le cas discret } |X| = \sum_{\omega \in X} \mu_X(\omega) \quad \text{et pour le cas non discret } |X| = \int_{\omega \in X} \mu_X(\omega) d\omega \quad (2.3)$$

Cela suppose que l'ensemble X ne soit pas infini car sinon la somme (ou l'intégrale) n'est pas directement calculable. C'est pour cela que nous avons supposé E fini car, dans ce cas, l'un de ses sous-ensembles est aussi fini.

Les expressions de s , u et v peuvent maintenant être données. Nous n'utilisons pas les mêmes égalités qu'à la définition 2 du paragraphe 2.1.1 car pour des ensembles flous, les égalités utilisant la complémentarité ne sont pas toujours valables :

$$X \cup X^C = E \quad X \cap X^C = \emptyset \quad (2.4)$$

1. Ces fonctions sont supposées ne pas être discrètes sans perte de généralité.

En évitant l'utilisation de la complémentation, nous supprimons par ailleurs le besoin de choisir un opérateur de complémentation.

$$\begin{aligned} s &= |X \cap Y| & t &= |E| - u - s - v \\ u &= |X| - s & p &= s + u + v + t \\ v &= |Y| - s \end{aligned} \quad (2.5)$$

Lorsque E est infini et que X et Y sont des sous-ensembles finis, le calcul de cardinal par (2.3) est possible car la somme (ou l'intégrale) est calculable. Il reste alors à étendre t et p . L'une des idées est de se ramener au cas précédent, c'est-à-dire de déterminer un sous-ensemble fini P de E contenant tous les sous-ensembles flous dont nous aurons à calculer la similarité. Nous supposons alors connaître l'union de tous les sous-ensembles flous qui interviendront dans le calcul de la similarité. Il est possible de définir le sous-ensemble classique P construit sur le support de l'union de tous les sous-ensembles flous (voir figure 2.2 page précédente). Si μ_U désigne la fonction d'appartenance de l'union floue de tous les ensembles flous intervenant dans le calcul de la similarité, la fonction d'appartenance de P est donnée par :

$$\mu_P(\omega) = \begin{cases} 0 & \text{si } \mu_U(\omega) = 0 \\ 1 & \text{sinon} \end{cases} \quad (2.6)$$

P sert alors de référentiel fini et les définitions données en (2.5) sont applicables, en remplaçant E par P :

$$\begin{aligned} s &= |X \cap Y| & t &= |P| - u - s - v \\ u &= |X| - s & p &= s + u + v + t \\ v &= |Y| - s \end{aligned} \quad (2.7)$$

Souvent, les sous-ensembles flous possèdent une signification physique ce qui permet de déterminer un sous-ensemble P qui les englobe tous sans connaître exactement leur union. L'important étant d'avoir P fini et contenant tous les sous-ensembles. C'est le cas par exemple pour des capteurs de température flous dont le minimum et le maximum sont connus. Par exemple, nous désirons calculer la similarité entre les nombres flous X et Y de la figure 2.3 en supposant que tous les nombres dont nous aurons à calculer la similarité seront dans l'intervalle $[1, 9]$.

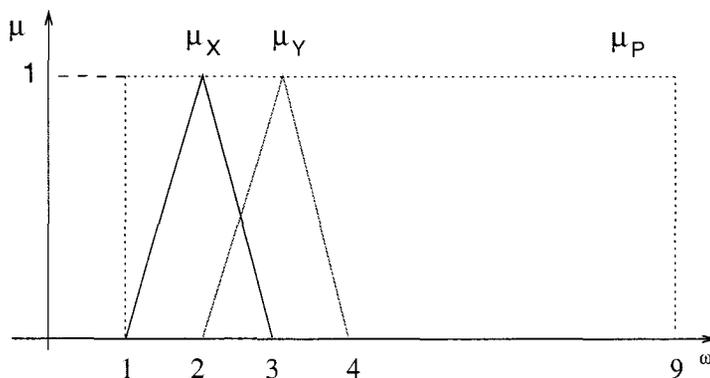


FIG. 2.3 – Calcul de la similarité sur deux nombres flous.

Nous avons donc $p = 9 * 1 = 9$, $s = 0.5 * 0.5 = 0.25$, $u = v = 1 - 0.25 = 0.75$, $t = 9 - 1.75 = 7.25$. L'indice de similarité suivant JACCARD est donc $S(X, Y) = \frac{0.25}{0.25 + 0.75 + 0.75} = 0.142$.

Si X et Y sont infinis ou si l'ensemble P est infini, il faudra trouver une autre manière d'étendre la similarité au cas flou, en s'assurant que la relation $p = s + u + v + t$ est toujours vérifiée.

La mesure de similarité entre deux sous-ensembles flous est une technique permettant de quantifier l'intensité avec laquelle deux sources d'informations s'accordent. Cependant, il existe d'autres techniques réalisant le même travail avec des distributions de possibilités. Ce sont ces indicateurs de conflit que nous décrivons maintenant.

2.1.3 Les indices de conflit

Un indice de conflit est un moyen d'évaluer le désaccord entre deux informations connues par leurs distributions de possibilités. La mesure d'indice de conflit prend la forme d'une fonction qui à deux distributions de possibilités sur un univers Ω associe un réel dans l'intervalle $[0, 1]$. Certains auteurs considèrent que la mesure de conflit doit être nulle lorsque les sources sont en accord total et d'autres [Deveughele et Dubuisson, 1996], [Dubois et Prade, 1994a] prennent la convention contraire. Nous suivons ici [Dubois et Prade, 1994a] et un conflit nul indiquera un désaccord très fort entre les sources.

Nous avons proposé dans [Vienne et al., 1996] une définition de l'indice de conflit et de la notion "d'indice plus pessimiste". L'indice de conflit est une fonction I telle que :

$$I : \Omega^2 \longrightarrow \mathbb{R}^+ \quad \text{avec} \quad \begin{cases} I \text{ continue} \\ I \text{ commutative} \\ I(\pi_1, \pi_2) = 1 \iff \text{Noyau}(\pi_1) \cap \text{Noyau}(\pi_2) \neq \emptyset \\ I(\pi_1, \pi_2) = 0 \iff \text{Support}(\pi_1) \cap \text{Support}(\pi_2) = \emptyset \end{cases} \quad (2.8)$$

Etant donné un indice de conflit I , il est alors facile de se ramener à la convention contraire (un indice de conflit nul indique des sources en parfait accord) en considérant l'indice I_R tel que :

$$I_R(\pi_1, \pi_2) = \left[\sup_{(\pi_1, \pi_2) \in \Omega^2} I(\pi_1, \pi_2) \right] - I(\pi_1, \pi_2) \quad (2.9)$$

La notion "d'indice de conflit plus pessimiste qu'un autre indice de conflit" est alors introduite :

Définition 3. *Indicateur plus pessimiste.*

Un indicateur de conflit I_1 sera dit plus pessimiste qu'un indicateur I_2 si :

$$\forall (\pi_1, \pi_2) \in \Omega^2, I_1(\pi_1, \pi_2) \leq I_2(\pi_1, \pi_2) \quad (2.10)$$

Un indicateur I_1 est plus pessimiste qu'un I_2 lorsqu'il fournit toujours des valeurs inférieures à I_2 .

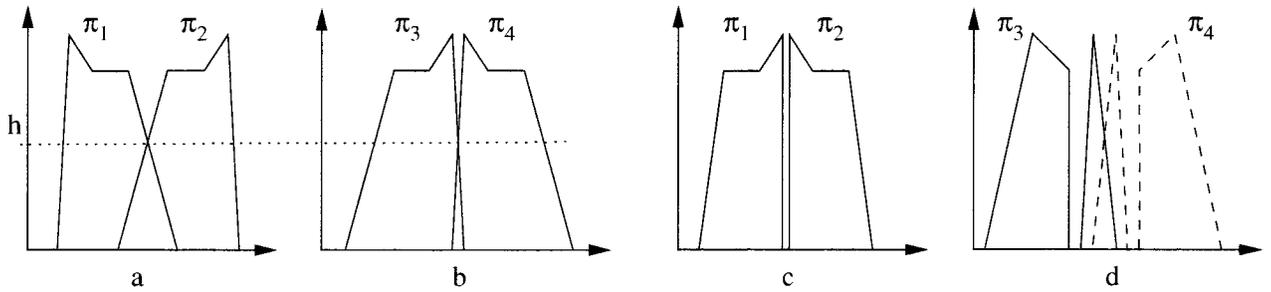
Hormis les mesures de similarités étendues au cas flou que nous avons déjà présentées et quelques distances, l'indice de conflit le plus usité est certainement h introduit dans [Dubois et Prade, 1994a] :

$$h(\pi_1, \pi_2) = \sup_{\omega \in \Omega} \min(\pi_1(\omega), \pi_2(\omega)) \quad (2.11)$$

qui a été étudié dans [Deveughele, 1993]. Cet indice donne la hauteur maximale de l'intersection des deux distributions de possibilité. Le calcul de h informatiquement est un algorithme qui doit calculer le maximum d'un tableau de valeurs (obtenues par discrétisation). Lorsque la forme des distributions n'est pas connue, la complexité² de cet algorithme est donc $O(n)$, n étant le nombre de points considérés dans le calcul. Par ailleurs, il est toujours difficile de fixer un pas de discrétisation car il s'avère très dépendant des distributions de possibilité.

Ces problèmes de mise en œuvre ne sont pas les seuls défauts de h . Considérons les distributions présentées figure 2.4 page suivante. Les distributions représentées sur la sous-figure *a* expriment un désaccord plus grand que celles de la sous-figure *b* et pourtant, la valeur de h est la même dans les deux cas. Les distributions de la

2. Le terme "complexité" [Sedgewick, 1991] désigne ici le temps utilisé par l'algorithme pour s'exécuter dans le pire des cas. Ce temps s'exprime en fonction du nombre de données traitées n . Cependant la notion de temps est trop liée à la machine sur laquelle s'exécute l'algorithme, au langage utilisé, à l'implantation qui en est faite . . . pour être réellement exploitable et le temps n'est donc pas une mesure permettant d'effectuer des démonstrations mathématiques. Il est plus intéressant d'essayer d'exprimer à l'aide d'une fonction comment la complexité évolue lorsque la taille des données manipulées change. Ainsi, on pourra dire que le temps mis par un algorithme pour traiter n données est proportionnel à n et en déduire que le traitement de $2n$ éléments sera donc deux fois plus long. Mathématiquement, cette notion de "proportionnalité" est exprimée par la notation O (prononcée "grand O "). Une fonction $g(n)$ est dite dans $O(f(n))$ (dans l'ordre de $f(n)$ ou encore proportionnelle à $f(n)$) s'il existe deux constantes c_0 et n_0 telles que $g(n) < c_0 f(n), \forall n > n_0$. La notation O donne un moyen pratique de majorer la complexité d'un algorithme.

FIG. 2.4 – Distributions pour lesquelles h est mal adapté.

sous-figure c sont proches mais leur intersection est vide et il pourrait être souhaitable d'obtenir un conflit non nul lorsque les distributions sont proches ce qui est impossible avec (2.11). Enfin, les distributions de la sous-figure d sont multimodales. Faut-il considérer tous les modes ou au contraire en considérer un particulier? Cette seconde alternative n'est pas possible avec la définition de h .

L'opérateur h n'est pas réellement un indicateur de conflit car il ne considère pas la distribution dans son ensemble mais s'intéresse uniquement au plus haut point de l'intersection. Dans [Zunino et al., 1997d] nous proposons une série d'indicateurs de conflits qui permettent de considérer complètement la forme des distributions.

2.1.4 De nouveaux indices de conflits

Pour considérer entièrement les distributions, il existe plusieurs solutions. Nous avons déjà exploré l'une d'entre elles qui consiste à utiliser la surface sous la fonction d'appartenance (ou la distribution de possibilité). Il en existe au moins deux autres. La première consiste à mesurer la distance existant entre les distributions (ou les fonctions d'appartenance). L'indice de conflit construit de la sorte disposera alors d'intéressantes propriétés comme la vérification de l'inégalité triangulaire. Une autre solution consiste à se limiter à une forme de distribution courante (par exemple les trapèzes) et à calculer des valeurs caractéristiques pour en déterminer le conflit. Cette approche permet d'évaluer très rapidement le conflit sans avoir à recourir à une technique d'intégration ou de distance, souvent coûteuse en temps de calcul. Enfin, le conflit est actuellement un réel. Il est tentant de considérer que le conflit lui même est imprécis et de le calculer de la sorte, le résultat étant par exemple un nombre flou. Les paragraphes suivants développent ces approches.

2.1.4.1 Indices de conflit et distances

Soit d une distance de Ω^2 vers \mathbb{R}^+ et n une fonction non décroissante monotone de d :

$$n : \begin{cases} \mathbb{R} & \rightarrow \mathbb{R}^+ \\ x & \mapsto n(x) \\ n(0) = 1 \text{ et } n(\infty) = 0. \end{cases} \quad (2.12)$$

Il existe de nombreuses fonctions vérifiant 2.12. En voici quelques exemples :

$$\begin{aligned} n(d) &= [1 + d]^{-1} \\ n(d) &= e^{-d} \\ n(d) &= 1 - \frac{d}{d_0} \end{aligned} \quad \text{où } d_0 \text{ s'il existe est défini par } \quad 0 < \sup_{(\pi_1, \pi_2) \in \Omega^2} d(\pi_1, \pi_2) \leq d_0 < \infty \quad (2.13)$$

En pratique et lorsque d_0 existe, la dernière formulation est choisie car la linéarité facilite la démonstration de propriétés, surtout si d est une distance vers $[0, 1]$. Sous ces conditions, l'indice de conflit associé à la distance d peut s'exprimer sous la forme :

$$I_d(\pi_1, \pi_2) = n(h(\pi_1, \pi_2)) \quad (2.14)$$

Construit comme cela, I_d possède de nombreuses propriétés intéressantes comme la robustesse. Intuitivement, étant donné deux couples de distributions (π_1, π_2) et (π_3, π_4) , les deux couples étant proches l'un de l'autre, la robustesse assure que le conflit entre (π_1, π_2) est peu différent du conflit entre (π_3, π_4) .

Propriété 21. *Robustesse.*

Supposons que Ω soit un univers numérique (par exemple \mathbb{R}). Soit ϵ un réel positif et π_1, π_2, π_3 trois distributions de possibilités. Supposons que les deux premières vérifient $d_{Max}(\pi_1, \pi_2) < \epsilon$. Soient les deux réels :

$$\underline{\pi}_{1\alpha} = \inf_{\omega \in \Omega} \{\omega / \pi_1(\omega) \geq \alpha\} \quad (2.15)$$

$$\overline{\pi}_{1\alpha} = \sup_{\omega \in \Omega} \{\omega / \pi_1(\omega) \geq \alpha\} \quad (2.16)$$

La distance d_{Max} appelée distance maximum entre π_1 et π_2 [Fortemps et Roubens, 1996] est définie par :

$$d_{Max}(\pi_1, \pi_2) = \max_{\alpha \in [0,1]} (\max \{|\underline{\pi}_{1\alpha} - \underline{\pi}_{2\alpha}|, |\overline{\pi}_{1\alpha} - \overline{\pi}_{2\alpha}|\}) \quad (2.17)$$

L'indicateur de conflit I_d est dit robuste lorsque :

$$\begin{cases} d_{Max}(\pi_1, \pi_2) < \epsilon \Rightarrow d(\pi_1, \pi_2) < \epsilon \\ |I_d(\pi_1, \pi_3) - I_d(\pi_2, \pi_3)| < \epsilon \end{cases} \quad (2.18)$$

Démonstration. La distance d vérifie l'inégalité triangulaire donc

$$\begin{aligned} d(\pi_2, \pi_3) &\leq d(\pi_2, \pi_1) + d(\pi_1, \pi_3) \\ \Rightarrow \underbrace{d(\pi_2, \pi_3) - d(\pi_1, \pi_3)}_{\alpha} &\leq \underbrace{d(\pi_2, \pi_1)}_{\beta} \end{aligned} \quad (2.19)$$

De même, il vient

$$\underbrace{d(\pi_1, \pi_3) - d(\pi_2, \pi_3)}_{-\alpha} \leq \underbrace{d(\pi_2, \pi_1)}_{\beta} \quad (2.20)$$

Donc $|\alpha| \leq \beta$. Cela implique

$$|d(\pi_2, \pi_3) - d(\pi_1, \pi_3)| < d(\pi_2, \pi_1) \leq \epsilon \quad (2.21)$$

□

Le choix de la distance a une grande importance sur le résultat final et il est impossible de présenter toutes les solutions possibles. Nous nous intéressons ici à la distance de HAMMING généralisée et à la distance de BHATTACHARYYA. La distance de HAMMING, qui n'est pas une métrique pour l'univers considéré, est définie par (lorsque l'intégrale converge) :

$$\begin{aligned} d_{Ham}(\pi_1, \pi_2) &= \frac{\int_{\Omega} |\pi_1(\omega) - \pi_2(\omega)| d\omega}{\delta} \\ \delta &= \max(\overline{\pi}_{10}, \overline{\pi}_{20}) - \min(\underline{\pi}_{10}, \underline{\pi}_{20}) \end{aligned} \quad (2.22)$$

Le résultat de cette distance varie dans l'intervalle $[0, 1]$. Si la fonction $n(d) = 1 - \frac{d}{d_0}$ est utilisée, nous avons alors $d_0 = 1$ et l'indice de conflit devient :

$$I_{d_{Ham}}(\pi_1, \pi_2) = 1 - d_{Ham}(\pi_1, \pi_2) \implies I_{d_{Ham}}(\pi_1, \pi_2) = 1 - \frac{u + v}{p} \quad (2.23)$$

Notre indice de conflit construit à l'aide de la distance de HAMMING se réduit donc à l'indice de similarité de KENDALL.

Lorsque la distance de BHATTACHARYYA [Kailath, 1967] est utilisée (si les intégrales convergent), l'indice de conflit I_{Bat} est défini par :

$$\begin{aligned} P_A(\omega) &= \frac{\pi_1(\omega)}{\int_{\Omega} \pi_1(\omega) d\omega} & R(P_A, P_B) &= \int_{\Omega} \sqrt{P_A(\omega) P_B(\omega)} d\omega & I_{Bat} &= 1 - d_{Bat} \\ P_B(\omega) &= \frac{\pi_2(\omega)}{\int_{\Omega} \pi_2(\omega) d\omega} & d_{Bat}(\pi_1, \pi_2) &= \sqrt{1 - R(P_A, P_B)} \end{aligned} \quad (2.24)$$

2.1.5 Un nouvel indice de conflit : le conflit horizontal

Les techniques d'évaluation du conflit présentées précédemment font appel à des techniques numériques très coûteuses en temps de calcul. Il est parfois intéressant d'avoir une idée du conflit entre deux distributions de manière très rapide, en effectuant très peu d'opérations mathématiques. Cette réduction du coût de calcul n'est possible que si la forme des distributions est connue a priori. Nous proposons [Vienne et al., 1996] une nouvelle évaluation du conflit lorsque les distributions sont des trapèzes. Dans ce paragraphe, π_1 and π_2 seront supposées être deux distributions de possibilités trapézoïdales. Chacune d'elle sera représentée par un quadruplet de réels (a, b, c, d) dont nous supposons qu'ils vérifient $a < b \leq c < d$. Nous avons alors :

$$\pi(\omega) = \begin{cases} \frac{\omega-a}{b-a} & \text{si } a \leq \omega < b, \\ 1 & \text{si } b \leq \omega \leq c, \\ \frac{d-\omega}{d-c} & \text{si } c \leq \omega < d, \\ 0 & \text{sinon.} \end{cases} \quad (2.25)$$

Lorsque π est triangulaire, nous posons $b = c$, ce qui simplifie la définition précédente. L'indication de conflit entre π_1 et π_2 (voir figure 2.5) peut être vue comme la distance maximale entre les noyaux (c'est-à-dire l_1 ou l_2) divisée par un facteur de renormalisation (c'est-à-dire $r_2 + r_3$ ou $r_1 + r_4$). Pour rester compatible avec l'indice h de DUBOIS et PRADE, H_Z est forcé à 1 dès que l'intersection des noyaux n'est pas vide et forcé à 0 dès que l'intersection des supports est vide (ce qui est compatible avec notre définition des indices de conflit 2.8 page 32). La définition de H_Z est alors :

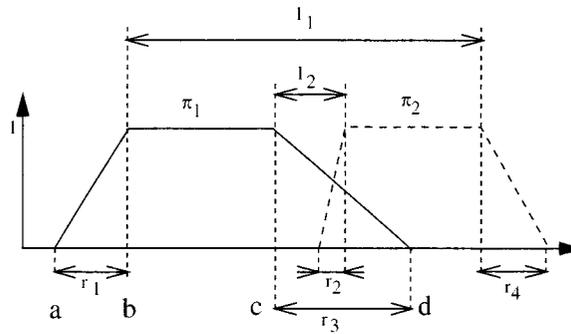


FIG. 2.5 – Représentation graphique de H_Z .

$$\begin{aligned} H_Z &= \max(H_{ZR}, H_{ZL}) & C_{\alpha\beta}(\pi_1, \pi_2) &= \begin{cases} 1 & \text{si Noyau}(\pi_1) \cap \text{Noyau}(\pi_2) \neq \emptyset \\ 0 & \text{si Support}(\pi_1) \cap \text{Support}(\pi_2) = \emptyset \\ 1 - g(\alpha, \beta) & \text{sinon} \end{cases} \\ H_{ZR} &= C_{12}(\pi_1, \pi_2) & \text{avec} & \\ H_{ZL} &= C_{21}(\pi_1, \pi_2) & g(\alpha, \beta) &= \frac{|c_\alpha - b_\beta|}{|d_\beta - c_\beta| + |b_\alpha - a_\alpha|} \end{aligned} \quad (2.26)$$

Défini comme précédemment, H_Z possède une propriété intéressante permettant de le calculer très rapidement. Pour détailler cette propriété, considérons une loi de combinaison externe \oplus entre un réel et une distribution de possibilité π

$$\begin{cases} (\mathbb{R}, \Omega) \rightarrow \Omega \\ (\lambda, \pi) \mapsto \pi_1 = \pi \oplus \lambda = \lambda \oplus \pi \\ \pi_1 = (a + \lambda, b + \lambda, c + \lambda, d + \lambda) \end{cases} \quad (2.27)$$

L'opérateur \oplus est utilisé pour obtenir une nouvelle distribution de possibilité π_2 décalée de π_1 d'un offset λ . Il s'agit d'un opérateur de translation de distribution de possibilités trapézoïdales. Cette opération possède

$\lambda = 0$ comme élément neutre dans \mathbb{R} . Nous pouvons maintenant détailler la propriété de linéarité qui simplifie fortement l'expression de H_Z .

Propriété 22. *Linéarité de H_Z .*

Soit π une distribution de possibilité trapézoïdale $\pi = (a, b, c, d)$ et $\pi_2 = \pi \oplus \lambda$ où λ est un réel vérifiant $|c - b| \leq |\lambda| < |d - a|$. Sous ces conditions, H_Z se réduit à

$$H_Z(\pi, \pi_2) = 1 - \frac{|c - b - |\lambda||}{|d - c| + |b - a|}$$

Démonstration. L'intersection des supports de π et π_2 n'est pas vide car λ vérifie $|c - b| \leq |\lambda| < |d - a|$.

▷ Lorsque $\lambda = |c - b|$, l'indicateur de conflit $H_Z(\pi, \pi_2) = 1$ ce qui est bien le résultat escompté.

▷ Dans les autres cas, (2.26) devient :

$$H_Z(\pi, \pi_2) = \max\left(1 - \frac{|c - b - \lambda|}{|d - c| + |b - a|}, 1 - \frac{|c - b + \lambda|}{|d - c| + |b - a|}\right)$$

Le réel $c - b$ étant positif :

$$H_Z(\pi, \pi_2) = 1 - \frac{|c - b - |\lambda||}{|d - c| + |b - a|}$$

□

Il est ainsi très facile et peu coûteux en opérations d'évaluer rapidement le conflit entre deux distributions de possibilités trapézoïdales.

Actuellement, tous les auteurs définissent et utilisent un indice de conflit qui est un réel. Cependant, il peut être intéressant de considérer l'indice de conflit comme une quantité imparfaite. Le conflit peut alors être vu comme un nombre flou. Ce nombre flou peut alors être utilisé tel quel ou devenir un indice de conflit tout à fait classique, après défuzification. Nous explorons cette voie nouvelle et proposons maintenant une définition d'un "indice de conflit flou".

2.1.6 Une nouvelle vision du conflit : l'indicateur de conflit flou

Tous les indicateurs de conflit que nous avons présentés sur des distributions de possibilités de Ω vers une partie de \mathbb{R}^+ . Parfois, une seule méthode de calcul du conflit ne suffit pas. Certains indicateurs sont trop pessimistes lorsque le conflit grandit, mais sont adaptés lorsque le conflit reste faible. D'autres sont un peu trop optimistes et l'idée de combiner judicieusement plusieurs indices pour en former un nouveau, meilleur que les autres, est attractive³. Il y a plusieurs moyens de faire cette combinaison. Tout d'abord, il est possible de combiner les réels issus de chaque indice pour en faire un autre réel. Il est alors difficile d'extraire les indices de conflits individuels qui ont servi à construire l'indice global. Une autre solution consiste à utiliser les résultats des divers indices pour former un nombre flou. Il est ainsi plus facile de traduire que le conflit est mal connu. Nous proposons donc un nouvel indice de conflit construit à partir de trois indices de conflit "classiques" et dont le résultat est un nombre flou.

Le résultat de cet indice peut être utilisé comme un nombre flou, dans les algorithmes qui le permettent, ou alors défuzifié pour fournir un réel utilisable comme tous les autres indices de conflit que nous avons présentés.

La définition de l'indice de conflit flou nécessite trois indices de conflits I_1, I_2, I_3 définis pour toutes distributions sur Ω . De plus, I_1 doit être plus pessimiste que I_2 qui doit être plus pessimiste que I_3 . Cela signifie que pour tout couple de distribution (π_1, π_2) :

$$I_1(\pi_1, \pi_2) \leq I_2(\pi_1, \pi_2) \leq I_3(\pi_1, \pi_2) \quad (2.28)$$

3. Cela revient finalement à fusionner les indices de conflit !

L'indicateur de conflit flou I_F est défini comme la combinaison des trois indicateurs I_1, I_2, I_3 . Pour tout couple de distributions (π_1, π_2) , I_F est donné par :

$$I_F(x) = \begin{cases} 1 & \text{si } \omega = I_2(\pi_1, \pi_2) \\ \frac{\omega - I_1}{I_2 - I_1} & \text{si } I_1 \leq \omega < I_2 \\ \frac{\omega - I_3}{I_2 - I_3} & \text{si } I_2 < \omega \leq I_3 \\ 0 & \text{sinon} \end{cases} \quad (2.29)$$

L'indice de conflit I_2 est celui qui est considéré comme étant le plus important dans l'association. C'est l'indice qui fournit l'évaluation la plus juste. Cette évaluation est ensuite complétée par les deux autres indices respectivement plus pessimistes et plus optimistes que I_2 . Les indices I_1, I_2 et I_3 peuvent par exemple être choisis parmi les mesures de similarité : $I_1 = \text{SOKAL-SNEATH}$, $I_2 = \text{JACCARD}$ et $I_3 = \text{KULZINSKY}$.

Le paragraphe suivant montre le comportement de ces indices de conflits de manière graphique.

2.2 Exemples

Le comportement des nouveaux indices de conflit présentés précédemment est étudié plus en détail dans ce paragraphe. Nous considérons les distributions de la figure 2.6 et de la figure 2.7 qui constituent les deux couples de distributions de test. Ces deux couples sont nommées par la suite "1" et "2". Nous procédons d'une manière similaire à celle déjà expliquée pour obtenir le comportement d'un opérateur de fusion de la disjonction à la conjonction. Le conflit entre π_1 et π_2 est calculé puis la distribution π_1 est décalée d'un réel δ vers la droite et le processus recommence jusqu'à ce que π_1 ait "traversé" complètement π_2 , c'est-à-dire que la distribution π_1 aie son support en $[4, 12]$.

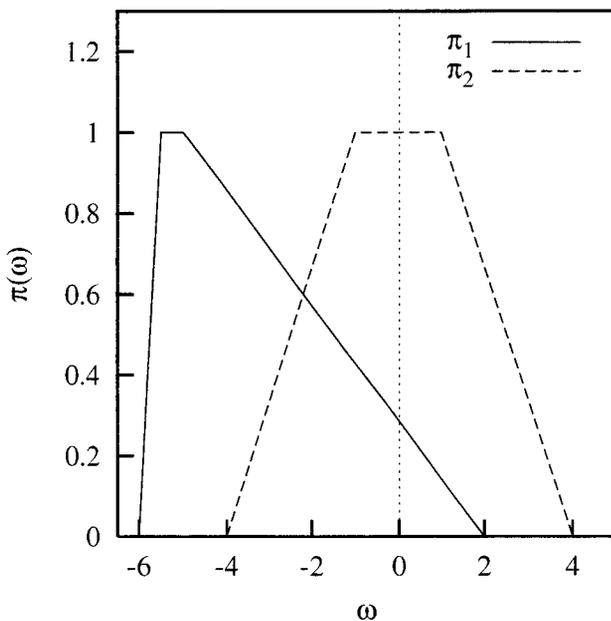


FIG. 2.6 – Distributions de test n° 1.

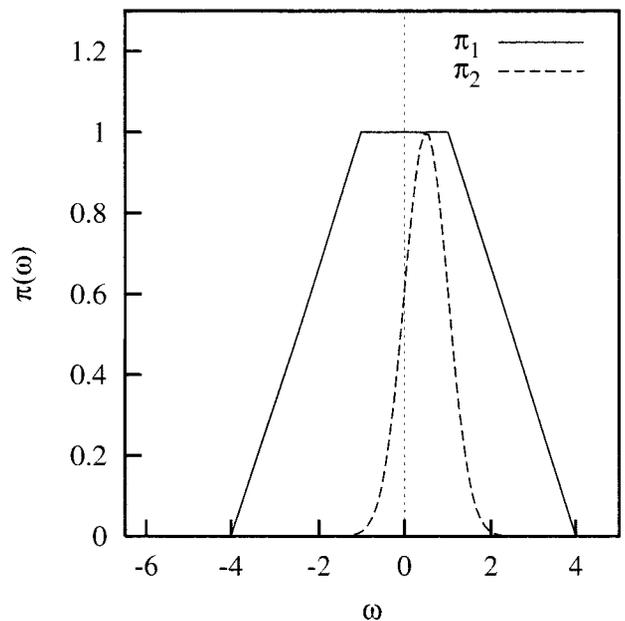


FIG. 2.7 – Distributions de test n° 2.

À chaque étape, un point de coordonnées $(\delta, I(\pi_1, \pi_2))$ est tracé sur un graphe. Les résultats sont présentés sur les figures 2.8 à 2.13. Pour faciliter la lecture, les indices provenant de l'analyse de données ont été regroupés sur les figures 2.8 et 2.9 tandis que les nouveaux indices tels que l'indice horizontal H_Z , les indices à base de distance et l'indice flou defuzifié à l'aide de la méthode du centre de gravité sont représentés figures 2.10 et 2.11.

Nous remarquons que les indices issus de l'analyse de données sont cohérents avec l'idée intuitive d'un indice de conflit : dès que les distributions n'ont aucun point en commun, l'indice de conflit est égal à 0 et il

4. Sauf l'indice de KENDALL à 0 qui ne vérifie pas cette règle.

est maximal lorsque les distributions s'accordent. Notons aussi que sur les figure 2.8 et 2.9 les indications de conflits sont maximales toutes en même temps, à une valeur $\delta = 3.45$, c'est-à-dire légèrement inférieure à la valeur minimale de δ pour laquelle l'intersection des noyaux des distributions n'est pas vide.

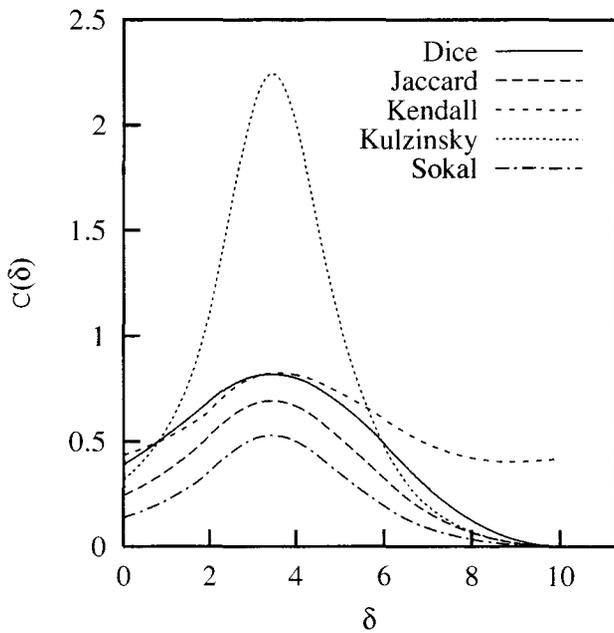


FIG. 2.8 – Indices de similarité (n° 1).

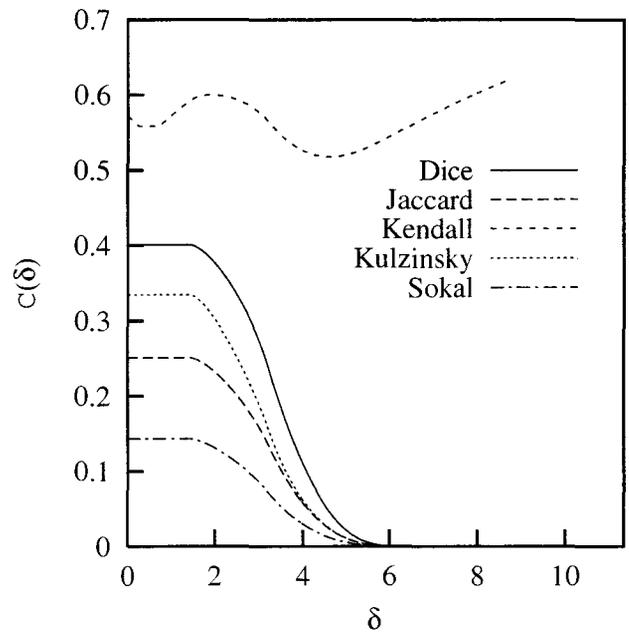


FIG. 2.9 – Indices de similarité (n° 2).

Les nouveaux indices sont présentés sur les figures ci-dessous. Nous constatons que l'indice horizontal H_Z est à 1 tant que les noyaux des distributions ont une zone en commun c'est-à-dire dans l'intervalle $[4, 7.5]$ pour le jeu de test n° 1. Par ailleurs, nous retrouvons aussi la linéarité de H_Z démontrée précédemment. D'une manière générale, peu d'indices de conflit varient entre $[0, 1]$ sauf h (non représenté ici) et H_Z (défini uniquement pour des trapèzes). Ces indices de conflits (hormis h et H_Z dont le comportement est proche de h) ne peuvent donc pas servir directement de facteur de normalisation. Ils indiquent seulement le degré avec lesquelles les sources s'accordent. Intuitivement, deux sources s'accordent uniquement lorsqu'elles sont identiques. Les figures ci-dessus représentent l'indice de conflit flou non défuzifié. Pour obtenir ces courbes,

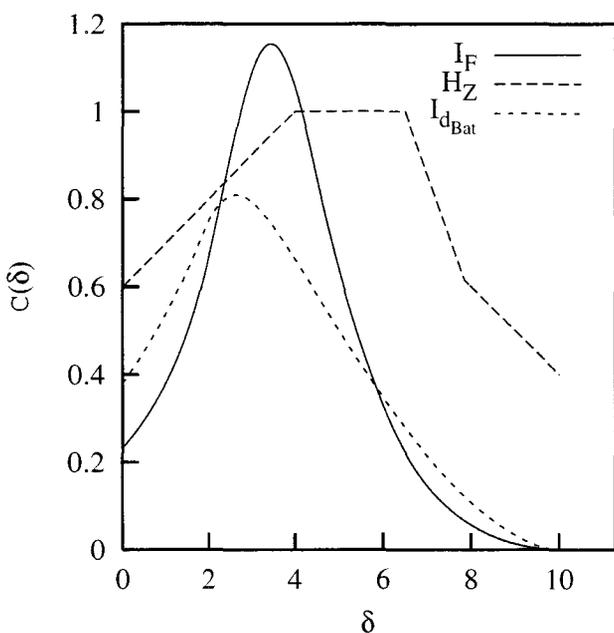


FIG. 2.10 – Nouveaux indicateurs (n° 1).

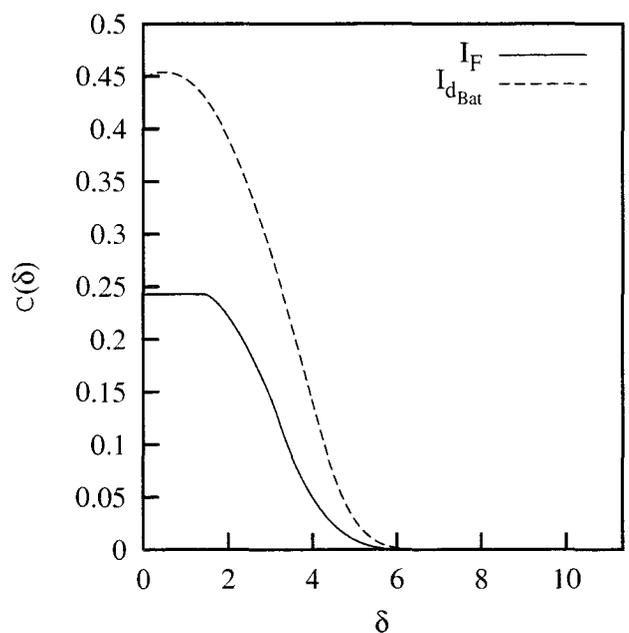
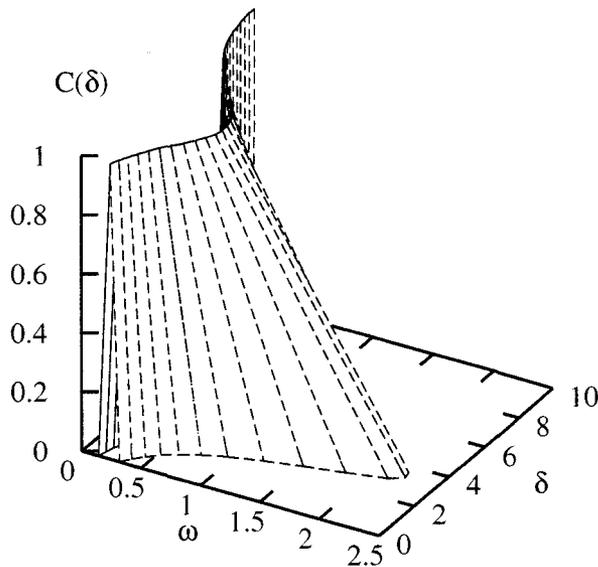
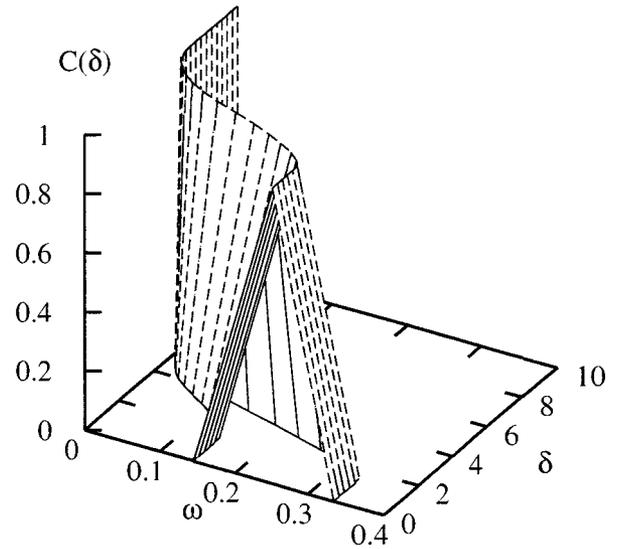


FIG. 2.11 – Nouveaux indicateurs (n° 2).

FIG. 2.12 – *Indice flou (n° 1).*FIG. 2.13 – *Indice flou (n° 2).*

nous avons procédé de la même manière que précédemment, mais à chaque étape, nous avons dessiné le nombre flou en totalité. Ainsi, l'axe des abscisses représente le support du nombre flou, l'axe des ordonnées porte le décalage δ par rapport à la position d'origine et l'axe des cotes porte le conflit $C(\delta)$.

Le paragraphe suivant s'intéresse aux opérateurs utilisant un indice de conflit pour déterminer automatiquement le mode de combinaison à utiliser. Ces opérateurs ont – pour la plupart – été définis par DUBOIS et PRADE, notamment dans [Dubois et Prade, 1992].

2.3 Les opérateurs CD

Le conflit entre les sources étant connu, le choix du mode de combinaison par l'opérateur semble possible. Une méthode naïve consisterait à effectuer la fusion en mode conjonctif lorsque l'indice de conflit est plus grand qu'un seuil ϵ et en mode disjonctif dans le cas contraire. Dans cette approche, le choix du seuil de discrimination ϵ est alors difficile et l'information fournie par l'indice de conflit n'est pas entièrement utilisée du fait du seuillage. Une idée plus intéressante consiste à créer un opérateur capable de passer de la conjonction à la disjonction de manière continue, en fonction de l'indice de conflit. De tels opérateurs ont été proposés dans [Dubois et Prade, 1994a] et étudiés par de nombreux auteurs tels que [Deveughele, 1993], [Oussalah, 1998], [Zunino et al., 1997b]. Les paragraphes suivants présentent la règle de DUBOIS et PRADE et quelques modifications.

2.3.1 La règle adaptative de DUBOIS et PRADE

DUBOIS et PRADE utilisent l'indicateur de conflit h (voir (2.11)) qui varie dans $[0, 1]$. Lorsque $h(\pi_1, \pi_2) = 1$, les sources sont considérées en accord et le mode conjonctif est utilisé. Sinon, il sera considéré que l'élément ω a un degré de possibilité $1 - h(\pi_1, \pi_2)$ d'être dans l'intersection des distributions. Il s'agit de la première règle de DUBOIS et PRADE dont l'expression est rappelée ci-dessous :

$$\begin{aligned}
 \pi_R(\omega) &= \max \left(\frac{\min(\pi_1(\omega), \pi_2(\omega))}{h(\pi_1, \pi_2)}, \min(1 - h(\pi_1, \pi_2), \max(\pi_1(\omega), \pi_2(\omega))) \right) \\
 &= \max \left(\frac{\pi_{Conj}(\omega)}{h(\pi_1, \pi_2)}, \min(1 - h, \pi_{Disj}(\omega)) \right) \\
 &= \max (\text{Terme conjonctif}(\omega), \text{Terme disjonctif}(\omega))
 \end{aligned} \tag{2.30}$$

Cette règle calcule tout d'abord l'intersection π_{Conj} et l'union π_{Disj} des deux sources. L'importance de l'union entre les sources est minorée par le $\min(1 - h, \pi_{Disj}(\omega))$ alors que le terme d'intersection est systématiquement privilégié du fait de la division par $h \leq 1$. Finalement, l'union de ce que l'on peut appeler un terme conjonctif et un terme disjonctif est calculée. Cet opérateur a donc un comportement prudent.

La distribution résultante est normalisée par construction, et son niveau de fiabilité est $h(\pi_1, \pi_2)$, c'est-à-dire la hauteur maximale de l'intersection des distributions. La zone commune aux deux distributions est toujours privilégiée lorsque les sources sont en accord. Lorsque $h = 1$, la règle conjonctive min est retrouvée. Lorsque $h = 0$ la disjonction max est obtenue. Cette règle possède quelques autres propriétés intéressantes auxquelles nous nous intéressons maintenant.

Propriété 23. *Commutativité et associativité.*

La règle (2.30) est commutative mais non associative.

Démonstration. La règle (2.30) est la combinaison de fonction commutatives par des fonctions commutatives. Elle est donc commutative. Il suffit de considérer un cas particulier comme les trois distributions de la figure 3.2 page 51 et de constater que $\pi_R(\pi_1, \pi_R(\pi_2, \pi_3)) \neq \pi_R(\pi_R(\pi_1, \pi_2), \pi_3)$. \square

Propriété 24. *Autodualité.*

La règle (2.30) est autoduale pour $h = 0$.

Démonstration. Le calcul de $\pi_R(\pi_1, \pi_2)(\omega) + \pi_R(1 - \pi_1, 1 - \pi_2)(\omega)$ conduit à la valeur 1 pour tout ω ce qui prouve l'auto-dualité. \square

Propriété 25. *Monotonie.*

Pour h constant, la règle (2.30) est monotone.

Démonstration. Soient $\pi_1, \pi_2, \pi_3, \pi_4$ quatre distributions de possibilités, $\pi = \pi_R(\pi_1, \pi_2)$ et $\pi' = \pi_R(\pi_3, \pi_4)$. A l'aide de calculs sur les inégalités, il est possible de montrer l'implication suivante qui prouve la monotonie :

$$\forall \omega \left\{ \begin{array}{l} \pi_3(\omega) \leq \pi_1(\omega) \\ \pi_4(\omega) \leq \pi_2(\omega) \\ h(\pi_3, \pi_4) = h(\pi_1, \pi_2) \end{array} \right. \implies \pi' \leq \pi \quad (2.31)$$

\square

Propriété 26. *Ignorance, préservation du zéro et plausibilité maximale.*

L'impossibilité est un cas limite pour lequel aucune information n'est disponible. La distribution de possibilité indiquant l'impossibilité est $\pi_{Imp}(\omega) = 0$. L'ignorance est un autre cas limite dans lequel tous les événements de l'univers sont possibles. La distribution de possibilité indiquant l'ignorance est $\pi_{Ign}(\omega) = 1, \forall \omega$.

La combinaison d'une distribution quelconque avec π_{Imp} ne change pas le résultat donné par (2.30) (si cette dernière est prolongée par continuité pour $h = 0$ avec (2.32)).

Démonstration. Pour toute distribution π_1 , le calcul de $\pi_R(\pi_1, \pi_{Imp})$ conduit à $h = 0$. Dans ce cas, nous avons $\pi_R(\pi_1, \pi_{Imp}) = \max(\pi_1, \pi_{Imp}) = \pi_1$. \square

La combinaison d'une distribution quelconque avec π_{Ign} ne change pas le résultat donné par (2.30).

Démonstration. Pour toute distribution π_1 , le calcul de $\pi_R(\pi_1, \pi_{Ign})$ conduit à $h = 1$. Dans ce cas, nous avons $\pi_R(\pi_1, \pi_{Ign}) = \max(\min(\pi_1, 1), \min(\max(1, \pi_1)), 0) = \pi_1$. \square

La règle (2.30) préserve la plausibilité maximale car si un ω est considéré comme possible dans une des sources, il est aussi considéré comme possible dans la combinaison.

Propriété 27. Idempotence.

La règle (2.30) est idempotente.

Démonstration. Le calcul de $\pi_R(\pi_1, \pi_1)$ conduit à évaluer $\max(\min(\pi_1, \pi_1), \min(0, \max(\pi_1, \pi_1))) = \pi_1$. La règle est donc idempotente. \square

Propriété 28. Convexité.

Le résultat de la règle (2.30) n'est en général pas convexe au sens de ZADEH. Selon lui, un ensemble flou est dit convexe si toutes ses α -coupes sont convexes. Pour rétablir la convexité, [Oussalah, 1998] propose plusieurs solutions.

Démonstration. L'union de deux ensembles flous convexes n'est en général pas convexe, contrairement à l'intersection qui fournit toujours un ensemble convexe. La figure 2.14 donne de nombreux contre-exemples montrant que le résultat de (2.30) n'est en général pas convexe. \square

Propriété 29. Continuité.

La règle (2.30) est de classe C^0 sauf pour $h = 0$. Lorsque $h = 0$, la règle (2.30) n'est pas définie.

Démonstration. L'équation (2.30) est la combinaison de fonctions de classe C^0 pour $h \neq 0$. \square

Une solution pour définir la règle de DUBOIS et PRADE lorsque $h = 0$ [Oussalah, 1998] consiste à remarquer que dans ce cas, la règle doit opérer en mode disjonctif. Il est alors possible de poser (valable lorsque $h = 0$):

$$\pi_R(\pi_1, \pi_2)(\omega) = \begin{cases} \max\left(\frac{\min(\pi_1(\omega), \pi_2(\omega))}{h(\pi_1, \pi_2)}, \min(1 - h(\pi_1, \pi_2), \max(\pi_1(\omega), \pi_2(\omega)))\right) & \text{si } h \neq 0 \\ \max(\pi_1(\omega), \pi_2(\omega)) & \text{sinon} \end{cases} \quad (2.32)$$

La figure 2.14 présente le comportement de l'opérateur de DUBOIS et PRADE de la conjonction à la disjonction. Les discontinuités lorsque h est faible sont nettement visible, pour δ proche de 0.

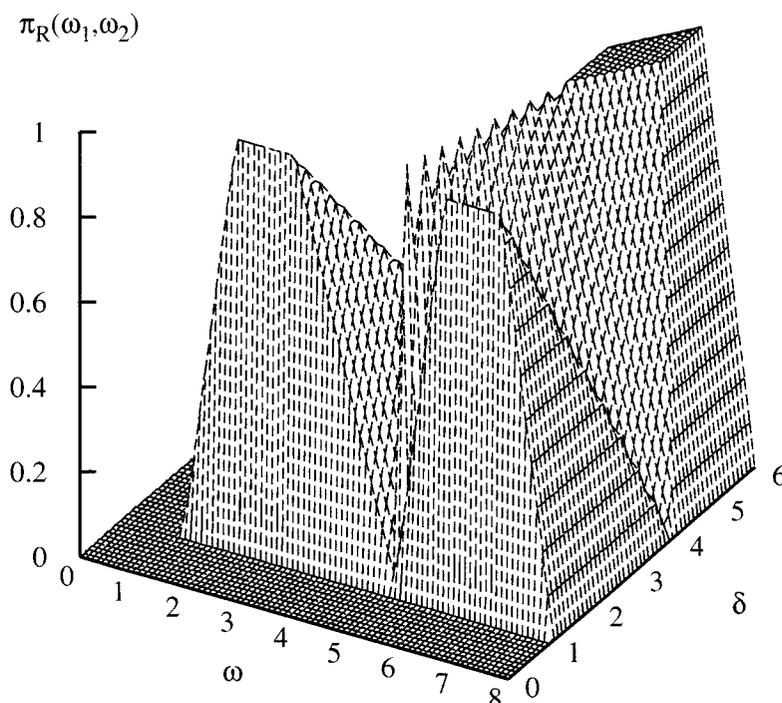


FIG. 2.14 – De la disjonction à la conjonction pour π_R .

La règle de DUBOIS et PRADE est non continue et certains auteurs ont tenté de la modifier de manière à limiter la discontinuité. Nous présentons rapidement une modification de la règle de DUBOIS et PRADE proposée par [Deveughele et Dubuisson, 1996].

2.3.2 La règle “saturée” de [Deveughele et Dubuisson, 1996]

Pour réduire la discontinuité, [Deveughele et Dubuisson, 1996] remarquent que le problème provient de la renormalisation pour $h = 0$ du terme $\pi_{Conj}(\omega)$. D’après eux, il suffit alors d’introduire un paramètre $s \in [0, 1]$ dit “seuil de saturation” de manière à ce qu’en-dessous de cette valeur s , la renormalisation ne s’effectue plus. Le terme conjonctif de l’équation (2.30) modifié devient alors :

$$\frac{\pi_{Conj}(\omega)}{h(\pi_1, \pi_2)} \text{ est remplacé par } \pi_{DEV} = \frac{\pi_{Conj}(\omega)}{\max(s, h(\pi_1, \pi_2))} \quad (2.33)$$

Lorsque s n’est pas nul, la discontinuité disparaît. Cependant, le terme conjonctif π_{DEV} n’est plus normalisé pour $h(\pi_1, \pi_2) < s$. Cela a donc les conséquences suivantes :

- ▷ Le “déficit” de normalisation de π_{DEV} est alors de $\frac{1 - h(\pi_1, \pi_2)}{\max(s, h(\pi_1, \pi_2))}$,
- ▷ La hauteur maximale de π_{DEV} est alors de $\rho = \frac{h(\pi_1, \pi_2)}{\max(s, h(\pi_1, \pi_2))}$

Il est impossible de diviser par ρ car cela conduirait à multiplier par ρ^{-1} et donc à diviser de nouveau par $h(\pi_1, \pi_2)$. L’idée est alors de reporter ce facteur de normalisation sur le terme disjonctif, en augmentant la valeur de $1 - h(\pi_1, \pi_2)$ dans (2.30). Le calcul du facteur de normalisation se fait en remarquant que :

$$\begin{aligned} &\text{puisque } \frac{1 - \max(s, h(\pi_1, \pi_2))}{1 - s} \text{ vaut } 1 \text{ si } h(\pi_1, \pi_2) < s \\ &\text{alors } \min\left(\frac{1 - \max(s, h(\pi_1, \pi_2))}{1 - s}, \max(\pi_1(\omega), \pi_2(\omega))\right) \text{ est toujours normalisé} \end{aligned} \quad (2.34)$$

La règle saturée de [Deveughele et Dubuisson, 1996] devient alors :

$$\pi_R(\omega) = \max\left(\frac{\min(\pi_1(\omega), \pi_2(\omega))}{\max(s, h(\pi_1, \pi_2))}, \min\left(\frac{1 - \max(s, h(\pi_1, \pi_2))}{1 - s}, \max(\pi_1(\omega), \pi_2(\omega))\right)\right) \quad (2.35)$$

En pratique, cette modification n’est pas très convaincante car la détermination automatique du paramètre s s’avère plus que hasardeuse. Les figures 2.15 et 2.16 (dont l’orientation a été modifiée pour faciliter la lecture) montrent le résultat de la fusion des distributions figure 1.5 page 21 avec cet opérateur pour deux valeurs de s . Le paramètre s agit donc bien comme un seuil, en-deçà duquel la règle saturée à un comportement purement disjonctif.

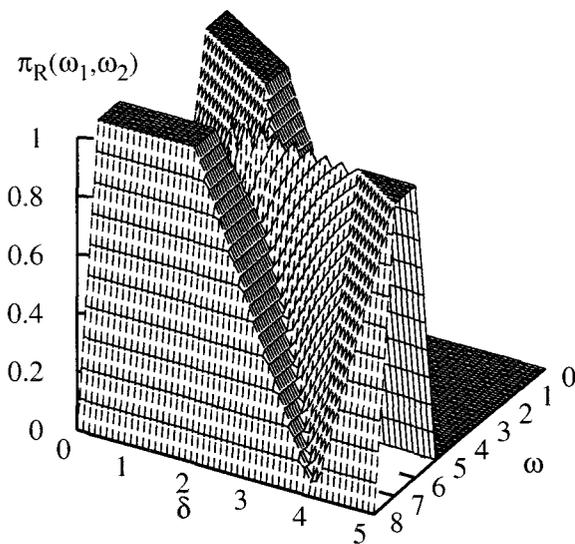


FIG. 2.15 – Fusion saturée ($s = 0.2$).

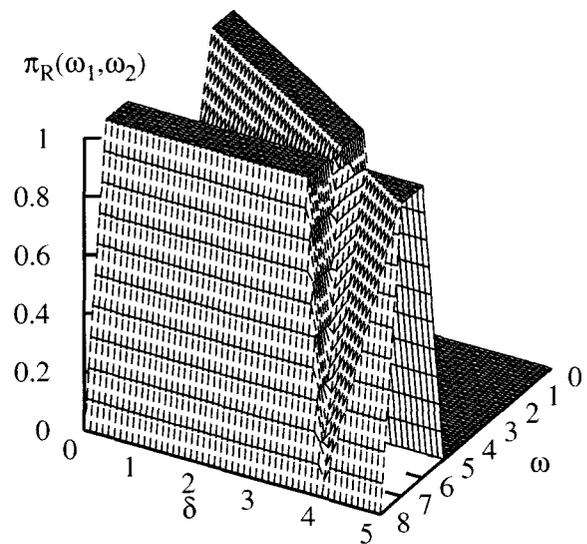


FIG. 2.16 – Fusion saturée ($s = 0.8$).

La règle de fusion adaptative de [DUBOIS] et [PRADE] repose sur une fusion logique car elle utilise l’union et l’intersection. Selon DUBOIS et PRADE, il est aussi possible de définir une règle de fusion adaptative arithmétique que nous présentons maintenant.

2.3.3 La règle de fusion linéaire de DUBOIS et PRADE

L'idée de cette règle est d'utiliser une combinaison linéaire de deux modes de fusion : la conjonction et la disjonction. Il reste alors à déterminer les coefficients pour obtenir un résultat normalisé. La règle est donc la suivante :

$$\pi_{Lin}(\pi_1, \pi_2)(\omega) = No(\min(\pi_1(\omega), \pi_2(\omega)) + (1 - h(\pi_1, \pi_2)) \max(\pi_1(\omega), \pi_2(\omega))) \quad (2.36)$$

La fonction No étant un opérateur de normalisation (voir paragraphe 1.4.1 page 15). Cette règle est linéaire mais la variation des degrés de possibilités ne l'est pas. L'intersection est privilégiée par rapport à l'union car l'intersection est associée à un coefficient 1 alors que l'union est associée à un coefficient $1 - h$. Une autre formulation permet d'obtenir une transition entre l'union et l'intersection pour $h = 0.5$ exactement :

$$\pi_{Lin}(\pi_1, \pi_2)(\omega) = No(h(\pi_1, \pi_2) \min(\pi_1(\omega), \pi_2(\omega)) + (1 - h(\pi_1, \pi_2)) \max(\pi_1(\omega), \pi_2(\omega))) \quad (2.37)$$

Les figures 2.17 et 2.18 présentent les résultats des deux règles précédentes. Sur ces figures, le résultat de la fusion n'a volontairement pas été normalisé pour bien montrer où le résultat est sous-normal.

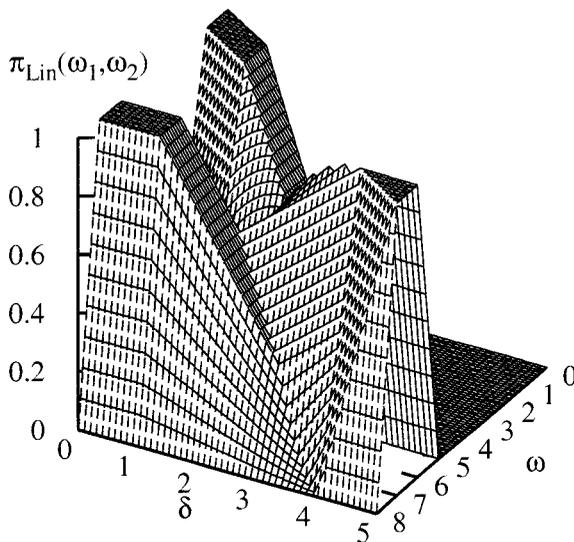


FIG. 2.17 – Fusion linéaire (avec (2.36)).

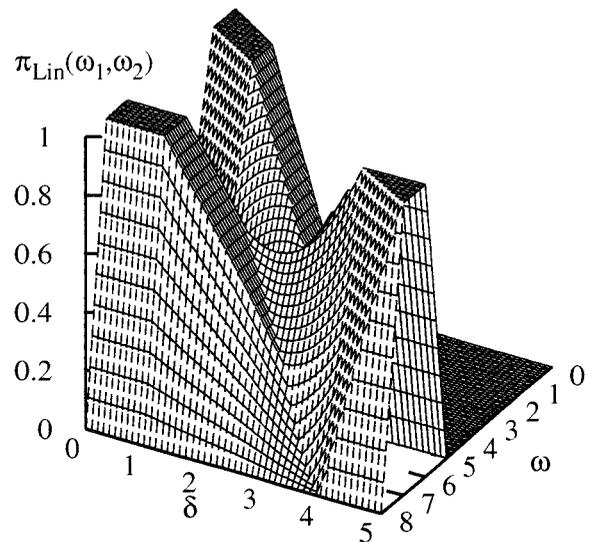


FIG. 2.18 – Fusion linéaire (avec (2.37)).

Les règles adaptatives sont donc en général construites en combinant le résultat de l'intersection et de l'union pondérés par un coefficient modulant la prépondérance d'un résultat sur l'autre. Cependant, la fusion peut opérer dans trois modes distincts : la conjonction, la disjonction et le compromis. Dans les paragraphes suivants, nous proposons d'étudier l'introduction du compromis dans les règles de fusion adaptatives et d'étudier l'effet de cette introduction.

2.4 Nouvelles règles

La règle adaptative de DUBOIS et PRADE donne entière satisfaction dans de nombreuses applications cependant, il est possible de se demander pourquoi cette règle est construite autour de l'intersection et de l'union, en ignorant le troisième grand mode de combinaison : le compromis. Dans les paragraphes suivants nous introduisons le compromis. Par ailleurs, il est intéressant d'observer l'impact de l'utilisation d'un nouvel indice de conflit dans les règles de fusion adaptatives. Enfin, lorsque $h = 0$, la règle 2.30 n'est pas continue et lorsque h est proche de 0, les résultats numériques obtenus sont quelques peu arbitraires. Nous imposons à nos règles d'être continues pour supprimer cet inconvénient.

Il existe deux grands types de règles adaptatives : les règles logiques basées sur l'intersection et l'union et les règles arithmétiques dans lesquelles les degrés de possibilités sont combinés à la manière de nombres réels, à l'aide d'opération classiques comme la somme, la différence, le produit ... Nous nous intéressons tout d'abord à quelques nouvelles règles logiques avant d'aborder le cas des règles arithmétiques.

2.4.1 Une nouvelle règle logique

Nous nous intéressons à la construction d'une nouvelle règle logique. Nous désirons faire intervenir un terme de compromis et utiliser un nouvel indice de conflit prenant en compte la totalité des distributions. Dans un premier temps, supposons un instant que l'opérateur n'ait à opérer que dans un unique mode précis : la conjonction, le compromis ou la disjonction. Les opérateurs logiques les plus populaires pour fusionner dans ces modes sont rappelés dans le tableau 2.3. Notre règle doit pouvoir passer d'un mode à l'autre de

Conjonction	min(ω_1, ω_2)
Compromis	$\sqrt{\omega_1 \omega_2}$
Disjonction	max(ω_1, ω_2)

TAB. 2.3 – Les opérateurs dans chacun des trois modes de fusion.

manière continue. Une idée qui vient à l'esprit consiste à prendre l'union des trois modes et à renormaliser la distribution résultante, ce qui revient à considérer que les résultats fournis par les trois modes de fusion ne sont pas concordants. Cependant, cela est insuffisant car l'union étant la plus grande des opérations, l'influence de l'opérateur de conjonction et de compromis ne seront pas visibles. Il est donc nécessaire d'affecter des poids à chacun des résultats fournis par les opérateurs du 2.3.

Le mode à privilégier est déterminé à l'aide de l'indice de conflit. Des entrées qui s'accordent rendront le mode conjonctif plus prépondérant, des entrées en conflit relatif augmenteront l'importance du mode compromis et des entrées divergentes accentueront l'effet du mode disjonctif. Nous choisissons l'indice de conflit fondé sur la distance de BHATTACHARYYA (voir paragraphe 2.24 page 34). Cet indice de conflit présente l'avantage de considérer les distributions dans leur totalité et par ailleurs, son résultat évolue dans $[0, 1]$, ce qui facilite son utilisation pour manipuler des degrés de possibilités. Tout autre indice de conflit présenté avant peut cependant être utilisé. Par la suite, $I_{d_{Bat}}$ désignera le conflit entre les sources π_1 et π_2 .

L'importance du terme conjonctif sur le résultat doit croître avec le conflit (plus l'indice de conflit est grand, plus les source s'accordent dans notre convention). Nous lui affectons donc le poids $I_{d_{Bat}}$. L'importance du mode disjonctif est inversement proportionnelle à celle du monde conjonctif et nous affectons donc le coefficient $1 - I_{d_{Bat}}$. Il s'agit là des choix classiques déjà réalisés par DUBOIS et PRADE dans leurs règles adaptatives. Reste à déterminer le poids affecté à l'opérateur de compromis. Le compromis doit s'activer lorsque les distributions ne sont ni en total conflit ni en total accord. Une solution est d'évaluer la médiane entre $I_{d_{Bat}}$, $1 - I_{d_{Bat}}$ et $\min(\pi_1(\omega), \pi_2(\omega))$. Les poids choisis sont rappelés dans le tableau 2.4 :

Conjonction	$I_{d_{Bat}}$
Compromis	$\gamma = \text{med}(I_{d_{Bat}}, 1 - I_{d_{Bat}}, \min(\pi_1(\omega), \pi_2(\omega)))$
Disjonction	$1 - I_{d_{Bat}}$

TAB. 2.4 – Les poids pour chacun des trois modes de fusion.

Une fois les poids choisis, il reste encore à déterminer comment combiner entre eux les résultats des trois modes pour obtenir la distribution résultat. Là encore, plusieurs possibilités s'offrent à nous. Tout d'abord, nous pouvons choisir le mode ayant l'importance la plus grande. La règle obtenue est alors :

$$\begin{aligned} \pi_Z(\pi_1, \pi_2)(\omega) = No(\max(I_{d_{Bat}} \min(\pi_1(\omega), \pi_2(\omega)), \\ \gamma \sqrt{\pi_1(\omega) \pi_2(\omega)}, \\ (1 - I_{d_{Bat}}) \max(\pi_1(\omega), \pi_2(\omega)))) \end{aligned} \quad (2.38)$$

No étant un opérateur de normalisation⁵. Cette règle permettra donc de passer de manière continue (mais très brutalement) d'un mode de combinaison à l'autre. Elle montre comment il est possible de construire un opérateur qui choisit automatiquement le mode de combinaison le plus approprié, les autres modes n'intervenant

5. Nous avons utilisé $No(\omega) = \frac{\omega}{\max_{\tau \in \Omega} \pi_Z(\tau)}$.

alors pas dans le résultat. Il s'agit donc d'une solution au problème présenté dans le premier chapitre : le choix automatique du mode dans lequel la combinaison doit se faire.

La figure 2.19 présente les résultats fournis par cette règle de combinaison. Nous remarquons le caractère brutal de la transition entre les modes. Il est possible de distinguer le mode disjonctif (le plus prépondérant) et le mode compromis transition au mode conjonctif. Cette reformulation de la règle ne donne pas entière

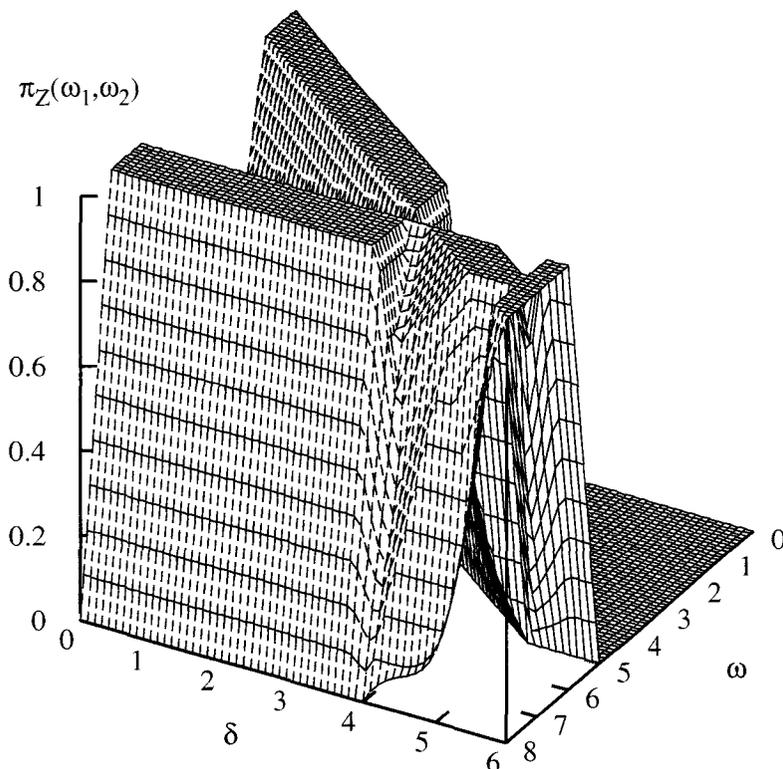


FIG. 2.19 – De la disjonction à la conjonction pour π_Z .

satisfaction en ce qui concerne l'inclusion du mode disjonctif. En effet, le conflit doit être très élevé pour faire basculer la règle dans le mode compromis ou conjonction. L'introduction du compromis dans la règle logique ne fournit pas réellement d'amélioration dans le résultat. Nous étudions maintenant l'introduction du compromis dans la règle arithmétique.

2.4.2 Une nouvelle règle arithmétique

L'introduction du compromis peut aussi s'envisager dans les règles linéaires. La construction est réalisée de la même manière que précédemment. L'opérateur effectuera la somme des résultats de la combinaison dans les trois modes de base (conjonction, disjonctions et compromis), chaque résultat étant pondéré par un coefficient [Zunino et al., 1997b] :

$$\pi_{ZLin}(\pi_1, \pi_2)(\omega) = \alpha \min(\pi_1(\omega), \pi_2(\omega)) + \beta \max(\pi_1(\omega), \pi_2(\omega)) + \gamma \sqrt{\pi_1(\omega)\pi_2(\omega)} \quad (2.39)$$

La détermination des coefficients α , β et γ est faite en considérant le conflit entre les sources, d'une manière analogue à celle choisie au paragraphe 2.4.1. L'indice de conflit retenu est aussi la distance de BHATTACHARYYA. L'intersection, résultat de la fusion en mode conjonctif aura d'autant plus d'importance sur le résultat final que le conflit sera faible. Le coefficient $\alpha = I_{d_{Bat}}$ sera donc un choix raisonnable. Par dualité, l'importance de résultat disjonctif sera donc $\beta = 1 - I_{d_{Bat}}$. Enfin, le mode compromis doit contribuer au résultat lorsque l'accord entre les distributions est faible : $\gamma = \text{med}(I_{d_{Bat}}, 1 - I_{d_{Bat}}, \min(\pi_1(\omega), \pi_2(\omega)))$.

L'opérateur ainsi construit est donc :

$$\begin{aligned} \pi_{ZLin}(\pi_1, \pi_2)(\omega) = & No(I_{d_{Bat}} \min(\pi_1(\omega), \pi_2(\omega)) + \\ & (1 - I_{d_{Bat}}) \max(\pi_1(\omega), \pi_2(\omega)) + \\ & \text{med}(I_{d_{Bat}}, 1 - I_{d_{Bat}}, \min(\pi_1(\omega), \pi_2(\omega))) \sqrt{\pi_1(\omega)\pi_2(\omega)}) \end{aligned} \quad (2.40)$$

Le résultat de la fusion par une telle règle est fournie figure 2.20. L'effet de la médiane sur le terme de compromis est identique à celui observé pour la règle logique. Il y a transition brutale (mais continue) entre les trois modes de base de la combinaison. Cette règle permet aussi de répondre au problème du choix du

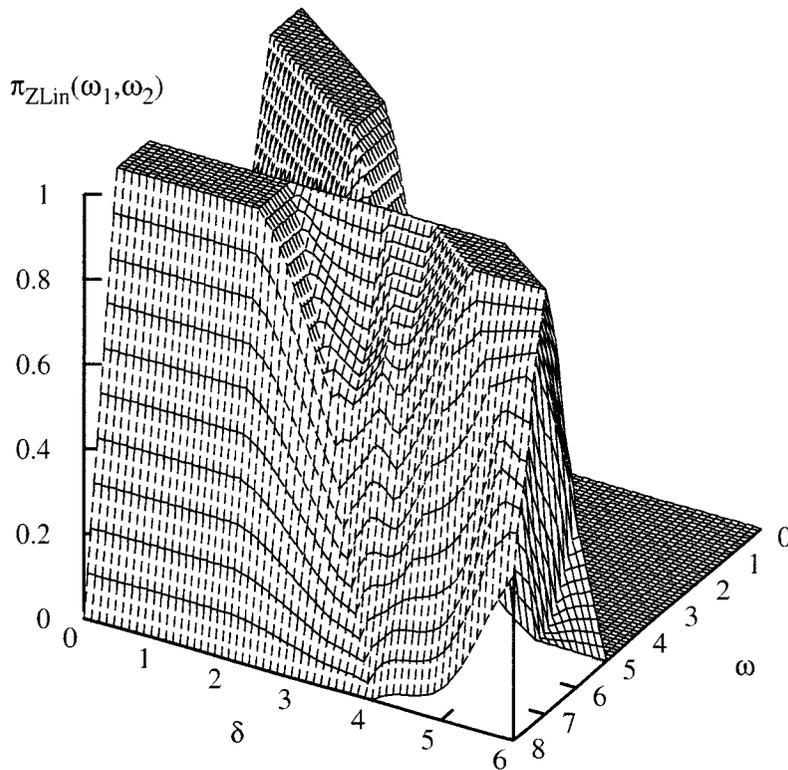


FIG. 2.20 – De la disjonction à la conjonction pour π_{ZLin} .

mode de combinaison abordé au paragraphe 1.6. L'opérateur π_{ZLin} choisit automatiquement le mode de combinaison adapté et donc l'opérateur correspondant. Il est bien entendu possible de jouer sur l'importance de γ en choisissant une autre valeur de pondération, mais la manière dont les sources sont combinées n'est pas très convaincante.

Les opérateurs adaptatifs présentés jusqu'à présent privilégient l'intersection des sources. C'est en effet là que l'information a intuitivement le plus de chance d'être correcte. Cependant, quel serait l'effet du remplacement du terme d'intersection par un terme de compromis? C'est à cette question que nous apportons quelques éléments de réponse dans le paragraphe suivant.

2.4.3 Une règle union/compromis

Intuitivement, le remplacement du terme d'intersection par un terme de compromis change l'interprétation du résultat. Le compromis est utilisé lorsque les sources ne sont pas réellement en conflit et ne sont pas non plus en accord total. Cette sorte de combinaison à mi-chemin entre l'union et l'intersection aura pour effet de faire diminuer la spécificité du résultat, ce qui signifie que le résultat sera plus imprécis qu'avec les règles utilisant l'intersection.

Nous avons choisi la règle arithmétique dans laquelle le terme d'union est conservé mais son comportement est accentué par le choix d'une T-Conorme "plus grande" que le maximum : $\min(1, \omega_1 + \omega_2)$. Le terme

d'intersection est remplacé par un terme de compromis : $\sqrt{\pi_1(\omega)\pi_2(\omega)}$. La règle est donc :

$$\pi_{UC}(\pi_1, \pi_2)(\omega) = No((1 - I_{d_{Bat}}(\pi_1, \pi_2)) \min(1, \pi_1(\omega) + \pi_2(\omega)) + \sqrt{\pi_1(\omega)\pi_2(\omega)}) \quad (2.41)$$

Les résultats de la fusion par cette règle de nos distributions de test montrent l'effet de l'opérateur de compromis : l'intersection adopte la forme d'une parabole. Par ailleurs, la spécificité a baissé, ce qui peut se vérifier graphiquement en constatant que le volume sous la surface est plus grand que pour la règle arithmétique de DUBOIS et PRADE. L'introduction d'un terme de compromis contribue plutôt à ajouter une l'incertitude dans le résultat alors que l'intersection réduit l'incertitude dans le résultat. L'influence d'un opérateur de compromis sera donc assez marginale.

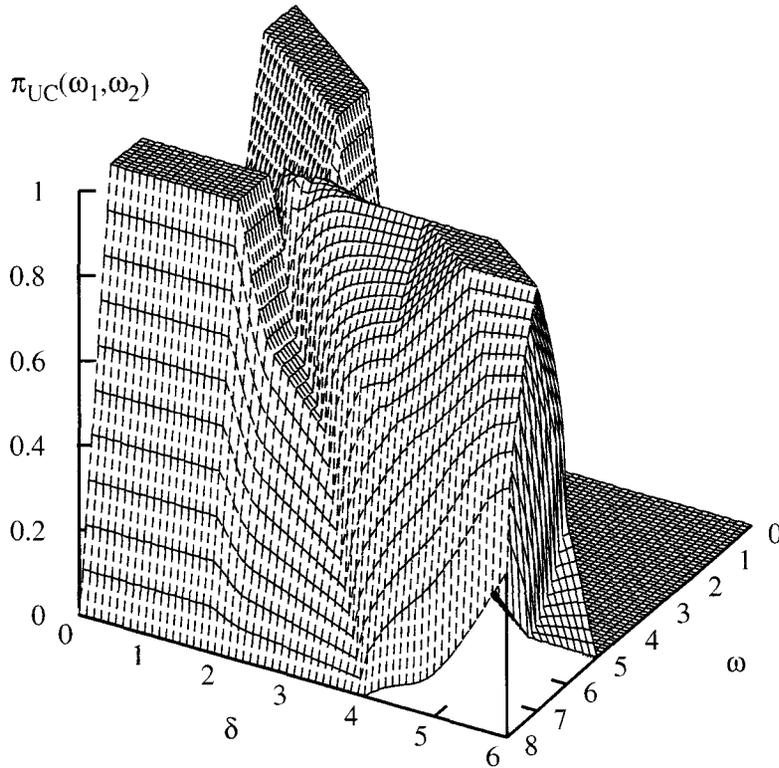


FIG. 2.21 – De la disjonction à la conjonction pour π_{UC} .

2.5 Conclusion

Nous avons décrit les opérateurs adaptatifs qui sont capables de déterminer automatiquement le mode de combinaison adéquat. Le choix du mode s'appuie sur la notion d'indice de conflit qui mesure l'accord entre les sources. Cette notion a été précisée et de nombreuses nouvelles techniques de calcul du conflit ont été proposées. Les nouveaux indices de conflit considèrent les deux distributions en totalité ce qui permet d'obtenir une évaluation plus juste de la ressemblance entre les sources. Par ailleurs, la notion d'indice de conflit permet aussi de déterminer rapidement la cohérence de deux sources et nous avons développé un indice de conflit très rapide permettant de "trier" les sources en fonction de leur compatibilité : l'indice horizontal. L'utilisation d'un indice de conflit flou, c'est-à-dire d'une mesure de cohérence dont le résultat serait lui-même imprécis a été évoqué et un exemple de définition utilisant un nombre flou a été donné.

La création d'un indice de conflit est intimement liée aux opérateurs de fusion adaptatifs car il indique l'importance avec laquelle l'intersection et l'union des sources interviendront dans le résultat. Surpris par l'absence d'un terme de compromis dans les règles de fusion adaptatives, nous proposons l'introduction de ce dernier dans de nouvelles règles de fusion. L'étude du comportement de ces règles a montré qu'elles permettaient le passage brutal (mais de manière continue) de la disjonction au compromis et à la disjonction et donc

qu'elles proposent une solution au choix automatique du mode évoqué à la fin du chapitre premier. Cependant, sous l'hypothèse faite au début selon laquelle une source au moins est cohérente, l'introduction explicite d'un terme de compromis n'apporte pas réellement de fonctionnalité supplémentaire, le résultat fourni par une telle règle étant moins spécifique et ne privilégiant pas l'intersection. Enfin, nous avons attiré l'attention sur le fait que la notion d'adaptativité dans les opérateurs de fusion est incompatible avec la notion d'associativité, ce qui pose un gros problème : comment faire pour combiner plus de deux sources ? Le but du chapitre suivant sera d'essayer d'apporter des éléments de réponse à cette interrogation.

Chapitre 3

Fusionner un grand nombre de sources

DANS les chapitres précédents, nous nous sommes volontairement limités à la description des techniques de fusion de données entre deux sources. En pratique, il est rare que seulement deux sources soient considérées. Les sources d'information à combiner sont multiples et toutes doivent être considérées dans le processus de fusion. Ce chapitre s'intéresse aux techniques de fusion utilisables lorsque plus de deux sources sont mises en jeu.

Comme dans le cas de la fusion entre deux sources, il est nécessaire qu'au moins une des sources soit cohérente – même s'il est impossible de préciser laquelle – sinon, le résultat sera erratique. Cette hypothèse étant faite, les techniques de fusion multi-sources peuvent grossièrement être classées en quatre grandes familles :

- Fusionner la première et la seconde source, puis combiner le résultat avec la troisième . . . Il s'agit de “cascader” des applications successives de la règle jusqu'à ce que toutes les sources aient été incorporées dans le résultat. Cette méthode est parfois dite “fusion modulaire”,
- Se ramener au cas où il n'y a que deux sources. Cela consiste à utiliser les n sources pour créer deux sources fictives qui seront fusionnées avec une des règles précédemment étudiées,
- Construire une nouvelle règle capable d'agréger n sources lors de la même opération. Cette règle devra éventuellement considérer le conflit entre les sources,
- Utiliser une combinaison des trois techniques précédentes (c'est pour cela que ces méthodes sont dites “hybrides”), par exemple se ramener à trois sources et utiliser une règle spécialement conçue pour trois sources.

Lorsque plus de deux sources doivent être combinées, de nouveaux problèmes apparaissent. Il faudra donc être prudent lors de l'utilisation de techniques de fusion pour deux sources et en particulier à l'adaptation de la notion d'indice de conflit à plus de deux sources, à la prise en compte de la fiabilité des sources et à l'ordre d'agrégation des sources.

Dans ce chapitre, nous nous intéressons aux quatre techniques suggérées plus haut : la fusion “cascade”, la réduction du nombre de sources, les règles de fusion conçues pour n sources et les méthodes “hybrides”. Le premier paragraphe de ce chapitre aborde les règles de combinaison de type “cascade”. Puis, dans une seconde partie, les techniques de réduction du nombre de sources sont présentées. Un opérateur de fusion de n sources est alors présenté. Les techniques précédentes sont alors utilisées conjointement pour construire une nouvelle méthode de fusion multi-sources dite “hybride” qui intègre le degré de confiance associé à chaque source dans le processus de fusion. Enfin, nous concluons sur l'utilisation des règles de fusion pour de multiples sources.

Dans tout ce chapitre, nous supposons disposer d'un ensemble $K = \{S_1, \dots, S_n\}$ de n sources connues par leurs distributions de possibilités $\pi_1(\omega) \dots \pi_n(\omega)$ qu'il faudra combiner. Les techniques décrites ici fonctionnent aussi pour des sous-ensembles flous connus par leurs fonctions d'appartenance. Lorsque cela sera nécessaire, $\tau_i \in [0, 1]$ indiquera le degré de confiance associé à la source i .

3.1 La fusion par “cascade” ou fusion modulaire

Lorsque l’on désire fusionner plus de deux sources, la première idée qui vient à l’esprit consiste à utiliser l’une des règles de fusion (entre deux sources) étudiées précédemment. Cette règle est alors appliquée à un couple de sources autant de fois que nécessaire jusqu’à obtention du résultat. La figure 3.1 présente un exemple de fusion “cascade” pour quatre sources. La première S_1 et la seconde S_2 sont fusionnées, puis la troisième S_3 et la quatrième S_4 . Les résultats intermédiaires sont alors fusionnés entre eux pour fournir le résultat S_R . Dans cet exemple, l’ordre de fusion choisi est tout à fait arbitraire. Il aurait été possible de combiner

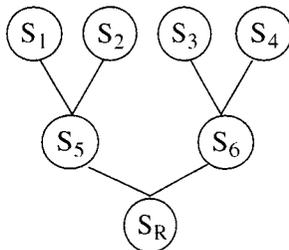


FIG. 3.1 – Exemple de fusion en “cascade”

les sources dans un ordre différent. Par exemple, calculer $S_{R_1} = \text{Fusion}(S_1, S_2)$ puis $S_{R_2} = \text{Fusion}(S_{R_1}, S_3)$ et enfin $S_R = \text{Fusion}(S_{R_2}, S_4)$. D’une manière générale, lorsque l’ordre de combinaison des sources est changé, le résultat lui aussi est changé ce qui est contraire à l’idée intuitive que l’on a de la fusion de données. Face à n sources, la fusion de données est censée utiliser les redondances pour fournir une information plus fiable et l’ordre dans lequel les informations sont analysées ne doit pas influencer sur le résultat.

Il existe alors deux approches : la première consiste à utiliser des opérateurs pour lesquels l’ordre n’a aucune importance c’est-à-dire qui sont associatifs et commutatifs. La seconde consiste à définir un opérateur qui classe lui-même correctement les sources pendant l’opération de fusion. Les paragraphes suivants s’intéressent à ces deux approches.

3.1.1 Les opérateurs associatifs et commutatifs

Pour s’assurer que l’ordre dans lequel les sources sont combinées n’a pas d’influence, l’opérateur doit être associatif et commutatif : associatif pour assurer que l’ordre dans lequel les résultats intermédiaires sont combinés n’importe pas et commutatif car les sources sont considérées comme équivalentes. De nombreux opérateurs ne possèdent pas ces deux propriétés :

- ▷ les moyennes pour lesquelles la notion d’associativité n’est pas compatible avec les notion de moyenne,
- ▷ les règles de DUBOIS et PRADE (2.30) et (2.36) qui ne sont pas associatives¹,
- ▷ Toutes les sommes symétriques non associatives.

Il ne reste donc que les T-Normes, les T-Conormes et quelques sommes symétriques utilisables pour effectuer une fusion en cascade qui se fait alors à l’aide de l’une des définitions de (3.1) dans laquelle \top , \perp et σ désignent respectivement une T-Norme, une T-Conorme et une somme symétrique associative.

$$\pi_R(\omega) = \top_{i=1}^n \pi_i(\omega) \quad \text{ou} \quad \pi_R(\omega) = \perp_{i=1}^n \pi_i(\omega) \quad \text{ou} \quad \pi_R(\omega) = \sigma_{i=1}^n \pi_i(\omega) \quad (3.1)$$

L’associativité et la commutativité permettent donc la construction facile de T-Normes, T-Conormes ou de sommes symétriques à n paramètres à partir d’opérateurs à deux paramètres.

Il est parfois nécessaire de faire suivre l’étape de fusion d’une étape de normalisation (voir 1.4.1) notamment lorsque l’on combine des distributions de possibilités qui – par nature – doivent être normalisées. Le

1. Contrairement à ce qui a été fait dans [Deveughele, 1993].

besoin d'effectuer une normalisation traduit ici encore un désaccord entre les sources et donc que l'une des deux sources se trompe.

Par sa construction, c'est-à-dire par l'utilisation d'opérateurs CICB ou CIVB associatifs et commutatifs, cette règle de fusion ne permet pas la prise en compte directe du conflit entre les sources. Il faudrait généraliser la notion de conflit à un ensemble de sources et modifier la manière avec laquelle les résultats intermédiaires sont combinés. Il ne s'agirait alors plus d'une règle de fusion modulaire car l'ordre avec lequel les sources sont combinées deviendrait important.

Dans le paragraphe suivant, nous présentons un exemple de fusion entre trois sources à l'aide d'une règle conjonctive et nous montrons les dangers de l'utilisation d'un opérateur non associatif dans (3.1).

3.1.2 Exemple

Nous considérons trois distributions de possibilités π_1, π_2, π_3 de la figure 3.2. Les distributions sont tout d'abord combinées en utilisant le minimum, opérateur associatif et commutatif dans l'ordre 1-2-3 puis dans l'ordre 2-3-1. Les résultats - non normalisés - sont présentés figure 3.3 où nous constatons graphiquement que l'ordre n'a pas d'importance.

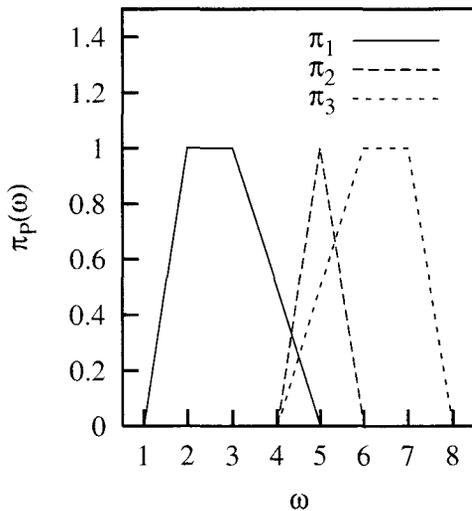


FIG. 3.2 - Distributions de test.

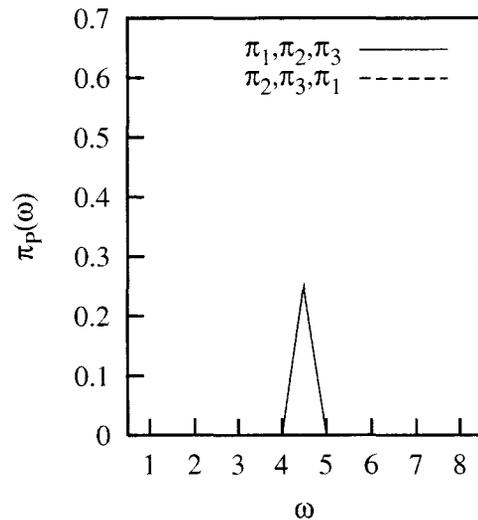


FIG. 3.3 - Fusion en "cascade" avec $\min(\omega_2, \omega_1)$.

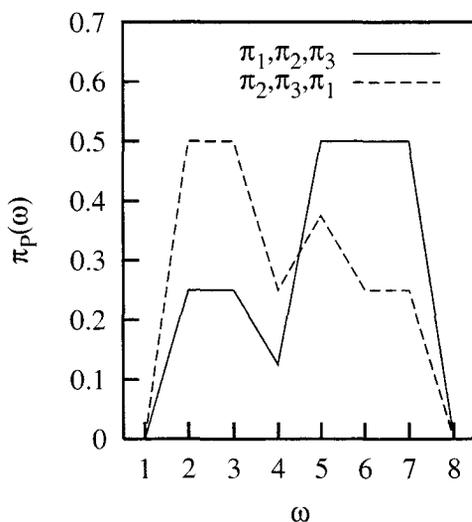


FIG. 3.4 - Fusion en "cascade" avec $\frac{\omega_1 + \omega_2}{2}$.

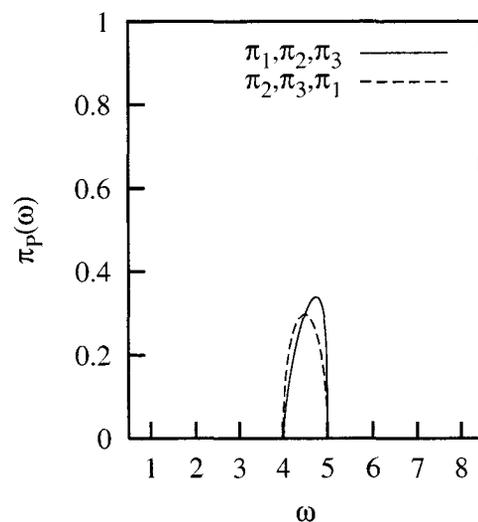


FIG. 3.5 - Fusion en "cascade" avec $\frac{\sqrt{\omega_2 + \omega_1}}{2}$.

La même opération est alors effectuée avec la moyenne arithmétique et la moyenne géométrique qui sont deux opérateurs non associatifs. Les résultats présentés sur les figures 3.4 et 3.5 permettent de vérifier graphiquement que l'ordre de combinaison a une grande importance.

Nous abordons maintenant l'autre possibilité de fusion modulaire : le classement des sources dans un ordre spécial préalablement à la fusion.

3.1.3 Opérateur effectuant un classement des sources

Nous faisons ici l'hypothèse que parmi les n sources, $j \geq 1$ sont fiables, sans pouvoir dire lesquelles. Ceci n'est pas une condition très restrictive car nous avons supposé qu'une source au moins était cohérente. L'idée présentée dans [Dubois et Prade, 1994a] consiste à construire une suite de distributions de possibilités $\pi_{(j)}$ telle que $\pi_{(1)}$ soit l'union de toutes les sources et $\pi_{(n)}$ l'intersection de toutes les sources. Pour j entre 1 et n , les $\pi_{(j)}$ sont alors telles que $\pi_{(j)} \leq \pi_{(j-1)}$. La construction de cette suite se fait alors de la manière suivante : puisqu'il y a j sources fiables, la combinaison de ces j sources doit se faire en mode conjonctif et les résultats intermédiaires doivent être combinés disjonctivement. Nous désignons par J l'ensemble des indices des sources fiables et par \top une T-Norme. La règle proposée par DUBOIS et PRADE est alors :

$$\pi_{(j)}(\omega) = \max_{\substack{|J|=j \\ J \subseteq K}} \top_{i \in J} \pi_i(\omega) \quad (3.2)$$

$\pi_{(1)}$ est bien l'union des sources, $\pi_{(n)}$ est bien l'intersection des sources et il est facile de vérifier que :

$$\pi_{(n)} \leq \pi_{(n-1)} \leq \dots \leq \pi_{(1)} \quad (3.3)$$

Pour effectuer le calcul qui peut sembler combinatoire, l'algorithme suivant peut être utilisé :

1. Trier $\{\pi_i(\omega), i = 1 \dots k\}$ dans l'ordre décroissant $\pi_{i_1}(\omega) \geq \pi_{i_2}(\omega) \dots \geq \pi_{i_n}(\omega)$
2. $\pi_{(j)}(\omega) = \pi_{i_j}(\omega)$ si $\top = \min$ ou $\pi_{(j)}(\omega) = \pi_{i_1}(\omega)\pi_{i_2}(\omega) \dots \pi_{i_j}(\omega)$ si $\top = xy$

Il y a donc bien classement des sources avant fusion (dans l'ordre décroissant des degrés de possibilité au point ω). Cette suite de distributions de possibilité fournit n résultats de fusion échelonnés entre la conjonction et la disjonction. Lorsque j sources sont considérées comme fiables, $\pi_{(j)}$ est la distribution dans laquelle les j sources ont été combinées conjonctivement, les résultats intermédiaires ayant été combinés disjonctivement.

Dans la définition (3.2) il est possible de remplacer le max par un autre opérateur, par exemple une moyenne. Les résultats intermédiaires seront alors combinés dans un mode de compromis. Souvent, le nombre de sources fiables n'est pas exactement connu et cette règle est difficile à utiliser. Par ailleurs, notons que cette règle nécessite le choix de deux opérateurs de fusion.

La fusion de n sources à l'aide de fusion modulaire (ou en "cascade") est simple mais le choix d'un ou de plusieurs opérateurs simples est difficile car fortement lié à l'application. Par ailleurs, la fusion modulaire considère – par construction – les sources deux par deux. L'introduction de la notion d'indice de conflit dans ce type d'opérateur de fusion n'est pas aisée. Pour y parvenir, une méthode consiste à utiliser l'indice de conflit pour diviser K en deux sous-ensembles. Les éléments d'un même sous ensemble sont alors combinés entre eux dans le mode adapté². Il reste donc à fusionner deux sources entre elles. C'est cette technique de réduction à deux sources qui nous intéresse maintenant.

3.2 La réduction du nombre de sources

L'idée de ces méthodes provient d'une remarque simple : il existe déjà de nombreux outils bien maîtrisés pour fusionner deux sources. Pourquoi ne pas utiliser les n sources pour construire deux sources fictives que l'on combinera entre elles à l'aide d'une des techniques du chapitre 1 ou 2 ?

2. En utilisant toute technique permettant de combiner l sources comme par exemple la fusion modulaire.

La difficulté de cette approche est double. Tout d’abord, il faut trouver le procédé qui permet de passer de n sources à deux sources tout en conservant l’information, puis il faut choisir la règle de fusion entre deux sources la plus adaptée à la situation. La règle “quantifiée” de DUBOIS et PRADE présentée dans [Dubois et Prade, 1994a] appartient à cette famille.

Dans cette approche, la première étape consiste à estimer le nombre de sources en accord de manière optimiste (soit k cette estimation) et de manière pessimiste (soit m cette estimation). Ainsi, la règle (3.2) peut être utilisée pour fusionner les n sources en deux sources virtuelles $\pi_{(k)}$ et $\pi_{(m)}$ qui seront alors combinées entre elles en utilisant la première règle adaptative (2.30) de DUBOIS et PRADE.

La détermination des deux entiers k et m tels que $k \geq m$ est effectuée en utilisant l’indice de conflit h . L’estimation optimiste (k) est obtenue en comptant le nombre de sources ayant au moins une zone en commun, même si cette dernière est très petite. Il s’agit de tous les sources qui, considérées deux à deux, conduisent à $h > 0$. L’estimation pessimiste (m) est réalisée en comptant les sources dont l’intersection des noyaux n’est pas vide. Il s’agit des toutes les sources qui, considérées deux à deux, conduisent à $h = 1$. Soit J un ensemble de sources, les entiers k et m sont alors donnés par :

$$h(J) = \sup_{\omega \in \Omega} \min_{i \in J} \pi_i(\omega) \quad \begin{array}{l} m = \sup\{|J|, h(J) = 1\} \\ k = \sup\{|J|, h(J) > 0\} \end{array} \quad (3.4)$$

Il semble alors plausible de supposer qu’il y a m sources fiables, mais en aucun cas plus de k . La définition (3.2) permet de déterminer $\pi_{(k)}$ et $\pi_{(m)}$ combinées ensuite à l’aide de la première règle adaptative :

$$\pi_R(\omega) = \max \left(\frac{\pi_{(k)}(\omega)}{h(k)}, \min(1 - h(k), \pi_{(m)}(\omega)) \right) \quad \text{avec} \quad h(k) = \max\{h(J), |J| = k\} \quad (3.5)$$

Lorsque K ne contient que deux sources, la première règle adaptative est bien retrouvée. Les propriétés de cette règle sont fortement comparables à celles dont dispose la première règle adaptative de DUBOIS et PRADE (2.30) présentée dans les propriétés 23 à 29 paragraphe 2.3.1.

Les solutions présentées tentent d’adapter à $n \geq 2$ des règles prévues pour $n = 2$. Une autre approche consiste à développer un opérateur capable de fusionner n sources, la fusion de $n = 2$ sources étant alors un cas particulier de la règle. [Delmotte, 1998] propose une règle de fusion fondée sur la notion d’opérateur logique ou-exclusif flou. Nous rappelons cette règle dans le paragraphe suivant.

3.3 Fusionner n sources : la règle “ou-exclusif”

La motivation de la construction de cette règle repose sur une observation simple : rares sont les opérateurs permettant de fusionner $n \geq 2$ sources directement, tenant compte la fiabilité des sources tout en ayant un comportement logique. [Delmotte, 1998] propose donc la construction d’un tel opérateur comparable à la règle linéaire de DUBOIS et PRADE (voir 2.3.3).

Chaque source est supposée associée à un coefficient $\alpha_i \in [0, 1]$ qui indique le degré de confiance accordé à cette source. L’opérateur devant être adaptatif, il sera nécessaire d’introduire des coefficients indiquant dans quelle mesure il faut appliquer le mode conjonctif ou le mode disjonctif ou le compromis. L’idée est alors d’utiliser un polynôme du premier degré en min et max :

$$\begin{aligned} \pi_{Del}(\omega) = & No(\theta(\alpha_i) \max_i(\alpha_i \pi_i(\omega)) + \\ & \beta(\alpha_i) \min_i(\alpha_i \pi_i(\omega)) + \\ & \varphi(\alpha_i) \max_i(\alpha_i \pi_i(\omega)) \min_i(\alpha_i \pi_i(\omega)) + \\ & \delta(\alpha_i)) \end{aligned} \quad (3.6)$$

Le terme en $\min \max$ n'ayant pas de signification physique, il est supprimé et l'opérateur prend alors la forme :

$$\pi_{Del}(\omega) = No(\theta(\alpha_i) \max_i(\alpha_i \pi_i(\omega)) + \beta(\alpha_i) \min_i(\alpha_i \pi_i(\omega)) + \delta(\alpha_i)) \quad (3.7)$$

No étant un opérateur de normalisation. Chaque source a alors une importance croissant linéairement en fonction de son degré de confiance. La détermination des fonctions θ, β, δ se fait en considérant quelques conditions limites qui reposent sur le nombre de sources fiables. Les coefficients α_i permettent de classer les sources en deux groupes : les sources fiables dont le groupe sera noté F et les sources non fiables NF . Les conditions limites s'expriment donc par (voir tableau 3.1):

Condition	Action désirée
$NF = \emptyset$	Toutes les sources sont fiable, une intersection est donc souhaitée
$NF \neq \emptyset$ et $F \neq \emptyset$	Il existe des sources fiables et non fiables. L'union est donc le mode le plus approprié
$F = \emptyset$	Aucune source n'est fiable et aucune forme de combinaison n'est à privilégier. Trois choix existent alors : <ul style="list-style-type: none"> ▷ Le résultat est une union. ▷ Le résultat est l'ignorance complète, ▷ Rien n'est imposé au résultat,

TAB. 3.1 – Les conditions aux limites

Le choix réalisé dans la dernière condition aux limites permet donc de définir trois opérateurs. Nous décrivons chacun d'entre eux dans les trois paragraphes suivants.

3.3.1 F est vide et l'union des sources est choisie

Dans ce cas, [Delmotte, 1998] montre que le choix des fonctions θ, β, δ sont respectivement un “non et” logique, un “et” logique et δ est alors la fonction nulle. L'opérateur devient donc :

$$\pi_{Nand}(\omega) = No(Nand(\alpha_i) \max_i(\alpha_i \pi_i(\omega)) + And(\alpha_i) \min_i(\alpha_i \pi_i(\omega))) \quad (3.8)$$

L'opérateur Nand est construit comme en logique classique par $Nand(.) = \overline{And(.)}$.

3.3.2 F est vide et l'ignorance complète est choisie

Dans ce cas, [Delmotte, 1998] montre que le choix des fonctions θ, β, δ sont respectivement un “ou exclusif” logique, un “et” logique et δ est alors un “non ou” logique. L'opérateur devient alors :

$$\pi_{Nor}(\omega) = No(Xor(\alpha_i) \max_i(\alpha_i \pi_i(\omega)) + And(\alpha_i) \min_i(\alpha_i \pi_i(\omega)) + Nor(\alpha_i)) \quad (3.9)$$

L'opérateur Xor est construit comme en logique classique par $Xor = Or(.) \overline{And(.)}$ et l'opérateur Nor étant défini par $Nor = \overline{Or(.)}$.

3.3.3 F est vide et rien n'est imposé au résultat

Dans ce cas, [Delmotte, 1998] montre que le choix des fonctions θ, β, δ sont respectivement un “ou exclusif” logique, un “et” logique et δ la fonction nulle. L'opérateur devient alors :

$$\pi_{Xor}(\omega) = No(Xor(\alpha_i) \max_i(\alpha_i \pi_i(\omega)) + And(\alpha_i) \min_i(\alpha_i \pi_i(\omega))) \quad (3.10)$$

3.3.4 Définition des opérateurs dans le cadre flou et dans le cadre possibiliste

Pour définir les opérateurs And, Or et $\overline{(\cdot)}$ nécessaires à la création de Xor, Nor et Nand, il suffit d'utiliser les propriétés rappelées au paragraphe 1.4.2.1 page 16. Les définitions les plus utilisées sont :

$$\text{And}(\cdot) = \min(\cdot) \quad \text{Or}(\cdot) = \max(\cdot) \quad \overline{(\cdot)} = 1 - (\cdot) \quad (3.11)$$

Nous présentons maintenant une nouvelle méthode qui combine les techniques précédentes pour construire des règles de fusion “hybrides” : nous nous proposons dans le prochain paragraphe une nouvelle méthode hybride fondée sur l'analyse de données.

3.4 Les méthodes “hybrides”

Intuitivement, en présence d'un grand nombre de sources, une partie d'entre elles est souvent concordante. Il semble intéressant de déterminer avant fusion – s'ils existent – les sous-ensembles de sources concordantes de manière à traiter les sources concordantes et les sources discordantes séparément. Cette voie a déjà été explorée dans le cadre de la théorie de DEMPSTER-SHAFFER par [Schubert, 1994]. Classer les sources en sous-ensembles (ou classes) est typiquement un problème d'analyse de données (voir [Bandemer et Näther, 1992]). Quelle que soit la technique choisie, il faudra définir un moyen de mesurer la concordance d'une source avec une autre et la concordance d'une source avec un ensemble de sources. Une fois les classes formées, il faudra alors réaliser la fusion “intra classes” puis “inter classes” à l'aide d'opérateurs différents pour prendre en compte les concordances puis les éventuelles discordances. Nous combinons ici les techniques de fusion modulaire (combiner rapidement un grand nombre de sources mais seulement de manière conjonctive ou de manière disjonctive) pour la fusion “intra classes” et la technique de réduction du nombre de sources (introduisant implicitement la notion de groupes de sources concordantes au travers des estimations du nombre de sources fiables) dans la fusion “inter classes”.

La construction de cette nouvelle règle fait donc appel à deux domaines : l'analyse de données et la fusion de données. Nous devons tout d'abord choisir un algorithme de classification (ce qui nécessitera le choix d'un indice de similarité entre deux groupes de sources), un algorithme de fusion de sources cohérentes et un algorithme de fusion de sources en désaccord.

Les paragraphes suivants proposent un algorithme de formation de classes permettant de déterminer automatiquement le nombre de classes k et détaillent les techniques de fusion utilisées pour combiner les sources dans une même classe (fusion “intra classe”) et combiner les résultats intermédiaires entre eux (fusion “inter classe”).

Si des sources discordantes ont été trouvées, il devient plus simple de les isoler pour vérifier leur cohérence et déterminer si elles doivent être intégrées ou non au processus de fusion.

3.5 Formation des classes

Pour grouper les sources, nous utilisons la classification hiérarchique ascendante dans laquelle chaque source appartient totalement³ à une classe et une seule à un instant donné. Dans ce type de classification, chaque source est tout d'abord placée seule dans une classe. Au début de la classification, n classes notées C_i sont alors présentes. Le processus de classification va grouper les classes concordantes au sens d'un indice de conflit ce qui aura pour effet de diminuer le nombre de classes jusqu'à ce que le rangement ainsi effectué soit considéré comme optimal au sens d'un critère à définir. Notons k le nombre de classes existantes à chaque étape du processus de classement. Etant donnée notre méthode de classification non floue dans laquelle une

3. Cela signifie qu'il s'agit d'une appartenance “classique” et non pas d'une appartenance floue. L'appartenance floue pourrait être envisagée et une source appartiendrait alors à l classes différentes, avec des degrés d'appartenances éventuellement différents. Il s'agirait alors d'un classement flou des sources dont l'algorithme “Fuzzy C-Mean” (voir [Chi et al., 1996]) est un exemple.

source appartient à une classe et une seule, nous aurons toujours $K = \bigcup_{i=1}^m S_i = \bigcup_{i=1}^k C_i$. L'approche est dite descendante car le nombre de classes diminue au fur et à mesure du processus de classification.

Pour réaliser ce classement, trois outils doivent être définis : l'indice de conflit capable de mesurer le conflit entre deux ensembles de sources, le critère d'arrêt qui décidera de l'optimalité du classement réalisé et donc du nombre de classes formées et enfin de la stratégie de combinaison de deux groupements de sources jugés en accord. Les paragraphes suivants s'intéressent à la définition de ces trois outils.

3.5.1 Conflit pour deux ensembles de sources

Il s'agit de mesurer la concordance entre deux ensembles de sources. Nous avons déjà abordé la notion d'indice de conflit entre deux ensembles flous ou distributions de possibilité au paragraphe 2.1. Ce paragraphe propose une extension de cette notion à deux ensembles de sources A et B ([Zunino et al., 1997a] et [Zunino et al., 1998a]). Soit J l'indice de conflit entre deux ensembles de sources. Le critère J doit au moins vérifier trois propriétés :

1. J est une fonction à valeurs positives,
2. Le conflit entre un ensemble de sources et lui même est maximal et égal à 1.
3. Le conflit entre deux ensembles A et B est égal au conflit entre B et A .

De manière générale, J n'est donc pas obligatoirement une relation d'équivalence car la transitivité est absente mais il est possible de trouver des fonctions J qui soient des relations de similarité. De nombreuses définitions vérifiant ces propriétés existent mais dépendent toutes du problème à traiter. La démarche la plus utilisée dans la littérature consiste à construire le conflit entre deux ensembles à partir du conflit entre les éléments de ces ensembles, ce qui correspond dans notre cas à l'indice de conflit I . Voici quelques définitions classiques du critère $J(A, B)$:

$$\begin{aligned} J_{min}(A, B) &= \min_{S_i \in A, S_j \in B} I(S_i, S_j) & J_{med}(A, B) &= \text{med}_{S_i \in A, S_j \in B} I(S_i, S_j) \\ J_{max}(A, B) &= \max_{S_i \in A, S_j \in B} I(S_i, S_j) & J_{moy}(A, B) &= \text{moy}_{S_i \in A, S_j \in B} I(S_i, S_j) \end{aligned} \quad (3.12)$$

Le minimum permet d'être prudent en fixant la similarité de deux ensembles comme étant la plus petite de toutes celles obtenues en prenant les éléments deux à deux. L'effet de "chaîne" est limité. En utilisant le maximum, deux ensembles se ressemblent lorsqu'il existe au moins un élément de l'un qui ressemble à un élément de l'autre. L'effet de "chaîne" peut apparaître. La médiane (moyenne entre la valeur la plus petite et la plus grande) et la moyenne sont des compromis habituels. Il est donc nécessaire de choisir un indice de conflit I (voir paragraphe 2.1) et une méthode de combinaison J . Le choix de ces deux éléments est cependant très lié à l'application et nécessite en pratique de nombreux essais.

```

1      Conflit entre deux groupes de sources  $C_1$  et  $C_2$ 
2      function Conflit( $C_1, C_2$ : Liste de sources) : real
3          var mini : real = REEL_MAXIMUM
4          begin
5              for  $i$  in 1 ... taille( $C_1$ ) do
6                  for  $i$  in 1 ... taille( $C_2$ ) do
7                       $mini = \min(\text{Conflit}(S_{i(C_1)}, S_{j(C_2)}))$ 
8                  done
9              done
10             return mini
11         end

```

FIG. 3.6 – Calcul du conflit entre deux ensembles de sources.

L'algorithme permettant le calcul du conflit entre deux groupes de sources est donné figure 3.6. La fonction "taille" donne le nombre de sources dans une classe et "REEL_MAXIMUM" désigne le réel le plus grand représentable sur la machine. Le minimum entre les conflits a été choisi. Nous nous intéressons maintenant à la stratégie de classement.

3.5.2 La stratégie de groupement et le critère d'arrêt

Avec un processus de classification descendant, il est tout d'abord nécessaire de former la partition atomique des sources. Chaque source est alors seule dans sa classe comme le schématise la figure 3.7. Le clas-

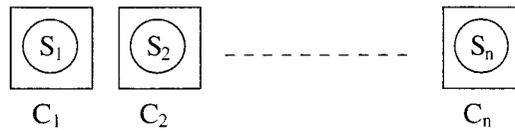


FIG. 3.7 – La partition atomique.

sement s'effectue alors selon l'algorithme décrit figure 3.11. L'analyse commence par la première classe C_1 qui à cet instant ne contient qu'une source. Nous calculons alors le conflit entre C_1 et toutes les autres classes $C_i, i \in 2 \dots n$ à la recherche du conflit le plus grand, c'est-à-dire de la classe la plus proche de C_1 . Ceci est symbolisé figure 3.8 où les rectangles représentant les classes ont été supprimés pour les classes ne comportant qu'un élément dans le but de faciliter la lecture. Nous cherchons donc le maximum des J_i . Il est intéressant de remarquer que ce calcul de conflit entre deux classes se réduit à un calcul de conflit entre une classe et une source.

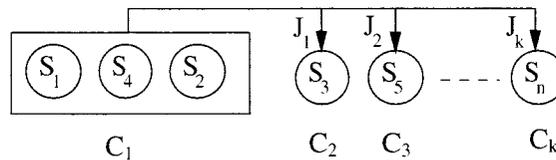


FIG. 3.8 – Calcul du critère de classement.

Une fois le maximum J_{max} des J_i trouvé, deux cas sont alors possibles :

1. Soit J_{max} est supérieur à un seuil ϵ et la classe correspondante C_{max} est compatible avec C_1 ,
2. Soit J_{max} est inférieur à ϵ et aucune classe n'est alors jugée suffisamment compatible avec C_1 .

Supposons que C_{max} soit jugée compatible avec C_1 . Nous nous trouvons par exemple dans le cas de la figure 3.9. La classe C_1 va être combinée avec C_{max} et le résultat sera celui présenté figure 3.10. La combi-

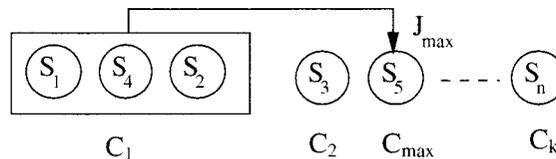


FIG. 3.9 – Avant combinaison de C_1 et de C_{max} .

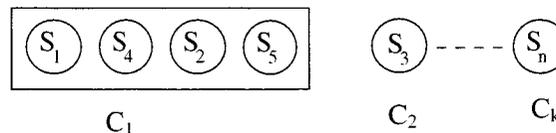


FIG. 3.10 – Après combinaison de C_1 et de C_{max} .

naison consiste donc à extraire C_{max} de la liste des classes non encore appariées et à placer toutes les sources qu'elle contient dans C_1 . Il est intéressant de remarquer que cette opération ajoute simplement une source à une classe car nous savons qu'à cette étape, la classe C_{max} ne contient qu'une seule source. Lorsque la source a été ajoutée à la classe C_1 , l'algorithme recommence et parcourt à nouveau la liste des sources à la recherche d'un nouveau J_{max} et de la C_{max} associée mais en considérant la nouvelle classe C_1 et la nouvelle liste (qui comporte une source de moins que précédemment).

Dans le second cas évoqué ci-dessus, la classe C_1 est considérée comme complète et l'analyse continue avec la prochaine classe dans la liste, c'est-à-dire avec la classe C_2 contenant la source S_3 sur les figures précédentes. Cet algorithme essaie donc toujours d'ajouter une source à une classe et parcourt la liste des sources autant de fois que nécessaire pour en extraire toutes les sources pouvant être mises dans la classe courante (la classe dans laquelle il y aura ajout, dans notre exemple précédent, il s'agissait de C_1). Par ailleurs, à l'étape i , l'algorithme partage la liste des sources en deux : de $1 \dots i - 1$ une liste contenant les classes définitives et de $i + 1$ jusqu'à la fin, une liste contenant des sources non encore définitivement groupées. L'algorithme s'arrête lorsque $i = k$ où k désigne le nombre de classes formées.

Nous procédons d'une manière un peu différente de l'habitude existant en analyse de données où le nombre de classes est d'abord déterminé (soit par l'analyste, soit automatiquement) et où ensuite, ces classes sont formées. Ici, la décision d'ajouter une source à une classe est entièrement prise à l'aide d'un seuil ϵ qui représente le J_{max} minimal conduisant à l'ajout d'une source dans la classe courante. Il est en pratique plus facile d'indiquer quel est le conflit minimal autorisant le classement plutôt que le nombre de classes lui-même. Cet

```

1   Classement des sources
2   X est la partition atomique
3   T est l'ensemble des classes formées
4   function Classement(X : Liste de sources) : Liste de classes
5       var T : Liste de classes =  $\emptyset$ 
6       C, Source : Source
7       begin
8         if |X|  $\geq$  2 then
9           begin
10            C = Première classe de X; X = X - C
11            while X  $\neq$   $\emptyset$  do
12              begin
13                J = 0, Source =  $\emptyset$ 
14                for S  $\in$  X do
15                  begin
16                     $\alpha$  = CConflit(C,S)
17                    if  $\alpha >$  J then
18                      begin
19                        Source = S; J =  $\alpha$ 
20                      end
21                  end
22                  if Source  $\neq$   $\emptyset$  and J  $>$   $\epsilon$  then
23                    begin
24                      C = C  $\cup$  Source; X = X - Source
25                    end
26                  else
27                    begin
28                      T = T  $\cup$  C; C = Suivante de X
29                    end
30                end
31              end
32            return T
33          end

```

FIG. 3.11 – Algorithme de classification.

algorithme nécessite quelques explications. La partition atomique est stockée dans la liste X . La première classe de cet ensemble X est alors retirée de X et comparée au sens d'un indice de conflit à toutes les autres classes de X . Si un appariement est possible, celui-ci est réalisé, faisant grandir la classe C . Si aucun appariement n'est possible, le contenu de la classe C est alors ajouté au résultat T . Avec ce procédé, nous ne classons toujours qu'une source à la fois. Cette heuristique permet de réduire le temps nécessaire à l'exécution en limitant le nombre de conflits calculés. En pratique, lorsqu'il faut réaliser la fusion d'un grand nombre de sources, la majorité d'entre elles sont concordantes. Elles seront donc classées dans C_1 .

Il reste cependant à déterminer la valeur de ϵ . Fixer ce paramètre consiste à fixer la limite inférieure en-dessous de laquelle deux sources sont considérées comme discordantes. Nous proposons maintenant une méthode automatique de détermination de ce seuil.

3.5.3 La détermination du seuil

La détermination du seuil est une phase critique du processus. Trop faible, il conduira à la formation de classes contenant de nombreuses sources peu concordantes tandis que trop élevé, il empêchera la formation des classes et la partition atomique restera inchangée.

Pour déterminer le seuil, nous commençons par le calcul du conflit entre toutes les sources. Puisque le calcul du conflit est réflexif et que le conflit entre deux sources identiques est par définition 1, il n'y a que $\frac{n(n-1)}{2}$ calculs du conflit à réaliser. Nous obtenons donc les a_i qui sont les éléments du triangle inférieur du tableau :

	S_1	S_2	S_3	...	S_n
S_1	1				
S_2	a_1	1			
S_3	a_2	a_3	1		
\vdots	1	
S_n	a_l	1

Ces conflits croisés sont alors classés par ordre décroissant. Nous disposons d'une suite de réels a_i tels que $a_1 \geq a_2 \geq \dots a_l$. Il est alors possible de calculer les différences successives $d_j = a_{i-1} - a_i$. Cela revient à calculer l'accroissement (qui dans ce cas est négatif puisque les a_i ont été triés) ou la dérivée de la courbe des conflits croisés. Il est possible de tracer le graphe des conflits croisés et de la dérivée, comme sur la figure 3.13. En abscisse, nous plaçons i et en ordonnée⁴ a_i . La dérivée se trace en plaçant en abscisse $j + \frac{1}{2}$ (la dérivée est calculée par différence, elle s'applique au milieu du segment) et en ordonnée d_j .

A l'aide de cette dérivée, nous recherchons le premier extrémum de la fonction définie par les points des conflits croisés. Ce extrémum étant trouvé (dans notre exemple, il est obtenu pour la seconde différence), nous considérons que le seuil est la valeur du conflit a_{j-1} , c'est-à-dire le plus petit des deux conflits ayant servi à calculer la différence faisant apparaître un extrémum local. Sur ce graphique 3.13, il est possible de "lire" visuellement le seuil qui est donné par le troisième point de la courbe C en partant de la gauche, c'est-à-dire un peu moins que 0.8.

Le seuil déterminé, nous sommes donc capables de former k classes. Il faut maintenant étudier le processus de fusion. Celui-ci se divise en deux étapes : la fusion "intra classes" consistant à combiner les sources dans une même classe et la fusion "inter classes" consistant à combiner ensemble les résultats des fusion "inter classes". Les paragraphes suivants décrivent ces deux fusions.

3.6 La fusion intra classes

Nous disposons maintenant de classes contenant des sources concordantes entre elles car, par construction, toutes les classes C_i contenant plus d'une source vérifient :

$$\min_{\substack{S_i \in C, S_j \in C \\ S_i \neq S_j}} I(S_i, S_j) > \epsilon \quad (3.13)$$

Si la classe C_i contient au moins deux sources, il est possible de fusionner toutes les sources de cette même classe à l'aide d'une règle de fusion modulaire comme cela est décrit au paragraphe 3.1 ou alors de la règle quantifiée de DUBOIS et PRADE du paragraphe 3.2.

4. En fait, sur la figure 3.13 page 61, les a_i ont tout d'abord été recadrés à l'aide d'une transformation linéaire de manière à ce que le plus petit des conflits soit représenté par 0 et le plus grand par 1. Ce recadrage – effectué pour faciliter la lecture – ne change rien à l'algorithme, il faut seulement en tenir compte dans l'interprétation de la figure.

La combinaison dans un mode conjonctif permet de remplacer les l sources de la classe par une nouvelle source virtuelle. Soit $|C_i|$ le nombre de sources dans la classe C_i , la fusion “intra classe” peut être réalisée à l’aide d’un opérateur de fusion modulaire comme le min. Nous allons donc remplacer chaque classe pour laquelle $|C_i| > 1$ par une source S_{r_i} définie par :

$$S_{r_i} = No(\min_{S_i \in C_i} S_i) \quad (3.14)$$

où No est un opérateur de normalisation. Si une classe C_j ne contient qu’une source S_j , la source virtuelle sera alors obtenue sans fusion, par simple égalité $S_{r_j} = S_j$. Nous sommes donc en présence de $k \leq n$ sources “virtuelles” qu’il est nécessaire de fusionner entre elles.

3.7 La fusion inter classes

A cet instant, nous espérons que k est très inférieur à n , ce qui signifie que de nombreuses sources étaient concordantes et ont été fusionnées dans un mode conjonctif qui est plus sûr qu’un mode disjonctif. Les k sources sont donc en désaccord et la fusion en mode disjonctif s’impose.

La règle de fusion peut-être une fusion modulaire, une règle quantifiée de DUBOIS et PRADE ou l’opérateur “ou-exclusif”. Nous choisissons ici d’utiliser le max pour la fusion disjonctive. Le résultat général de la fusion sera donc :

$$\pi_R = \max_{i \in \{1 \dots k\}} S_{r_k} \quad (3.15)$$

Nous appliquons maintenant notre méthode à cinq distributions triangulaires aléatoirement générées.

3.8 Exemple

Nous avons choisi – dans un but de simplification des figures – des distributions triangulaires. Le jeu de sources à fusionner est présenté figure (3.12). Il a été généré aléatoirement. La borne inférieure du support a été choisi dans l’intervalle $[-1, 1]$, l’abscisse du support dans $[1, 2]$ et la borne supérieure du support dans $[2, 3]$. Nous avons volontairement restreint le nombre de sources à 5 pour faciliter la compréhension et la lecture des figures. La figure (3.12) présente les sources au début du processus : chaque source est dans une classe. Puis, le processus de recherche du seuil calcule les différences des conflits croisés (figure 3.13) et calcule le seuil (ici un peu moins que 0.8). La figure (3.14) présente les trois classes formées. Les deux premières (dans le sens des abscisses croissantes) contiennent chacune deux sources. La dernière est un singleton. Les sources de chacune des deux classes sont alors fusionnées à l’aide de l’équation (3.14) et les résultats sont visibles sur la figure (3.15). La forme de la distribution F_{C_1} montre clairement qu’elle a été obtenue par la fusion de plusieurs sources en mode conjonctif. Enfin, la fusion des trois sources de la figure (3.15) à l’aide de la règle (3.15) fournit le résultat visible figure (3.16).

Les opérateurs de fusion utilisés ici sont des opérateurs “simples”. L’utilisation d’autres opérateurs est possible, ce qui permet l’ajout de fonctionnalités au processus de fusion comme par exemple la prise en compte de sources plus ou moins fiables.

3.8.1 Prise en compte de la fiabilité des sources

Parfois, les sources à fusionner sont plus ou moins fiables. Lorsque la fiabilité des sources peut être quantifiée, il semble intéressant de tenir compte de ces coefficients de fiabilité dans le processus de fusion en renforçant l’influence d’une source en laquelle on a confiance et en réduisant l’influence des sources considérées moins sûres. Notre processus de classification/fusion permet de prendre en compte la fiabilité de chaque source. La prise en compte s’effectue au moment de la fusion et donc à deux niveaux : pendant la fusion “intra classes” et pendant la fusion “inter classes”. Il faut donc disposer d’opérateurs capables de gérer la fiabilité des sources comme la règle “ou-exclusif”, les opérateurs OWA ou les intégrales floues abordés au chapitre prochain.

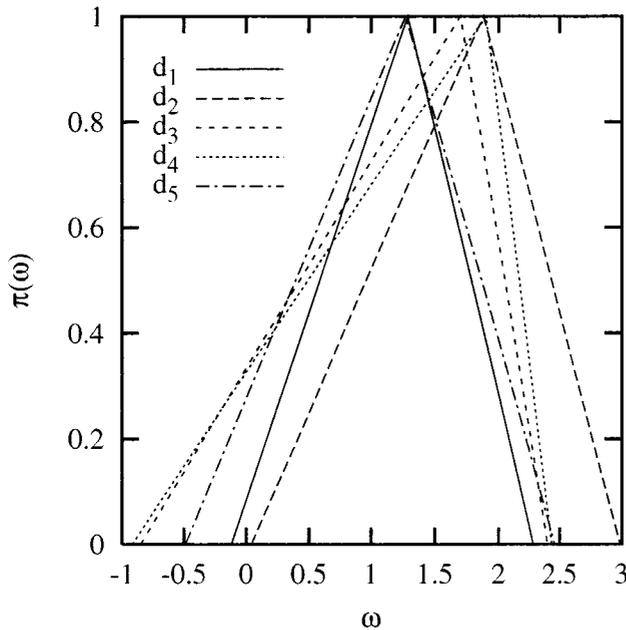


FIG. 3.12 – Les sources à fusionner.

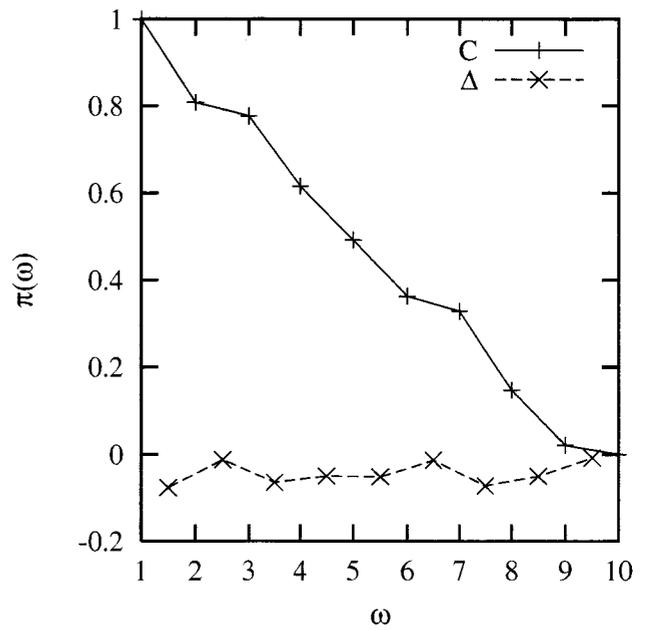


FIG. 3.13 – Le graphe des conflits.

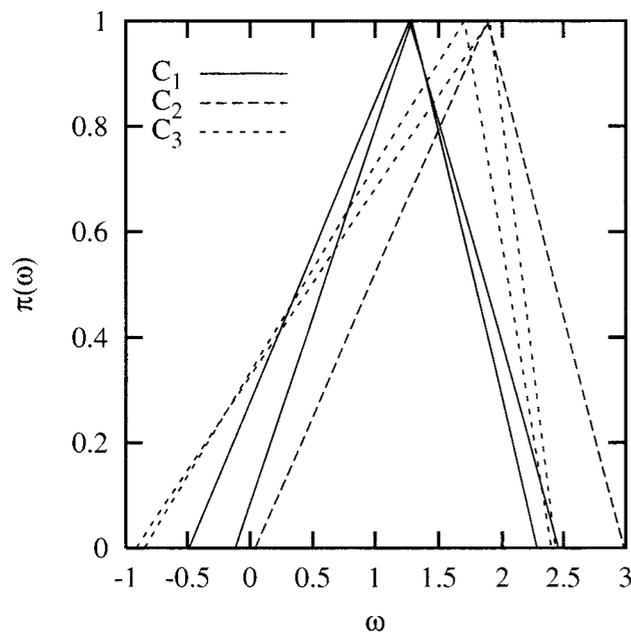


FIG. 3.14 – Les classes formées.

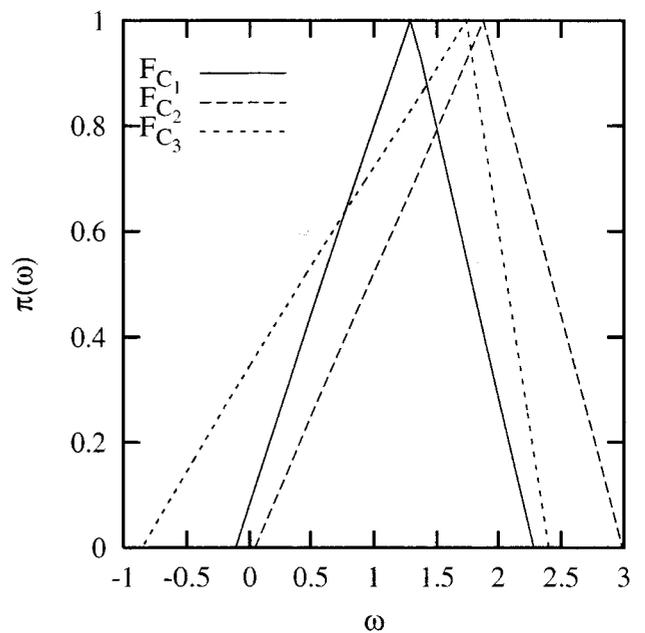
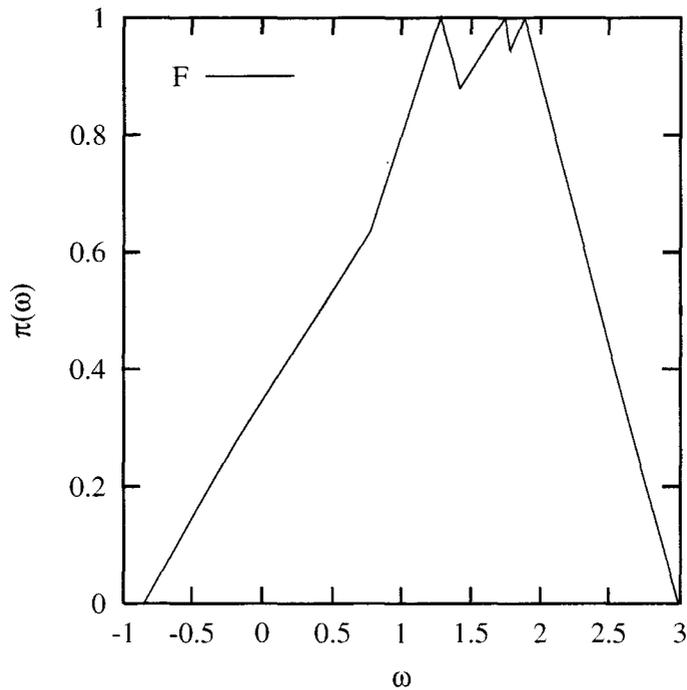


FIG. 3.15 – Résultat de la fusion intra-classes.

3.9 Conclusion

Ce chapitre a décrit les méthodes de fusion permettant de combiner plus de 2 sources et force est de constater que finalement, bien peu de techniques dédiées au cas particulier 2 sources peuvent être étendues pour $n \geq 2$. De manière générale, cela est possible pour tous les opérateurs associatifs ou quasi-associatifs par l'intermédiaire des techniques de fusion modulaire. Une autre approche consiste à se ramener à deux sources ou à définir un opérateur capable de fusionner n sources, mais la manœuvre n'est pas aisée. Enfin, il est possible de combiner ces trois techniques pour former un opérateur "hybride" qui classe les sources avant de les combiner.

Nous avons présenté une nouvelle technique de formation de classes de sources concordantes. Les sources

FIG. 3.16 – *Résultat final.*

erronées sont plus facilement détectables et peuvent être supprimées avant le processus de fusion. A l'intérieur des classes, les sources sont fusionnées à l'aide d'un opérateur conjonctif simple. Tous les résultats sont alors combinés de manière disjonctive. Notre stratégie de classement des sources – bien que naïve – montre l'intérêt de grouper les sources selon un critère de concordance avant d'effectuer la fusion et ce pour au moins deux raisons : il est plus facile d'écarter une classe fortement différente des autres que de combiner toutes les sources, même avec une notion de fiabilité. En effet, toute mesure clairement incorrecte peut potentiellement perturber la fusion. La deuxième raison concerne le procédé de combinaison lui-même : il est toujours plus intéressant d'effectuer une combinaison en mode conjonctif qu'en mode disjonctif car le mode conjonctif crée un résultat plus spécifique que les entrées, ce qui signifie que le processus de fusion affine l'information.

Nous avons décrit maintenant un grand nombre d'opérateurs, mais il existe encore quelques techniques un peu particulières comme les opérateurs construits sur le principe d'extension ou les intégrales floues dont le chapitre suivant aborde l'étude.

Chapitre 4

Les opérateurs spéciaux

LES opérateurs de fusion de données étudiés jusqu'à présent étaient a priori conçus pour travailler sur des distributions de possibilités. Cependant, d'autres formalismes disposent de nombreux opérateurs bien maîtrisés et il semble légitime d'essayer de les adapter au cadre possibiliste tout en conservant dans ce formalisme – si cela est possible – les propriétés dont ils disposent dans l'univers d'origine. L'un de ces univers est le corps des réels dans lequel nous effectuons quotidiennement toutes sortes d'opérations de fusion de données sans même nous en rendre compte : le calcul du revenu imposable, l'évaluation d'un étudiant au travers de sa moyenne, le calcul de la surface sous une courbe par intégration . . . Ces exemples suggèrent que la fusion est une technique très utilisée et qui peut prendre la forme d'un simple calcul entre réels.

Si l'intérêt d'étendre le calcul du revenu imposable aux ensembles flous peut laisser dubitatif, d'autres processus de décision peuvent avantageusement être traités dans l'univers des possibilités. C'est le cas du choix d'un nouveau véhicule par un particulier. Dans un tel problème de décision, l'acheteur dispose de nombreuses alternatives (tous les véhicules disponibles à la vente sur le marché) et de critères de choix personnels comme le prix, le type de carburant, la consommation, l'aspect esthétique . . . Chacun de ces critères de choix n'a pas la même importance aux yeux de l'acheteur qui préférera par exemple un véhicule consommant peu à un véhicule qu'il juge très beau. Pour effectuer son choix, notre client va mentalement "noter" chaque alternative entre 0 et 1 en fonction de ses critères et retenir celle qui a obtenu la note la plus grande. Il s'agit donc de définir une fonction qui associe un réel de $[0, 1]$ à toute alternative, le réel attribué à l'alternative remplissant au mieux les critères étant le plus grand. Si l'acheteur est exigeant, il désirera que tous ses critères soient satisfaits au maximum et il réalisera donc une agrégation des critères en mode "et" : le prix le plus faible *et* la consommation la plus faible *et* la plus belle ligne. A l'extrême, un autre client pourra considérer que dès que l'un de ses critères est rempli, le véhicule est parfait. Il réalisera donc une agrégation des critères en mode "ou" : le prix est faible *ou* la consommation est faible *ou* l'esthétique est agréable. Dans le cas de l'achat d'une nouvelle voiture, il y a de fortes chances pour que le mode d'agrégation des critères se trouve quelque part entre ces deux extrêmes ! Réaliser ce type de notation dans un univers flou peut faciliter la modélisation de critères comme l'esthétique qui sont par nature difficilement représentables par un unique réel. Dans ce cas, il est tentant de déterminer la fonction de notation dans le corps des réels et de l'adapter avec ses propriétés au cas flou.

Cette adaptation est réalisée à l'aide du principe d'extension qui permet de calculer l'image d'une distribution de possibilité (ou d'un ensemble flou) par une fonction f de \mathbb{R}^n vers \mathbb{R}^p avec $n \geq 1, p \geq 1$ ce qui donne une autre distribution de possibilité (ou un autre ensemble flou), les propriétés connues sur f dans le corps des réels étant – lorsque faire se peut – propagées dans l'univers possibiliste (ou flou).

Dans un premier paragraphe, nous présentons les opérateurs OWA (Ordered Weighted Average) introduits par [Yager, 1988] et permettant d'évaluer l'intensité avec laquelle une alternative satisfait les exigences d'un ensemble de critères¹. Le second paragraphe détaille le principe d'extension imaginé par [Zadeh, 1977] pour calculer l'image d'une distribution de possibilité par une fonction réelle classique. Nous en décrivons

1. De nombreux travaux s'intéressent à cette évaluation. Ces opérations peuvent être comparés à des moyennes. La théorie de l'utilité est une des approches les plus utilisées dont [Fishburn, 1967] et [Vincke, 1967] sont d'excellentes sources d'information.

une implantation informatique possible, calculons la complexité de cet algorithme et présentons à titre d'illustration la moyenne de trois nombres flous obtenue à l'aide du principe d'extension. Le troisième paragraphe détaille l'utilisation du principe d'extension pour transformer toute fonction f du corps des réels à valeurs réelles (et donc tout opérateur OWA) en un opérateur de fusion entre distributions de possibilités (ou sous-ensembles flous). La notion de compatibilité entre les sources (proche de la notion d'indice de conflit décrite au chapitre 2.1 page 25) et la notion d'agrégation partielle sont alors introduites dans le processus de fusion. Muni de ces deux notions, la complexité de l'algorithme de fusion dépasse de loin les limites du raisonnable. Dans le quatrième paragraphe, nous présentons les intégrales floues.

4.1 L'agrégation de critères ou les opérateurs OWA

Nous nous intéressons ici aux opérateurs OWA introduits par YAGER en supposant implicitement que les objets à combiner sont des réels et que le résultat de la combinaison est un réel. D'autres opérateurs présentés par YAGER existent, ce sont les WM (Weighted Median) pour lesquels la fusion est possible sur tout type d'objet dès qu'une relation d'ordre existe entre eux. La définition de ces opérateurs [Yager, 1998] et leurs propriétés sont proches de celle des OWA.

Soit Γ l'ensemble de toutes les alternatives. La définition d'un opérateur OWA passe par la construction d'une fonction D permettant de noter les alternatives X et Y relativement à un ensemble de n critères C_1, \dots, C_n . L'intensité avec laquelle l'alternative X remplit les exigences exprimées par le critère C_i est désignée par $C_i(X) \in [0, 1]$. D est donc fonction des notes individuelles fournies par les $C_i(X)$:

$$D : \begin{cases} \Gamma \longrightarrow [0, 1] \\ X \longmapsto D(X) = F(C_1(X), \dots, C_n(X)) \end{cases} \quad F : \begin{cases} [0, 1]^n \longrightarrow [0, 1] \\ (x_1, \dots, x_n) \longmapsto F(x_1, \dots, x_n) \end{cases} \quad (4.1)$$

Pour effectuer correctement la notation, la fonction F doit présenter quelques propriétés. Tout d'abord, l'ordre dans lequel les notes de satisfactions des critères individuels $C_i(X)$ apparaissent ne doit pas avoir d'incidence sur le résultat de F . Etant donné n réels (a_1, \dots, a_n) de $[0, 1]$, P_1 et P_2 deux permutations quelconques des a_i , nous devons donc avoir $F(P_1) = F(P_2)$. Cette propriété est encore appelée commutativité généralisée. Par ailleurs, si une alternative satisfait mieux les critères qu'une autre, la satisfaction totale en cette alternative doit être plus grande. Mathématiquement, ces deux propriétés se traduisent par :

$$\begin{aligned} 1 \quad & \forall j \in \{1 \dots n\}, C_j(X) \geq C_j(Y) \implies D(X) \geq D(Y) \\ 2 \quad & \forall P_1, P_2 \text{ deux permutations des } a_i, F(P_1) = F(P_2) \end{aligned} \quad (4.2)$$

Enfin, F doit permettre de préciser le nombre de critères qui doivent être satisfaits. Trois cas sont à considérer :

1. tous les critères doivent être satisfaits ce qui suppose un comportement de type "et",
2. un au moins des critères doit être satisfait ce qui suppose un comportement de type "ou",
3. une situation intermédiaire aux deux extrêmes précédents où $1 \leq p \leq n$ critères doivent être satisfaits.

Pour forcer la satisfaction de tous les critères, il s'agit d'effectuer une opération "et" entre toutes les notes et nous avons déjà présenté au paragraphe 1.4.2.1 page 16 une famille d'opérateurs adaptés à ce travail : les T-Normes. T étant une T-Norme, nous pouvons alors poser $D(X) = T(C_1(X), \dots, C_n(X))$. Il est facile de vérifier que les T-Normes remplissent les conditions exprimées par (4.2). Le minimum étant la plus grande de toutes les T-Normes, nous avons toujours $T(a_1, \dots, a_n) \leq \min(a_1, \dots, a_n)$. Du point de vue de la décision, un critère faiblement satisfait ne peut être compensé par un autre même s'il est totalement satisfaisant.

Si un critère au moins doit être satisfait, il s'agit alors d'effectuer une opération "ou" entre les notes. Les opérateurs duaux des T-Normes (les T-Conormes présentées au paragraphe 1.4.2.1 page 16) sont adaptés. S étant une T-Conorme, nous pouvons alors poser $D(X) = S(C_1(X), \dots, C_n(X))$. Il est facile de vérifier que les T-Conormes remplissent les conditions exprimées par (4.2). Le maximum étant la plus petite de toutes les T-Conormes, nous avons toujours $S(a_1, \dots, a_n) \geq \max(a_1, \dots, a_n)$. Du point de vue de la décision,

la bonne satisfaction d'un critère suffit pour considérer l'alternative comme satisfaisant bien à l'ensemble de critères.

Le plus souvent, le preneur de décision désirera une réaction plus nuancée que les deux extrêmes que nous venons de présenter : "la plupart des critères doivent être satisfaits" ou "au moins la moitié des critères doivent être satisfaits" sont deux exemples classiques des besoins en matière de décision multicritère. Par ailleurs, fixer le nombre de critères à satisfaire n'est pas suffisant pour une bonne prise de décision. Certains critères sont plus importants que d'autres au yeux du preneur de décision et la fonction F doit en tenir compte. Il est difficile en utilisant les T-Normes et T-Conormes de prendre en compte à la fois le nombre de critères et leur importance relative. Cependant, les opérateurs OWA introduits dans [Yager, 1988] permettent la prise en compte de ces deux aspects.

Dans un premier paragraphe, nous expliquons comment ces opérateurs prennent en compte le nombre de critères devant être satisfaits avant d'aborder dans un second paragraphe la prise en compte de l'importance des critères.

4.1.1 F et le nombre de critères à vérifier

Un opérateur OWA de dimension n est une fonction F de $[0, 1]^n$ vers $[0, 1]$ (voir (4.1)) associée à un vecteur de pondération $W = [w_1, \dots, w_n]^T$ vérifiant :

$$\begin{aligned}
 &1. \quad \forall i, w_i \in [0, 1] \\
 &2. \quad \sum_{i=1}^n w_i = 1 \quad \text{et} \quad \forall (x_1, \dots, x_n) \in [0, 1]^n, F(x_1, \dots, x_n) = \sum_{i=1}^n w_i x_{(i)} = W.B \quad (4.3)
 \end{aligned}$$

Le vecteur $B = [b_1, \dots, b_n]^T$ (dit "vecteur ordonné") est construit à partir des x_i de telle sorte que b_j soit le $j^{\text{ème}}$ plus grand élément des x_i pour $i \in \{1..n\}$. Cela signifie que le vecteur B contient les arguments x_i du plus grand au plus petit, le plus grand étant à la position de b_1 et le plus petit à la position de b_n . Sous ces conditions, $F(\vec{x})$ est donc un élément de $[0, 1]$. En pratique, la "note" de l'alternative X relativement au vecteur W est donnée par $F(C_1(X), \dots, C_n(X))$.

L'opérateur défini par (4.3) est donc un simple produit scalaire entre le vecteur W et le vecteur B . Il est important de noter que le poids w_i est affecté au $i^{\text{ème}}$ plus grand des arguments x_i plutôt qu'à une position particulière. Cet opérateur vérifie les propriétés de base exprimées par (4.2), c'est-à-dire :

- 1 B_1 et B_2 étant deux vecteurs ordonnés, $\forall i, b_{1i} \geq b_{2i} \implies F(B_1) \geq F(B_2)$
- 2 F vérifie la commutativité généralisée car deux permutations ont le même vecteur ordonné (4.4)
- 3 F est idempotente, $\forall a \in [0, 1], F(a, \dots, a) = a$

Il existe trois vecteurs remarquables pour lesquels les opérateurs OWA correspondants ont des comportements déjà présentés au paragraphe 4.1 page précédente. Le premier agit comme une T-Norme (un "et"), le second effectue un calcul de moyenne (dont le paragraphe 1.4.2.2 page 17 décrit les propriétés) et le dernier agit comme une T-Conorme (un "ou"). Ces vecteurs particuliers sont résumés dans le tableau 4.1 :

Vecteur	Comportement	Remarques
$W_* = [0 \dots 0 \ 1]^T$	"et pur" (min)	la plus mauvaise note individuelle est affectée à l'alternative
$W_M = [\frac{1}{n} \dots \frac{1}{n}]^T$	"moyenne pure"	la moyenne des notes individuelles est affectée à l'alternative
$W^* = [1 \ 0 \dots 0]^T$	"ou pur" (max)	la meilleure note individuelle est affectée à l'alternative

TAB. 4.1 – Trois vecteurs OWA remarquables.

Numériquement, il est facile de s'apercevoir que W_* ne considère que la plus petite note individuelle, que W_M réalise la moyenne des notes individuelles et que W^* ne s'intéresse qu'à la plus grande note individuelle. Ainsi, pour tout vecteur ordonné A , $F_* \leq F(A) \leq F^*(A)$ (F_* et F^* étant les opérateurs OWA associés respectivement aux vecteurs W_* et W^*).

Lorsque le comportement désiré n'est ni un "et", ni une moyenne et ni un "ou", il est légitime de se demander comment calculer les coefficients du vecteur W . Pour cela, au moins deux techniques sont utilisables. La première est fondée sur le principe de l'apprentissage : disposant d'alternatives et des notes pour chacune d'elles, les w_i peuvent être déduits en utilisant une technique de régression. Cela implique la connaissance d'alternatives de tests, d'un modèle de régression et soulève tous les problèmes liés aux techniques d'apprentissages comme le choix de l'ensemble d'apprentissage, le choix du modèle de régression ... Une autre solution possible consiste à donner un sens aux poids w_i . Le preneur de décision pourra alors fournir directement leurs valeurs. C'est cette seconde approche que nous développons.

Supposons que l'alternative X vérifie complètement $k \leq n$ critères (le degré de satisfaction pour chacun de ces critères individuels est alors 1) et que les $n - k$ critères restants ne soient pas du tout vérifiés par X (le degré de satisfaction pour chacun de ces critères est alors 0). Le calcul de $F(C_1(X), \dots, C_n(X))$ est alors égal à la somme des w_i pour i dans $\{1 \dots n\}$. Définissons la suite S_n par :

$$S_k = \sum_{i=0}^k w_i \quad \text{avec} \quad \begin{array}{l} S_0 = 0 \\ S_n = 1 \end{array} \quad (4.5)$$

Dans notre cas particulier, $F(C_1(X), \dots, C_n(X)) = S_k$. La valeur de S_k peut donc être vue comme la satisfaction obtenue lorsqu'une alternative vérifie exactement k des n critères. Par ailleurs, puisque $S_k = S_{k-1} + W_k$, le coefficient W_k représente l'augmentation de satisfaction obtenue lorsque l'on passe de $k - 1$ critères totalement satisfaits à k critères totalement satisfaits.

Il est alors plus facile de préciser les valeurs des poids w_i en fonction du nombre de critères que l'on désire voir satisfaits. Pour tout $0 \leq i \leq n$, w_i représente l'augmentation de la satisfaction globale lorsque l'on passe de $i - 1$ critères vérifiés à i critères vérifiés. L'expression des poids w_i peut aussi se faire en fixant les n premiers termes de la suite S_k car $W_i = S_i - S_{i-1}$. Bien entendu, fixer les S_i peut parfois être aussi difficile que de déterminer les poids w_i .

Le comportement des opérateurs OWA variant de manière continue du "et" au "ou", il est intéressant de quantifier un vecteur OWA par son degré de "et" et son degré de "ou".

4.1.2 Quelques propriétés : la mesure du caractère "et"/"ou" d'un vecteur OWA

La lecture du tableau 4.1 suggère que le comportement des opérateurs OWA peut évoluer de manière continue entre le "et" et le "ou". Il semble intéressant de tenter de quantifier ce comportement en définissant un degré de "et" et un degré de "ou" duaux. Intuitivement, plus une grande fraction de la somme totale des w_i est située dans les premiers éléments du vecteur W , plus le comportement de l'opérateur se rapprochera d'un "ou" (obtenu avec $[1 \ 0 \ \dots \ 0]^T$). A l'inverse, plus une grande fraction de la somme des w_i sera proche du dernier élément w_n , plus l'opérateur aura le comportement d'un "et" (obtenu avec $[0 \ \dots \ 0 \ 1]^T$). La figure ci-dessous présente à gauche un vecteur qui donnera à l'opérateur OWA un comportement plutôt de type "ou" et à droite un vecteur qui donnera un comportement plutôt de type "et". La fraction de la somme des w_i et sa position dans le vecteur sont présentées :

$$\begin{array}{ccc} 1 & & n \\ \underbrace{[0.5 \ 0.3 \ 0 \ \dots \ 0.1 \ 0.1]^T}_{80\%} & & \underbrace{[0.1 \ 0 \ \dots \ 0 \ 0.9]^T}_{90\%} \end{array}$$

Le degré de "ou" est alors obtenu en calculant la fraction de la somme des coefficients apparaissant au début du vecteur W et en la comparant à la fraction totale. Le degré de "et" est calculé par dualité :

$$OR(W) = \frac{1}{n-1} \sum_{i=1}^n (n-i)w_i \quad AND(W) = 1 - OR(W) \quad (4.6)$$

Il est facile de vérifier que $AND(W_*) = 1$, que $OR(W_*) = 1$ et que pour un vecteur ayant un comportement de type moyenne, les degrés de "et" et de "ou" sont tous les deux égaux à $\frac{1}{2}$.

Dans la définition de $OR(W)$, il est intéressant de noter que lorsque $i = n$, le produit $(n - i)w_i = 0$. C'est intentionnellement que le dernier poids w_n n'est pas pris en compte dans cette définition de YAGER car le dernier poids w_n représente l'intensité du comportement "et". Il ne semble pas cohérent de l'introduire dans le calcul du comportement du degré de "ou".

La connaissance du degré de "ou" n'est pas encore suffisante. Considérons les vecteurs $[0\ 0\ 1\ 0\ 0]^T$ et $[\frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}\ \frac{1}{5}]^T$. Ces deux vecteurs ont le même degré de "ou" et pourtant, le premier ne s'intéresse qu'au degré de satisfaction du troisième critère alors que le second utilise toutes les entrées. La mesure de dispersion va permettre de différencier ces deux vecteurs :

$$DISP(W) = - \sum_{j=1}^n w_j \ln w_j \quad (4.7)$$

La dispersion telle qu'elle est définie ici est une mesure d'entropie. Cela implique qu'elle est maximale (et égale à $\ln n$) lorsque pour tout i , $w_i = \frac{1}{n}$ et minimale (et égale à 0) lorsque les w_i non nuls valent 1. D'une manière générale, plus W est dispersé, plus la valeur de chaque critère individuel est utilisé dans le calcul de la satisfaction d'une alternative.

Etant donné un degré de "ou" τ , il est possible de déterminer les coefficients w_i pour obtenir le vecteur W tel que $OR(W) = \tau$ en maximisant ζ :

$$DISP(W) = - \sum_{i=1}^n w_i \ln w_i \quad \text{sous les contraintes} \quad \begin{cases} \frac{1}{n-1} \sum_{i=1}^n w_i (n-i) = \tau \\ \sum_{i=1}^n w_i = 1, \forall i, w_i \geq 0 \end{cases} \quad (4.8)$$

Les opérateurs OWA disposent de nombreuses autres propriétés comme la compensation, la stabilité, la réduction à une intégrale de CHOQUET (voir à ce sujet la définition 12) et peuvent – sous certaines conditions – être décomposés. Le détail de ces propriétés et leurs démonstrations présentées dans [Fodor et al., 1995] dépassent le cadre de ce travail.

Toujours dans l'optique d'aider notre acheteur dans le choix de sa nouvelle voiture, nous abordons maintenant la dernière étape permettant de préciser l'importance des critères.

4.1.3 F et l'importance de chaque critère

La stratégie de calcul des w_i admet implicitement que chaque critère a la même importance aux yeux du preneur de décision. Chaque fonction d'évaluation d'un critère $C_i(X)$ est maintenant supposée être associée à un réel $\alpha_i \in [0, 1]$ indiquant le degré d'importance associé au critère i . La note de l'alternative X est alors donnée par $F(X) = F(H(C_1(X), \alpha_1), \dots, H(C_n(X), \alpha_n))$. Dans [Yager, 1988], H est définie par :

$$H(C_i(X), \alpha_i) = \max(\alpha_i, p) C_i(X)^{\max(\alpha_i, q)} \quad \text{avec} \quad \begin{cases} p = 1 - q \\ q = \text{degré de "ou" du vecteur } W \end{cases} \quad (4.9)$$

Pour un vecteur W ayant un comportement de type "et pur", $H(C_i(X), \alpha_i) = C_i(X)^{\alpha_i}$ et pour un vecteur W ayant un comportement de type "ou pur", $H(C_i(X), \alpha_i) = \alpha_i C_i(X)$. Lorsque tous les poids sont à 1, nous retrouvons le comportement précédent. Enfin, nous pouvons remarquer que la somme des coefficients α_i varie entre 0 et n .

D'autres techniques permettent de prendre en compte l'importance de chaque critère dans un opérateur de type OWA. D'une manière générale, toutes utilisent une pondération ; effectuée soit sur le résultat fourni par les fonctions d'évaluation C_i (comme c'est le cas dans la méthode de YAGER ci-dessus), soit sur les

coefficients w_i . Cette dernière solution est adoptée par la famille d'opérateurs WOVA (Weighted Ordered Weighted Average) décrits dans [Torra, 1997]. Le vecteur W est calculé en utilisant :

$$w_i = w^* \left(\sum_{j=1}^i p_j \right) - w^* \left(\sum_{j=1}^{i-1} p_j \right) \tag{4.10}$$

où w^* est une fonction² monotone croissante qui interpole les points de coordonnées $(\frac{i}{n}, \sum_{j=1}^i p_j)$ avec le point de coordonnées $(0, 0)$ et les p_i étant les éléments d'un vecteur ordonné que l'on peut interpréter comme l'importance de chaque critère. Il y a donc deux ensembles de coefficients à déterminer : les coefficients de la fonction d'interpolation w^* et les coefficients p_i . Les opérateurs WOVA se réduisent aux opérateurs OWA lorsque pour tout $i, p_i = \frac{1}{n}$

Nous disposons maintenant de tous les éléments pour aider notre automobiliste à faire son choix. Dans un premier temps, nous présentons la résolution de ce problème de décision sans tenir compte de l'importance des critères puis nous introduisons l'importance des critères avec la technique de YAGER.

4.1.4 Exemple

Notre automobiliste désire effectuer son choix sur quatre critères : le prix, la consommation, la disponibilité du modèle et l'allure (c'est-à-dire le plaisir visuel fourni par le modèle). Il est tout d'abord nécessaire de préciser quelles sont les alternatives disponibles et quelles sont les fonctions $C_i(X)$. Les alternatives sont résumées dans le tableau 4.2.

Alternative	Description			
	Prix	Consommation	Disponibilité	Esthétique
1	50000	6	Disponible	$\frac{1}{10}$
2	100000	7.5	Non disponible	$\frac{6}{10}$
3	135000	7	Non disponible	$\frac{4}{10}$
4	150000	12	Disponible	$\frac{9}{10}$

TAB. 4.2 – Les alternatives retenues.

Les fonctions $C_i(X)$ sont présentés dans le tableau 4.3 sont toutes linéaires elle sont suffisantes pour traduire sa manière de choisir une voiture. Il est possible d'utiliser des fonctions non linéaires. Une fonction

Alternative	Evaluation
Prix	$C_1(X) = 1 - \frac{\text{Prix}}{150000}$
Consommation	$C_2(X) = 1 - \frac{\text{Consommation}}{12}$
Disponibilité	$C_3(X) = 1$ si disponible 0 sinon
Esthétique	$C_4(X) = \text{Esthetique}$

TAB. 4.3 – Fonction d'évaluation de chaque critère.

d'évaluation doit donner un réel de $[0, 1]$, la meilleure note étant 1. Notre preneur de décision considère qu'en ce qui concerne le prix, la meilleure note doit être attribuée à une voiture gratuite³. Par conséquent, le critère de prix est entièrement validé uniquement si le vendeur offre la voiture. Par contre, notre acheteur désire pénaliser fortement la voiture la plus chère car elle dépasse le budget qu'il s'est fixé. Nous procédons d'une manière similaire pour le critère de consommation (la voiture idéale ne consomme pas). La disponibilité est un critère binaire et il est traité comme tel. L'esthétique est directement évaluée par l'acheteur.

Reste maintenant à calculer les poids w_i et à fixer l'importance des critères. Notre acheteur cherche la voiture la moins chère, consommant le moins, disponible et dont la ligne est agréable ! Il désire donc vérifier tous les critères. Nous fixons directement les coefficients w_i en fonction des désirs de l'automobiliste, en se

2. Ne pas confondre cette fonction avec le vecteur W^* désignant l'opérateur OWA au comportement de "ou" pur.

3. Cela est un peu utopiste mais certains acheteurs peuvent être tentés de se faire offrir le véhicule par un proche !

souvenant que w_i représente l'augmentation de satisfaction lorsque l'on passe de $i - 1$ à i critères entièrement vérifiés. La vérification d'un seul critère est trop faible pour notre acheteur et il donne dans ce cas la satisfaction de $w_1 = 0.1$. Lorsque deux critères sont remplis, l'acheteur considère l'alternative encore trop peu intéressante et la vérification d'un critère supplémentaire entraîne selon lui une augmentation de la satisfaction de $w_2 = 0.2$. La vérification de trois critères est jugée acceptable et mérite une augmentation de satisfaction de $w_3 = 0.3$. Enfin, la vérification d'un dernier critère est jugée importante. L'augmentation de satisfaction sera de $w_4 = 0.4$ ce qui donne le vecteur $W = [0.1 \ 0.2 \ 0.3 \ 0.4]^T$.

Alternative	Critères				Note
	Prix	Consommation	Disponibilité	Esthétique	
1	$\frac{2}{3}$	$\frac{1}{2}$	1	$\frac{1}{10}$	0.423
2	$\frac{3}{10}$	$\frac{4.5}{12}$	0	$\frac{6}{10}$	0.235
3	$\frac{1}{10}$	$\frac{10}{12}$	0	$\frac{4}{10}$	0.193
4	0	0	1	$\frac{9}{10}$	0.28

TAB. 4.4 – Les alternatives et leurs notes.

En utilisant (4.3) nous pouvons calculer la note attribuée à chaque alternative. Les résultats sont présentés dans le tableau 4.4. La voiture la moins onéreuse, disponible et consommant peu est bien celle qui a été retenue. Il est important de remarquer que même si cette voiture a une ligne peu agréable, elle est préférée aux autres car trois critères sur quatre sont nettement remplis.

Par contre, le modèle 4, cher et consommant beaucoup est pourtant préféré au modèle 3 consommant beaucoup moins et moins onéreux. Cela est dû à la grande satisfaction du critère "Esthétique" et à la disponibilité du modèle par rapport au modèle 3 non disponible et moins beau. L'excellente satisfaction de critères que l'on pourrait juger secondaires (l'esthétique et la disponibilité) a compensé la mauvaise performance obtenue par les critères que l'on peut supposer importants : le prix et la consommation. Pour fournir des résultats encore plus sûrs, l'introduction de l'importance des critères est indispensable.

Notre acheteur reformule ses règles de choix, ayant bien compris qu'il ne pouvait pas trouver une voiture disposant de toutes les qualités à la fois. Le modèle préféré sera économique à l'usage (faible consommation), peu cher à l'achat et éventuellement moins beau. La disponibilité devient finalement un "plus" qui peut contribuer à augmenter la satisfaction, mais ne doit pas pénaliser une alternative. L'importance des critères peut alors être : $\alpha_{\text{Prix}} = \alpha_{\text{Consommation}} = 1$ pour exprimer la très grande importance du prix à l'achat et de la consommation. L'esthétique sera faiblement prise en compte avec $\alpha_{\text{Esthétique}} = 0.3$ et la disponibilité très faiblement considérée en posant $\alpha_{\text{Disponibilité}} = 0.1$. Muni de ces poids et à l'aide de (4.3) et (4.9), nous pouvons évaluer la note de chaque alternative et grouper les résultats dans le tableau (4.5). L'importance des critères

Alternative	Critères				Note
	Prix	Consommation	Disponibilité	Esthétique	
1	$\frac{2}{3}$	$\frac{1}{2}$	1	$\frac{1}{10}$	0.473
2	$\frac{3}{10}$	$\frac{4.5}{12}$	0	$\frac{6}{10}$	0.231
3	$\frac{1}{10}$	$\frac{10}{12}$	0	$\frac{4}{10}$	0.211
4	0	0	1	$\frac{9}{10}$	0.195

TAB. 4.5 – Les alternatives retenues.

étant prise en compte, les alternatives sont maintenant classées d'une manière compatible avec l'intuition. L'alternative la plus chère (à l'achat et à l'utilisation) est bien la dernière.

Ces opérateurs ont aussi été utilisés avec succès dans des problèmes de segmentation de photos aériennes, notamment pour effectuer l'extraction de zones représentant des lacs, des routes, des forêts ... Le lecteur intéressé trouvera une description précise des algorithmes utilisés dans [Sobrevilla et al., 1996].

Les opérateurs OWA ne sont que des fonctions de \mathbb{R}^n vers \mathbb{R} et ils ne sont pas utilisables directement pour manipuler des sous-ensembles flous connus par leurs fonctions d'appartenance ou des distributions de possibilité. Le prochain paragraphe s'intéresse à l'extension des notions précédentes au cas flou.

4.2 Principe d'extension et opérateurs de fusion

Connaissant un sous-ensemble $A_{\mathbb{R}}$ de \mathbb{R} et une application f , il est relativement aisé de déterminer l'ensemble $f(A_{\mathbb{R}})$, image de $A_{\mathbb{R}}$ par f . Les propriétés connues sur la relation entre $A_{\mathbb{R}}$ et $f(A_{\mathbb{R}})$ sont exploitables directement car le corps des réels, $A_{\mathbb{R}}$ et son image par f sont connus par leurs éléments. A_X étant un sous-ensemble flou (ou une distribution de possibilité) d'un univers Ω_X , la détermination de $A_Y = f(A_X)$ est plus délicate. En général, A_X ne représente pas un élément x précis mais un ensemble flou auquel x appartient fortement. Les propriétés connues sur les relations entre A_X et un autre ensemble flou ne sont pas directement utilisables car Ω_X n'est pas connu par ses éléments mais au travers de ses sous-ensembles flous. L'adaptation des propriétés est réalisée à l'aide du principe d'extension proposé par [Zadeh, 1977].

Le problème posé est alors le suivant : étant donnés deux référentiels Ω_X et Ω_Y (éventuellement confondus) et une application f de Ω_X dans Ω_Y , comment déterminer le sous-ensemble A_Y lié à A_X par la relation $A_Y = f(A_X)$? La fonction d'appartenance de A_X étant notée μ_{A_X} , le principe d'extension permet de définir A_Y par sa fonction d'appartenance :

$$\forall y \in \Omega_Y, \mu_{A_Y}(y) = \begin{cases} \sup_{\{x \in \Omega_X / y=f(x)\}} \mu_{A_X}(x) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad (4.11)$$

Nous pouvons alors distinguer trois cas :

1. f est monodimensionnelle $f : \begin{cases} \mathbb{R} \longrightarrow \mathbb{R} \\ x \longmapsto y = f(x) \end{cases}$
2. f est multidimensionnelle $f : \begin{cases} \Omega_X = \Omega_{X_1} \times \dots \times \Omega_{X_n} \subseteq \mathbb{R}^n \longrightarrow \Omega_Y = \mathbb{R} \\ \vec{x} = (x_1, \dots, x_n) \longmapsto y = f(x) \end{cases}$
3. f est vectorielle $f : \begin{cases} \Omega_X = \Omega_{X_1} \times \dots \times \Omega_{X_n} \subseteq \mathbb{R}^n \longrightarrow \Omega_Y = \Omega_{Y_1} \times \dots \times \Omega_{Y_n} \subseteq \mathbb{R}^n \\ \vec{x} = (x_1, \dots, x_n) \longmapsto \vec{y} = (y_1, \dots, y_n) = f(x) \end{cases}$

Les paragraphes suivants détaillent l'expression de la distribution de possibilité pour chacun des trois cas.

4.2.1 Lorsque f est monodimensionnelle

En remplaçant la fonction d'appartenance μ_{A_X} par la distribution de possibilité $\pi_X(x)$, l'équation (4.11) devient :

$$\pi(y) = \begin{cases} \sup_{\substack{x=f^{-1}(y) \\ x \in \Omega_X}} \pi_X(x) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad \text{ou encore} \quad \pi(y) = \begin{cases} \sup_{\substack{y=f(x) \\ x \in \Omega_X}} \pi_X(x) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad (4.12)$$

En pratique il est souvent difficile d'accéder à f^{-1} et l'expression de droite ci-dessus est souvent préférée car seule f intervient. Lorsque f est bijective, l'expression précédente se réduit à $\pi(y) = \pi_X(h^{-1}(y))$.

4.2.2 Lorsque f est multidimensionnelle

En posant $A_Y = \{y\}$, l'équation (4.11) devient :

$$\pi(y) = \begin{cases} \sup_{\substack{\vec{x}=f^{-1}(y) \\ \vec{x} \in \Omega_X}} \pi_{X_1 \dots X_n}(\vec{x}) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad \text{ou} \quad \pi(y) = \begin{cases} \sup_{\substack{y=f(\vec{x}) \\ \vec{x} \in \Omega_X}} \pi_{X_1 \dots X_n}(\vec{x}) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad (4.13)$$

4. En écrivant $f(A_X)$, nous utilisons la même notation pour désigner l'application de Ω_X dans Ω_Y et celle de l'ensemble des parties de Ω_X vers l'ensemble des parties de Ω_Y . Cette notation abusive est cependant couramment employée dans la littérature. Nous utiliserons cette notation simplifiée pour l'application réciproque de f notée f^{-1} .

où $\pi_{X_1 \dots X_n}(\vec{x})$ désigne la distribution de possibilité conjointe des distributions π_1, \dots, π_n . En pratique il est souvent difficile d'accéder à f^{-1} et l'expression de droite ci-dessus (équivalente à l'expression de gauche) est souvent préférée car seule f intervient.

L'expression (4.13) peut encore être simplifiée lorsque les variables associées aux distributions de possibilités ne sont pas interactives. La notion de non-interaction entre deux variables possibilistes est équivalente à la notion d'indépendance pour les variables aléatoires :

Définition 4. *Non-interaction.*

Les variables possibilistes π_1, \dots, π_n sont dites non-interactives si leur distribution conjointe est égale au produit cartésien des distributions marginales π_{X_i} :

$$\pi_X(x) = \pi_{X_1 \dots X_n}(x) = \min_{(x_1, \dots, x_n) \in X_1 \times \dots \times X_n} (\pi_{X_1}(x_1), \dots, \pi_{X_n}(x_n))$$

Les variables étant supposées non-interactives, (4.13) s'écrit :

$$\pi(y) = \begin{cases} \sup_{\substack{\vec{x}=f^{-1}(y) \\ \vec{x} \in \Omega_X}} \min(\pi_{X_1}(x_1), \dots, \pi_{X_n}(x_n)) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad (4.14)$$

Ou encore lorsque l'accès à f^{-1} est impossible :

$$\pi(y) = \begin{cases} \sup_{\substack{y=f(\vec{x}) \\ \vec{x} \in \Omega_X}} \min(\pi_{X_1}(x_1), \dots, \pi_{X_n}(x_n)) & \text{si } f^{-1}(y) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad (4.15)$$

4.2.3 Lorsque f est vectorielle

En posant $A_Y = \{\vec{y}\}$, l'équation (4.11) devient :

$$\pi(\vec{y}) = \begin{cases} \sup_{\substack{\vec{x}=f^{-1}(\vec{y}) \\ \vec{x} \in \Omega_X}} \pi_{X_1 \dots X_n}(\vec{x}) & \text{si } f^{-1}(\vec{y}) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad \text{ou } \pi(\vec{y}) = \begin{cases} \sup_{\substack{\vec{y}=f(\vec{x}) \\ \vec{x} \in \Omega_X}} \pi_{X_1 \dots X_n}(\vec{x}) & \text{si } f^{-1}(\vec{y}) \neq \emptyset \\ 0 & \text{sinon} \end{cases} \quad (4.16)$$

l'expression de droite étant adéquate lorsque l'accès à f^{-1} est impossible. Dans le cas vectoriel, l'opérateur sup doit calculer le suprémum d'une famille de vecteurs de \mathbb{R}^n et plusieurs définitions existent. La plus classique consiste à choisir le vecteur de la famille ayant la plus grande norme.

Avant de nous intéresser d'une manière plus formelle à la complexité de ces algorithmes, nous comprenons bien que le principe d'extension est extrêmement coûteux en temps de calcul et d'une manière générale (4.16) est en pratique incalculable en un temps raisonnable lorsque n dépasse 5. L'idée de simplifier (4.16) en considérant un problème de dimension n comme n problèmes de dimension 1 est en général à éviter⁵ bien qu'elle soit parfois possible, lorsque la fonction f n'introduit pas de perte de non-interactivité.

Le principe d'extension tel qu'il est présenté ci-dessus est difficile à implémenter informatiquement car pour obtenir $\pi(y)$, il faut déterminer l'ensemble $\Gamma = \{x \in \Omega_X / y = f(x)\}$ et calculer le suprémum de $\pi_X(x)$ pour x variant dans Γ . Il existe cependant une autre technique pour traduire informatiquement les définitions précédentes. En faisant varier x dans l'ensemble de départ, nous pouvons calculer $\pi(f(x)) = \pi_X(x)$. Lorsque x_1 et x_2 distincts ont la même image par f , il suffit d'affecter à $\pi(f(x))$ le plus grand degré de possibilité c'est à dire $\max(\pi_X(x_1), \pi_X(x_2))$. Si nous supposons la distribution de possibilité résultante initialisée à

5. Supposons par exemple que f soit une l'application affine $f(x) = Ax + b$ où A est une matrice de rotation et b un vecteur de translation. Dans ce cas $\pi(y) = \pi_X(A^{-1}(y - b))$. La distribution de possibilité étendue est donc une "copie conforme" de π_X mais qui a pivoté et été déplacée. Si π_X était non-interactive, le résultat ne l'est plus en général et il devient alors impossible d'utiliser les distributions de possibilité marginales.

0, le principe d'extension s'écrit alors : "pour tout x de Ω_X , $\pi(f(x)) = \max(\pi_X(x), \pi(f(x)))$ ". Lorsque f est bijective, l'espace dans lequel x doit évoluer est réduit au support de la distribution π_X ce qui est particulièrement intéressant pour diminuer le coût en temps de calcul du principe d'extension. L'extension de π_A pour f monodimensionnelle est alors traduite informatiquement figure 4.1.

```

1      Extension de  $A$  par  $f$ 
2       $\pi_A$  : distribution de possibilité de  $A$  sur  $\Omega_X$ 
3       $f$  : fonction à étendre
4      function Extension( $\pi_A$  : red-black tree,  $f$  : function) : red-black tree
5          var  $\pi_B$  [] : red-black tree = 0.0
6          begin
7              for  $\omega$  in  $\Omega$  do
8                   $\pi_B[f(\omega)] = \max(\pi_B[f(\omega)], \pi_A[\omega])$ 
9              done
10             return  $\pi_B$ 
11         end

```

FIG. 4.1 – L'algorithme du principe d'extension – f monodimensionnelle.

Le "red black tree" utilisé dans l'algorithme est une structure associative permettant de stocker une donnée associée à une clef. Il ne peut y avoir qu'une seule donnée associée à une clef et toutes les clefs sont différentes. Il doit exister une relation d'ordre total sur les clefs ce qui permet d'ordonner de manière totale tous les éléments dans la structure. Pour nous, la clef sera un réel (la valeur de x) et la donnée sera le degré de possibilité de x dans l'ensemble. Le "red black tree" pouvant grandir au fur et à mesure de l'insertion de nouvelles données, nous ne sommes pas limités par le nombre de points comme cela serait le cas dans un tableau. Le chapitre 6 donne de plus amples informations sur le stockage informatique des distributions de possibilité (ou des fonctions d'appartenance).

Lorsque π_A n'est pas une distribution de possibilité discrète, l'espace peut être discrétisé et les ω choisis tous les $\delta \approx 10^{-3}$ par exemple. Cette valeur dépend de la précision recherchée et elle n'est donnée ici qu'à titre indicatif. La complexité de cet algorithme est $O(n)$ où n est le nombre des ω ayant fait l'objet d'un calcul.

```

1      Extension de  $\pi_{X_1}, \dots, \pi_{X_n}$  (interactives) par  $f$ 
2       $\pi_{X_1 \dots X_n}$  : distribution de possibilité conjointe sur  $\Omega_X$ 
3       $f$  : fonction à étendre
4      function Extension( $\pi_{X_1 \dots X_n}$  : red-black tree,  $f$  : function) : red-black tree
5          var  $\pi_B$  [] : red-black tree = 0.0
6          begin
7              for  $\omega_1$  in  $\Omega_{X_1}$  do
8                  for  $\omega_2$  in  $\Omega_{X_2}$  do
9                      ...
10                     for  $\omega_n$  in  $\Omega_{X_n}$  do
11                          $\pi_B[f(\omega_1, \dots, \omega_n)] = \max(\pi_{X_1 \dots X_n}[\omega_1, \dots, \omega_n], \pi_B[f(\omega_1, \dots, \omega_n)])$ 
12                     done
13                 ...
14             done
15         done
16         return  $\pi_B$ 
17     end

```

FIG. 4.2 – L'algorithme du principe d'extension – f multidimensionnelle et $\pi_{X_1}, \dots, \pi_{X_n}$ interactives.

L'algorithme de calcul pour f multidimensionnelle en supposant les distributions en interaction est présenté figure 4.2. Il y a n variables différentes donc n boucles différentes et éventuellement n pas de discrétisation différents. En supposant que chaque boucle "for" soit faite p fois (p lié au pas de discrétisation choisi), la complexité de ces algorithmes est $O(n^p)$. L'ajout d'une variable supplémentaire multiplie la complexité par

n . En pratique, le temps nécessaire à l'exécution de cet algorithme dépasse les limites du raisonnable pour $p \geq 5$. Lorsque les distributions sont non-interactives, l'algorithme de la figure 4.3 peut être employé. Pour les distributions non-interactives, nous fournissons à la fonction les red black tree en paramètre. Pour des raisons de taille de pile, le passage d'un red black tree à une fonction doit se faire par référence uniquement.

```

1   Extension de  $\pi_{X_1}, \dots, \pi_{X_n}$  par  $f$ 
2    $\pi_{X_1}, \dots, \pi_{X_n}$  :  $n$  distributions de possibilité sur  $\Omega_X$ 
3    $f$  : fonction à étendre
4   function Extension( $\pi_{X_1}, \dots, \pi_{X_n}$  : red-black tree,  $f$  : function) : red-black tree
5       var  $\pi_B$  [] : red-black tree = 0.0
6       begin
7           for  $\omega_1$  in  $\Omega_{X_1}$  do
8               for  $\omega_2$  in  $\Omega_{X_2}$  do
9                   ...
10                  for  $\omega_n$  in  $\Omega_{X_n}$  do
11                       $\pi_B[f(\omega_1, \dots, \omega_n)] = \max(\min(\pi_{X_1}[\omega_1], \dots, \pi_{X_n}[\omega_n]), \pi_B[f(\omega_1, \dots, \omega_n)])$ 
12                  done
13              ...
14          done
15      done
16      return  $\pi_B$ 
17  end

```

FIG. 4.3 – L'algorithme du principe d'extension – f multidimensionnelle et $\pi_{X_1}, \dots, \pi_{X_n}$ non interactives.

L'algorithme de calcul pour f vectorielle est donné figure 4.4. La structure de données utilisée pour stocker

```

1   Extension de  $\pi_X$  (vectorielle) par  $f$ 
2    $\pi_X(x_1, \dots, x_n)$  : distribution de possibilité sur  $\Omega_X$ 
3    $f$  : fonction à étendre
4   function Extension( $\pi_X$  : multimap[ $n$ ],  $f$  : function) : multimap[ $n$ ]
5       var  $\pi_B$  [] : multimap[ $n$ ] = 0.0
6       begin
7           for  $\omega_1$  in  $\Omega_{X_1}$  do
8               for  $\omega_2$  in  $\Omega_{X_2}$  do
9                   ...
10                  for  $\omega_n$  in  $\Omega_{X_n}$  do
11                       $\pi_B[f(\omega_1, \dots, \omega_n)] = \max(\pi_X[\omega_1, \dots, \omega_n], \pi_B[f(\omega_1, \dots, \omega_n)])$ 
12                  done
13              ...
14          done
15      done
16      return  $\pi_B$ 
17  end

```

FIG. 4.4 – L'algorithme du principe d'extension – f vectorielle.

la distribution de possibilité vectorielle $\pi_X(\vec{x})$ est la "multimap" d'ordre n . Elle permet d'associer n clefs (dans notre cas n réels) à un seul élément (dans notre cas le degré de possibilité). L'opérateur max de la ligne 11 fournit le plus grand de deux vecteurs, c'est-à-dire celui dont la norme est la plus grande. De nombreux choix sont possibles pour le calcul de la norme, mais souvent, la norme euclidienne est retenue :

$$E(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (4.17)$$

En supposant que chaque boucle "for" soit faite p fois, la complexité de cet algorithme est $O(n^p)$ et il est plus lent que le précédent car il nécessite le calcul d'une norme à chaque étape, ce qui ralentit encore le processus.

En pratique, pour ne pas trop augmenter le temps de calcul, l'expression de la norme euclidienne est élevée au carré, faisant ainsi gagner le temps de calcul de la racine carrée. En pratique, le temps nécessaire à l'exécution de cet algorithme dépasse les limites du raisonnable pour $p \geq 5$.

Pour éviter la répétition inutile de l'opérateur sup, nous utiliserons une notation plus compacte proposée par [Yager et Kelman, 1996] où le numérateur représente l'argument de l'opérateur sup et le dénominateur représente l'élément de Ω_Y considéré :

$$A_Y = \left\{ \frac{\text{Degré de possibilité}}{\text{Élément de } \Omega_Y} \right\}_{\text{Domaine}} \quad \text{soit pour } f \text{ monodimensionnelle} \quad A_Y = \left\{ \frac{\pi_X(x)}{f(x)} \right\}_{x \in \Omega_X} \quad (4.18)$$

Il est alors possible de noter le principe d'extension pour f multidimensionnelle et vectorielle par :

$$A_Y = \left\{ \frac{\pi_{X_1 \dots X_n}(\vec{x})}{f(\vec{x})} \right\}_{\vec{x} \in \Omega_X} \quad A_Y = \left\{ \frac{\min(\pi_{X_1}(x_1), \dots, \pi_{X_n}(x_n))}{f(\vec{x})} \right\}_{\vec{x} \in \Omega_X} \quad A_Y = \left\{ \frac{\pi_X(x)}{f(\vec{x})} \right\}_{\vec{x} \in \Omega_X} \quad (4.19)$$

4.2.4 Exemple

Considérons le dispositif industriel présenté figure 4.5. Une cuve contenant un liquide est équipée d'un dispositif de brassage, d'une arrivée en haut de cuve et d'une vidange en fond de cuve. Ce système est équipé

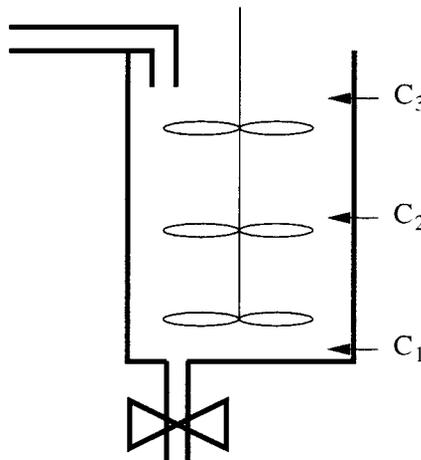


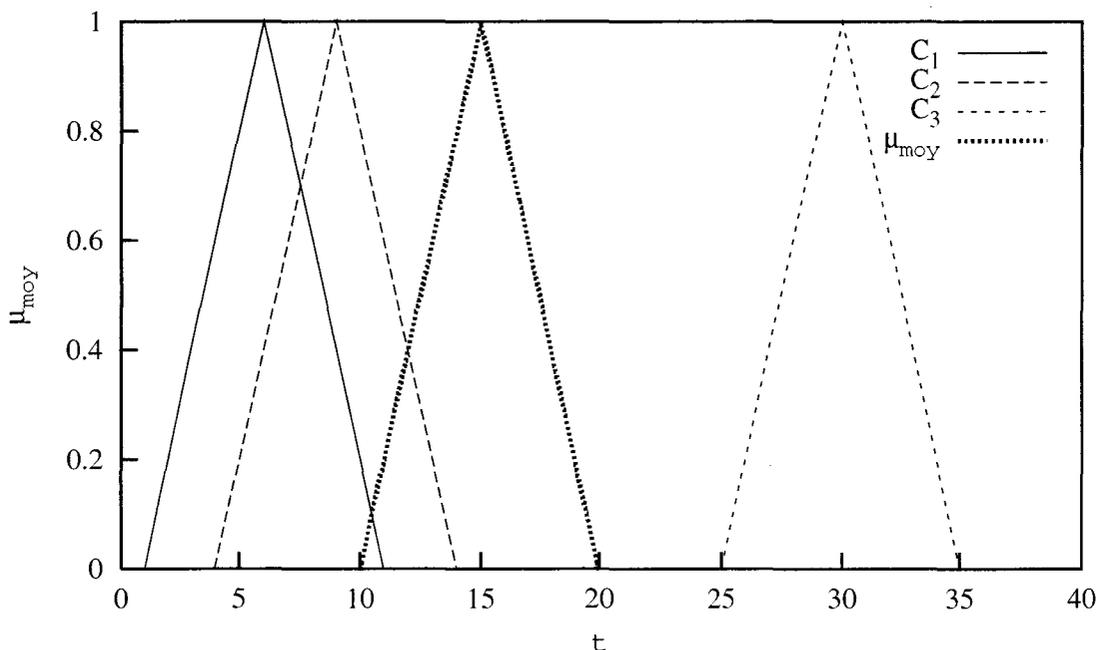
FIG. 4.5 – La cuve et ses capteurs de température.

de trois capteurs de température C_1 , C_2 et C_3 qui mesurent respectivement la température du liquide au fond, au milieu et au sommet. Supposons que ces capteurs fournissent un nombre flou défini par sa fonction d'appartenance. Nous nous intéressons ici à la combinaison des informations fournies par les trois capteurs.

Pour obtenir la température du liquide, la première idée consiste à calculer la moyenne des trois valeurs C_1 , C_2 , C_3 ce qui donnera une idée globale de la température dans la cuve. La fonction de fusion dans le corps des réels est donc $f(c_1, c_2, c_3) = \frac{c_1 + c_2 + c_3}{3}$ et il faut l'étendre au cas flou. Dans cet exemple particulier, nous manipulons des nombres flous, les distributions de possibilités seront donc remplacées par des fonctions d'appartenance sans perte de généralité. Le principe d'extension s'écrit alors :

$$\mu_{temp} = \left\{ \frac{\min(\mu_{C_1}(c_1), \mu_{C_2}(c_2), \mu_{C_3}(c_3))}{\frac{c_1 + c_2 + c_3}{3}} \right\}_{\vec{c} = (c_1, c_2, c_3) \in \mathbb{R}^3} \quad (4.20)$$

où les μ_{C_i} représentent les fonctions d'appartenance des valeurs fournies par les capteurs C_i . La figure 4.6 présente le résultat du principe d'extension⁶ pour trois nombres flous C_1 , C_2 et C_3 . Nous obtenons bien un résultat cohérent et concordant avec l'intuition : un nombre flou qui est la moyenne des trois nombres

FIG. 4.6 – Moyenne arithmétique “étendue” des trois nombres flous C_1 , C_2 et C_3 .

flous étendus. Intuitivement, nous avons fusionné ces trois nombres flous avec une moyenne arithmétique. Le principe d’extension permet donc de passer d’une fonction réelle à une fonction “floue” tout en maintenant ses propriétés. Nous pouvons alors facilement obtenir un opérateur de fusion “flou” à partir d’un opérateur de fusion sur des réels. Toute fonction f réelle est alors potentiellement un opérateur de fusion de données floues. Comme tout opérateur de fusion, f doit posséder certaines propriétés dont la définition a déjà été abordée (les numéros entre parenthèses renvoient à la propriété correspondante au paragraphe 1.3).

$$\begin{aligned}
 (10) \text{ Monotonie} & : \forall i \in \{1 \dots n\}, x_i \geq y_i, \implies f(x_1, \dots, x_n) \geq f(y_1, \dots, y_n) \\
 (2) \text{ Commutativité} & : \forall \vec{x} = (x_1, \dots, x_n) \in \mathbb{R}^n \text{ et } \vec{y} \text{ toute permutation des } x_i, f(\vec{x}) = f(\vec{y}) \\
 (6) \text{ Idempotence} & : \forall i \in \{1 \dots n\}, x_i = a \implies f(x_1, \dots, x_n) = a \\
 (9) \text{ Continuité} & : \forall \vec{x}_0 \in \mathbb{R}^n, \lim_{\vec{x} \rightarrow \vec{x}_0} f(\vec{x}) = f(\vec{x}_0)
 \end{aligned} \tag{4.21}$$

Etant donnée une fonction f vérifiant les propriétés de (4.21), l’opérateur de fusion s’exprime alors par (pour des distributions en interaction à gauche et pour des distributions non interactives à droite) :

$$\pi_F = \left\{ \frac{\pi_{X_1 \dots X_n}(\vec{x})}{f(\vec{x})} \right\}_{\vec{x} \in \Omega_X} \quad \pi_F = \left\{ \frac{\min(\pi_{X_1}(x_1), \dots, \pi_{X_n}(x_n))}{f(\vec{x})} \right\}_{\vec{x} \in \Omega_X} \tag{4.22}$$

Les opérateurs OWA décrits au paragraphe 4.1 sont des fonctions monotones, continues, idempotentes et commutatives. Elles sont donc utilisables comme fonction de fusion dans (4.22).

Il est important de noter que comme dans tous les opérateurs de fusion, la fiabilité du résultat est fortement conditionnée par la qualité des entrées. Rien dans la définition de l’opérateur 4.1 ne l’empêche d’effectuer la combinaison d’informations lorsque cela est aberrant. Le résultat est indépendant d’une quelconque mesure de conflit (voir 4.22). Par conséquent, (4.22) est indépendant du contexte et à comportement constant (CICB) comme une T-Norme et cela avec un coût en temps de calcul nettement plus élevé qu’une simple T-Norme. Le prochain paragraphe s’attache à introduire la notion de compatibilité dans cet opérateur.

4.2.5 La compatibilité entre les sources

Sur la figure 4.6, la valeur fournie par C_3 semble être incompatible avec celles fournies par C_1 et C_2 . L’idée la plus simple consiste à évaluer la compatibilité entre toutes les sources et à diminuer la confiance

6. Le calcul de la moyenne de ces nombres flous peut aussi être obtenu avec des méthodes de calcul sur des nombres flous comme cela est détaillé dans [Dubois et Prade, 1988].

accordée au résultat en fonction de cette compatibilité. Si les sources étaient compatibles, la confiance est maximale et lorsque toutes les sources ne sont compatibles qu'avec elles-mêmes, la confiance doit être nulle. La compatibilité est une mesure de similarité (voir 3.5.1 page 56) possédant une propriété supplémentaire.

Définition 5. Compatibilité.

Une fonction $R : \Omega_X^2 \rightarrow [0, 1]$ est une mesure de compatibilité sur Ω_X^2 si elle vérifie :

$$\begin{aligned} \forall x \in \Omega_X, R(x, x) &= 1 \\ \forall (x, y) \in \Omega_X^2, R(x, y) &= R(y, x) \\ \forall (x, y) \in \Omega_X^2, d(x, z) \geq d(x, y) &\implies R(x, z) \leq R(x, y), d \text{ distance sur } \Omega_X, \end{aligned} \quad (4.23)$$

Comme pour la similarité, la compatibilité de $n > 2$ éléments est définie à l'aide d'une fonction qui combine les mesures de compatibilité calculées entre tous les éléments pris deux à deux. Le choix fait par YAGER est celui de la prudence puisque la compatibilité de n éléments est la plus petite des compatibilités.

Définition 6. Compatibilité d'ordre n .

Une fonction $R_n : \Omega_X^n \rightarrow [0, 1]$ est une mesure de compatibilité sur Ω_X^n si $n \geq 2$ et si elle vérifie $R(x_1, \dots, x_n) = \min_{i,j \in \{1..n\}} R(x_i, x_j)$, R étant une mesure de compatibilité sur Ω_X^2 .

L'évaluation de la compatibilité entre n éléments peut être coûteuse en temps de calcul mais les propriétés de R^n permettent de ne pas effectuer le calcul pour toutes les combinaisons, mais seulement pour $\frac{n(n-1)}{2}$. Nous devons pondérer le degré de possibilité avec le résultat de la compatibilité. Il suffit donc de calculer le minimum entre le degré de compatibilité et le degré d'appartenance qui aurait été affecté si la compatibilité n'avait pas été introduite. L'expression de l'opérateur de fusion devient alors [Yager et Kelman, 1996] :

$$\pi_F = \left\{ \frac{\min(R(\vec{x}), \min(\pi_{X_1}(x_1), \dots, \pi_{X_n}(x_n)))}{f(\vec{x})} \right\}_{\vec{x} \in \Omega_X} \quad (4.24)$$

Lorsque toutes les informations sont compatibles, tout couple (x, y) de Ω_X^2 est donc tel que $R(x, y) = 1$ et l'expression (4.24) se réduit alors au principe d'extension. Dans le cas où les informations sont toutes incompatibles, $R(x, y) = 0$ pour tout couple (x, y) tel que $x \neq y$ et (4.24) se réduit à l'intersection des distributions de possibilités.

Toujours en considérant les nombres flous de la figure 4.6, introduisons maintenant la notion de compatibilité dans la fusion. Nous devons définir une fonction R . L'analyse du processus industriel révèle que le liquide doit être conservé aux alentours de 10° . Deux températures séparées de plus de 20° ne seront pas compatibles. Un expert exprime donc la compatibilité à l'aide de la fonction R (qui en l'occurrence n'est pas continue) :

$$R(x, y) = \begin{cases} 0 & \text{si } |x - y| > 20 \\ 1 - 0.005|x - y| & \text{si } |x - y| \leq 20 \end{cases} \quad (4.25)$$

Les résultats de la fusion des nombres flous C_1 , C_2 et C_3 de la figure 4.6 avec l'opérateur (4.24) sont présentés figure 4.7. La confiance a diminué et l'ensemble auquel la température appartient s'est réduit, l'information est donc un peu plus spécifique. Cela est dû à la mauvaise compatibilité entre C_3 et les autres nombres flous. La fonction d'appartenance est "décapitée" à la hauteur de 0.2 valeur de la compatibilité entre les trois nombres flous. Cette compatibilité est fixée par le nombre flou C_3 . L'idéal serait cependant de traiter les données qui semblent erronées avec une importance moindre par rapport aux autres. Cette notion de choix parmi les sources au moment de la combinaison est nommée combinaison partielle. Le paragraphe suivant s'intéresse à sa mise en œuvre.

4.2.6 L'agrégation partielle

Dans un processus de fusion, certaines entrées sont parfois clairement identifiables comme étant hors des limites et ne peuvent que perturber la combinaison. Nous avons déjà proposé au chapitre 3.5 une méthode

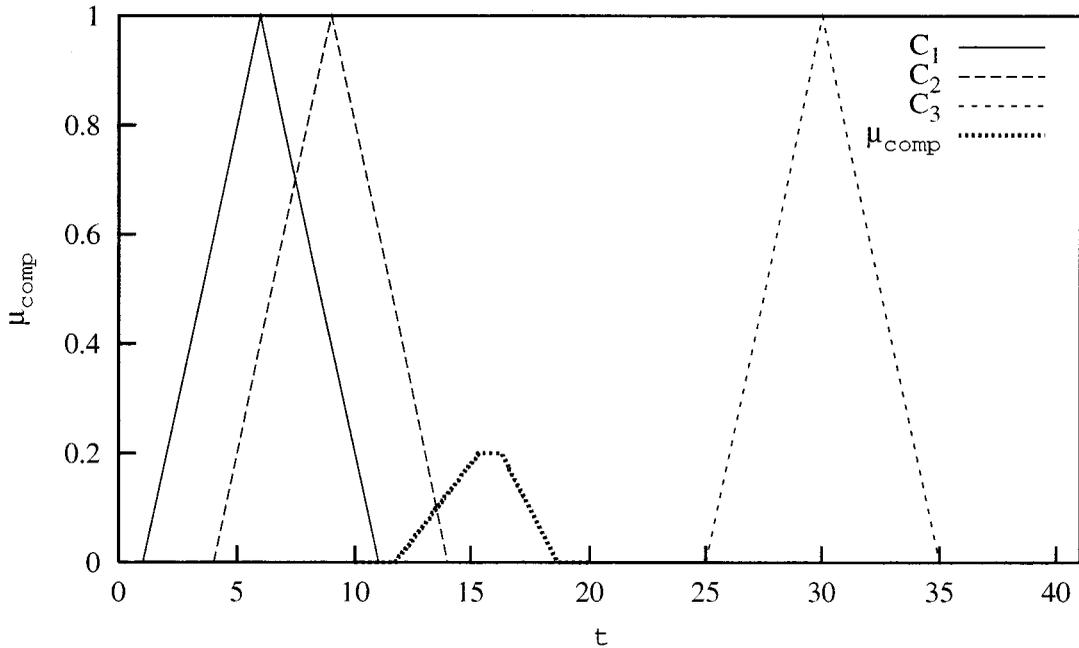


FIG. 4.7 – Moyenne arithmétique “étendue” des trois nombres flous C_1 , C_2 et C_3 avec compatibilité.

pour éliminer les sources incohérentes en formant des groupes de sources similaires et écartant les groupes incompatibles. Cette opération était une étape préliminaire à la combinaison proprement dite. Nous nous intéressons maintenant à une technique capable de considérer l'importance des sources directement pendant la phase de combinaison.

Il existe plusieurs manières d'introduire la partialité dans la fusion. Nous décrivons ici la méthode proposée par YAGER qui utilise la notion de mesure floue. Une mesure floue peut être interprétée comme une évaluation de la confiance que l'on a en un sous-ensemble d'un même univers mesurable Ω_X . Il s'agit donc d'une fonction de l'ensemble des parties de Ω_X vers $[0, 1]$.

Définition 7. Mesure Floue.

Une mesure floue $\nu : \mathbb{P}(\Omega_X) \rightarrow [0, 1]$ sur un espace mesurable Ω_X est une fonction vérifiant :

$$\begin{aligned} \nu(\emptyset) &= 0, \nu(\Omega_X) = 1 \\ \forall (X_i, X_j) \in \mathbb{P}(\Omega_X)^2, X_i \subseteq X_j &\implies \nu(X_i) \leq \nu(X_j) \end{aligned} \quad (4.26)$$

Pour introduire l'agrégation partielle, désignons par A_1, \dots, A_n les sources à fusionner (qu'il s'agisse de sous-ensembles flous ou de distributions de possibilité). Tous ces ensembles sont des sous-ensembles d'un même univers Ω_X . Si T désigne un groupement de sources quelconque (par exemple $T = \{A_1, A_2\}$), $\nu(T)$ indiquera la confiance que l'on a dans le résultat de la combinaison des éléments de T (par exemple de A_1 et A_2). Plus $\nu(T)$ sera grand plus nous considérerons la combinaison de ses éléments fiable. D'une manière plus théorique, nous notons I_T l'ensemble des indices d'un groupement de sources T . Ainsi, étant donné $T, I_T = \{i \in \mathbb{N} / A_i \in T\}$. Pour chaque élément T de $\mathbb{P}(\Omega_X)$, nous calculons la combinaison des sources contenues dans T , nommé B_T :

$$B_T = \left\{ \frac{\min(R(\vec{x}), \min_{i \in I_T} \pi_i(x_i))}{f(\vec{x})} \right\}_{\vec{x} \in \prod_{i \in I_T} x_i} \quad (4.27)$$

Le résultat final est obtenu en combinant les résultats partiels à la hauteur de la confiance que l'on a en eux :

$$\forall y \in \Omega_Y, \pi_F(y) = \max_{T \in \mathbb{P}(\Omega_X)} \min(B_T(y), \nu(T)) \quad (4.28)$$

ou encore en utilisant la notation de YAGER pour le principe d'extension :

$$\pi_F = \left\{ \frac{\min(\nu(T), R(\vec{x}), \min_{i \in I_T} \pi_i(x_i))}{f(\vec{x})} \right\}_{T \in \mathbb{P}(\Omega_X), \vec{x} \in \prod_{i \in I_T} x_i} \quad (4.29)$$

Intéressons nous à la complexité de cet algorithme. Nous devons calculer les 2^n agrégés partiels à l'aide d'une technique comparable à celle de l'équation (4.24). Supposons que l'entier p désigne le nombre de pas de chaque boucle (toutes les boucles sont supposées avoir le même pas de discrétisation). Il faut réaliser 2^n fois un calcul dont la complexité varie entre p^2 (le cas le plus favorable où il y a deux sources dans T) et p^n (le cas le plus défavorable où il y a n sources dans T) de sorte que la complexité pour évaluer tous les B_T est $O(2^n \frac{p^2+n^p}{2})$ ce qui est déjà énorme. Cela étant fait, il faut maintenant calculer le résultat à l'aide du principe d'extension. Finalement, l'algorithme décrit par YAGER donne d'excellents résultats théoriques mais, en pratique, il est inutile d'essayer de mettre en œuvre cet algorithme lorsque le nombre de sources dépasse trois. Cette solution s'adapte cependant aux nombres flous pour lesquels le calcul du principe d'extension peut être remplacé par un calcul d'intervalles [Dong et Wong, 1987]. Ce calcul n'est cependant utilisable que lorsque les fonctions d'appartenance (ou les distributions de possibilité) ne sont pas multimodales.

Toujours en considérant les nombres flous de la figure 4.6, introduisons maintenant la notion d'agrégation partielle dans la fusion. Nous devons définir une fonction ν . Nous définissons ici une mesure floue basée sur le cardinal de l'ensemble. Lorsqu'il s'agit de combiner une seule source, nous n'avons aucune confiance (pourquoi utiliser un processus de fusion si l'on ne considère qu'une source). Lorsqu'il s'agit de combiner deux ou trois sources, nous attachons toute notre confiance au groupement.

$$\nu(T) = \begin{cases} 0 & \text{si } |T| < 2 \\ 1 & \text{si } |T| \geq 2 \end{cases} \quad (4.30)$$

Les résultats de la fusion des nombres flous C_1 , C_2 et C_3 de la figure 4.6 page 75 avec l'opérateur (4.29) sont

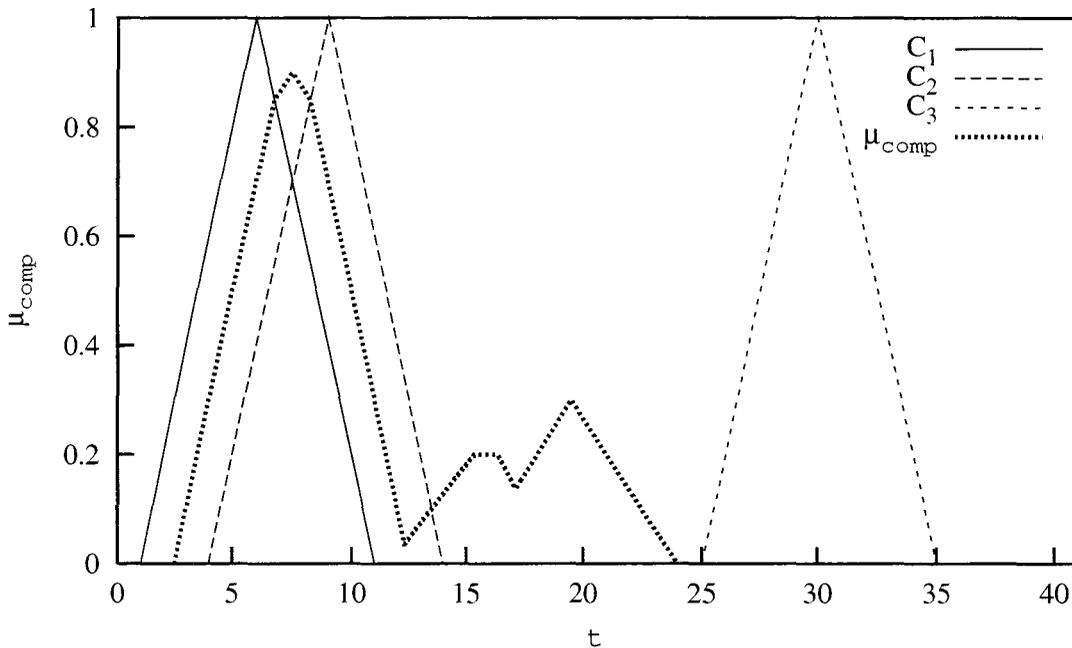


FIG. 4.8 – Fusion de C_1 , C_2 et C_3 avec compatibilité et agrégation partielle.

présentés figure 4.8. La complexité et l'espace mémoire nécessaire rendent l'algorithme de YAGER difficilement utilisable en pratique. Le prochain paragraphe s'intéresse à la fusion par intégrales floues.

4.3 Les intégrales floues

Le calcul d'une intégrale dans \mathbb{R} peut être vu comme un processus de fusion de données qui combine les points d'une courbe pour obtenir une surface ou un volume. Les intégrales floues [Sims et Wang, 1996], [Grabisch et al., 1995] sont une extension et une généralisation des intégrales réelles. Le but de ce paragraphe est de présenter succinctement les intégrales floues sur des espaces discrets et de montrer qu'un opérateur OWA est un cas particulier d'une intégrale floue.

Les intégrales floues sont définies à partir des mesures floues déjà abordée au paragraphe 7. La définition de l'intégrale est intimement liée à la mesure floue choisie et il existe donc potentiellement un très grand nombre d'intégrales floues. Les plus connues sont l'intégrale de SUGENO et l'intégrale de CHOQUET qui furent les premières publiées. L'intégrale à T-Conormes plus générale a été présentée plus tard. Dans ce paragraphe, $X_1, \dots, X_n \in \Omega_X^n$ représentent n sources d'informations. Les paragraphes suivants rappellent l'expression de quelques mesures floues et celles des trois types d'intégrales floues.

4.3.1 Quelques mesures floues

Une mesure floue est une fonction associant un ensemble de sous-ensembles à un réel. Ainsi, pour tout ensemble discret Ω_X comportant n éléments, l'ensemble des parties $\mathbb{P}(\Omega_X)$ comporte alors $2^n - 1$ éléments à chacun desquels la mesure floue doit associer un réel. Pour faciliter la mise en œuvre, de nombreux auteurs ont défini des mesures floues permettant de calculer les 2^n réels uniquement à l'aide de la connaissance de n d'entre eux judicieusement choisis. Par exemple, étant donnée la confiance en chaque source c'est-à-dire la valeur de la mesure floue pour tous les singletons $\{X_i\}$, il est possible de déterminer facilement la confiance de tous les groupements de sources. Cela réduit donc le nombre de coefficients à calculer de 2^n à n . Les mesures floues de SUGENO et de WIENER-SHANON) possèdent cette propriété.

Pour les besoins de l'intégration, la définition (7) de la notion de mesure floue doit être complétée par la propriété suivante :

Définition 8. *Mesure Floue.*

Une mesure floue adaptée à l'intégration $\nu : \mathbb{P}(\Omega_X) \longrightarrow [0, 1]$ sur un espace mesurable Ω_X est une fonction vérifiant :

$$\begin{aligned} \nu(\emptyset) &= 0, \nu(\Omega_X) = 1 \\ \forall (X_i, X_j) \in \mathbb{P}(\Omega_X), X_i \subseteq X_j &\implies \nu(X_i) \leq \nu(X_j) \\ \text{Soient } (X_1, \dots, X_n) \in \Omega_X^n / X_i \subseteq X_{i+1} &\implies \lim_{i \rightarrow n} \nu(X_i) = \nu(\lim_{i \rightarrow n} (X_i)) \end{aligned} \quad (4.31)$$

Définition 9. *Mesure Floue de SUGENO.*

Soient $(X_i, X_j) \in \Omega_X^2$ quelconques vérifiant $X_i \cap X_j = \emptyset$ et λ un réel de l'intervalle $[-1, \infty)$. Une mesure floue ν_λ sur Ω_X est dite mesure floue de SUGENO (ou λ -mesure) si ν_λ vérifie la λ -règle suivante :

$$\nu_\lambda(X_i \cup X_j) = \nu_\lambda(X_i) + \nu_\lambda(X_j) + \lambda \nu_\lambda(X_i) \nu_\lambda(X_j) \quad (4.32)$$

Posons $\nu_\lambda^j = \nu_\lambda(X_j)$. Le réel λ vérifie alors :

$$\nu_\lambda \left(\bigcup_{i=1}^n X_i \right) = 1 = \frac{1}{\lambda} \left(1 - \prod_{j=1}^n (1 + \lambda \nu_\lambda^j) \right) \quad (4.33)$$

Définition 10. *Mesure de WIENER-SHANON.*

Soit un espace de probabilité (Ω_X, X, P) , où P est une mesure de probabilité dans l'espace $f(\Omega_X, X)$. La mesure floue de WIENER-SHANON $\nu : X \longrightarrow [0, +\infty[$ est définie par

$$\nu(X_i) = (-c - \log P(X_i))^{-1} \quad (4.34)$$

et vérifie la règle suivante qui permet la détermination de tous les coefficients :

$$X_i \cap X_j = \emptyset \implies \nu(X_i \cup X_j) = \min \left(\infty, -c - \ln \left(e^{\left(\frac{-1}{c\nu(X_i)}\right)} + e^{\left(\frac{-1}{c\nu(X_j)}\right)} \right) \right) \quad (4.35)$$

Si l'espace Ω_X est infini, l'ensemble des parties $\mathbb{P}(\Omega_X)$ a un cardinal infini et les définitions présentées ici ne s'appliquent plus. Les paragraphes suivants rappellent succinctement les expressions des intégrales floues dans le cadre d'espaces discrets.

4.3.2 L'intégrale de SUGENO

L'intégrale floue de SUGENO utilise une mesure floue de SUGENO. Dans le cas d'un ensemble Ω_X discret dont x_i sont les éléments, cette intégrale est définie ci-dessous.

Définition 11. *Intégrale Floue de SUGENO.*

L'intégrale de SUGENO d'une fonction mesurable $f : \Omega_X \longrightarrow [0, 1]$ par rapport à une mesure floue ν est définie pour Ω_X discret dont x_i sont les éléments par :

$$\int f \circ \nu = \bigcup_{i=1}^n \min(a_i, \nu(X_i)) \quad \text{avec} \quad \begin{array}{l} a_i = f(x_i) \\ 0 \leq a_1 \leq \dots \leq a_n \\ X_i = \{x_i, \dots, x_n\} \end{array} \quad (4.36)$$

L'intégrale de SUGENO fournit donc un autre ensemble discret qui est le résultat de l'intégration. La fusion de n ensembles discrets est un ensemble discret.

4.3.3 L'intégrale de CHOQUET

Définition 12. *Intégrale Floue de CHOQUET.*

L'intégrale de CHOQUET d'une fonction mesurable $f : \Omega_X \longrightarrow [0, 1]$ par rapport à une mesure floue ν est définie pour Ω_X discret dont x_i sont les éléments par :

$$\int f d\mu = \sum_{i=1}^n (a_i - a_{i-1})\mu(X_i) \quad \text{avec} \quad \begin{array}{l} a_i = f(x_i) \\ 0 \leq a_1 \leq \dots \leq a_n \\ X_i = \{x_i, \dots, x_n\} \end{array} \quad (4.37)$$

L'intégrale de CHOQUET sur un espace discret est très fortement liée aux opérateurs OWA comme le montre [Fodor et al., 1995]. Nous donnons ci-après l'idée générale de la démonstration. Soit F un opérateur OWA de vecteur W . Considérons la suite de valeurs e_i telle que :

$$e_i = F(\underbrace{0, \dots, 0}_i \text{ zéros}, \underbrace{1, \dots, 1}_{n-i} \text{ uns}) \quad (4.38)$$

Etant données les propriétés de F (notamment l'idempotence), les valeurs e_i sont alors telles que :

$$0 = e_n \leq \dots \leq e_i \leq \dots \leq e_0 = 1 \quad \text{avec} \quad e_{i-1} - e_i = \omega_i \quad (4.39)$$

Le résultat de l'agrégation d'un vecteur ordonné $\vec{x} = (x_1, \dots, x_n)$ peut donc s'écrire sous la forme :

$$F(\vec{x}) = \sum_{i=1}^n \omega_i a_i = \sum_{i=1}^n (e_{i-1} - e_i) x_i = \sum_{i=1}^n e_{i-1} (x_i - x_{i-1}) \quad (4.40)$$

Considérons alors l'ensemble $X_j = \{x_j, \dots, x_n\}$ et posons $X_{n+1} = \emptyset$. Nous pouvons définir une mesure floue sur les sous-ensembles X_i en posant $\nu(X_j) = e_{j-1}$. La mesure floue ν s'exprime alors sous la forme :

$$\begin{array}{l} \nu(X_1) = \nu(x_1, \dots, x_n) = e_0 = 1 \\ \nu(X_j) \geq \nu(X_{j+1}) \quad \text{avec} \quad X_j \supset X_{j+1} \\ \nu(X_{n+1}) = \nu(\emptyset) = e_0 = 0 \end{array} \quad (4.41)$$

Ce qui peut se formuler d'une manière plus proche de la définition (12) :

$$F(x_1, \dots, x_n) = \sum_{i=1}^n \nu(X_i)(x_i - x_{i-1}) \quad (4.42)$$

Les opérateurs OWA sont donc bien un cas particulier de l'intégrale de CHOQUET. Par ailleurs, le résultat de l'intégrale de CHOQUET est un réel. Comparativement à l'intégrale de SUGENO, le résultat est plus pauvre. En fait, l'intégrale de CHOQUET peut être rapprochée d'un opérateur OWA, la pondération étant fournie par une mesure floue et les réels à combiner (nommés a_1, \dots, a_n au paragraphe 4.1 page 64 traitant des opérateurs OWA) étant fournis par les éléments de l'ensemble flou intégré.

Nous présentons maintenant la définition de l'intégrale à T-Conormes qui est la forme la plus générale des intégrales floues.

4.3.4 L'intégrale à T-Conormes

Cette intégrale est basée sur trois T-Conormes (voir 1.4.2.1 pour une description de cet opérateur). Avant de présenter la définition de l'intégrale à T-Conormes, nous devons décrire le système de T-Conormes et l'opération " $-\Delta$ " associée à la T-Conorme Δ .

Définition 13. *Système de T-Conorme.*

$F = (\Delta, \perp, \underline{\perp}, \diamond)$ est un système de T-Conormes pour l'intégration si $\Delta, \perp, \underline{\perp}$ sont des T-Conormes continues et que l'opération \diamond vérifie :

$$\diamond : ([0, 1], \Delta) \times ([0, 1], \perp) \longrightarrow ([0, 1], \underline{\perp}) \quad \text{avec} \quad \begin{array}{l} \diamond \text{ est continue sur }]0, 1[\times]0, 1[\\ a \diamond x = 0 \iff a = 0 \text{ ou } x = 0 \end{array} \quad (4.43)$$

Les espaces $([0, 1], \Delta)$, $([0, 1], \perp)$ et $([0, 1], \underline{\perp})$ sont respectivement les espaces de valeurs de l'intégrande, de la mesure et de l'intégrale. Les trois opérations $\perp, \underline{\perp}, \diamond$ d'un système de T-Conormes possèdent deux propriétés intéressantes qui sont comparables à la distributivité :

$$\begin{aligned} x \perp y < 1 &\implies a \diamond (x \perp y) = (a \diamond x) \perp (a \diamond y) \text{ pour } a \in [0, 1] \\ a \diamond b < 1 &\implies (a \Delta b) \diamond x = (a \diamond x) \underline{\perp} (b \diamond y) \text{ pour } x \in [0, 1] \end{aligned} \quad (4.44)$$

Définition 14. *Opération " $-\Delta$ ".*

Pour une T-Conorme donnée Δ , on définit une opération " $-\Delta$ " sur $[0, 1] \times [0, 1]$ par :

$$a - \Delta b = \inf\{c : b \Delta c \geq a\} \quad \text{et pour } \Delta = \max, \text{ il vient } a - \Delta b = \begin{cases} a & \text{si } a > b \\ 0 & \text{si } a < b \end{cases} \quad (4.45)$$

Définition 15. *Intégrale Floue à T-Conormes.*

L'intégrale à T-Conormes associé à un système de T-Conormes $F = (\Delta, \perp, \underline{\perp}, \diamond)$ d'une fonction mesurable $f : \Omega_X \longrightarrow [0, 1]$ par rapport à une mesure floue ν est définie par :

$$\int f \diamond d\nu = \underline{\perp}_{i=1}^n ((a_i - \Delta a_{i-1}) \diamond \nu(X_i)) \quad \text{avec} \quad \begin{array}{l} a_i = f(x_i) \\ 0 \leq a_1 \leq \dots \leq a_n \\ X_i = \{x_i, \dots, x_n\} \end{array} \quad (4.46)$$

L'intégrale floue de SUGENO (respectivement de CHOQUET) est un cas particulier de l'intégrale à T-Conormes obtenu pour $F = (\max, \max, \max, \min)$ (respectivement pour $F = (\hat{+}, \hat{+}, \hat{+}, .)$).

4.3.5 Quelques exemples

Nous utilisons ici les intégrales floues dans deux applications : l'évaluation d'un étudiant et la localisation d'un objet dans un espace à deux dimensions.

4.3.5.1 L'évaluation d'étudiants

Nous considérons trois étudiants A , B et C ayant chacun été interrogé dans trois matières : les mathématiques, la physique et la littérature. Les notes obtenues sont présentées dans le tableau 4.6. Traditionnellement, une telle évaluation est effectuée à l'aide d'une moyenne pondérée. Nous effectuons ici le calcul à l'aide de la moyenne pondérée puis à l'aide de l'intégrale floue de CHOQUET.

Etudiant	Maths	Physique	Littérature
A	18	16	10
B	10	12	18
C	14	15	15

TAB. 4.6 – Les notes des trois étudiants.

Pour effectuer le calcul avec une moyenne pondérée, il est nécessaire de définir les coefficients de chaque matière. Pour le doyen de cet établissement, les mathématiques et la physique sont plus importants que la littérature. En tant que chef d'établissement, il choisit donc les coefficients suivants : $\alpha_M = 40$, $\alpha_P = 40$ et $\alpha_L = 20$. Le calcul de la moyenne pondérée s'effectue donc pour chaque étudiant à l'aide de :

$$\bar{x} = \frac{40N_M + 40N_P + 20N_L}{40 + 40 + 20} \quad (4.47)$$

N_M , N_P et N_L représentant respectivement la note obtenue en mathématiques, physique et littérature. Les résultats sont présentés dans le tableau 4.7. Avec cette manière de procéder, nous pouvons donner l'importance

Etudiant	Note
A	15.6
B	12.4
C	14.6

TAB. 4.7 – Les résultats des trois étudiants.

d'une matière par rapport à une autre. Supposons maintenant que le doyen soit un peu plus exigeant dans les critères de calcul. Il considère que :

- Les mathématiques et la physique ont la même importance, la littérature a une importance plus faible,
- Les étudiants bien notés en mathématiques obtiennent souvent de bonnes notes en physique car les matières sont proches. De tels étudiants ne doivent pas être privilégiés,
- Les étudiants bien notés en physique (ou en mathématiques) et bien notés en littérature sont rares et doivent être privilégiés.

De tels critères peuvent être plus facilement représentés par une mesure floue. L'agrégation se fera alors en utilisant l'intégrale floue de CHOQUET. Le doyen – qui joue dans cet exemple le rôle de l'expert – définit les valeurs de la mesure floue μ pour chacun des éléments de l'ensemble des parties⁷ de $\{M, P, L\}$. Ces valeurs sont visibles dans le tableau 4.8.

Ensemble	Mesure floue	Commentaire
$\{M\}$ et $\{P\}$	0.45	Physique et Mathématiques ont la même importance
$\{L\}$	0.30	La Littérature est une matière moins importante
$\{M, L\}$ et $\{P, L\}$	0.9	Un étudiant bon en sciences et en Littérature est privilégié
$\{M, P\}$	0.5	Un étudiant bon en Mathématiques et Physique n'est pas privilégié
$\{M, L, P\}$	1	Un étudiant brillant partout doit être favorisé

TAB. 4.8 – La mesure floue retenue.

Il est alors possible de calculer les notes de chaque étudiant à l'aide de l'intégrale de CHOQUET comme décrit au paragraphe 4.3.3. Nous détaillons le calcul pour l'étudiant A , les résultats de l'agrégation étant visibles dans le tableau 4.9 page ci-contre.

7. C'est-à-dire pour tous les éléments de $\{\{M\}, \{P\}, \{L\}, \{M, P\}, \{M, L\}, \{L, P\}, \{M, P, L\}\}$.

Les x_i représentent ici les trois disciplines considérées et les a_i les notes obtenues dans chaque discipline. Etant donnée la définition de l'intégrale de CHOQUET, les a_i doivent être dans l'intervalle $[0, 1]$. Nous procédons tout d'abord à une phase de normalisation consistant à diviser toutes les notes par 20, constante naturelle pour l'application car la meilleure note possible est 20.

Puis, nous ordonnons les notes de la plus petite à la plus grande, formant ainsi la suite des a_i :

$$\begin{aligned}
 a_0 &= 0 && \text{par définition} && a_2 &= \frac{16}{20} && \text{obtenue en } P \\
 a_1 &= \frac{10}{20} && \text{obtenue en } L && a_3 &= \frac{18}{20} && \text{obtenue en } M
 \end{aligned}
 \tag{4.48}$$

Nous calculons alors le résultat pour A en utilisant la définition du paragraphe 4.3.3 :

$$\begin{aligned}
 \sum_{i=1}^3 (a_i - a_{i-1})\mu(X_i) &= (a_1 - a_0)\mu(\{M, P, L\}) + (a_2 - a_1)\mu(\{P, M\}) + (a_3 - a_2)\mu(\{M\}) \\
 &= \left(\frac{10}{20} - 0\right) \times 1 + \left(\frac{16}{20} - \frac{10}{20}\right) \times 0.5 + \left(\frac{18}{20} - \frac{16}{20}\right) \times 0.45 \\
 &= \frac{13.9}{20}
 \end{aligned}
 \tag{4.49}$$

En effectuant les calculs nous obtenons les résultats présentés dans le tableau 4.9.

Etudiant	Maths	Physique	Littérature	CHOQUET
A	18	16	10	$\frac{13.9}{20}$
B	10	12	18	$\frac{13.6}{20}$
C	14	15	15	$\frac{14.9}{20}$

TAB. 4.9 – Les évaluations des trois étudiants par l'intégrale floue.

4.3.5.2 Localisation d'un objet dans un espace à deux dimensions

Nous nous intéressons ici à la localisation d'objets dans un espace à 2 dimensions, où l'on réalise la fusion de données numériques. Nous cherchons à localiser un objet dont les coordonnées (connues pour les besoins de l'exemple) sont $(8, 5)$. Pour cela nous disposons de trois sources d'informations qui nous indiquent la position de l'objet dans un même référentiel. Les trois sources évoluent dans des milieux différents comme le présente la figure 4.9. La première source A est placée sur un avion, la seconde source B est placée sur un bateau et la troisième source C est placée sur un char. De part le relief du sol, le point recherché est partiellement (ou totalement) masqué pour certaines sources (B ne "voit" pas directement l'objet cherché).

En utilisant les connaissances d'experts, nous affectons un degré de confiance à chaque source. La mesure fournie par le char C sur terre est considérée comme meilleure que les deux autres, la mesure de l'avion A est associée à une confiance plus faible et enfin, la mesure du bateau B sera associée à une confiance minimum car l'objectif n'est pas directement visible de B .

Nous supposons que les données fournies par les sources sont des fonctions d'appartenance de sous-ensembles flous qui indiquent la position en x et en y . Les informations mesurées par les sources sont regroupées dans le tableau 4.10 où les formes et les points caractéristiques des fonctions d'appartenance sont donnés. Chaque source fournit deux fonctions d'appartenance : la position en x et la position en y . Par exemple, pour

Source	Information	Type de la distribution
A	$[(7.81, 7.97, 8.13), (4.92, 5.07, 5.24)]$	Triangle
B	$[(8.0, 8.08, 8.12, 8.2), (4.70, 4.77, 4.82, 4.90)]$	Trapèze
C	$[(7.84, 8.0, 8.17), (4.85, 5.01, 5.17)]$	Triangle

TAB. 4.10 – Les mesures fournies par les trois sources.

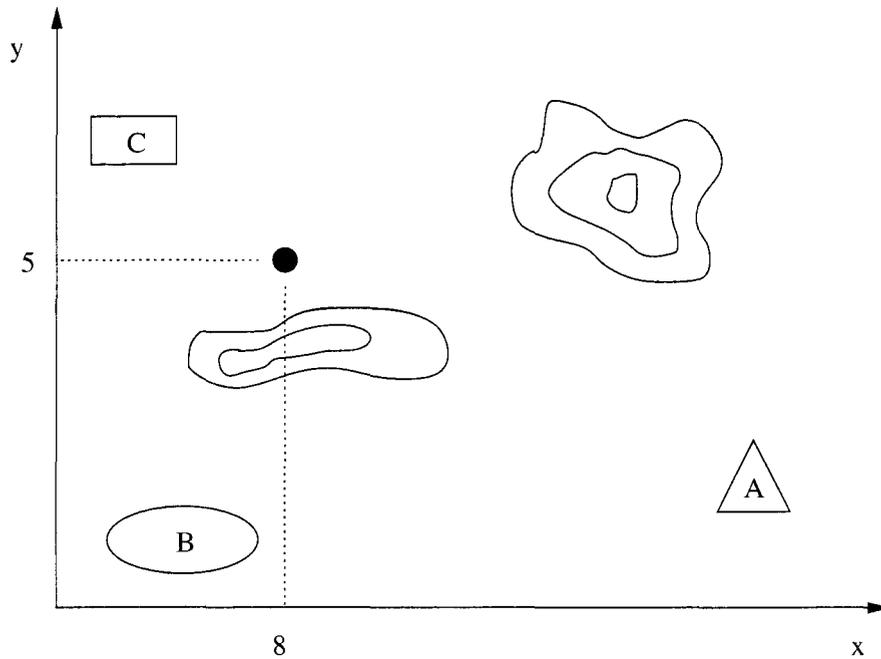


FIG. 4.9 – La scène expérimentale.

le x fourni par, A il s'agit d'un triangle dont les points ont pour coordonnées $(7.81, 0)$, $(7.97, 1)$ et $(8.13, 0)$. Pour le trapèze x fournit par B , les coordonnées des quatre points caractéristiques sont : $(8.0, 0)$, $(8.08, 1)$, $(8.12, 1)$, $(8.2, 0)$.

Il est nécessaire de définir les degrés de confiance que l'on a en chacune des sources. Nous utilisons ici la mesure de SUGENO (définition 9). En utilisant les connaissances des experts, nous déterminons : $\mu_\lambda(A) = 0.7$, $\mu_\lambda(B) = 0.6$, $\mu_\lambda(C) = 0.8$. Le problème va être résolu en utilisant la λ -mesure de SUGENO (définition 9). En vertu des propriétés de cette mesure, il est possible de déterminer le coefficient λ de en résolvant :

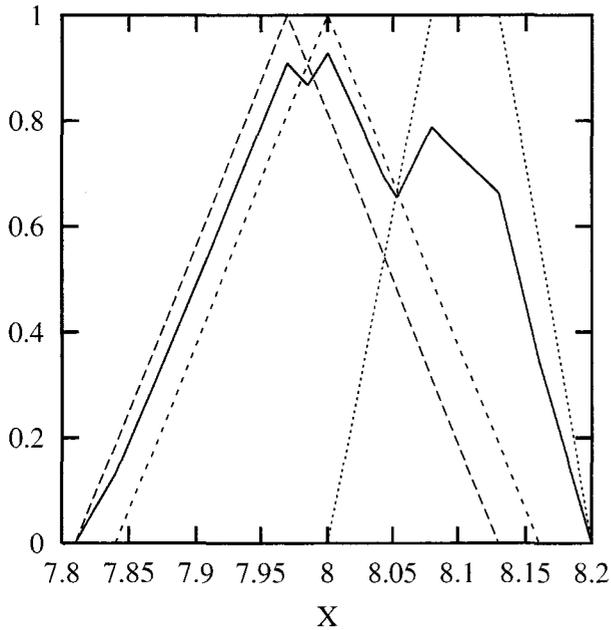
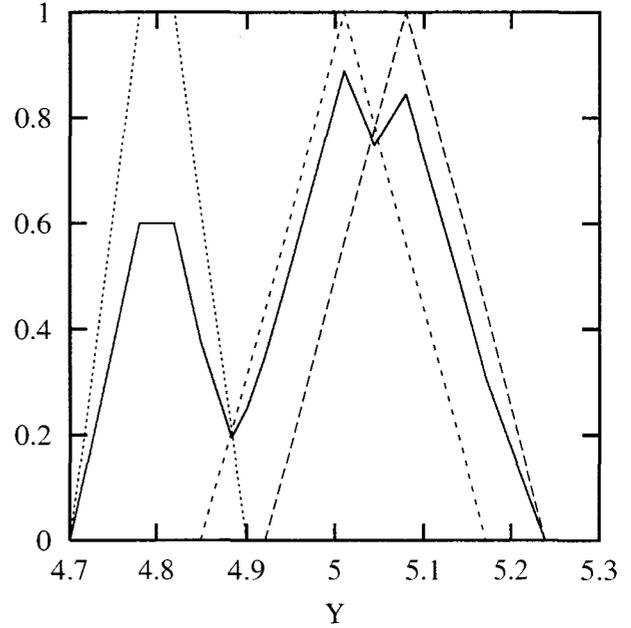
$$\lambda + 1 = (1 + 0.8\lambda)(1 + 0.6\lambda)(1 + 0.7\lambda) \implies 0.336\lambda^2 + 1.46\lambda + 1.1 = 0 \quad (4.50)$$

La recherche des deux racines conduit à une unique valeur possible $\lambda = -0.969$ car l'autre racine est hors de l'intervalle $[-1, \infty[$. Toutes les valeurs de la mesure floues sont ensuite calculées en utilisant les propriétés de cette mesure :

$$\begin{aligned} \mu_\lambda(A, B) &= \mu_\lambda(A) + \mu_\lambda(B) + \lambda\mu_\lambda(A)\mu_\lambda(B) = 0.893 \\ \mu_\lambda(A, C) &= \mu_\lambda(A) + \mu_\lambda(C) + \lambda\mu_\lambda(A)\mu_\lambda(C) = 0.957 \\ \mu_\lambda(B, C) &= \mu_\lambda(B) + \mu_\lambda(C) + \lambda\mu_\lambda(B)\mu_\lambda(C) = 0.935 \\ \mu_\lambda(A, B, C) &= \mu_\lambda(A, B) + \mu_\lambda(C) + \lambda\mu_\lambda(A, B) \\ \mu_\lambda(C) &= 1 \\ \mu_\lambda(\emptyset) &= 0 \end{aligned} \quad (4.51)$$

La résolution est effectuée à l'aide de l'intégrale de CHOQUET (voir paragraphe 4.3.3). Les informations en abscisses et en ordonnées sont traitées séparément. Pour effectuer le calcul, nous procédons d'une manière similaire à celle utilisée dans l'exemple précédent. Par exemple, pour de l'abscisse, nous choisissons un réel τ élément de l'union des supports. Le degré d'appartenance de τ à l'avion, le char et le bateau est alors traité comme les notes en Mathématiques, Physique et Littérature dans l'exemple précédent. Après calcul de la valeur C de l'intégrale floue, nous disposons donc d'un couple (τ, C) qu'il est possible de représenter graphiquement. Lorsque τ varie, nous obtenons le nombre flou résultat de la fusion. Le calcul en ordonné est réalisé de la même manière. Les résultats sont présentés sur la figure 4.12.

Il est alors possible de calculer les coordonnées en utilisant une méthode de défuzzyfication, par exemple la méthode du maximum. Nous remarquons que les fonctions d'appartenance en x et en y passent respectivement

FIG. 4.10 – Le résultat en x .FIG. 4.11 – Le résultat en y .FIG. 4.12 – Les résultats en x et y de la fusion.

par un maximum aux points de coordonnées $(8.00, 0.927)$ et $(5.01, 0.888)$. En utilisant la méthode du centre de gravité, les position en x et y sont respectivement 8.015 et 4.975.

4.4 Conclusion

Dans ce chapitre, nous avons présenté les opérateurs de fusion que nous utilisons chaque jour de manière inconsciente pour combiner plusieurs nombres réels entre eux et obtenir un autre nombre réel. Le professeur calculant la moyenne de l'un de ses étudiants illustre très bien ce processus de fusion dans le corps des réels. Combiner plusieurs réels peut aussi servir à choisir la meilleure alternative au sens d'un ensemble de critères donnés dont certains sont éventuellement plus importants que d'autres. C'est le but des opérateurs OWA présentés au début de ce chapitre mais aussi des opérateurs WOWA (Weighted Ordered Weighted Average) décrits dans [Torra, 1997]. Ils sont souvent utilisés pour effectuer une décision multicritères.

A l'aide du principe d'extension de ZADEH, ces opérateurs bien connus dans le corps des réels ont été adaptés à l'univers des ensemble flous et des distributions de possibilités. Grâce aux travaux de YAGER, il est possible de construire un opérateur de fusion sur \mathbb{R} et de l'utiliser pour combiner des sous-ensembles flou. L'agrégation prend alors en considération la compatibilité avec les sources et réalise une agrégation partielle, c'est-à-dire la prise en compte partielle des sources lorsque celles-ci sont en désaccord. La complexité des algorithmes présentés à été étudiée et une nouvelle méthode d'introduction de la compatibilité et de l'agrégation partielle à été introduite. Cette technique réduit la complexité de l'algorithme rendant ainsi la méthode utilisable sur calculateur.

Les opérateurs présentés sont tous commutatifs et permettent de fusionner plus de deux sources en une même opération. Les problèmes rencontrés précédemment au sujet du nombre de sources n'existent pas ici ce qui offre un grand champ d'application aux opérateurs construits à l'aide du principe d'extension.

Les intégrales floues – opérateurs de fusion plus marginaux – on été présentés et leur relation avec les opérateurs OWA précisée. Les intégrales floues sont fortement dépendantes d'une mesure floue et pour certaines du choix de plusieurs T-Conormes rendant leur utilisation difficile. Le nombre de paramètres à déterminer et à régler : valeurs de la mesure floues ou T-Conormes est souvent un frein au développement de ces techniques. Un exemple de fusion par intégrale floue a été détaillé.

Jusqu'à présent, les informations combinées étaient redondantes ou complémentaires. Dans le prochain chapitre, nous nous intéressons à la fusion dissymétrique, c'est à dire à la combinaison d'informations produites séquentiellement par une même source.

Chapitre 5

La fusion dissymétrique

LA fusion de données permet de combiner des informations afin d'obtenir un jugement plus fondé que celui obtenu en considérant chaque donnée isolément. Jusqu'à présent, nous supposions que toutes les informations à combiner étaient acquises au même instant de sources différentes. Les informations à fusionner étaient donc redondantes ou complémentaires. La fusion était dite symétrique.

Lorsque les informations à combiner sont produites séquentiellement par une même source ou lorsque certaines sources ont un rôle différent des autres, la fusion symétrique est inadaptée et des opérateurs spéciaux sont nécessaires. La fusion est alors dite dissymétrique. Une règle de fusion dissymétrique est une loi qui ajuste la connaissance courante du monde à la lumière d'une nouvelle information acquise. La fusion dissymétrique s'applique notamment à la mise à jour de bases de données [Gärdenfors, 1991] et au filtrage de données sensorielles en robotique mobile [Strauss et Aimé, 1996].

La fusion dissymétrique est utilisée lorsque le phénomène étudié est tel qu'une information acquise à t_{i+1} doit servir à modifier une information acquise à t_i , sous la contrainte $t_{i+1} > t_i$. C'est par exemple le cas d'informations acquises par un capteur ultrasonique monté sur un robot mobile. Plus le robot s'approche d'un obstacle, plus la perception ultrasonique est de bonne qualité. La force de ce type de combinaison réside dans sa capacité à faciliter la décision même en présence d'informations manquantes. Supposons qu'un robot mobile cherche à connaître la nature de l'obstacle face à lui. Au début, l'obstacle peut être de toute nature, puis au fur et à mesure que le robot s'approche, les informations sont toujours incomplètes mais deviennent plus précises jusqu'à ce qu'elles soient suffisantes pour en déduire la nature de l'obstacle. Le processus de fusion dissymétrique suppose cependant l'existence d'une information *a priori* pour initialiser le processus.

Ce chapitre présente tout d'abord les concepts de la fusion dissymétrique hors de tout cadre mathématique. Puis, nous nous intéressons plus particulièrement aux techniques de fusion dissymétrique dans le cadre de la théorie des possibilités. Les opérateurs existant sont présentés et nous étudions l'influence du choix de la T-Norme dans le processus de révision ce qui conduit à la proposition de nouveaux opérateurs dérivés des opérateurs existant. Un exemple simple est alors proposé.

5.1 La fusion dissymétrique d'informations

L'idée de modifier une connaissance à la lumière d'une nouvelle information existe depuis fort longtemps et ce dans de nombreux formalismes comme par exemple les espaces logiques [Gärdenfors, 1988], les espaces probabilistes [Lewis, 1976] et plus récemment les espaces possibilistes [Dubois et Prade, 1994b]. Les univers de modélisation étant plus ou moins riches, il existe donc de nombreuses techniques de modification d'une information existante à la lumière d'une nouvelle évidence. Chacune de ces techniques ayant une sémantique différente des autres, nous tentons dans ce paragraphe de présenter tous les concepts existants.

La littérature [Dubois et Prade, 1994b] distingue classiquement les règles de conditionnement et de mise-à-jour. Cette distinction est comparable à celle existant dans le cadre de la logique entre la notion de révision et de projection (voir à ce sujet [Katsuno et Mendelzon, 1992]). Nous explicitons ces notions dans les deux

paragraphes suivants en conservant la décomposition proposée dans [Dubois et Prade, 1994b] reprise dans [Delplanque et al., 1997].

5.1.1 Les règles de conditionnement (révision)

Les règles de conditionnement généralisent le conditionnement BAYSIEN. Elles sont réservées aux cas où le monde est statique. Dans ce cas, la nouvelle information ne signifie pas que la connaissance actuelle est fautive mais donne un autre point de vue sur l'univers ou une nouvelle information. Le type d'information manipulé – générique ou factuelle – permet de diviser cette famille en deux sous-familles :

- Une connaissance générique fait état d'une vérité valable dans tout l'univers, sans que cette connaissance soit liée à une situation particulière. Ce type de connaissance est souvent exprimé sous forme de règle : “Les stratus sont des nuages de basse altitude”,
- Une connaissance factuelle est une description de l'univers à un instant, fondée sur des faits constatés comme par exemple la liste des nuages dans une zone particulière du ciel.

Il est alors possible de réaliser la distinction entre révision et focalisation.

5.1.1.1 Révision

La révision modifie la connaissance actuelle à l'aide d'une nouvelle connaissance de même type (factuelle ou générique). Par exemple, la connaissance existante (générique) “Les stratus sont des nuages de basse altitude” pourra être modifiée par la nouvelle connaissance (générique) “Les stratus sont les plus bas des nuages”.

Lorsque les informations sont représentées par des distributions de possibilités, il est possible de mettre en évidence trois modes de révisions selon la spécificité¹ de la distribution résultante : l'expansion, la contraction et la révision “simple” :

- l'expansion est appliquée lors de la réception d'une information cohérente avec l'information *a priori*, la distribution de possibilité résultante est alors plus spécifique que la distribution *a priori*.
- la contraction consiste à oublier si B est vraie ou non. Après contraction, la distribution résultante doit exprimer toute ignorance vis-à-vis de la nouvelle information B . Le résultat sera donc moins spécifique que la distribution *a priori*. La contraction est appliquée lorsque la nouvelle information B n'est pas consistante avec l'information *a priori*
- La révision “classique” fournit des résultats ni plus ni moins spécifiques et qui contredisent partiellement l'information *a priori*. Contrairement aux lois précédentes traduites par des heuristiques, la révision classique est une extension du conditionnement probabiliste.

La terminologie “expansion”, “contraction” et “révision classique” peut sembler contraire à l'intuition mais c'est celle choisie dans [Dubois et Prade, 1994b] et nous utiliserons cette terminologie dans tout ce chapitre.

Les notions d'expansion et de contraction sont adaptés à des référentiels symboliques et il est difficile de les interpréter en termes de distributions de possibilités.

5.1.1.2 Focalisation

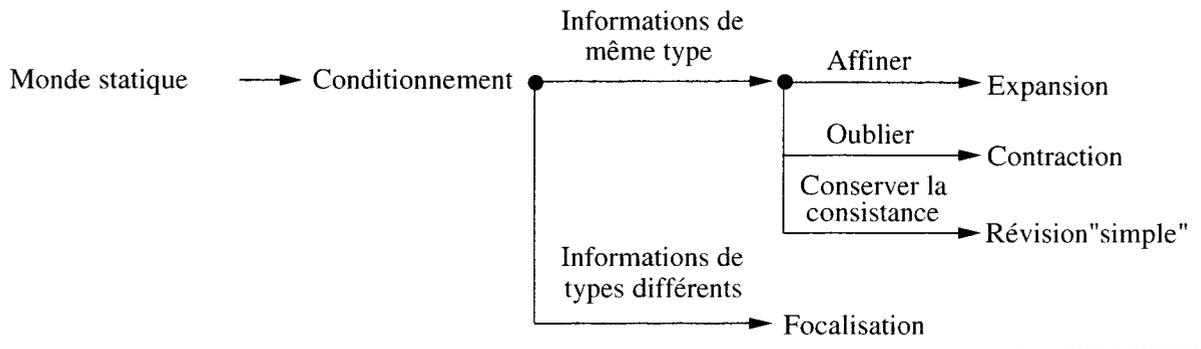
La focalisation utilise une connaissance générique (valable pour l'ensemble des objets de l'univers) et une connaissance factuelle sur un objet particulier pour améliorer les conclusions plausibles au sujet de cet objet particulier. La connaissance générique “Les stratus sont des nuages de basse altitude” et l'information factuelle : “Le nuage au-dessus de Strasbourg a une altitude de 0 m” permettent de déduire que “Le nuage au-dessus de Strasbourg est un stratus”. La connaissance factuelle ne modifie en rien la connaissance générique mais elle permet de préciser la connaissance relative à un objet particulier. Cette technique trouve son application dans les systèmes à base de règles de connaissances.

1. Selon [Yager, 1983], une distribution π_1 est plus spécifique que π_2 lorsque $\forall \omega \in \Omega, \pi_1(\omega) \leq \pi_2(\omega)$. La distribution de possibilité la plus spécifique est définie par $\pi(\omega_0) = 1$ et $\forall \omega \in \Omega - \omega_0, \pi(\omega) = 0$. Il existe d'autres définitions de la spécificité et certaines sont présentées propriété 17 du paragraphe 7.1.3.

5.1.2 Les règles de projection (mise-à-jour)

Les règles de projection sont utilisées lorsque la situation courante change. Le système peut apprendre la réalisation d'un événement E ce qui rend alors impossible un ensemble de situations. Les croyances affectées à ces situations doivent donc être ré-affectées en les projetant sur celles qui sont toujours possibles. Cela ne signifie pas que la croyance passée était fausse mais qu'elle l'est devenue. Par exemple, il est possible de calculer la probabilité de présence d'un sous-marin nucléaire en une zone donnée. Si maintenant un pays a désarmé quelques uns de ses sous-marins nucléaires, il faut mettre à jour la connaissance en conséquence.

La figure 5.1 résume les différents cas d'emploi des règles de fusion dissymétriques.



Monde dynamique → Projection (mise-à-jour)

FIG. 5.1 – Les lois de fusion dissymétriques.

Après avoir défini le comportement des opérateurs nous nous intéressons maintenant à leurs expressions mathématiques dans le contexte de la théorie des possibilités. Nous présentons succinctement les règles de mise-à-jour avant de décrire de manière plus complète les règles de conditionnement.

5.2 Opérateurs dissymétriques et théorie des possibilités

Nous rappelons ici quelques opérateurs de fusion dissymétriques de la famille des règles de mise-à-jour et des règles de conditionnement. L'information est notée A et la nouvelle information B .

5.2.1 Les règles de mise-à-jour

Les règles de mise à jour possibilistes étendent le concept de projection probabiliste encore connu sous le nom de "LEWIS' imaging" et présenté dans [Lewis, 1976]. Des travaux comparables [Smets et Keenes, 1990] ont été effectués la théorie de DEMPSTER-SHAFER. Ce paragraphe présente le principe de la mise-à-jour dans le cadre de la théorie des possibilités dont l'article [Dubois et Prade, 1994b] est une description plus complète.

Soit Ω l'ensemble de toutes les situations possibles, chacune étant affectée d'un coefficient p (représentant une probabilité, une masse élémentaire de probabilité, un degré de possibilité...). Une nouvelle information E est disponible et indique que l'ensemble des situations possibles se réduit à un sous ensemble $B \subset \Omega$. On suppose que pour tout élément ω de Ω il existe un ω_B de B tel que ω_B soit supposé "le plus proche" de ω (au sens d'un critère à choisir).

La règle de mise-à-jour opère un transfert des masses p de ω sur les éléments ω_B en respectant le principe de changement minimal. Cette règle est nommée "imaging" parce que p_B est l'image de p sur B obtenue par transfert des masses p pour les ω n'appartenant pas à B vers les ω_B de B . La figure 5.2 page suivante schématise ce transfert. Plus formellement, la mise-à-jour de π_A dans B est alors :

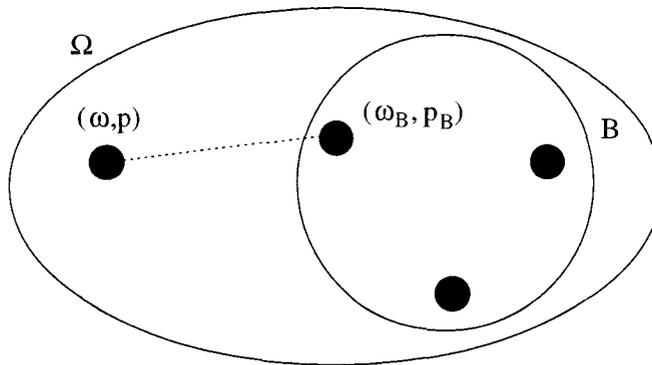


FIG. 5.2 – Le principe de la mise-à-jour.

$$\begin{aligned} \pi_B^o(\omega') &= \max_{\omega'=\omega_B} \pi_A(\omega) \text{ si } \omega' \in B \\ &= 0 \text{ sinon} \end{aligned} \tag{5.1}$$

Il est aussi possible qu'il existe plusieurs ω_B proches de ω , il est possible d'affecter les poids $\pi_A(\omega)$ à chacun des éléments les plus proches. Soit $C(\omega)$ l'ensemble des ω_B proches de ω :

$$\begin{aligned} \pi_B^o(\omega') &= \max_{\omega' \in C(\omega)} \pi_A(\omega) \text{ si } \omega' \in B \\ &= 0 \text{ sinon} \end{aligned} \tag{5.2}$$

La mise-à-jour définie par (5.1) satisfait les postulats de GÄRDENFORS rappelés dans le tableau 5.1.

	Postulat
1	$\pi_B^o \leq \mu_B$
2	$\pi_A \leq \mu_B \implies \pi_B^o = \pi_A$
3	$B = \emptyset$ et π_A normalisée alors π_B^o est normalisée
4	$A = B \implies \pi_B^o = \pi_A^o$
5	$\min(\pi_B^o, \mu_A) \leq \pi_{A \cap B}^o$
6	$\pi_B^o \leq \mu_A, \pi_A^o \leq \mu_B \implies \pi_A^o = \pi_B^o$
7	Si π_A est la plus spécifique, alors $\min(\pi_A^o, \pi_B^o) \leq \pi_{A \cup B}^o$
8	$[\max(\pi, \pi')]_B^o = \max(\pi_B^o, \pi_B^{\prime o})$

TAB. 5.1 – Les postulats de la mise-à-jour.

Si nous définissons une relation R_B qui à chaque ω associe ω_B , son plus proche dans B , l'équation (5.2) se réduit alors au principe d'extension déjà étudié au paragraphe 4.2 :

$$\begin{aligned} \pi_B^o &= \mu_{R_B \circ F} \quad \text{avec} \quad \begin{aligned} \pi_A &= \mu_F \\ \mu_{R_B}(\omega_1, \omega_2) &= \mu_{C(\omega)}(\omega_2) \\ \mu_{R_B}(\omega_1, \omega_2) &= 0 \quad \text{si } \omega_2 \notin B \end{aligned} \end{aligned} \tag{5.3}$$

Si la relation $\circ = \min = \wedge$, il vient alors :

$$\begin{aligned} \pi_B^o(\omega') &= \max_{\omega} \pi_A(\omega) \wedge \mu_{C(\omega)}(\omega') \text{ si } \omega' \in B \\ &= 0 \text{ sinon} \end{aligned} \tag{5.4}$$

5.2.2 Les règles de conditionnement

Nous rappelons maintenant quelques règles de conditionnement en supposant que les informations à combiner sont de même type (génériques ou factuelles). Par conséquent, seules les règles de révision seront étudiées.

5.2.2.1 L'expansion

Supposons tout d'abord que la nouvelle information B est un fait sûr. Une expansion est appliquée lorsque B est cohérente avec A . Dans ce cas, la distribution résultante π_B^+ doit être plus spécifique :

$$\begin{aligned}\pi_B^+(\omega) &= \pi_A(\omega), \forall \omega \in B \text{ si } \exists \omega_0 \in B / \pi_A(\omega_0) = 1 \\ &= 0 \text{ sinon}\end{aligned}\quad (5.5)$$

Lorsque la nouvelle information B est telle que $\Pi(B) = 1$, B est cohérente avec l'information *a priori* et π_A subit une expansion π_B^+ donnée par :

$$\pi_B^+(\omega) = \min(\mu_B(\omega), \pi_A(\omega)) \quad (5.6)$$

où μ_B est la fonction caractéristique du sous-ensemble B . Dans le cas contraire, (5.6) n'est plus la description d'un état cognitif cohérent et π_B^+ n'est plus normalisée. Dans ce cas, l'expansion fournit l'état absurde² $\pi_B^+(\omega) = \pi_\perp$. Cela est discutable et certains auteurs préfèrent fournir l'ignorance totale $\pi_B^+(\omega) = \pi_\top$. Cette définition vérifie les six postulats de GÄRDENFORS exprimés dans le tableau 5.2 en termes de distributions de possibilités :

	Postulat	Signification
1	π_B^+ est une distribution de possibilité π_B^+ décrit un état cognitif	Stabilité
2	$N_B^+(B) = 1$ N_B^+ mesure de nécessité basée sur π_B^+	Priorité à la nouvelle information
3	$\pi_B^+ < \pi_A$	Amélioration de la connaissance
4	$N(B) = 1 \implies \pi_B^+ = \pi_A$	Invariance si l'entrée est déjà connue
5	$\pi_1 \leq \pi_2 \implies \pi_1^+ \leq \pi_2^+$	Monotonie
6	Pour tout π_A et B , π_B^+ est la distribution la moins spécifique vérifiant 1 à 5	Principe de spécificité minimale

TAB. 5.2 – Les postulats de l'expansion.

5.2.2.2 La contraction

Si A contient l'information “ B est vrai” et que B devient faux, une contraction peut être appliquée. Il s'agit d'oublier B et la distribution résultante π_B^- doit exprimer une ignorance totale vis-à-vis de B . En termes de possibilités, l'ignorance totale vis-à-vis de B se traduit par $\Pi_B^-(B) = \Pi_B^-(\bar{B}) = 1$. Intuitivement, si nous avons déjà $\Pi(B) = \Pi(\bar{B}) = 1$, il semble intéressant d'obtenir $\pi_B^- = \pi_A$. Lorsque $\Pi(B) = 1 > \Pi(\bar{B})$ nous devrions obtenir $\pi_B^-(\omega) = 1$ pour quelques ω de \bar{B} et notamment pour ceux vérifiant $\Pi(\bar{B}) = \pi_A(\omega)$. Cela conduit à la proposition suivante :

$$\begin{aligned}\pi_B^-(\omega) &= 1, \text{ si } \pi_A(\omega) = \Pi(\bar{B}), \omega \notin B \\ &= \pi_A(\omega) \text{ sinon}\end{aligned}\quad (5.7)$$

La distribution π_B^- est alors moins spécifique que π_A . Cette définition répond au critère de changement minimal de l'information *a priori*. La tableau 5.3 présente les postulats de la contraction.

Une autre définition de la contraction proche des règles de révision (voir paragraphe suivant) est :

$$\begin{aligned}\pi_B^-(\omega) &= \frac{\pi_A(\omega)}{\Pi(B)}, \text{ si } \omega \in \bar{B} \\ &= \pi_A(\omega) \text{ sinon}\end{aligned}\quad (5.8)$$

2. L'impossibilité est traduite par $\pi_\perp(\omega) = 0, \forall \omega \in \Omega$ et l'ignorance par $\pi_\top(\omega) = 1, \forall \omega \in \Omega$.

	Postulat	Signification
1	π_B^- est une distribution de possibilité π_B^- décrit un état cognitif	Stabilité
2	$N(B) = 0 \implies \pi_B^- = \pi_A$	Invariance si l'entrée n'est pas connue
3	$\pi_B^- \geq \pi_A$	π_B^- n'est pas plus informative que π_A
4	$N_B^-(B) = 0$ sauf si $B \subseteq \Omega$ N_B^- mesure de nécessité basée sur π_B^-	B est oublié si possible
5	$N(B) > 0 \implies \pi_A \geq \min(\mu_B, \pi_B^-)$	Contraction sur B puis expansion sur B doit amener dans un état cohérent avec l'état originel
6	$A = B \implies \pi_B^- = \pi_A^-$	
7	$\max(\pi_B^-, \pi_A^-) \geq \pi_{A \cap B}^-$	Contracter $A \cap B$ conduit à un état au moins aussi informatif que la rétraction de A ou de B
8	$N_{A \cap B}^-(B) = 0 \implies \pi_{A \cap B}^- \geq \pi_B^-$	Contracter $A \cap B$ lorsque B n'est plus certain ne supprime pas plus d'information que contracter B directement

TAB. 5.3 – Les postulats de la contraction.

5.2.2.3 La révision “classique”

Toute règle de révision ne donnant pas des résultats plus spécifiques que l'expansion est dite “classique”. Dans la théorie des possibilités, la révision est comparable au conditionnement BAYSIEN. Il est donc nécessaire de définir la notion de possibilité conditionnelle entre deux ensembles flous.

La notion de *possibilité conditionnelle* est inspirée de la notion de probabilité conditionnelle. Elle a été introduite par ZADEH puis affinée dans [Hisdal, 1978a] et plus récemment par [Dubois et Prade, 1992]. Nous rappelons ici rapidement les concepts (décrits de manière plus détaillée dans [de Cooman, 1997]). Soient A et B deux ensembles au sens classique, A étant supposé non vide. Selon la loi de BAYES, la probabilité jointe $P(A \cap B)$ est donnée par :

$$\forall A, P(A \cap B) = P(A|B).P(B) \quad (5.9)$$

D'après [Hisdal, 1978b], l'expression de la possibilité jointe peut-être calculée à partir des possibilités conditionnelles et marginales en utilisant une analogie avec la théorie des probabilités. Supposons A et B flous. En remplaçant P par Π et le produit par une T-Norme T dans l'équation (5.9), il vient :

$$\forall A, \Pi(A \cap B) = T(\Pi(A|B), \Pi(B)) \quad (5.10)$$

Le processus de révision implique la modification d'une connaissance existante à la lumière d'une nouvelle évidence. Il s'agit de minimiser les changements apportés à l'information *a priori*. La T-Norme T pourra donc être choisie parmi tous les opérateurs d'intersection. Le tableau 1.2 page 18 rappelle les T-Normes les plus utilisées.

Pour calculer l'expression de la possibilité conditionnelle $\Pi(A|B)$ et celle de la distribution de possibilité associée $\pi(\omega|B)$ pour une T-Norme donnée, il est nécessaire de procéder en deux étapes. Nous supposons tout d'abord que la nouvelle information B est un ensemble au sens classique. Cela permet d'obtenir l'expression de la distribution de possibilité $\pi(\omega|B)$. Cette expression est alors utilisée pour calculer la distribution de possibilité $\pi(\omega|B)$ pour B flou (voir [Dubois et Prade, 1992]). Nous nous intéressons tout d'abord à l'expression de $\pi(\omega|B)$ pour quelques T-Normes et B classique [Zunino et al., 1998b] avant de nous intéresser au cas B flou.

5.2.2.3.1 La nouvelle information est un ensemble classique Ce paragraphe traite de la possibilité conditionnelle pour B ensemble classique (B connu par sa fonction caractéristique), A restant un ensemble flou. L'expression de $\pi(\omega|B)$ est donnée pour plusieurs T-Normes.

5.2.2.3.1.1 $T = \min(x, y)$ Lorsque $T = \min(x, y)$, l'équation (5.10) devient :

$$\forall A, A \cap B \neq \emptyset, \Pi(A \cap B) = \min(\Pi(A|B), \Pi(B)) \quad (5.11)$$

Lorsque $\Pi(A|B)$ est strictement inférieur à $\Pi(B)$, nous obtenons $\Pi(A|B) = \Pi(A \cap B)$. Dans le cas contraire, nous avons $\Pi(B) = \Pi(A \cap B)$ et par conséquent $\Pi(A|B)$ est un réel α choisit arbitrairement dans l'intervalle $[\Pi(A \cap B), 1]$. L'expression de la possibilité conditionnelle est alors :

$$\Pi(A|B) = \begin{cases} \Pi(A \cap B) & \text{si } \Pi(B) > \Pi(A \cap B), \\ \alpha \in [\Pi(A \cap B), 1] & \text{si } \Pi(B) = \Pi(A \cap B) \end{cases} \quad (5.12)$$

Il est impossible d'extraire une unique valeur de $\Pi(A|B)$ à partir de (5.12). Un choix doit être fait pour définir $\Pi(A|B)$ lorsque $\Pi(B) = \Pi(A \cap B)$. Pour éviter toute perte d'information, le résultat le plus imprécis (celui qui écarte le moins d'alternatives) semble le plus raisonnable. En terme de possibilités, la solution la plus imprécise (encore nommée "solution la moins spécifique") est celle qui a les degrés de possibilité les plus élevés. Ce choix est celui fait par DUBOIS et PRADE dans [Dubois et Prade, 1992]. Ainsi, lorsque $\Pi(B) = \Pi(A \cap B)$, nous choisirons le degré de possibilité le plus élevé possible, soit 1. Lorsque les deux ensembles n'ont aucun élément en commun, il vient $\Pi(A \cap B) = \Pi(\emptyset) = 0$ avec par hypothèse $B \neq \emptyset$ donc $\Pi(B) > 0$. Le degré de possibilité retenu sera donc 0 en vertu de l'équation (5.12). Dans les autres cas, il n'y a aucun choix à faire et nous poserons $\Pi(A|B) = \Pi(A \cap B)$. Ainsi, la mesure de possibilité de A conditionnelle à la nouvelle connaissance B est donnée par :

$$\Pi(A|B) = \begin{cases} 1 & \text{si } \Pi(A \cap B) = \Pi(B), \\ 0 & \text{si } A \cap B = \emptyset, \\ \Pi(A \cap B) & \text{sinon} \end{cases} \quad (5.13)$$

Calculer $\Pi(B)$ consiste à évaluer la possibilité de B étant donnée la connaissance préalable A . Si Ω désigne l'univers du discours, la possibilité de B relative à A s'exprime par :

$$\Pi(A|B) = \sup_{x \in \Omega} \min(\pi_A(x), \pi_B(x)) \quad (5.14)$$

Le degré de possibilité d'avoir ω sachant B [Dubois et Prade, 1988] est déterminé en posant $A = \{\omega\}$ dans (5.13) et en faisant varier ω dans A . Lorsque ω n'est pas un élément de B , $\{\omega\} \cap B = \emptyset$ et $\pi(\omega|B)$ est nul en vertu de l'équation (5.13). En choisissant les valeurs maximales pour $\Pi(\{\omega\}|B) = \pi(\omega|B)$ il vient :

$$\forall \omega \in A, B \neq \emptyset, \pi(\omega|B) = \begin{cases} 1 & \text{si } \pi_A(\omega) = \Pi(B), \omega \in B \\ \pi_A(\omega) & \text{si } \pi_A(\omega) < \Pi(B), \omega \in B \\ 0 & \text{si } \omega \notin B \end{cases} \quad \text{avec } \Pi(B) = \sup_{s \in B} \pi_A(s) \quad (5.15)$$

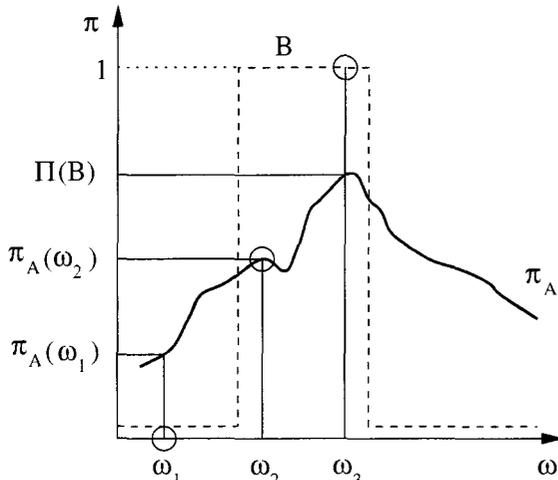
Lorsque la nouvelle information B est en totale contradiction avec l'information existante représentée par π_A , nous avons $A \cap B = \emptyset$ et donc $\Pi(B) = \sup_{s \in B} \pi_A(s) = 0$. L'équation (5.15) s'interprète alors comme suit :

- soit ω n'est pas dans B . Nous obtenons $\pi(\omega|B) = 0$,
- soit ω est dans B et $\pi_A(\omega) = \Pi(B) = 0$. Nous obtenons $\pi(\omega|B) = 1$.

Cela revient à poser $\pi(\cdot|B) = \pi_A$ dans tous les autres cas. L'ancienne information est totalement oubliée et la priorité est donnée à la nouvelle information. Pour obtenir l'information révisée, il suffit de faire varier s dans B et de faire $\pi(s|B) = 1$ lorsque $\Pi(B) = \pi_A(s)$. Dans les autres cas, il faut faire $\pi(s|B) = \pi_A(s)$, les autres degrés de possibilités étant forcés à 0. La figure 5.3 illustre ce procédé.

La définition de l'opérateur introduit des discontinuités, notamment au(x) point(s) ω_0 tels que $\pi_A(\omega_0) = \Pi(B) < 1$ ainsi qu'aux extrémités du support de l'intersection de π_A et de B (Le point noté ω_1 sur la figure 5.4 en est un exemple).

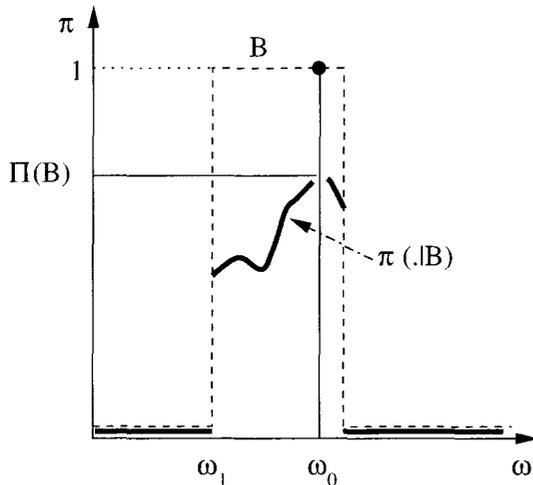
Propriété 30 (Discontinuité). Si $\Pi(B) \neq 0$ et $\Pi(B) \neq 1$ et qu'il existe $\omega \notin B$ tel que $\pi_A(\omega) > 0$ alors la distribution $\pi(\cdot|B)$ est discontinue. La figure 5.4 présente la distribution de possibilité conditionnelle obtenue avec π_A et B définis sur la figure 5.3. Hors de B , $\pi(\cdot|B)$ est nul. Il existe au moins un point ω_1 où $\pi(\cdot|B)$ n'est pas continue.



$\pi(\omega_1|B) = 0$ car $\omega_1 \notin B$,
 $\pi(\omega_2|B) = \pi_A(\omega_2)$ car $\pi_A(\omega_2) < \Pi(B)$,
 $\pi(\omega_3|B) = 1$ car $\pi_B(\omega_3) = \Pi(B)$.

FIG. 5.3 – Possibilité conditionnelle $\pi(.|B)$ pour B classique ($T=\min$).

Démonstration. Soit w_1 défini comme sur la figure 5.4. Au voisinage de w_1 , nous avons :



$$\begin{cases} \lim_{\omega \rightarrow \omega_1^-} \pi(\omega|B) = 0 \\ \lim_{\omega \rightarrow \omega_1^+} \pi(\omega|B) = \pi_A(\omega_1) \neq 0 \end{cases} \quad \square$$

FIG. 5.4 – Un exemple de discontinuité.

Cette T-Norme conduit donc à une expression de la possibilité conditionnelle peu adaptée aux ensembles continus mais plus adaptée aux ensembles discrets dans lesquels ces problèmes de discontinuité ont moins d'influence.

Le comportement de cet opérateur de révision a été testé avec les ensembles A et B présentés figure 5.5 page ci-contre. Le mode opératoire est le suivant. La révision de l'information A a priori par la nouvelle information B est effectuée. Puis, la distribution π_A est décalée d'un offset δ vers la droite et la révision est de nouveau calculée. Lorsque π_A a "traversé" totalement B , le processus est arrêté. Les résultats sont présentés sur la figure 5.6 page suivante.

5.2.2.3.1.2 $T = x.y$ Lorsque $T = x.y$, l'équation (5.10) devient alors :

$$\forall A, A \cap B \neq \emptyset, \Pi(A \cap B) = \Pi(A|B) \cdot \Pi(B) \tag{5.16}$$

Si $\Pi(B)$ n'est pas nul, l'équation précédente permet de déterminer $\Pi(A|B)$ de manière unique :

$$\Pi(A|B) = \begin{cases} \frac{\Pi(A \cap B)}{\Pi(B)} & \text{si } A \cap B \neq \emptyset, \\ 0 & \text{sinon.} \end{cases} \tag{5.17}$$

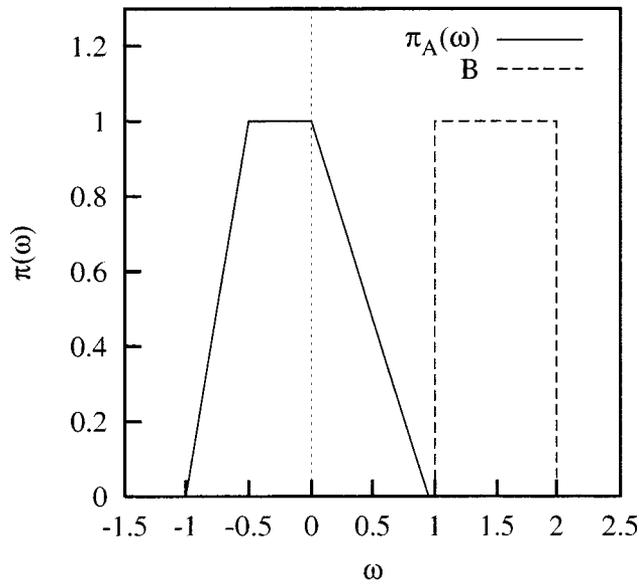


FIG. 5.5 – Fonctions d'appartenances des ensembles utilisés lors de la simulation.

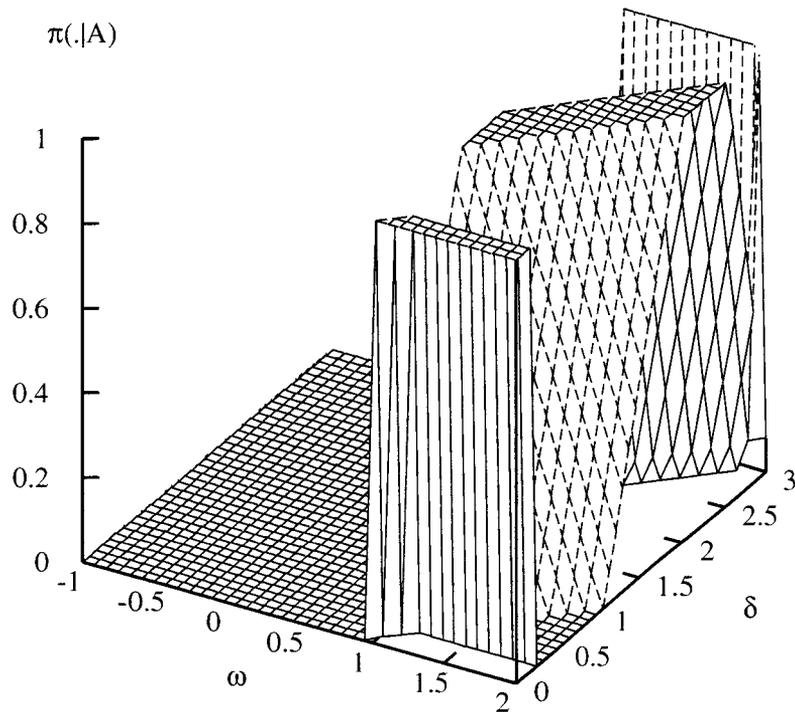


FIG. 5.6 – Possibilité conditionnelle $\pi(.|B)$ pour B classique ($T=\min$).

La distribution possibilité conditionnelle $\pi(\omega|B)$ est déterminée en posant $A = \{\omega\}$ dans (5.17) :

$$\forall \omega \in A, B \neq \emptyset, \pi(\omega|B) = \begin{cases} \frac{\pi_A(\omega)}{\Pi(B)} & \text{si } \omega \in B \\ 0 & \text{si } \omega \notin B \end{cases} \quad \text{avec} \quad \Pi(B) = \sup_{\omega \in B} \pi_A(\omega) \quad (5.18)$$

Cette définition qui est plus adaptée que la précédente pour des univers continus renormalise tous les degrés de possibilité de π_A présents dans B . Un exemple est présenté figure 5.7.

Nous utilisons maintenant les définitions précédentes pour calculer $\pi(.|B)$ lorsque B est une information imprécise connue par sa distribution de possibilité π_B .

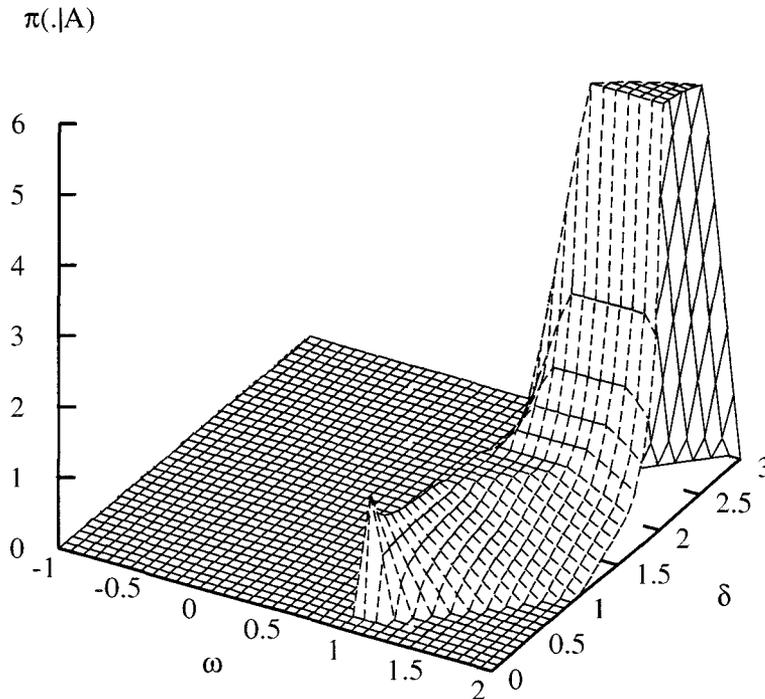


FIG. 5.7 – Possibilité conditionnelle $\pi(.|B)$ pour B classique ($T=x.y$).

5.2.2.3.2 La nouvelle information est un ensemble flou Nous nous intéressons maintenant au cas général où la nouvelle connaissance est imprécise. L'ensemble B est désormais un ensemble flou. Dans l'approche probabiliste, la règle de BAYES (5.9) est étendue par la règle de JEFFREY (voir [Jeffrey, 1965]) dans laquelle l'observation B est entachée d'une incertitude α . La mesure de probabilité révisée est alors :

$$P(A) = \alpha \frac{P(A \cap B)}{P(A)} + (1 - \alpha) \frac{P(\bar{A} \cap B)}{P(\bar{A})} \tag{5.19}$$

Si les observations possibles B_1, \dots, B_n forment une partition et si α_i est la certitude d'avoir observé B_i :

$$P(A) = \sum_{i=1}^n \alpha_i P(A|B_i) \quad \text{avec} \quad \sum_{i=1}^n \alpha_i = 1 \tag{5.20}$$

Il est possible de voir la nouvelle information B connue par sa distribution de possibilité π_B sous la forme de la fonction d'appartenance μ_B d'un ensemble flou U [Dubois et Prade, 1991]. La possibilité $\Pi(A|B_\alpha)$ de l' α -coupe B_α du sous-ensemble U joue alors un rôle comparable à la probabilité $P(A|B_i)$ et α un rôle comparable à α_i . Les paragraphes précédents nous permettent de calculer $\Pi(A|B_\alpha)$ et il reste donc à combiner ces possibilités pour α dans $(0, 1]$ pour former une autre possibilité. La seule manière de réaliser cette combinaison d'une manière compatible avec la manipulation d'événements est :

$$\forall A, \Pi(A) = \max(f_1(\Pi_1(A)), \dots, f_n(\Pi_n(A))) \tag{5.21}$$

où les f_i sont des fonctions monotones croissantes telle que $f_i(0) = 0$ pour tout i et il doit exister j tel que $f_j(1) = 1$. L'opération max pondérée vérifie ces caractéristiques³ ce qui conduit alors à :

$$\forall A, \Pi(A) = \max_{j=1}^n \min(\lambda_j, \Pi_j(A)) \quad \text{avec} \quad \sum_{j=1}^n \lambda_j = 1 \tag{5.22}$$

3. Il est possible de remplacer la T-Norme min par une autre T-Norme mais l'opération max pondérée est l'équivalent possibiliste de la combinaison convexe des probabilités, c'est pour cela que cette opération est choisie ici.

les λ_i représentant l'importance relative des Π_i . La mesure de possibilité $\Pi(A|B)$ peut alors être⁴ :

$$\Pi(A|B) = \sup_{\alpha \in (0,1]} \min(\alpha, \Pi(A|B_\alpha)) \quad \text{où} \quad B_\alpha = \{s \in S, \mu_B(s) \geq \alpha\} \quad (5.23)$$

La distribution de possibilité associée étant alors donnée par :

$$\forall \omega \in A, B \neq \emptyset, \pi(\omega|B) = \sup_{\alpha \in (0,1]} \min(\alpha, \pi(\omega|B_\alpha)) \quad (5.24)$$

Ainsi, la représentation de l'observation floue : “ x est B ” peut être vue comme étant équivalente à la collection d'événements non flous “ x est B_α ” où les poids $\alpha = \inf_{s \in B_\alpha} \mu_B(s)$ indiquent jusqu'à quel point il est possible que le sous-ensemble B_α soit une représentation au sens non flou de B . L'équation (5.24) peut être vue comme une extension du théorème classique de la représentation d'un ensemble flou B par ses α -coupes :

$$\mu_B(s) = \sup_{\alpha \in (0,1]} \min(\alpha, \mu_{B_\alpha}(s)) \quad (5.25)$$

D'une manière pratique, le calcul de la révision de A par la nouvelle information B est réalisé par (5.24) mais nécessite le choix d'une méthode de calcul de $\pi(\omega|B_\alpha)$ parmi celles déjà détaillées dans les paragraphes précédents. Finalement, le calcul peut être implanté sur un calculateur de deux manières différentes :

- utiliser (5.24) pour étendre la définition au cas où B est flou. L'algorithme est alors général (changer de manière d'évaluer $\pi(\omega|B_\alpha)$ est presque instantané) mais nécessite l'évaluation d'un grand nombre d' α -coupes,
- simplifier mathématiquement (si cela est possible) (5.24) pour une définition particulière de $\pi(\omega|B_\alpha)$ ce qui supprime le besoin d'évaluer de nombreuses α -coupes et conduit à un algorithme plus rapide, mais moins général.

Lorsque $\pi(\omega|B_\alpha)$ est calculé à l'aide de la T-Norme minimum (voir (5.15)):

$$\forall \omega, \pi(\omega|B) = \begin{cases} \mu_B(\omega) & \text{si } \exists \alpha > 0, \Pi(B_\alpha) = \pi_A(\omega) \\ \min(\mu_B(\omega), \pi_A(\omega)) & \text{sinon} \end{cases} \quad (5.26)$$

Le résultat fourni est normalisé et le suprémum est atteint pour $\alpha = \mu_B(\omega)$ quelque soit ω . En utilisant la T-Norme produit (voir (5.18)) :

$$\forall \omega, \pi(\omega|B) = \min \left(\mu_B(\omega), \frac{\pi_A(\omega)}{\Pi(B_{\mu_B(\omega)})} \right) \quad (5.27)$$

qui est aussi normalisée. Les deux définitions possèdent des propriétés intéressantes :

- ▷ $\forall \omega, \pi(\omega|B) \leq \mu_B(\omega)$
- ▷ si $\text{Noyau}(B) \cap \text{Noyau}(A) \neq \emptyset$ alors $\pi(\omega|B) = \min(\mu_B, \pi_A(\omega))$

Nous nous intéressons maintenant à un exemple d'application de la révision dans le domaine de la détection d'obstacle sur véhicule routier.

5.3 Un exemple

Un véhicule est équipé à l'avant d'un dispositif capable de distinguer un objet immobile (ou animé d'une vitesse faible) sur la route jusqu'à une distance de 200m, même en pleine nuit. Si un objet est détecté, le dispositif donne une évaluation de la largeur de l'objet. Les tailles susceptibles d'être fournies sont au nombre de trois : entre 0m et 1.50m (un triangle de signalisation d'accident . . .) entre 1.50m et 3m (un véhicule léger, ou un camion . . .) et enfin entre 3m et 5m (un objet entrave la route et empêche tout passage).

4. L'ensemble des α -coupes pouvant être infini, la notation sup a été utilisée à la place du max.

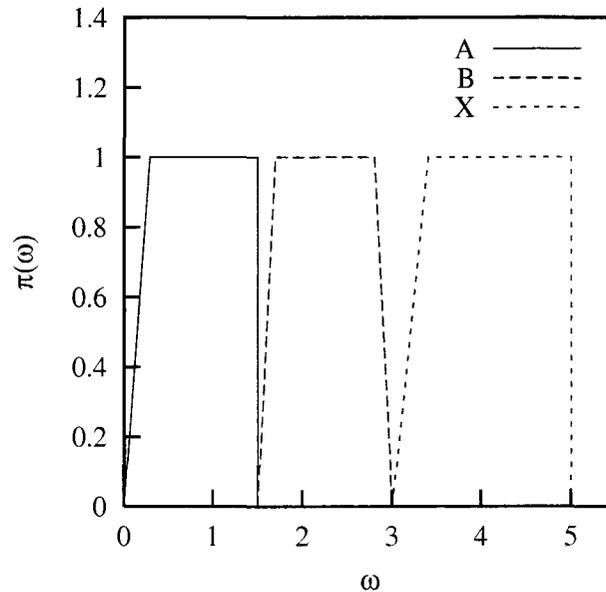


FIG. 5.8 – Les distributions de possibilité des trois obstacles.

Le but du système est de fournir une bonne estimation de la taille de l'obstacle pour permettre au conducteur d'agir en conséquence. La décision d'avertir le conducteur est prise par un autre module non détaillé ici. Ici, la nouvelle information est prioritaire sur l'ancienne information car plus le temps passe, plus l'information fournie par le capteur est pertinente car le capteur s'approche de l'obstacle. L'estimation de la largeur de l'obstacle sera réalisée à l'aide d'un processus de fusion dissymétrique ce qui implique une information *a priori* pour initialiser le processus. Lorsque le dispositif ne perçoit aucun objet, il semble raisonnable de supposer que tout obstacle peut se présenter : l'ignorance totale semble donc un choix correct pour l'information *a priori*. Cependant, pour faciliter la lecture des résultats, l'ignorance totale sera réduite à l'ignorance totale sur une distance plus grande que la largeur de la route soit par exemple 20m sans aucune perte de généralité.

Nous supposons qu'aucun autre type d'obstacle ne peut venir entraver la route et que les obstacles considérés ne peuvent pas changer au cours du processus. Cela signifie que notre univers est statique, nécessitant l'utilisation d'une règle de conditionnement. Par ailleurs, l'information *a priori* est factuelle (elle décrit le type d'objet entravant la route, devant notre véhicule, à un instant donné, sur une route donnée, dans un sens de circulation bien donné) et l'information fournie par le capteur est aussi factuelle. Nous utiliserons alors une technique de révision.

Lorsqu'un obstacle est détecté, une expansion est effectuée sur la connaissance actuelle (au début du processus, l'information actuelle est en fait l'ignorance totale). Le résultat de l'expansion remplace alors l'information actuelle en vue d'une nouvelle étape dans le processus de fusion. Nous choisissons ici l'expansion pour ses capacités d'affinement de l'information. Cependant, si la nouvelle évidence est en désaccord avec l'information courante, une expansion conduit alors à un état impossible. Pour supprimer ce phénomène, en cas de désaccord, une union simple est appliquée en lieu et place de l'expansion. Ce processus se poursuit jusqu'à ce que le module de décision considère la spécificité de l'information suffisante.

L'algorithme utilisé pour réaliser l'expansion est celui décrit dans (5.5). Il est plus adapté aux univers symboliques (donc discrets). En effet, les figures 5.16 et 5.17 montrent nettement les discontinuités de la règle.

Les figures (5.9) à (5.18) montrent chacune des étapes précédant la rencontre du véhicule avec un obstacle de type *B* (voir figure (5.8)). Lors de cette simulation, le capteur a envoyé dans l'ordre les informations suivantes : *B, A, B, A, A, A, B, B, B, B*. L'entrée était donc fortement bruitée. La première étape a été obtenue par expansion de l'ignorance totale. Les courbes figures (5.10) (étape 2), (5.12) (étape 4) et (5.15) (étape 7) ont été obtenues par union de l'information *a priori* et de la nouvelle information. Les autres courbes

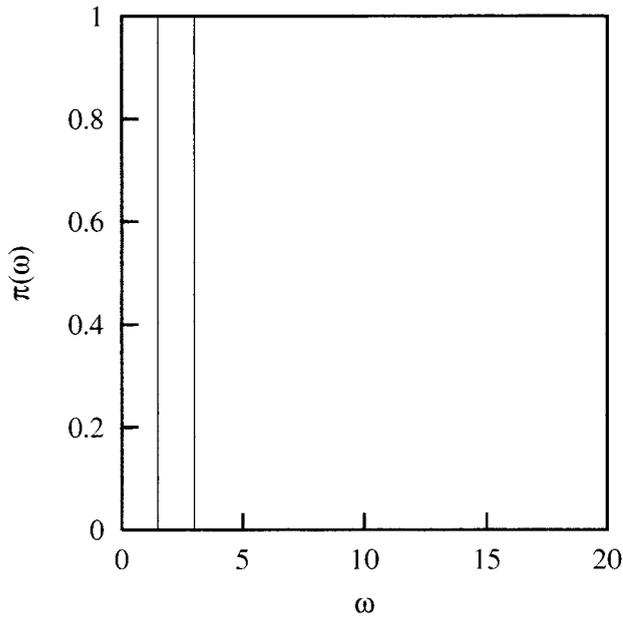


FIG. 5.9: Etape 1.

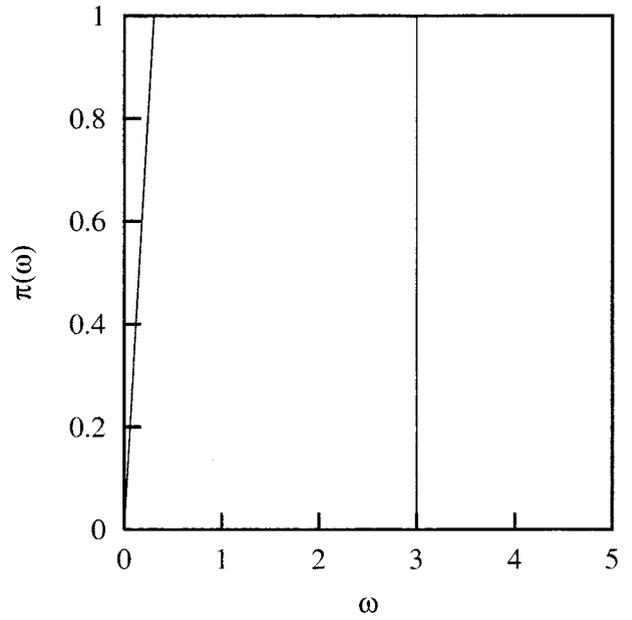


FIG. 5.10: Etape 2 (union simple).

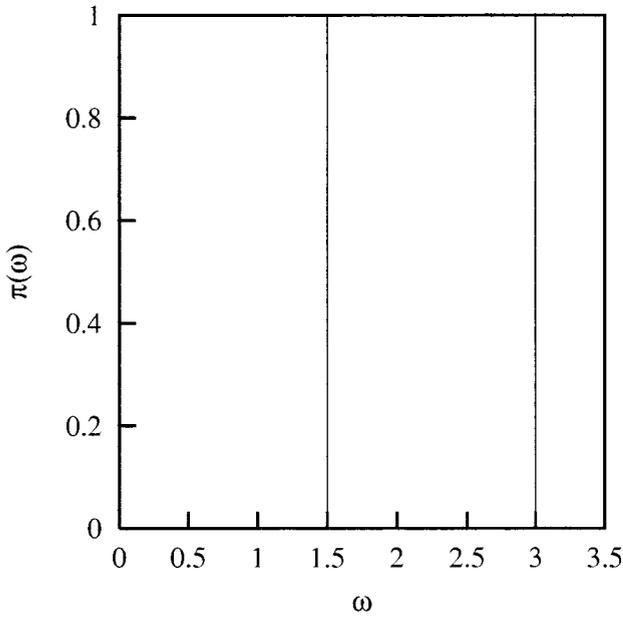


FIG. 5.11: Etape 3.

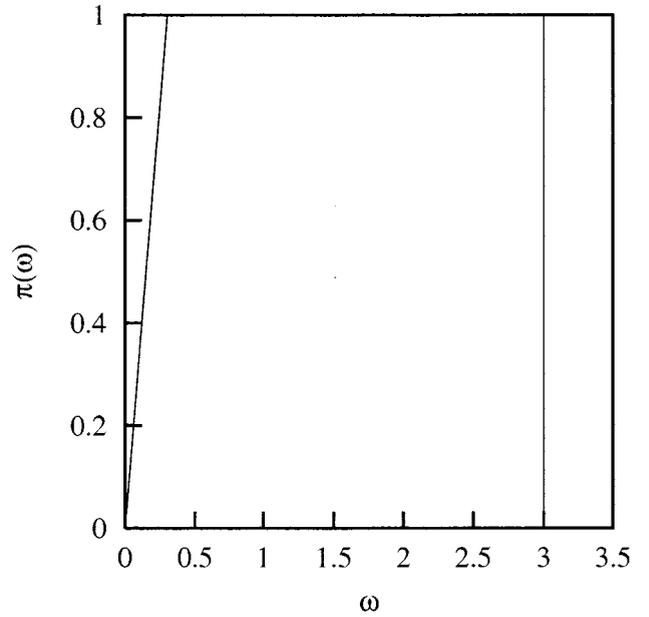


FIG. 5.12: Etape 4 (union simple).

résultent l'une expansion entre l'information *a priori* et la nouvelle évidence.



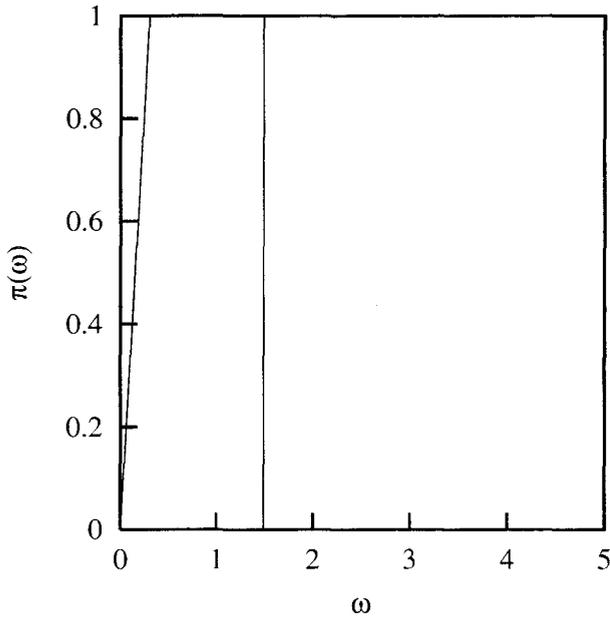


FIG. 5.13: Etape 5.

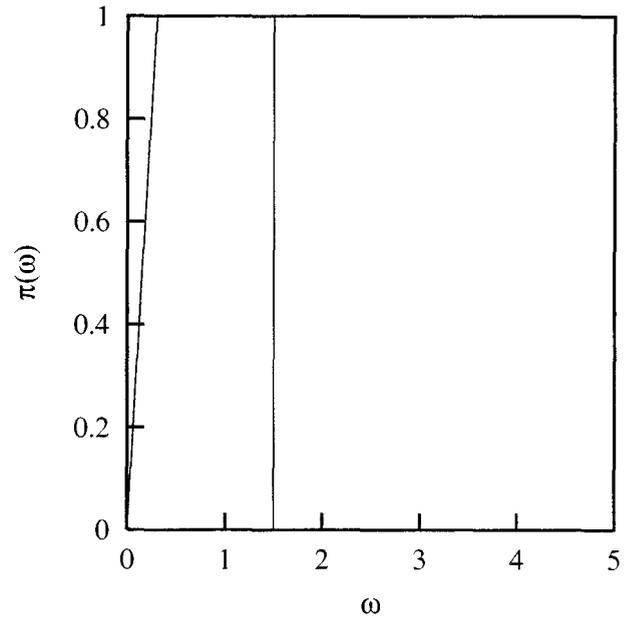


FIG. 5.14: Etape 6.

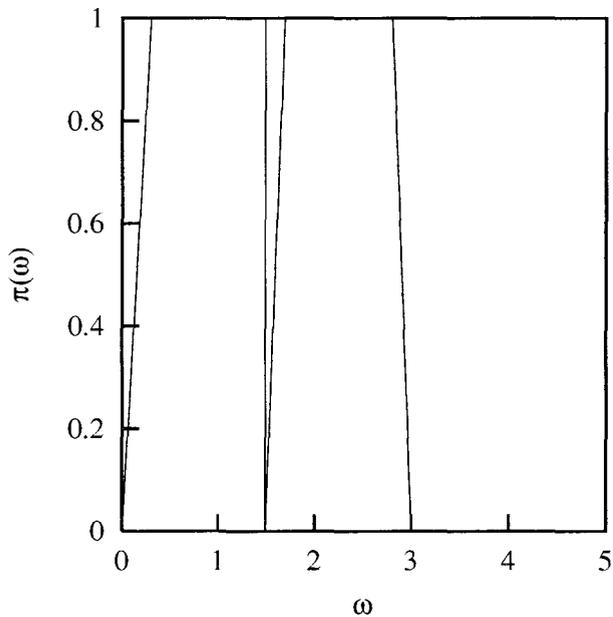


FIG. 5.15: Etape 7 (union simple).

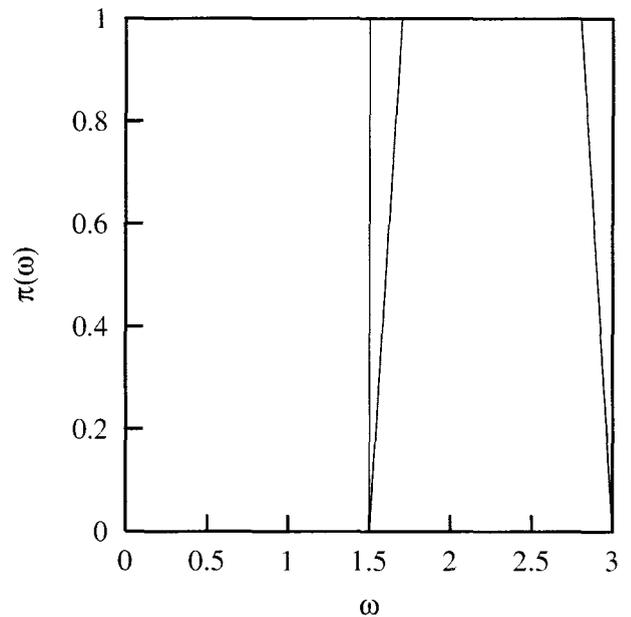


FIG. 5.16: Etape 8.

5.4 Conclusion

La fusion dissymétrique permet de modifier la croyance d'une information *a priori* à la lumière d'une nouvelle évidence. Il existe plusieurs familles de règles de fusion dissymétriques suivant que l'univers est statique ou non, que l'information et la nouvelle évidence sont du même type (générique ou factuelle) ou de types différents et suivant le type d'opération à faire : un oubli, un renforcement ou révision. Ces techniques sont utilisées dans toutes les applications où plus le temps passe, plus les informations sont supposées pertinentes. C'est le cas notamment dans l'exemple présenté où un processus de fusion dissymétrique tente de déterminer le plus rapidement possible la nature de l'obstacle barrant la route d'un véhicule automobile.

Le tour d'horizon des opérateurs de fusion existant dans la théorie des possibilités est maintenant terminé. Il semble alors judicieux de s'intéresser aux problèmes d'implantation des algorithmes de fusion sur calculateur numérique.

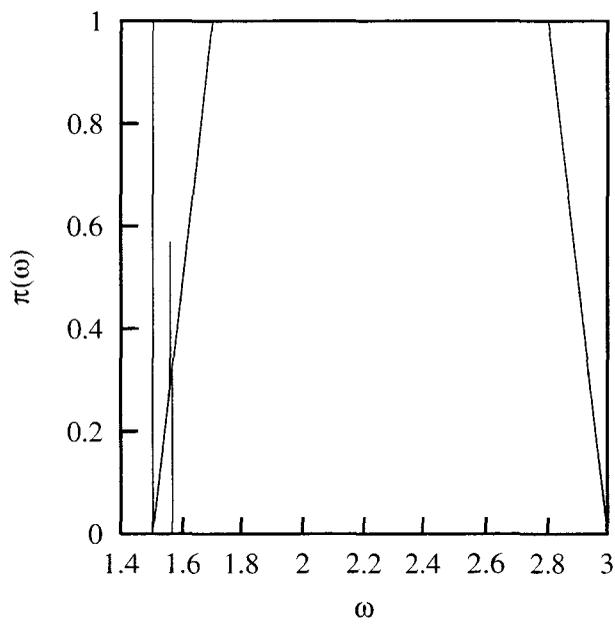


FIG. 5.17: Etape 9.

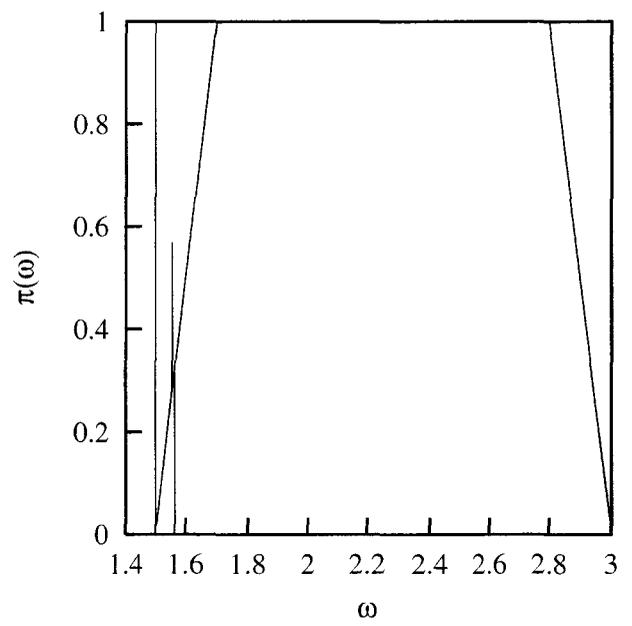


FIG. 5.18: Etape 10.



Chapitre 6

DFOL : A Data Fusion Object Library

La théorie des possibilités est typiquement un univers de modélisation numérique permettant la représentation de phénomènes discrets ou continus. Dans ce formalisme, chaque élément se voit associer un réel de l'intervalle $[0, 1]$ quantifiant le degré avec lequel il se peut qu'il corresponde à l'événement considéré. Les algorithmes associés à la théorie des possibilités et en particulier les techniques de fusion symétrique ou dissymétrique effectuent des calculs arithmétiques (addition, soustraction, minimum . . .) sur des réels de l'intervalle $[0, 1]$. La mise en œuvre de ces algorithmes sur un ordinateur nécessite donc une attention toute particulière.

Pour effectuer d'une manière fiable la fusion de deux informations modélisées à l'aide de distributions de possibilité, l'implémentation doit résoudre trois problèmes :

- le stockage des distributions de toute forme (par exemple triangulaire, quelconque . . .). Il est nécessaire de trouver un compromis entre l'espace mémoire utilisé et la précision de la représentation,
- la précision des calculs. Multiplier, diviser, soustraire des réels proches et éléments de l'intervalle $[0, 1]$ est toujours source de lourdes erreurs numériques. Par ailleurs, le résultat d'un algorithme est souvent utilisé comme entrée d'un autre algorithme favorisant ainsi la propagation des erreurs numériques,
- la vitesse des calculs. Le résultat d'une fusion est habituellement utilisé pour prendre des décisions. Il faut donc trouver là encore un compromis entre la vitesse de calcul et la précision du résultat. Ce compromis dépend de l'ordre du temps disponible pour prendre la décision donc du champ applicatif.

Actuellement, pour manipuler des distributions de possibilité sur un ordinateur, il existe deux solutions : les logiciels "intégrés" (par exemple le logiciel **MATLAB**) ou les bibliothèques logicielles écrites dans un langage de programmation classique comme C ou Pascal. Les logiciels intégrés fournissent un ensemble de fonctions pré définies (des possibilités de représentation graphique puissantes par exemple) et fonctionnent sur plusieurs type de machines (PC, Macintosh . . .). Le portage d'applications est alors instantané mais implique l'installation d'un logiciel coûteux et gourmand en mémoire. Les bibliothèques logicielles fournissent des facilités plus ciblées sur un aspect donné et sont prévues pour une machine particulière de sorte que leur utilisation est presque impossible si l'on ne dispose pas de l'environnement de travail de l'auteur (machine, compilateur, . . .).

Dans le cas particulier de la théorie des possibilités, les logiciels intégrés existant ne supportent aucun algorithme de fusion de données. Ils se contentent tout au plus de permettre la modélisation de distributions discrètes (c'est pas exemple le cas du logiciel **MATLAB**). Ces logiciels permettent cependant d'effectuer les calculs avec un grand nombre de décimales au prix d'une substantielle perte de performances.

Les boîtes à outils de manipulation des distributions de possibilité sont elles aussi limitées aux cas discret et ne fournissent que des "assistants" à la saisie des points et à l'affichage d'une distribution sous forme graphique. Enfin, peu de boîtes à outils sont livrées avec le programme source ce qui rend impossible l'extension des fonctionnalités de la librairie logicielle.

Toutes ces remarques nous ont conduit à réaliser une bibliothèque logicielle de manipulation de distributions de possibilités et à implémenter une grande partie des algorithmes de fusion disponibles dans ce

formalisme dans la mesure ou d'autres membres de l'équipe étaient amenés à utiliser les techniques de fusion pour diverses applications.

Ce chapitre présente donc notre composant logiciel de fusion de données sur des distributions de possibilités (et des sous-ensembles flous). Dans une première partie nous abordons le cahier des charges de la bibliothèque et le cadre de réalisation choisi. Nous présentons alors l'architecture du composant logiciel. La troisième partie décrit les différentes fonctionnalités mises en œuvre. Enfin, avant de conclure, nous commentons à titre d'exemple le code source de quelques programmes de fusion dans un cas concret.

6.1 Le cahier des charges

Le composant logiciel doit permettre d'effectuer la fusion de données dans le cadre de la théorie des possibilités. Cela implique l'existence de deux grandes familles de fonctionnalités : la représentation concise de distributions de possibilités et la fusion des distributions. Le module de représentation et le module de fusion seront donc totalement dissociés. Cette séparation permet l'utilisation de la librairie uniquement à des fins de représentation de distributions sans devoir alourdir l'exécutable avec des fonctionnalités non nécessaires.

Le module de modélisation des distributions comprend l'ensemble des opérations nécessaires sur les distributions comme la création, la destruction, l'affichage, le décalage la vérification de l'égalité . . . mais aussi des opérations plus complexes comme le calcul exact du noyau et du support par exemple. Le module fusion est divisé en deux sous modules : le sous module de fusion optimisé pour deux sources uniquement et le sous module de fusion sur plus de deux sources. Chacun d'eux implémente (lorsque cela est possible théoriquement) tous les algorithmes de fusion qu'ils soient symétriques ou dissymétriques.

Ces fonctionnalités doivent être implémentées en s'assurant que cinq critères sont respectés :

1. L'accès aux distributions sera le plus rapide possible,
2. Les algorithmes seront optimisés pour s'exécuter le plus rapidement possible,
3. Les calculs seront réalisés avec la précision maximale,
4. Le composant logiciel sera totalement portable quelle que soit l'architecture de la machine, le système d'exploitation utilisé et si possible indépendant du compilateur. L'utilisation de la librairie sera donc possible sur des systèmes embarqués,
5. Le composant sera facile d'utilisation et permettra d'ajouter simplement de nouvelles fonctionnalités.

La première contrainte doit être nuancée. En général, accéder rapidement à une information nécessite l'utilisation d'un très gros volume de mémoire. C'est par exemple le cas avec des tables de hachage. Une telle consommation de mémoire n'est pas toujours possible, notamment sur des systèmes embarqués. Les distributions seront donc stockées en utilisant une technique qui offre le meilleur compromis espace mémoire occupée par rapport à la vitesse d'accès.

Pour optimiser la vitesse d'exécution, les algorithmes ont été analysés dans les moindres détails et tout au long du développement un outil de mesure de vitesse permettra de valider l'analyse. La qualité du compilateur utilisé influe naturellement beaucoup sur les performances du composant logiciel¹.

La précision des calculs dépend énormément de la machine utilisée. Il aurait été possible d'écrire une bibliothèque de calcul numérique indépendante de la machine mais ce travail aurait été long et fastidieux. Après de nombreux tests, la précision standard des machines les plus couramment employées (stations de travail, PC . . .) a été jugée satisfaisante. Pour assurer une bonne stabilité numérique, toutes les algorithmes seront rédigés en évitant les constructions numériques instables comme par exemple les erreurs de cancellation².

La quatrième contrainte est actuellement la plus difficile à respecter. Il est humainement impossible de supporter tous les systèmes informatiques existants. Pour fonctionner, le composant DFOL nécessitera un

1. Sur PC, le compilateur CodeWarrior semble générer du code 20% plus rapide que son homologue Visual C++.

2. Une erreur de cancellation survient lors de la soustraction de deux nombres réels proches.

compilateur C++ respectant la norme C++ ANSI 2³ et disposant de la bibliothèque STL⁴. Actuellement, c'est le cas de toutes les machines fonctionnant sous UNIX puisque le compilateur gratuit egcs⁵ dispose de tous les composants nécessaires. Pour un PC, de nombreux outils satisfont ces contraintes. C'est par exemple le cas de Microsoft Visual C++⁶ répondent aussi aux exigences. Sur Macintosh, il n'y a guère à l'heure actuelle que l'environnement commercial CodeWarrior⁷ qui compile DFOL. Les architectures les plus répandues sont donc supportées.

L'utilisation et l'extension du composant logiciel devant être facile, une approche objet a été choisie. Il est ainsi facile d'étendre les fonctionnalités en dérivant d'une classe existante. De nombreux langages permettent la programmation par objets. Les plus répandus sont Pascal, Ada 95, C++, Java et smalltalk. Le langage C++ a été choisi car il offre le meilleur niveau de performances dans l'exécutable final et qu'il jouit au même titre que C d'une grande diffusion et d'une portabilité accrue.

Les trois paragraphes suivants détaillent les choix techniques réalisés pour résoudre le problème du stockage des distributions, la représentation des intervalles et la fusion des données.

6.1.1 La représentation des distributions

Pour réaliser la fusion avec le plus de précision possible, nous devons représenter les distributions de la manière la plus concise possible. D'une manière générale, les distributions de possibilités peuvent se classer en trois catégories :

1. Les distributions discrètes. Le stockage de n couples de la forme $(x, \pi(x))$ permet de représenter précisément et complètement une distribution discrète,
2. Les distributions de forme "géométriques" comme par exemple les triangles, trapèzes, gaussiennes . . . Il faut stocker les éléments caractéristiques de la distribution pour représenter cette dernière précisément et complètement,
3. Les distributions de forme quelconque pour lesquelles il est difficile de déterminer automatiquement les points caractéristiques. Il est par ailleurs impensable de stocker tous les points de la distribution. Le mode de stockage de ces distributions est stratégique car tout algorithme de fusion fournit une distribution de forme a priori quelconque.

Du point de vue informatique, stocker précisément une distribution discrète ou une distribution de forme "géométrique" implique la mémorisation de la forme de la distribution (trapèze, triangle . . .) et la mémorisation de n points caractéristiques. Cela s'implémente très facilement en œuvre avec un langage à objets en utilisant par exemple un arbre "red-black".

Le stockage de distributions quelconques nécessite un compromis entre la place mémoire occupée et la précision désirée. Nous avons choisi de stocker les distributions sous forme de segments connectés les uns aux autres comme le montre la figure 6.1 page suivante. L'échantillonnage des points de connexion n'est pas forcément régulier et l'utilisateur peut préciser lui-même les points de connexion comme bon lui semble. Il est donc possible d'augmenter localement le nombre de points de connexions pour mieux représenter une distribution qui le nécessiterai. Le nombre de points de connexion est limité uniquement par la mémoire disponible.

Les distributions disposent donc d'un bon compromis mémoire occupé rapporté au temps d'accès à l'information. Pour obtenir le degré d'appartenance d'un point, deux cas sont à considérer. Soit le point existe dans la structure (c'est donc l'extrémité d'un segment) soit il n'existe pas et il est nécessaire de réaliser une interpolation linéaire. Dans le premier cas, l'accès à la structure fournit le résultat. Dans le cas contraire,

3. Disponible sur <ftp://ftp.research.att.com/dist/c++std/WP/CD2>

4. "Standard Template Library" disponible sur <http://www.sgi.com/Technology/STL/>

5. Disponible sur <http://www.cygnum.com/egcs>

6. Informations sur <http://msdn.microsoft.com/vstudio>

7. Informations sur <http://www.metrowerks.com>

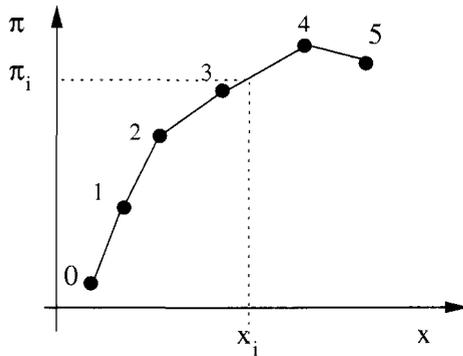


FIG. 6.1 – Représentation d'une distribution quelconque à l'aide de segments.

Le degré d'appartenance π_i au point x_i est donné par :

$$\frac{\pi_4 - \pi_3}{x_4 - x_3} = \frac{\pi_4 - \pi_i}{x_4 - x_i} \iff \pi_i = \pi_4 - \frac{(\pi_4 - \pi_3)(x_4 - x_i)}{x_4 - x_3}$$

il est nécessaire de trouver deux points connus qui encadrent le point inconnu et de réaliser une interpolation linéaire. Pour accélérer l'accès aux points, les extrémités des segments sont stockés dans une structure arborescente associative : le "reb-black tree" qui permet une recherche rapide de l'information.

Après fusion, les distributions obtenues comporteront en général un grand nombre de points. Tous ces points ne sont pas significatifs. Certains encombrant inutilement la mémoire et ralentissent les algorithmes. Il est donc nécessaire de prévoir un algorithme de simplification des distributions destiné à supprimer les points jugés inutiles. Cette technique de simplification doit pouvoir être automatique (chaque distribution est simplifiée automatiquement dès que possible) ou manuelle (le programmeur demande explicitement la simplification d'une distribution donnée). Par ailleurs, la "sévérité" de cet élagage doit pouvoir être modifiée facilement. Cette technique de simplification s'articule autour de deux étapes :

1. Réduction du nombre de points,
2. Elimination des paliers.

La première étape tente de supprimer tous les points non nécessaires de manière à réduire l'occupation mémoire et à accélérer les traitements. La seconde étape consiste à supprimer les faibles perturbations locales pour limiter les parasites dans le résultat final. La figure 6.2 présente, à gauche la réduction du nombre de points et à droite, la réduction des paliers.

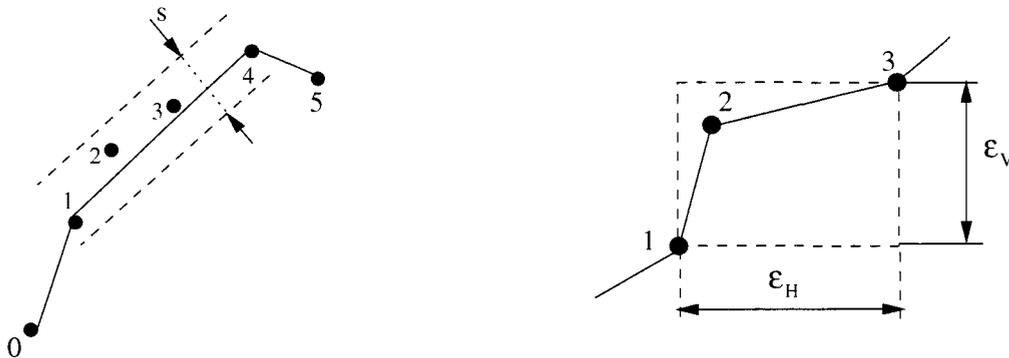


FIG. 6.2 – Deux possibilités de simplification des distributions quelconques.

6.1.1.1 Réduction du nombre de points

La réduction du nombre de points consiste à rechercher tous les points consécutifs pouvant être alignés sur une même droite de régression, à un seuil s près. Lorsqu'un alignement est détecté, tous les points le composant sont supprimés à l'exception du premier et du dernier. Sur la figure 6.2, les points 1 à 4 peuvent être alignés sur une même droite en respectant un seuil s . Dans la distribution résultante, seuls les points 0, 1 et 4, 5 seront conservés. Cela revient donc à éliminer les points contenus dans un "tube" de diamètre s .

L'alignement est vérifié de proche en proche en calculant la droite de régression linéaire d'équation $y = ax + b$ et le coefficient de corrélation r . Soit k le nombre de points actuellement alignés sur la droite de régression ($k > 1$), \bar{x} et \bar{y} les moyennes en x et en y des k points alignés. Les coefficients a_k , b_k , r_k sont donnés par :

$$a_k = \frac{\sum_{i=1}^k x_i y_i - k \cdot \bar{x} \cdot \bar{y}}{\sum_{i=1}^k x_i^2 - k \cdot \bar{x}^2}, \quad b_k = \bar{y} - a_k \bar{x}, \quad r_k = \frac{\left(\sum_{i=1}^k \frac{x_i y_i}{k} - \bar{x} \cdot \bar{y} \right)^2}{\left(\sum_{i=1}^k \frac{x_i^2}{k} - \bar{x}^2 \right) \cdot \left(\sum_{i=1}^k \frac{y_i^2}{k} - \bar{y}^2 \right)} \quad (6.1)$$

Pour détecter si les points d'indices k et $k + 1$ sont alignés, nous évaluons les nouveaux coefficients a_{k+1} , b_{k+1} , r_{k+1} à l'indice $k + 1$. Lorsque $|a_k - a_{k+1}| < s$ et $|b_k - b_{k+1}| < s$, le point d'indice $k + 1$ est considéré comme aligné et la recherche continue. Le coefficient de corrélation r permet de vérifier la décision. Lorsque le point $k + 1$ est détecté comme aligné, le coefficient r_{k+1} doit être le plus proche possible de 1. Dans le cas contraire, le point est rejeté et le segment est considéré comme terminé.

Le but de cette étape étant la diminution du nombre de points, le seuil s doit être le plus bas possible de manière à ne supprimer que les points réellement alignés. En pratique, ce seuil sera fixé à celui de la précision de la machine. Les différents tests que nous avons menés montrent une réduction du nombre de points d'environ 45% (en moyenne), le maximum étant atteint pour les ensembles obtenus à l'aide d'algorithmes utilisant le principe d'extension.

6.1.1.2 Elimination des paliers

Cette phase consiste à supprimer les "marches d'escaliers" souvent introduites par les algorithmes de fusion fondés sur le principe d'extension. Considérons trois points 1, 2, 3 consécutifs disposés comme sur la partie droite de la figure 6.2 page précédente. Ces trois points définissent un rectangle (en pointillés) de dimensions $\epsilon_H \times \epsilon_V$. Lorsque les deux dimensions ϵ_H et ϵ_V sont petites, le point central (noté 2) peut être supprimé. Ainsi, la fonction d'appartenance devient linéaire entre les points 1 et 3. Le choix des seuils ϵ_H et ϵ_V dépend du pas de calcul Δ_{Ext} utilisé lors de l'extension. En pratique, toute valeur vérifiant les inégalités suivantes donne de bons résultats :

$$\begin{cases} \epsilon_H & \leq \Delta_{Ext} \\ \epsilon_H & \leq \epsilon_V \leq 2\epsilon_H \end{cases} \quad (6.2)$$

Pour obtenir un ensemble réduit au maximum, il est nécessaire de répéter ces deux étapes jusqu'à ce que le résultat soit inchangé. Cette approche est très consommatrice de temps de calcul. Pour éviter une trop grande dégradation des performances, la simplification peut être réalisée en 3 phases : réduction du nombre de points, élimination des paliers, réduction du nombre de points.

6.1.2 La représentation de segments et d'intervalles

Le support du résultat de la fusion de deux distributions est en général un sous-ensemble de l'union des supports des distributions fusionnées. Il est donc impératif de connaître et de représenter avec précision les noyaux et supports des distributions et plus généralement tout intervalle de la droite réelle. Deux classes de DFOL réalisent ce travail : CSegment et CIntervalle.

6.1.2.1 Les segments

Un segment est un ensemble contigu de réels compris entre une borne inférieure et une borne supérieure. Les bornes supérieures ou inférieures peuvent être incluses ou non. La figure 6.3 page suivante présente quelques exemples de segments : une borne exclue est représentée à l'aide d'une parenthèse tandis qu'une borne incluse est représentée par un point.

FIG. 6.3 – Exemples de segments $[0; 3[$ et $]0; 3[$ dans DFOL.

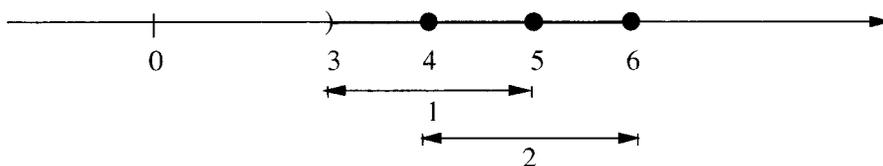
Il est nécessaire de définir un opérateur permettant de vérifier que deux segments sont identiques. Il s'agit de l'opérateur "=". Soient s_1 et s_2 deux segments, $\text{inf}(s)$ la borne inférieure du segment s , $\text{sup}(s)$ la borne supérieure du segment s , $\text{inclusion}(\text{sup}(s))$ le mode d'inclusion de la borne supérieure du segment s et $\text{inclusion}(\text{inf}(s))$ le mode d'inclusion de la borne inférieure du segment s , l'opérateur "=" est défini comme :

$$s_1 = s_2 \iff \begin{cases} \text{inf}(s_1) = \text{inf}(s_2) \\ \text{sup}(s_1) = \text{sup}(s_2) \\ \text{inclusion}(\text{inf}(s_1)) = \text{inclusion}(\text{inf}(s_2)) \\ \text{inclusion}(\text{sup}(s_1)) = \text{inclusion}(\text{sup}(s_2)) \end{cases} \quad (6.3)$$

Une relation d'ordre total entre les segments est aussi nécessaire. Nous choisirons une relation simple : ordonner deux segments consiste à les ranger dans l'ordre de leurs bornes inférieures :

$$\begin{aligned} s_1 < s_2 &\iff \text{inf}(s_1) < \text{inf}(s_2) \\ s_1 > s_2 &\iff \text{inf}(s_1) > \text{inf}(s_2) \end{aligned} \quad (6.4)$$

Cette relation d'ordre est exploitée en interne par le composant logiciel et l'utilisateur n'y a pas directement accès. Elle a été choisie car elle simplifie beaucoup les algorithmes de calcul d'union et d'intersection entre intervalles comme nous le verrons dans les paragraphes suivants. La figure 6.4 présente un exemple de la relation d'ordre entre $]3; 5]$ et $[4; 6]$:

FIG. 6.4 – Relation d'ordre entre les segments $]3; 5]$ et $[4; 6]$.

Outre les opérateurs décrits ci-dessus, d'autres fonctionnalités sont disponibles. C'est le cas par exemple des opérations d'union, d'intersection ... Les segments ne peuvent représenter qu'un ensemble contigu de réels ce qui est insuffisant pour stocker l'union de deux distributions. Ce problème est résolu à l'aide de la notion d'intervalles.

6.1.2.2 Les intervalles

Un intervalle est défini comme l'union de segments. Il devient alors possible de représenter des ensembles de réels non contigus. Dans DFOL, un intervalle est une liste ordonnée d'éléments de type `Csegment`. L'ordre des segments dans la liste est donc l'ordre "naturel" lorsque l'on dessine les segments sur la droite réelle. Par exemple, $[3; 5] \cup]-4; 1] \cup [5; 6]$ est un intervalle stocké sous la forme $] -4; 1] \cup [3; 5] \cup [5; 6]$.

Les opérations d'union et d'intersection entre intervalles sont disponibles mais les algorithmes utilisés nécessitent quelques explications. Quelle que soit l'opération (union ou intersection), une nouvelle liste temporaire contenant les segments des deux intervalles est créée⁸.

8. Pour des raisons de performances, la librairie ne crée pas cette liste : les structures existantes sont modifiées "en place". Il s'agit là d'une optimisation de la place mémoire qui ne change rien au principe du traitement. L'utilisation d'une nouvelle liste sert uniquement à faciliter l'explication et la compréhension de l'algorithme.

6.1.2.2.1 L'union. La liste temporaire est parcourue dans l'ordre croissant des segments (de la tête vers la queue). Dès que deux segments consécutifs peuvent être réduits en un seul segment, cette réduction est opérée et l'analyse reprend avec comme premier élément le résultat de la combinaison. Nous présentons les manipulations faites par cet algorithme sur la liste temporaire lors l'union de $[-10; -8] \cup [3; 5] \cup [7; 8]$ et de $[-9; 2] \cup]7; 8[$. Les segment soulignés sont ceux qui sont susceptibles d'être combinés.

$$\begin{array}{cccccc}
 \underline{[-10; -8]} & [-9; 2] & [3; 5] & [7; 8] &]7; 8[& \\
 & \underline{[-10; 2]} & [3; 5] & [7; 8] &]7; 8[& \\
 & [-10; 2] & \underline{[3; 5]} & [7; 8] &]7; 8[& \\
 & [-10; 2] & [3; 5] & \underline{[7; 8]} &]7; 8[& \\
 & [-10; 2] & [3; 5] & [7; 8] & \underline{]7; 8[} &
 \end{array}$$

Lors de la première étape, les deux segments peuvent être combinés pour former un seul segment. Les deux segments combinés sont remplacés par le résultat de la combinaison. L'analyse continue avec nouveau segment formé comme premier élément. La combinaison est impossible : le premier segment analysé est "sauté" et l'analyse reprend. Nouvel échec. La ligne quatre met en évidence une possibilité de combinaison. Il ne reste alors plus qu'un seul segment dans la liste, le processus s'arrête et la liste temporaire ainsi modifiée est le résultat cherché.

6.1.2.2.2 L'intersection. Elle fonctionne d'une manière comparable à l'union. La liste temporaire est parcourue dans l'ordre croissant des segments (de la tête vers la queue). Dès que deux segments consécutifs ont au moins un point en commun, ils peuvent être réduits en un seul segment, cette réduction est opérée et le résultat est stocké dans une autre liste temporaire. Nous présentons les manipulations faites par cet algorithme sur la liste temporaire (à gauche) et sur la liste résultat (à droite) lors de l'intersection de $[-10; -8] \cup [3; 5] \cup [7; 8]$ et de $[-9; 2] \cup]7; 8[$. Les segment soulignés sont ceux qui sont susceptibles d'être combinés.

$$\begin{array}{cccccc}
 \underline{[-10; -8]} & [-9; 2] & [3; 5] & [7; 8] &]7; 8[& \parallel & [-9; -8] \\
 [-10; -8] & \underline{[-9; 2]} & [3; 5] & [7; 8] &]7; 8[& \parallel & [-9; -8] \\
 [-10; -8] & [-9; 2] & \underline{[3; 5]} & [7; 8] &]7; 8[& \parallel & [-9; -8] \\
 [-10; -8] & [-9; 2] & [3; 5] & \underline{[7; 8]} &]7; 8[& \parallel & [-9; -8] \\
 [-10; -8] & [-9; 2] & [3; 5] & [7; 8] & \underline{]7; 8[} & \parallel & [-9; -8] &]7; 8[
 \end{array}$$

Lors de la première étape, les deux segments peuvent être combinés pour former un seul segment. Le résultat de la combinaison est placé dans une nouvelle liste vide (à droite). L'analyse continue avec le segment suivant dans la liste temporaire (à gauche). La combinaison est impossible : le premier segment analysé est "sauté" et l'analyse reprend. Nouvel échec. La ligne quatre met en évidence une possibilité de combinaison dont le résultat est ajouté au résultat. Il ne reste alors plus qu'un seul segment dans la liste, le processus s'arrête et la liste de droite contient le résultat.

6.1.3 La fusion de données

Dans les chapitres précédents, nous avons montré qu'il existait au moins deux grandes manières de fusionner : la fusion "classique" et la fusion dissymétrique. La fusion dissymétrique est implantée différemment de la fusion symétrique. En effet, la fusion dissymétrique nécessite la présence d'une information *a priori*. Une fois la fusion dissymétrique réalisée, le résultat devient alors l'information *a priori* pour la fusion suivante. Les algorithmes de fusion dissymétriques stockent l'information *a priori* pour simplifier la tâche de l'utilisateur. Il aurait été possible de laisser l'utilisateur fournir l'information *a priori* et la nouvelle information à chaque fusion, mais cela ouvre la porte à une modification involontaire de l'information *a priori*.

Certains algorithmes de fusion ne sont prévus que pour deux sources (c'est le cas de la règle adaptative de DUBOIS et PRADE 2.3.1 page 39 par exemple) il est donc nécessaire de subdiviser la fusion classique en deux grands types d'algorithmes : les algorithmes concernant deux sources et les autres. En ce qui concerne la

fusion dissymétrique, nous n’avons considéré que le conditionnement qui se décline en trois sous-familles : l’expansion, la contraction et la révision. La librairie disposera donc de modules permettant :

1. la fusion symétrique de deux sources exclusivement,
2. la fusion symétrique de plus de deux sources,
3. la fusion dissymétrique (comprenant l’expansion, la contraction et la révision).

Par ailleurs, il est nécessaire de pouvoir fusionner deux sources avec des techniques prévues pour plus de deux sources et ce, sans perte de performances. Les algorithmes de fusion sur n sources seront donc spécialisés pour pouvoir travailler sur deux sources sans utiliser les algorithmes “ n sources” forcément plus lourds.

Le grand nombre d’opérateurs ayant des comportements comparables (la fusion avec min ou avec max est faite par le même algorithme dans lequel une seule ligne change) suggère l’utilisation de la programmation générique. Dans ce paradigme, le programmeur définit le modèle (ou “template”) et le compilateur se charge de gérer les modifications (remplacer min par max, . . .) lorsque cela est nécessaire. Les paragraphes suivants s’intéressent aux spécificités de mise en œuvre de chacune des formes de fusion.

6.1.3.1 Fusion de deux sources

Du point de vue de DFOL, la fusion de deux sources peut se faire de trois manières différentes résumées dans le tableau 6.1. Il est donc nécessaire de définir un modèle pour chaque comportement A, B, C . Tout opérateur de fusion entre deux sources héritera au sens objet des propriétés du modèle et il ne restera qu’à programmer la fonction correspondante, par exemple $\min(\omega_1, \omega_2)$.

Type	Description
A	Le résultat au point ω est obtenu en combinant deux degrés de possibilités $\pi_1(\omega)$ et $\pi_2(\omega)$. C’est le cas des T-Normes simples.
B	Le résultat au point ω est obtenu en combinant deux degrés de possibilités $\pi_1(\omega)$ et $\pi_2(\omega)$ à l’aide d’une relation qui nécessite un paramètre réel γ constant, fixé à la compilation. C’est le cas des T-Normes paramétrées.
C	Le résultat au point ω est obtenu en combinant deux degrés de possibilités $\pi_1(\omega)$ et $\pi_2(\omega)$ à l’aide d’une relation qui nécessite un paramètre réel γ constant pendant la fusion, mais inconnu à la compilation. C’est le cas des règles adaptatives, le paramètre étant par exemple un indice de conflit.

TAB. 6.1 – Les trois manières de fusion (cas multisources) vues par DFOL.

6.1.3.2 Fusion de plus de deux sources

Du point de vue de DFOL, fusionner plus de deux sources nécessite la gestion d’une liste ordonnée de sources. Les algorithmes accèdent alors à la liste – éventuellement dans un ordre différent de celui de la liste – et fusionnent n sources. Le chapitre 3 nous apprend qu’il y a trois grands types d’opérateurs fusionnant plus de deux sources. Ces types sont détaillés dans le tableau 6.2 page suivante. Pour faciliter l’ajout d’algorithmes de fusion sur n sources, nous allons créer une classe dont le but sera de maintenir la liste de sources. Cela étant fait, nous créons trois modèles qui serviront de base à tous les opérateurs de fusion multisources. Le tableau 6.3 page ci-contre détaille les paramètres de ces modèles.

6.1.3.3 La fusion dissymétrique

Le chapitre 4 décrit de nombreux opérateurs dissymétriques. Nous nous limitons à l’implémentation des opérateurs de révision. Du point de vue de la programmation, ceux-ci sont comparables aux opérateurs fusionnant deux sources seulement, seule la manière de calculer le résultat change. Ce qui a été dit au paragraphe

Type	Description
A	Les sources sont combinées à l'aide d'une technique "cascade". Un objet de fusion entre deux sources est appliqué autant de fois que nécessaire. L'ordre dans lequel les sources sont considéré est éventuellement différent de celui de la liste de sources.
B	Les sources sont combinées à l'aide d'une technique de réduction à deux sources. Il existe donc deux type de fusion : la première utilise les sources réelles pour générer deux sources virtuelles et la seconde combine ses deux sources virtuelles à l'aide d'un algorithme de fusion entre deux sources .
C	Les sources sont combinées à l'aide d'une technique explicitement prévue pour n sources. Du point de vue de la programmation, les méthodes "hybrides" rentrent aussi dans cette catégorie.

TAB. 6.2 – Les trois manières de fusion vues par DFOL.

Type	Description
A	Dépend d'un opérateur de fusion sur deux sources
B	Dépend d'un opérateur de réduction et d'un opérateur de fusion sur deux sources.
C	Nécessite la création d'une classe spéciale.

TAB. 6.3 – Les paramètres des modèles dans le cas multisources.

6.1.3.1 est aussi valable ici. Nous allons créer un modèle pour tous les opérateurs de révision qui sera adapté à la compilation.

Le cahier des charges étant précisé, nous nous intéressons maintenant à l'implémentation. Nous présentons les classes créées et la hiérarchie entre elles. Les modèles de fusion sont aussi décrits.

6.2 L'architecture de DFOL

Nous décrivons maintenant l'implémentation [Zunino et al., 1997c] qui en a été faite et notamment la hiérarchie des objets et les différents modèles génériques. Certaines classes importantes sont décrites mais le grand nombre d'objets de DFOL interdit une description exhaustive de la librairie dans ce document. Le lecteur intéressé pourra consulter la documentation fournie avec le code source pour plus de détails.

Les paragraphes suivants adoptent certaines conventions de notation. Les éléments de programmes sont écrits en caractères "machine à écrire" et pour ces éléments, les majuscules et minuscules seront différentes. Tout élément de programme précédé de la double barre oblique "/" désigne un commentaire. Les noms de classes sont toujours précédés d'un "C" pour rappeler qu'il s'agit d'une classe comme par exemple CSegment. Les structures verront leur nom toujours précédé d'un "s" comme dans sAdaptativeDu-boisPrade.

Un modèle est reconnaissable à son nom qui toujours suivi de "<X>" dans lequel "X" est un type valide. Nous parlerons par exemple du modèle CFusionTNorme<> qui implémente la fusion par T-Norme/T-Conorme. L'opération d'utiliser un modèle pour créer un élément de programme est nommée "instanciation". Par exemple CFusionTNorme<minimum> désignera donc une classe capable de fusionner deux sources par la T-Norme min obtenue à l'aide du modèle CFusionTNorme<>.

Dans les paragraphes suivants, nous abordons tout d'abord l'architecture du composant logiciel en présentant le nom des fichiers d'en-tête, puis nous nous intéressons à quelques classes importantes comme les intervalles et segments, les distributions, les indicateurs de conflit et les algorithmes de fusion.

6.2.1 Introduction à la programmation avec DFOL

Pour utiliser le composant logiciel DFOL, il est nécessaire d'inclure certains fichiers⁹ définissant les structures de données et algorithmes disponibles. Ce paragraphe donne la liste de quelques fichiers d'en-tête et précise les fonctionnalités auxquelles ils donnent accès.

L'ensemble de la bibliothèque utilise largement les intervalles et segments. Les fichiers nécessaires sont donc automatiquement inclus par tous les algorithmes de fusion. Pour mémoire, nous présentons le nom des fichiers associés aux segments et intervalles dans le tableau 6.4. Les modules de gestion des segments et

Segments – Intervalles	
CSegment.h	Primitives de traitement sur les segments
CIntervalle.h	Primitives de traitement sur les intervalles

TAB. 6.4 – Fichiers d'en-tête pour les segments et les intervalles.

intervalles sont totalement dissociés des modules de fusion ce qui permet leur utilisation dans n'importe quel programme, sans à avoir à inclure les algorithmes de fusion.

Les distributions nécessitent elles aussi l'inclusion de fichiers en-tête. Il existe un fichier par type de distribution ce qui facilite l'utilisation. Le tableau 6.5 indique les fichiers à utiliser. Le fichier CDistribution.h ne décrit pas une distribution a proprement parler. Il contient l'ensemble des abstractions nécessaires pour toutes les distributions. Son inclusion est réalisée automatiquement dans les autres fichiers des distributions.

Distributions	
CDistribution.h	Déclaration générales. Inclus automatiquement par les autres fichiers
CTriangle.h	Représentation des distributions triangulaires
CTrapeze.h	Représentation des distributions trapézoïdales
CGaussienne.h	Représentation des distributions gaussiennes
CMembershipPoint.h	Représentation d'un point d'appartenance
CQuelconque.h	Représentation des distributions quelconques
CNNQuelconque.h	Représentation des distributions quelconques non normalisées
CDiscret.h	Représentation des distributions discrètes
CCrisp.h	Représentation des ensembles discrets classiques

TAB. 6.5 – Fichiers d'en-tête pour les distributions.

En ce qui concerne les algorithmes de fusion, les en-têtes disponibles sont présentés dans le tableau 6.6 page 114. Les classes CFusionAlg et CRevisionAlg sont deux classes abstraites destinées à implémenter les comportements communs de tous les opérateurs de fusion symétriques et dissymétriques.

Les trois catégories d'opérateurs de fusion décrit dans le cahier des charges sont bien présentes. La hiérarchie de toutes ces classes est présentée figure 6.5. Nous y distinguons deux modèles dont nous n'avons pas encore parlé : CNNFusionAdaptative et CNNMultiFusionAdaptative. Il s'agit respectivement des modèles permettant la mise en place d'opérateurs de fusion dont le résultat n'est pas normalisé, pour deux sources et pour n sources. Les opérateurs de fusion disposent tous d'un comportement commun implémenté par CFusionAlg.

Les paragraphes suivants s'intéressent aux méthode des classes les plus importantes : CSegment, CFusionTNorme, CFusionAdaptative. Nous abordons aussi les algorithmes de calcul des conflits.

9. En sus des fichiers d'en tête standards tels que iostream, exception, iomanip, cmath...

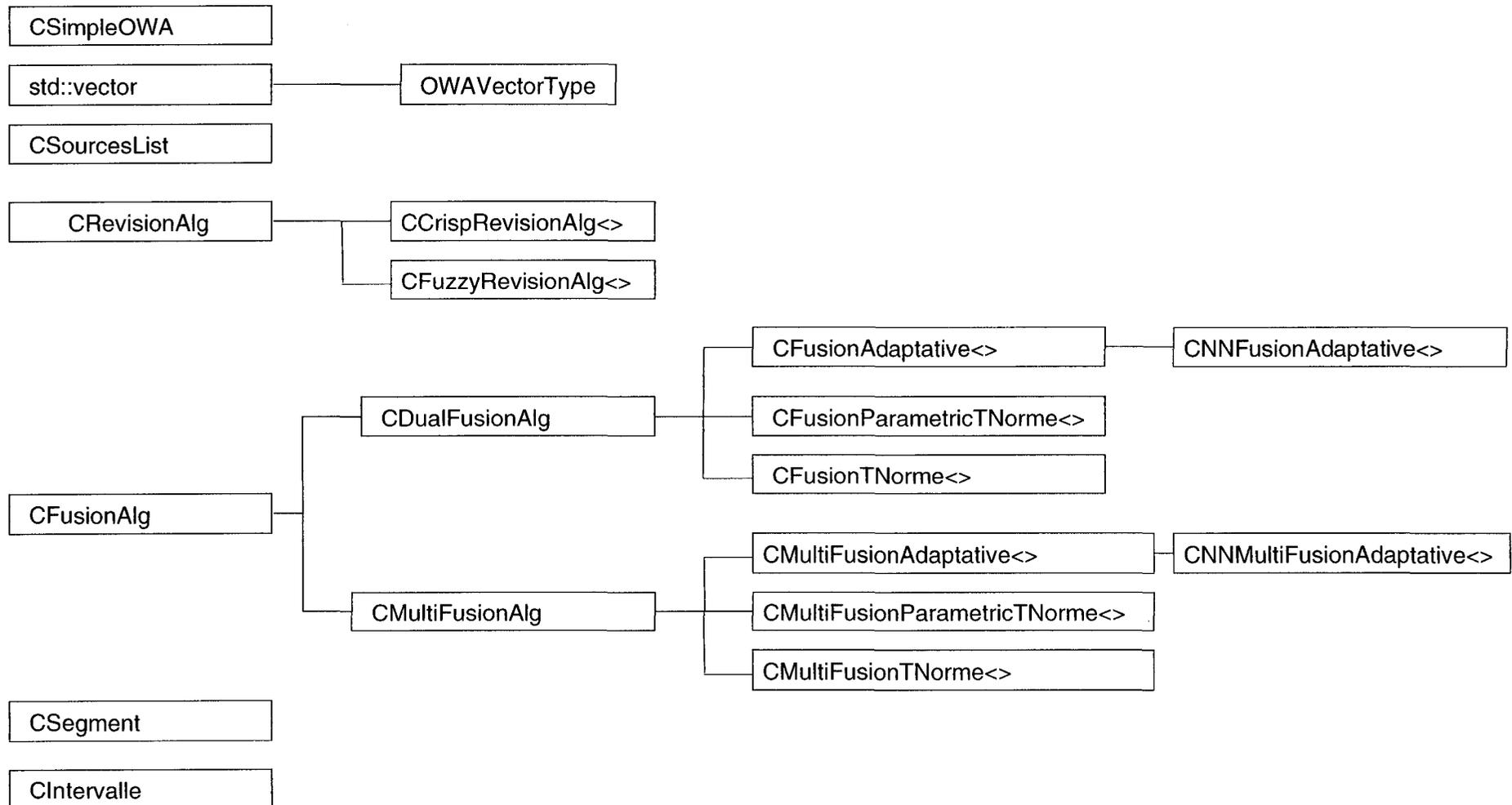


FIG. 6.5 – Hiérarchie des classes de gestion d'intervalles dans DFOL.

Fusion	
CFusionAlg.h	Déclaration générales.
CRRevisionAlg.h	Déclaration générales (fusion dissymétrique).
CDualFusionAlg.h	Modèle pour la fusion de deux sources
CMultiFusionAlg.h	Modèle pour la fusion de n sources
CFusionTNorme.h	Fusion de deux sources par T-Norme
CFusionParametricTNorme.h	Fusion de deux sources par T-Norme paramétrée
CFusionAdaptative.h	Fusion de deux sources par les règles adaptatives
CMutliFusionTNorme.h	Modèle de fusion de n sources par T-Norme
CMultiFusionParametricTNorme.h	Modèle de fusion de n sources par T-Norme paramétrée
CMutliFusionAdaptative.h	Modèle de fusion de n sources par les règles adaptatives
CSourcesList.h	Liste des sources pour fusion de n sources
CSimpleOWA.h	Opérateurs OWA simples

TAB. 6.6 – Fichiers d'en-tête pour les algorithmes de fusion entre deux sources.

6.2.2 Les algorithmes de gestion d'intervalles

Les intervalles de notre composant logiciel sont en tout points identiques à ceux manipulés en mathématiques et supportent les opérations classiques telles que l'intersection, l'union, l'égalité entre deux intervalles, la vérification de l'inclusion d'un intervalle dans un autre, ... Par ailleurs, les opérations d'entrée/sortie sur les intervalles sont disponibles.

La classe permettant la gestion des intervalles est nommée `CIntervalle`. De manière interne, la librairie représente un intervalle réel comme l'union de plusieurs objets de type `CSegment`. Un objet de type `CSegment` est chargé de représenter un segment au sens mathématique. Ces deux classes sont des classes de base : elles n'ont aucun parent. Leur hiérarchie est présentée figure 6.6. Les intervalles sont en tous points



FIG. 6.6 – Hiérarchie des classes `CIntervalle` et `CSegment`.

identiques à ceux manipulés en mathématiques et supportent les opérations classiques tels que l'intersection, l'union, l'égalité entre deux intervalles, la vérification de l'inclusion d'un intervalle dans un autre, ... Par ailleurs, les opérations d'entrée/sortie sur les intervalles sont disponibles. La liste des méthodes de la classe `CIntervalle` est visible dans le tableau 6.7.

6.2.3 Les indices de conflit

Ils permettent d'évaluer le conflit entre deux distributions. A la base des opérateurs de fusion, ils sont définis comme de toutes petites classes toutes descendantes d'une même classe abstraite : `CSimilarite`. La hiérarchie des classes de conflits est présentée figure 6.7 page 116.

Parmi ces indices de conflit, nous reconnaissons l'indice h de DUBOIS et PRADE (HauteurH) ainsi que les indices construits sur les distances et les indices relatifs à l'analyse de données. Les principales méthodes de la classe `CSimilarite` sont visibles dans le tableau 6.8 page 116.

Méthodes de la classe CIntervalle	
<code>friend ostream & operator << (ostream &os, const CIntervalle &S)</code>	Ecriture de l'intervalle vers le flux os
<code>CIntervalle(void)</code>	Constructeur par défaut
<code>CIntervalle(const CSegment &S)</code>	Constructeur à partir d'un segment
<code>CIntervalle(const double &lower, const double &upper)</code>	Constructeur à partir de la borne inférieure et supérieure
<code>double inf(void) const throw (Borne_Inexistante)</code>	Donne la borne inférieure
<code>double sup(void) const throw (Borne_Inexistante)</code>	Donne la borne supérieure
<code>bool In(const double &x) const</code>	x est-il dans le segment?
<code>CIntervalle & operator +=(const double &delta)</code>	Décale l'intervalle de delta
<code>CIntervalle & operator = (const CSegment &S)</code>	Calcule l'union avec un segment
<code>CIntervalle & operator &= (const CSegment &S)</code>	Calcule l'intersection avec un segment
<code>CIntervalle operator (const CSegment &S) const</code>	Calcule l'union avec un segment
<code>CIntervalle operator & (const CSegment &S) const</code>	Calcule l'intersection avec un segment
<code>CIntervalle & operator = (const CIntervalle &I)</code>	Calcule l'union avec un intervalle
<code>CIntervalle & operator &= (const CIntervalle &I)</code>	Calcule l'intersection avec un intervalle
<code>CIntervalle operator (const CIntervalle &I) const</code>	Calcule l'union avec un intervalle
<code>CIntervalle operator & (const CIntervalle &I) const</code>	Calcule l'intersection avec un intervalle
<code>CIntervalle & operator +=(const double &delta)</code>	Décale l'intervalle de delta
<code>bool operator ==(const CIntervalle &inter) const</code>	Vrai si les deux intervalles sont identiques

TAB. 6.7 – Quelques méthodes de la classe CIntervalle.

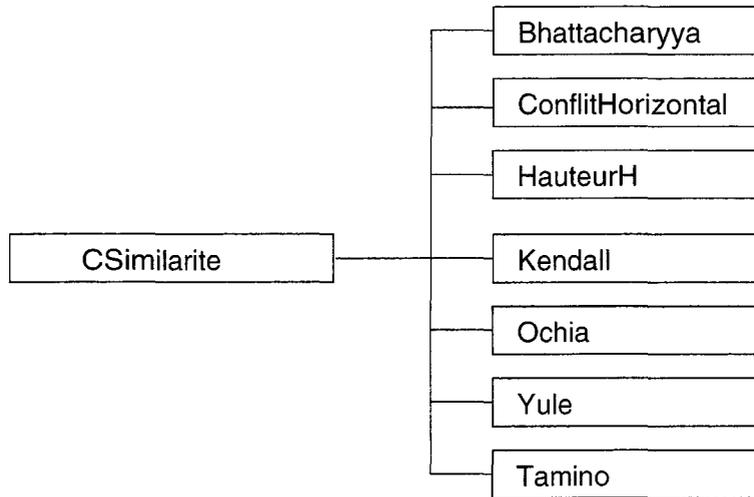


FIG. 6.7 – Hiérarchie (simplifiée) des classes de conflit dans DFOL.

Méthodes de la classe CSimilarite	
CSimilarite(void)	Constructeur
double Conflit(const CDistribution *d1,const CDistribution *d2) const	Calcule le conflit
const std::string & GetConflitName(void) const	Nom méthode

TAB. 6.8 – Méthodes de la classe CSimilarite.

6.2.4 Les distributions

Les classes permettant la représentation de distributions et leur hiérarchie sont représentées sur la figure 6.5. La classe CMembershipPoint est une classe utilitaire qui permet de représenter un point d'appartenance $(x, \pi(x))$. Elle est utilisée par les autres types de distributions. La classe CDistribution contient toutes les abstractions nécessaires utilisées par les autres distributions. Pour des raisons d'optimisation en place mémoire, il existe une classe par type de distribution : les gaussiennes, les trapèzes, les triangles, les distributions discrètes (CDiscret), les distributions quelconques toujours normalisées(CQuelconque), les distributions quelconques non normalisées (CNNQuelconque) et enfin, les ensembles au sens classique (CCrisp). La figure 6.9 présente quelques méthodes de la classe CNNQuelconque.

Outre les méthodes d'ajout et de destruction d'un point dans la distribution, nous remarquons la présence d'une méthode d'affichage de la distribution, de la fonction de normalisation et de la fonction de simplification. Les méthodes d'obtention des α -coupes, du noyau et du support de la distribution sont aussi présentes.

6.2.5 Les algorithmes de fusion

Les algorithmes de fusion descendant tous du même modèle, nous décrivons uniquement les méthodes communes (visibles tableau 6.10 page 118). L'objet de fusion est construit avec les deux distributions à combiner. Le résultat de la fusion s'obtient alors en exécutant l'objet fonction (c'est-à-dire en appelant `operator() (void)`) pour obtenir toute la distribution. Il est aussi possible d'obtenir le résultat de la fusion au point x . Enfin, chaque opérateur de fusion dispose d'une chaîne de caractères dans laquelle est stocké le nom de la méthode de fusion.

Méthodes de la classe CNNQuelconque	
friend ostream & operator << (ostream &os, const CNNQuelconque &S)	Ecriture vers le flux os
CNNQuelconque(void)	Constructeur par défaut
void InsertPoint(const MembershipPoint &inPoint)	Insère un point dans la distribution
void InsertPoint(const double &inX, const double &inY)	Insère un point dans la distribution
void RemovePoint(const MembershipPoint &inPoint)	Détruit un point de la distribution
void RemovePoint(const double &inX)	Détruit un point de la distribution
CDistribution *Clone(void) const	Copie la distribution
double GetMinimum(void)	Donne le plus petit degré de possibilité
double GetMaximum(void)	Donne le plus grand degré de possibilité
CNNQuelconque &operator +=(const double &delta)	Décale la distribution de delta
const double operator [] (const double &x)	Donne le degré d'appartenance de x
bool operator == (const CNNQuelconque &inNN) const	Comparaison de distributions
bool operator == (const CNNQuelconque *inNN) const	Comparaison de distributions
void Normalize(void)	Normalise la distribution
void Simplify(const double &eps)	Simplifie la distribution
double Surface(void) const	Surface sous la distribution
CIntervalle Kernel(void) const	Calcul du noyau
CIntervalle Support(void) const	Calcul du support
CCrisp AlphaCut(const double &x) const	Calcul de l' α -coupe

TAB. 6.9 – Quelques méthodes de la classe CIntervalle.

Méthodes du modèle CFusionTNorme	
CFusionTNorme(const CDistribution &d1,const CDistribution &d2)	Constructeur
CFusionTNorme(const CFusionTNorme &inSrc)	Constructeur par copie
CFusionTNorme &operator =(const CFusionTNorme &inSrc)	Affectation
CDistribution *operator () (void) const	Fusion
double operator () (const double &x)	Fusion en x
std::string GetMethodName(void) const	Méthode

TAB. 6.10 – Méthodes pour le modèle CFusionTNorme.

6.3 Exemples

Cette partie présente quelques exemples pratiques d'utilisation de DFOL. Le but n'est pas de fournir un exemple exhaustif d'utilisation de toutes les fonctionnalités mais de faciliter la prise en main de la librairie. Dans tous les exemples suivants, les distributions sont modélisées à l'aide de trapèzes ou de distributions quelconques sans aucune perte de généralité car changer le type d'une distribution se résume à changer la déclaration d'une variable.

Dans les paragraphes suivants, nous nous intéressons à la création de distributions et à la manipulation des intervalles, puis aux algorithmes de fusion entre deux sources : la fusion par T-Norme/T-Conorme, la fusion adaptative et la fusion dissymétrique. Le lecteur est invité à se reporter à la librairie pour des exemples de fusion pour plus de deux sources.

6.3.1 Les distributions, segments et intervalles

Le fragment de programme suivant déclare deux distributions, calcule l'intervalle union des supports et affiche à l'écran l'intervalle résultat. Puis, nous vérifions si l'union des supports est identique au noyau de la distribution t1 :

```
CTrapeze      t1(1,2,3,4);
CTrapeze      t2(0,1,2,4);
CIntervalle   union_supports = t1.Support() | t2.Support();

std::cout << "L'union des supports est "
           << union_supports << std::endl;

// Union des supports = noyau de t1 ?
if (union_supports == t1.Kernel())
    std::cout << "union_supports = Kernel(t1)" << std::endl;
else
    std::cout << "union_supports != Kernel(t1)" << std::endl;

// *****
// Résultat de l'exécution
// *****
L'union des supports est [0;4]
union_supports != Kernel(t1)
```

Les distributions permettent d'autres manipulations intéressantes comme le décalage de toute la distribution vers la droite ou vers la gauche. Le fragment de programme suivant déclare t1, l'affiche, le décale vers la droite d'une unité et affiche le résultat :

```
CTrapeze      t1(1,2,3,4);

std::cout << t1 << std::endl;
t1 += 1;
std::cout << t1 << std::endl;
```

```
// *****
// Résultat de l'exécution
// *****
Trapeze[a = 1, b = 2, c = 3, d = 4]
Trapeze[a = 2, b = 3, c = 4, d = 5]
```

Il est aussi possible de déterminer toute α -coupe pour toute distribution comme le montre l'exemple suivant dans lequel nous affichons la coupe de niveau 0.5 :

```
CTrapeze      t1(1,2,3,4);

      std::cout << t1.AlphaCut(0.5) << std::endl;

// *****
// Résultat de l'exécution
// *****
Crisp[1.5;3.5]
```

L' α -coupe est bien un ensemble au sens classique (Crisp). Les segments, intervalles et distributions disposent de nombreux autres utilitaires non présentés ici. Nous nous intéressons maintenant à quelques exemples de fusion.

6.3.2 La fusion

Dans cet exemple, nous déclarons deux distributions, que nous fusionnons entre elles en utilisant une T-Norme : le min. Le résultat de la fusion est alors affiché. Il est intéressant de noter la manière avec laquelle le modèle CFusionTNorme<> a été instancié pour le T-Norme minimum :

```
CTrapeze      t1(1,2,3,4);
CTrapeze      t2(0,1,2,4);
CFusionTNorme<minimum> Fusion(t1,t2);
CNNQuelconque *resultat;

      resultat = dynamic_cast<CNNQuelconque *>(Fusion());
      std::cout << resultat << std::endl;

// *****
// Résultat de l'exécution
// *****
NNQuelconque[(0, 0), (0.001, 0), (0.002, 0), (0.003, 0), (0.004, 0),
(0.005, 0), (0.006, 0), (0.007, 0), (0.008, 0), (0.009, 0),
(0.01, 0), (0.011, 0), (0.012, 0), (0.013, 0), (0.014, 0),
..... environ 200 lignes de couples .....
(4, 1.64757e-13)]
```

La fusion se fait donc en deux étapes. Tout d'abord, créer un objet (ici la variable "Fusion") et lui indiquer les distributions à combiner lors de la création. La seconde étape consiste à exécuter la fusion par Fusion() qui fournit le résultat. Le résultat d'un algorithme de fusion est un pointeur sur une distribution abstraite de manière à ce que le résultat puisse être utilisé partout par liaison dynamique. Cependant, les algorithmes de fusion renvoient toujours une distribution de type CNNQuelconque et il est donc possible de convertir de manière sûre le résultat d'un opérateur de fusion vers une CNNQuelconque à l'aide de dynamic_cast<CNNQuelconque *>().

La fusion de deux distributions de forme particulière n'est pas en général une distribution de forme particulière. C'est pour cela que l'affichage du résultat est une liste de couples $(x, \pi(x))$. Il est possible de normaliser et de simplifier le résultat. Pour cela, nous ajoutons à la fin de l'exemple précédent les deux lignes suivantes :

```
resultat->Normalize();
```

```

resultat->Simplify(1E-8);
std::cout << resultat << std::endl;

// *****
// Résultat de l'exécution
// *****
[sortie du programme précédent non reportée]
NNQuelconque[(0, 0), (1, 6.66134e-16), (2, 1), (4, 1.64757e-13)]

```

où 10^{-8} représente la précision de la machine sur laquelle ce programme a été exécuté. Le résultat est bien une distribution triangulaire, normalisée. Notons l'effet spectaculaire de la simplification qui est capable de retrouver la forme des distributions sur certains cas simples.

La fusion par la règle adaptative de DUBOIS et PRADE est tout aussi simple. Il est nécessaire d'instancier le modèle `CFusionAdaptative<>` avec l'indice de conflit à utiliser (ici h défini par `HauteurH`) et la structure de calcul de la règle définie par `sAdaptativeDuboisPrade`. Le fragment de programme est alors :

```

CTrapeze                               t1(1,2,3,4);
CTrapeze                               t2(0,1,2,4);
CFusionAdaptative<HauteurH,sAdaptativeDuboisPrade> Fusion(t1,t2);
CNNQuelconque                          *resultat;

    resultat = dynamic_cast<CNNQuelconque *>(Fusion());
    std::cout << resultat << std::endl;

// *****
// Résultat de l'exécution
// *****
Quelconque[(0, 0), (0.001, 1.09468e-13), (0.002, 1.09468e-13),
..... environ 300 lignes de couples .....
(4, 1.64757e-13)]

```

Il a simplement été nécessaire de modifier une ligne : la définition de l'objet qui réalise la fusion. Là encore, les opérations de normalisation (inutile ici) et de simplifications sont disponibles.

La fusion dissymétrique s'utilise de manière tout aussi simple. Le modèle `CREvisionAlg<>` est dédié à cette tâche. Il s'utilise comme type d'un objet de fusion, de la même manière que dans les exemples précédentes.

6.4 Conclusion

Ce chapitre a décrit notre contribution aux bibliothèques de manipulations de quantités floues et de distributions de possibilités. Notre composant peut représenter de nombreuses distributions comme les triangles, les gaussiennes ... et fournit de nombreuses fonctions de manipulations dont le calcul des α -coupes est un exemple. La possibilité de manipuler des segments et intervalles réels permet de représenter exactement le noyau et le support de toute distribution. Les algorithmes de fusion sont faciles à utiliser et il est aisé d'en ajouter de nouveaux. Enfin, le composant logiciel est utilisable sur plusieurs plateformes, éventuellement sur des systèmes embarqués à l'aide du compilateur croisé gratuit `gcc` qui peut générer un exécutable pour une trentaine de cibles différentes dont une bonne vingtaine sont des processeurs embarqués.

Après avoir présenté de nombreux opérateurs de fusion, force est de constater que, face à un problème pratique, le choix de la méthode de fusion n'est pas facile. L'utilisateur cherche l'opérateur "le plus performant" sans souvent être capable de définir précisément le sens de l'expression "le plus performant". Il est donc intéressant de tenter de définir plus précisément la notion de "gain" d'un processus de fusion et d'étudier quelles peuvent être les performances des opérateurs. Le chapitre suivant présente notre contribution relative à ce sujet.

Chapitre 7

L'évaluation des systèmes de fusion

A PRÈS une opération de fusion d'informations il est important de chercher à calculer le "gain" obtenu lors de la combinaison, c'est-à-dire de réaliser une quantification chiffrée de la prestation fournie par l'opérateur de fusion. Cette notation du travail de l'opérateur de fusion est paradoxalement un peu délaissée par la communauté¹ dont les travaux se concentrent plutôt sur la définition de nouvelles techniques de fusion. L'évaluation des performances des processus de fusion possède au moins deux applications intéressantes. Tout d'abord, mesurer le "gain" permet de comparer les performances des opérateurs pour une application donnée. Il devient alors plus simple de choisir l'opérateur de fusion adapté, par exemple celui qui introduit le "gain" le plus élevé. Par ailleurs, certains processus de fusion nécessitent le choix d'un indice de conflit et/ou de divers paramètres réels (un seuil, un pas de discrétisation . . .) et la mesure du "gain" peut faciliter le choix de ces paramètres.

Cependant, évaluer le "gain" d'un opérateur nécessite la définition préalable de la notion de gain, souvent fortement liée à l'application. De par leurs différences fondamentales, il est nécessaire de séparer la fusion symétrique de la fusion dissymétrique.

Un processus de fusion symétrique exploite la redondance entre les sources pour fournir un résultat plus fondé que celui obtenu en considérant chaque source séparément. La prise en compte de la redondance se réalise de trois manières différentes selon que la fusion s'effectue en mode conjonctif, disjonctif ou compromis et il sera donc nécessaire de définir la notion de gain pour chacun des trois modes. Un opérateur adaptatif ayant un comportement continu de la conjonction à la disjonction, la notion de gain pour un opérateur adaptatif pourra se déduire de la définition des trois modes de base.

Les algorithmes de fusion dissymétriques affinent une information existante à la lumière d'une nouvelle information supposée plus pertinente. Comme dans le chapitre 5, nous nous limitons ici à l'étude des techniques de révision. La révision peut s'effectuer suivant trois modes différents : la contraction, l'expansion ou la révision "simple" et la notion de gain sera donc indissociable du mode. Par ailleurs, tous ces modes doivent respecter le principe du changement minimal ce qui signifie qu'un processus de révision introduisant plus de changements qu'un autre devra être pénalisé.

Certains systèmes de fusion complexes peuvent se diviser en sous-systèmes de fusion. C'est le cas par exemple lorsque le résultat d'une fusion est présenté en entrée d'un autre algorithme de fusion (notre technique "hybride" détaillée au paragraphe 3.4 appartient à cette catégorie car il y a tout d'abord fusion "intra-classes" puis fusion "inter-classes"). Nous désignons par "système de fusion atomique" tout système de fusion disposant de $n \geq 1$ entrées (les sources à combiner) et d'une sortie (le résultat de la combinaison). Il peut s'agir d'un opérateur simple, d'un opérateur adaptatif, ou d'un assemblage d'opérateurs. Dans ce chapitre, nous supposons que tout système devant être évalué est atomique. Un système de fusion complexe pourra se décomposer en systèmes atomiques qui seront chacun évaluables. Cependant, l'utilisation des évaluations des systèmes atomiques pour obtenir l'évaluation globale du système complexe n'est pas abordée ici.

1. Seuls quelques auteurs (par exemple [Kewley, 1993], [Dubois et Kalfsbeek, 1990] et [Delmotte, 1998]) abordent une telle évaluation.

Nous proposons tout d'abord une précision de la notion de "gain" dans les processus de fusion symétriques et dissymétriques. Puis, dans un second paragraphe, nous rappelons quelques mesures sur les sous-ensembles flous et les distributions de possibilités comme le cardinal, l'indice de flou . . . qui sont alors utilisées pour évaluer les systèmes de fusion symétriques et dissymétriques. Les techniques d'évaluation proposées sont alors appliquées à des exemples simples. Munis des évaluations, nous proposons un algorithme destiné à faciliter le choix d'un opérateur de fusion pour quelques applications classiques.

7.1 Quantifier l'amélioration de la connaissance

La fusion symétrique et la fusion dissymétrique n'ont pas les mêmes objectifs et la notion de "gain" doit donc être différente. Nous nous intéressons tout d'abord à la fusion symétrique puis à la fusion dissymétrique.

7.1.1 Le gain en fusion symétrique

La notion de "gain" pour un système de fusion est difficile à définir car elle est souvent liée à l'application. Pour formaliser cette notion, observons le système de fusion atomique figure 7.1 disposant de n entrées I_j et d'une sortie S . Intuitivement, la notion de "gain" d'un système est associée au travail introduit par le système.

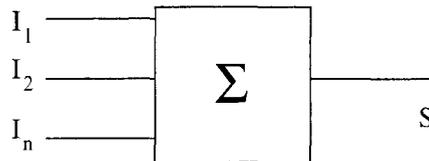


FIG. 7.1 – Un système de fusion atomique.

C'est le cas par exemple dans un amplificateur où le gain mesure le travail fourni par les transistors pour augmenter l'amplitude du signal d'entrée. Nous pouvons alors définir la notion de gain d'un système de fusion par analogie à celle du gain d'un amplificateur électronique². Cette approche nécessite alors la résolution de deux problèmes :

1. Déterminer une fonction J capable de mesurer une (ou plusieurs) caractéristique(s) sur les entrées (notée $J(I_1, \dots, I_n)$) et la sortie (notée $J(S)$). Cette détermination doit permettre la comparaison des résultats. La(es) caractéristique(s) déterminée(s) peut(vent) être représentée(s) par n'importe quel objet mathématique, mais le plus souvent – dans un but de simplification – les nombres réels sont choisis,
2. Déterminer une fonction de comparaison $Comp$ qui permet de déterminer l'augmentation du "travail" à partir de $J(I_1, \dots, I_n)$ et $J(S)$ comme la division ou la soustraction par exemple.

Les systèmes de fusion opèrent dans trois grands modes différents : la conjonction, la disjonction et le compromis. D'autres opérateurs n'ont pas un comportement constant : ils varient de manière continue de la conjonction à la disjonction au gré des informations à combiner (voir paragraphe 2.3). Il nous faudra donc résoudre le problème n° 1 ci-dessus pour chacun des modes de base et pour les opérateurs adaptatifs. La fonction de comparaison, elle, est commune à chacun des modes.

La conjonction est utilisée lorsque les sources sont en accord. Le but de l'opérateur de fusion est alors de supprimer le plus d'alternatives possibles, d'obtenir un résultat le moins flou possible.

La disjonction est utilisée lorsque les sources sont en désaccord. Le but de l'opérateur de fusion est alors de collecter toutes les alternatives possibles, sans pour autant tendre vers l'ignorance totale.

La compromis est utilisé pour les situations intermédiaires aux précédentes. Le but de l'opérateur de fusion est alors de refléter la décision moyenne ce qui consiste à s'assurer que toutes les alternatives sont présentes dans le résultat mais que la zone commune a plus d'importance que les autres.

2. Une autre approche consiste à voir le "gain" comme la possibilité de prendre une meilleure décision à partir du résultat S qu'à partir de chaque I_i . Ceci nécessite la formalisation de la notion de "meilleure décision". Cette autre vision du gain n'est pas abordée ici.

Les opérateurs adaptatifs ont un comportement qui varie continuellement de la conjonction à la disjonction. Leur but est donc d'être conjonctif lorsque les sources sont en accord et disjonctifs lorsque les sources sont en désaccord, en passant par toutes les nuances possibles. Un opérateur de ce type construit le résultat en combinant des termes conjonctifs, disjonctifs et de compromis. Cet opérateur introduit un "travail" d'autant plus fort qu'il a détecté le mode correct de combinaison et que ce mode introduit un travail important et prépondérant par rapport aux autres modes. L'évaluation de cet opérateur doit se réduire à celle des opérateurs conjonctifs lorsque les sources sont en accord total et disjonctifs lorsque les sources sont en désaccord total.

7.1.2 Le gain en fusion dissymétrique

Nous nous intéressons maintenant à l'évaluation des processus de fusion dissymétriques. Comme cela a été détaillé au chapitre 5 les règles dissymétriques sont des techniques de changement de croyance qui modifient une connaissance existante à la lumière d'une nouvelle information supposée plus pertinente. Ce type de fusion est adapté lorsque les informations sont acquises à différents instants par une même source.

Pour évaluer ces opérateurs, nous utilisons une approche comparable à celle utilisée pour l'évaluation symétrique. Nous évaluons un ou plusieurs critères sur l'entrée (c'est-à-dire l'information *a priori*) et sur le résultat. Nous effectuons alors une comparaison à l'aide d'une fonction *Comp* spécialement définie.

Le chapitre 5 a déjà présenté la fusion dissymétrique en détail. Nous nous intéressons ici uniquement aux règles de révision décrites dont le paragraphe 5.1.1.1 rappelle quelques propriétés. Les opérateurs de conditionnement opérant sur des informations de même type peuvent avoir trois comportements différents : la contraction, l'expansion et la révision "simple". Il s'agit – par définition – de modifier une information existante en minimisant le changements effectués. Cette propriété modulera la définition du gain. En effet, la seule considération des modifications introduites n'est pas suffisante : un processus de fusion ne modifiant pas l'information *a priori* aurait alors le meilleur gain, ce qui est intuitivement incorrect.

La contraction augmente le nombre d'alternatives contenues dans l'information *a priori*. Elle consiste à "oublier" dans l'information *a priori* l'information présente dans la nouvelle évidence. Le résultat contiendra alors plus d'alternatives que l'information *a priori*. Un opérateur introduira un travail d'autant plus important que la nouvelle information a été "oubliée" avec le moins de modifications possibles.

L'expansion diminue le nombre d'alternatives contenues dans l'information *a priori* à l'aide d'une nouvelle évidence. La priorité est donnée à la nouvelle information et l'opération ne transforme pas l'information *a priori* si l'entrée est déjà connue. L'évaluation devra considérer la réduction du nombre d'alternatives possibles lors du processus.

La révision "classique" est une règle de conditionnement sur des informations de mêmes types qui ne change pas ou peu le nombre d'alternatives contenues dans l'information *a priori*³. L'évaluation devra tenir compte de la variation du nombre d'alternatives lors du processus. Cette opération dépend du degré de concordance entre les sources qui interviendra donc dans l'évaluation.

Nous avons exprimé la notion de "gain" d'un opérateur de fusion en termes divers tels que "l'opérateur doit supprimer le plus d'alternatives possibles" mais, étant donnée une distribution de possibilité, nous ne sommes pas encore capables de chiffrer le nombres d'alternatives possibles et donc de construire la fonction *J*. Le paragraphe suivant s'intéresse aux mesures qui évaluent les caractéristiques des distributions.

7.1.3 Quelques techniques de mesure de la qualité de l'information

Dans le cadre possibiliste, il existe de nombreuses mesures comme le cardinal et la spécificité. De telles mesures évaluent une particularité de la distribution de possibilité comme par exemple l'indice de flou ou le cardinal. Ces mesures peuvent être utilisées pour évaluer le comportement d'un processus de fusion. Nous rappelons ci-dessous l'expression de quelques unes de ces mesures en nous limitant aux formes adaptées

3. Le résultat n'est ni plus ni moins spécifique que l'information *a priori*, la spécificité faisant l'objet de la définition 17 du paragraphe 7.1.3.

aux distributions ou sous-ensembles flous non discrets. Cependant, la plupart des mesures peuvent aussi être calculées sur des distributions ou sous-ensembles flous discrets (voir [Klir et Folger, 1988]).

Définition 16. Cardinal.

Le cardinal [De Luca et Termini, 1972] de π évalue le nombre d'éléments dans π :

$$|\pi| = \int_{\Omega} \pi(\omega) d\omega \tag{7.1}$$

Le cardinal peut être vu comme une mesure d'imprécision parce qu'il est minimal lorsque la distribution est réduite à un singleton et maximal lorsque π dénote l'ignorance totale.

Définition 17. Spécificité.

Soient deux distributions π_1 et π_2 , π_2 est dite plus spécifique que π_1 si $\forall \omega \in \Omega, \pi_1(\omega) \leq \pi_2(\omega)$. Une autre définition existe. Dans cette approche, seuls les noyaux des distributions sont considérés et dans cette approche π_2 est dite plus spécifique que π_1 si $\text{Noyau}(\pi_1) \subseteq \text{Noyau}(\pi_2)$. La figure 7.2 schématise cette approche. Dans

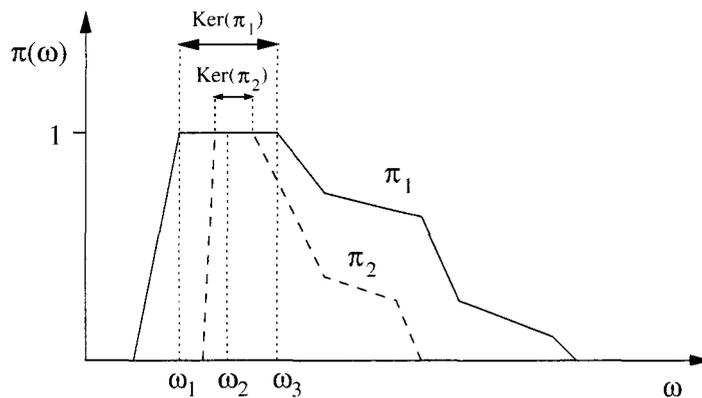


FIG. 7.2 – Interprétation de la spécificité (définition n° 2).

les deux approches, dire que π_2 est plus spécifique que π_1 signifie que π_2 traduit une connaissance plus riche, plus pertinente du point de vue de la précision. Une distribution plus spécifique contient moins d'alternatives comme cela est représenté sur la figure 7.2. La distribution π_1 contient les valeurs préférées $\omega_1, \omega_2, \omega_3$ tandis que la distribution plus spécifique π_2 ne contient que ω_2 .

Il est possible d'obtenir une quantification numérique de la spécificité en évaluant combien la distribution de possibilité est lointaine de la distribution la plus spécifique :

$$S_P(\pi) = \int_0^1 \frac{1}{|\pi_\alpha|} . d\alpha \quad \begin{cases} \pi_\alpha(\omega) = 1 \text{ si } \pi(\omega) \geq \alpha \\ = 0 \text{ sinon} \end{cases} \tag{7.2}$$

Nous constatons que $S_P(\pi_A) = 1$ si et seulement si il existe ω dans Ω tel que $A = \{\omega\}$. Pour deux ensembles A et B de distributions de possibilités respectives π_A et π_B , S_P vérifie $A \subseteq B \Rightarrow S_P(\pi_A) \geq S_P(\pi_B)$.

Définition 18. Non spécificité.

La non spécificité évalue le nombre d'alternatives dans l'information associée à une distribution de possibilité. C'est une quantification numérique directe de l'imprécision car plus une information contient d'alternatives, plus elle est imprécise :

$$H(\pi) = \int_0^1 \log_2 |\pi_\alpha| . d\alpha \tag{7.3}$$

H est minimum (et égal à 0) si et seulement si il existe ω de Ω tel que $A = \{\omega\}$. Pour deux ensembles A and B nous avons $A \subseteq B \Rightarrow H(\pi_A) \leq H(\pi_B)$. La non-spécificité peut se déduire de la spécificité.

Définition 19. Entropie de SHANNON.

L'entropie de SHANNON est très utilisée dans le domaine probabiliste. p étant une distribution de probabilité, l'entropie est donnée par :

$$H_S(p) = - \int_{\Omega} p(\omega) \log_2 p(\omega) d\omega \tag{7.4}$$

Certains auteurs ont proposé une extension de la notion d'entropie au contexte possibiliste comme par exemple [Dubois et Prade, 1988]. KLIR et YUAN montrent (un résumé est disponible dans [Cliff, 1997]) que la non spécificité déjà présentée précédemment peut remplacer l'entropie. Enfin, l'entropie peut aussi servir de mesure de spécificité.

Définition 20. Indicateur de flou.

Le but de cet indicateur est d'évaluer le caractère flou d'un ensemble. De nombreux auteurs ont proposé de tels indicateurs et parmi eux, [Kaufmann, 1977] utilise une distance d entre les distributions de possibilité π et π_{ref} déduite de π . La distribution de référence π_{ref} est définie comme l'ensemble ordinaire le plus proche de l'ensemble flou associé à π . Il peut être calculé de deux manières différentes : une manière optimiste et une manière pessimiste comme cela est présenté dans (7.5) et symbolisé figure 7.3. Le réel γ est utilisé comme facteur de normalisation car f doit être à valeurs dans $[0, 1]$:

$$f(\pi) = \gamma d(\pi, \pi_{ref}) \quad \text{avec} \quad \begin{cases} \pi_{opt}(\omega) &= 1 \text{ si } \omega \in \text{Support}(\pi) \\ &= 0 \text{ sinon} \\ \pi_{pess}(\omega) &= 1 \text{ si } \omega \in \text{Noyau}(\pi) \\ &= 0 \text{ sinon} \end{cases} \tag{7.5}$$

Le calcul optimiste est construit en utilisant le support de la distribution tandis que la cas pessimiste utilise le noyau de la distribution. Le cardinal de π_{opt} est toujours plus grand ou égal au cardinal de π_{pess} .

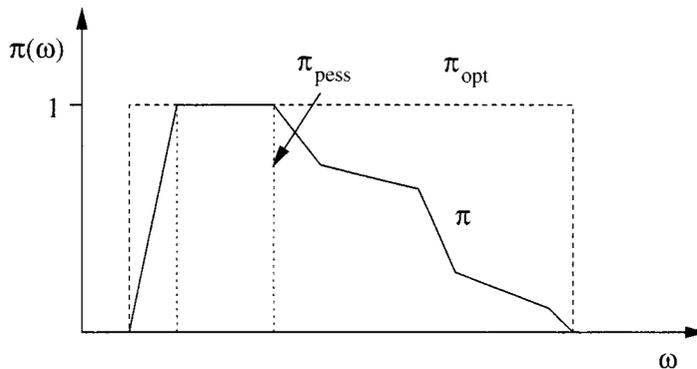


FIG. 7.3 – Calcul de π_{opt} et π_{pess} .

Si la distance de HAMMING est choisie pour d et si $\gamma = \frac{1}{\max_{\Omega} d(\pi, \pi_{ref})}$, f se réduit alors à l'indice de KAUFFMAN qui prend la valeur 0 dès que l'ensemble est un ensemble classique et croit jusqu'au maximum lorsque l'ensemble flou est tel que $\forall \omega \in \Omega, \mu(\omega) = 0.5$ (parfois encore dit "ensemble le plus flou").

Lorsque la distribution est spécifique et certaine ou peu spécifique et incertaine, l'indicateur donne 0. Si la distribution de possibilité n'est pas normalisée (même si ce scénario est exclu du fait de l'hypothèse du monde fermé), l'indice vaut aussi 0. L'indice de KAUFFMAN est incapable de distinguer une distribution normalisée d'une distribution non normalisée.

Définition 21. Distances.

Les distances servent souvent à mesurer la proximité entre deux éléments d'un même ensemble (l'utilisation

de distances dans ce but a déjà été abordé au paragraphe 2.1.4.1). Une distance est une fonction d définie par :

$$\begin{aligned} \Omega^2 &\longrightarrow \mathbb{R}^+ \\ \pi_1, \pi_2 &\longmapsto d(\pi_1, \pi_2) \end{aligned} \quad \text{vérifiant} \quad \begin{cases} \forall \pi \in \Omega, d(\pi, \pi) = 0 \\ \forall (\pi_1, \pi_2) \in \Omega^2, d(\pi_1, \pi_2) = d(\pi_2, \pi_1) \\ \forall (\pi_1, \pi_2, \pi_3) \in \Omega^3, d(\pi_1, \pi_3) \leq d(\pi_1, \pi_2) \otimes d(\pi_2, \pi_3) \end{cases} \quad (7.6)$$

où \otimes est une opération binaire sur \mathbb{R} . Parmi le grand nombre de distances existantes, citons pour mémoire la distance de HAMMING définie (lorsque l'intégrale converge) par :

$$d_{Ham}(\pi_1, \pi_2) = \frac{\int_{\Omega} |\pi_1(\omega) - \pi_2(\omega)| d\omega}{\delta} \quad \text{avec} \quad \begin{aligned} \delta &= \max_s \Gamma - \min_s \Gamma \\ \Gamma &= \{s / \max_{\omega \in \Omega} (\pi_1(\omega), \pi_2(\omega))\} \end{aligned} \quad (7.7)$$

La forme donnée ci-dessus est une version normalisée de la distance de HAMMING. Il est aussi possible de poser $\delta = 1$, d_{Ham} ne variant alors plus dans $[0, 1]$.

Nous sommes maintenant capables d'évaluer quelques caractéristiques sur les distributions ou sur les sous-ensembles flous. Les prochains paragraphes décrivent comment il est possible d'utiliser ces caractéristiques pour évaluer les systèmes de fusion symétriques puis dissymétriques.

7.2 Evaluation des systèmes de fusion symétrique

Le paragraphe 7.1.1 a présenté la notion de gain pour les systèmes de fusion symétriques. La fonction J associe un réel à n sources. Pour simplifier la détermination de cette fonction, nous supposons qu'elle associe un réel à *une* source dite "source de référence". Cette source notée π_A est tout d'abord choisie parmi les distributions d'entrée I_j et suivant certains critères. Cela permet à la fonction J d'être identique pour calculer la valeur des caractéristiques "avant" et "après". Nous présentons maintenant une définition des fonctions J et Com permettant d'évaluer le gain d'un système de fusion.

7.2.1 Evaluation d'opérateurs conjonctifs

Lorsque la fusion est faite en mode conjonctif, le but de l'opérateur est de supprimer le plus d'alternatives possibles. Cela signifie qu'il ne faut conserver que l'information commune à toutes les sources et diminuer l'incertitude des données comme indiqué dans [Kewley, 1993]. Cela conduit donc à augmenter la possibilité dans la zone d'intersection des distributions.

Supposons un instant que le processus de fusion n'existe pas. Une seule source π_A de l'ensemble des entrées serait alors considérée par le module de prise de décision. Cependant, le module de fusion existe et il fournit un résultat π_R . Un choix raisonnable consiste à poser $\pi_A = I_j$ où I_j désigne la source la plus proche au sens d'une métrique d du résultat π_R . Nous calculerons le gain sur π_A et sur π_R , nous plaçant ainsi dans le cas le plus défavorable pour l'opérateur de fusion. En effet choisir l'entrée la plus proche du résultat de l'opérateur de fusion conduit intuitivement à placer l'opérateur de fusion dans une situation où il a le moins de "travail" à effectuer.

Le processus de fusion a atteint son but lorsque la distribution résultante est plus spécifique que la distribution de référence π_A et que le cardinal a diminué. Ces mesures sont alors calculées sur π_A et π_R puis comparées à l'aide d'une combinaison linéaire :

$$E_C = \frac{|\pi_A| - |\pi_R|}{\max(|\pi_A|, |\pi_R|)} + \frac{S_p(\pi_R) - S_p(\pi_A)}{\max(S_p(\pi_R), S_p(\pi_A))} \quad (7.8)$$

Plus E_C est petit, moins le processus de fusion a introduit de valeur ajoutée. Lorsque $E = 0$, aucune valeur ajoutée n'a été introduite. L'opérateur est dans ce cas considéré comme inutile.

Les normalisations dans (7.8) ne sont présentes que pour faciliter la lecture du résultat. En effet, E_C varie dans $[0, 2]$. Cependant, comme avec toute normalisation par division, des discontinuités apparaissent lorsque le dénominateur est proche de 0. Dans ce cas, il est tout à fait possible de supprimer purement et simplement les normalisations de (7.8).

7.2.2 Evaluation d'opérateurs disjonctifs

Les sources sont maintenant en total désaccord. Cela nécessite l'utilisation d'un processus de fusion prudent, donc disjonctif. L'opérateur va donc conserver toutes les alternatives et rendre le résultat le plus spécifique possible. Si la distribution π_A est définie comme précédemment, l'évaluation est alors exprimée par :

$$E_D = \frac{|\pi_R| - |\pi_A|}{\max(|\pi_R|, |\pi_A|)} + \frac{H(\pi_R) - H(\pi_A)}{\max(\|H(\pi_R)\|, \|H(\pi_A)\|)} \quad (7.9)$$

Comme pour E_C , plus E_D est faible, moins le système de fusion a introduit de travail. Dans cette évaluation, π_A est souvent très différente du résultat π_R ce qui conduit donc à une évaluation un peu faussée. Une autre manière de procéder consiste à utiliser une distribution de référence qui n'existe pas dans les entrées, mais qui est construite à partir de ces dernières. L'un des choix les plus intuitifs du point de vue du module de prise de décision serait de considérer l'ensemble classique le plus proche de l'union (par la T-Conorme max) des entrées, ensemble que nous désignons par π_{NS} . Par rapport à la distribution de référence π_{NS} , le cardinal et la non-spécificité doivent avoir diminué. L'évaluation est alors donnée par :

$$E_D = \frac{|\pi_{NS}| - |\pi_R|}{\max(|\pi_{NS}|, |\pi_R|)} + \frac{H(\pi_{NS}) - H(\pi_R)}{\max(\|H(\pi_{NS})\|, \|H(\pi_R)\|)} \quad (7.10)$$

Comme pour E_C , plus E_D est faible, moins le système de fusion a introduit de "travail". Il faut être certain qu'un processus de fusion qui fournirait systématiquement une distribution de possibilité indiquant que tout l'univers est possible serait marqué comme n'introduisant que peu de travail. C'est pour cela que la condition de spécificité est présente. Là encore, les normalisations peuvent être supprimées.

7.2.3 Evaluation d'opérateurs de compromis

L'opérateur de compromis conserve toutes les alternatives comme un opérateur de disjonction mais renforce la croyance dans la zone où les distributions s'accordent. Le "travail" introduit peut alors s'évaluer en utilisant une distance entre le résultat et π_A et la variation de la non-spécificité :

$$E_M = d_{Ham}(\pi_A, \pi_R) + \frac{H(\pi_A) - H(\pi_R)}{\max(\|H(\pi_A)\|, \|H(\pi_R)\|)} \quad (7.11)$$

Comme pour E_C , plus E_M est faible, moins le système de fusion a introduit de valeur ajoutée.

Là encore, la distribution π_A choisie comme référence n'a pas grand chose à voir avec le résultat. Une autre solution consiste à considérer comme référence la distribution π_M moyenne arithmétique de toutes les distributions d'entrée. Le critère est donc :

$$E_M = d_{Ham}(\pi_M, \pi_R) + \frac{H(\pi_M) - H(\pi_R)}{\max(\|H(\pi_M)\|, \|H(\pi_R)\|)} \quad (7.12)$$

Comme précédemment, les normalisations (celle présente dans la distance de HAMMING n'apparaissant pas ici) peuvent être supprimées.

Il est intéressant de noter que la création d'une distribution fictive π_A de référence utilise un opérateur de fusion. L'évaluation sera donc dépendante de l'opérateur choisi.

7.2.4 Evaluation d'opérateurs adaptatifs

Nous pouvons maintenant évaluer les opérateurs dans chacun des modes de base. Un opérateur adaptatif évolue continuellement de la conjonction à la disjonction. Il est intéressant de calculer le gain introduit par l'opérateur en supposant qu'il agit en mode conjonctif, disjonctif ou compromis.

Supposons que la fusion soit faite non pas avec un opérateur adaptatif mais avec un opérateur conjonctif min. Le résultat aurait été $\pi_{R_{\min}}$. Si la fusion avait été faite avec un opérateur disjonctif comme max, le

résultat aurait été $\pi_{R_{\max}}$ et enfin, si la fusion avait été faite avec un opérateur de moyenne comme la moyenne arithmétique, le résultat aurait été $\pi_{R_{\text{mean}}}$. Nous mesurons alors la distance entre chacune des distributions $\pi_{R_{\min}}$, $\pi_{R_{\max}}$ et $\pi_{R_{\text{mean}}}$ et le résultat π_R fourni par l'opérateur adaptatif ce qui donne l'intensité avec laquelle l'opérateur adaptatif s'est comporté comme un opérateur de compromis, de conjonction ou de disjonction.

Le gain du résultat est alors évalué par notre méthode d'évaluation conjonctive (7.8) disjonctive (7.9) et de compromis (7.11). Chacune de ces évaluations est pondérée par l'intensité avec laquelle l'opérateur s'est comporté dans le mode correspondant :

$$\begin{cases} I_{Conj} = d_{Ham}(\pi_R, \pi_{R_{\min}}) \\ I_{Disj} = d_{Ham}(\pi_R, \pi_{R_{\max}}) \\ I_{Mean} = d_{Ham}(\pi_R, \pi_{R_{\text{mean}}}) \end{cases} \quad E_A = I_{Conj}E_C + I_{Disj}E_D + I_{Mean}E_M \quad (7.13)$$

Nous évaluons maintenant un système de fusion simple déjà étudié précédemment.

7.2.5 Exemple

Nous appliquons la technique d'évaluation précédente à un exemple de fusion très simple, similaire à celui abordé au paragraphe 4.2.4 page 74. Une cuve contenant un liquide est équipée d'un dispositif de brassage, d'une arrivée en haut de cuve et d'une vidange en fond de cuve (voir figure 7.4). Ce système est équipé de

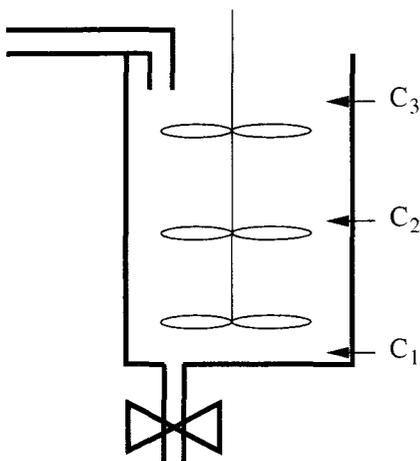


FIG. 7.4 – La cuve et ses capteurs.

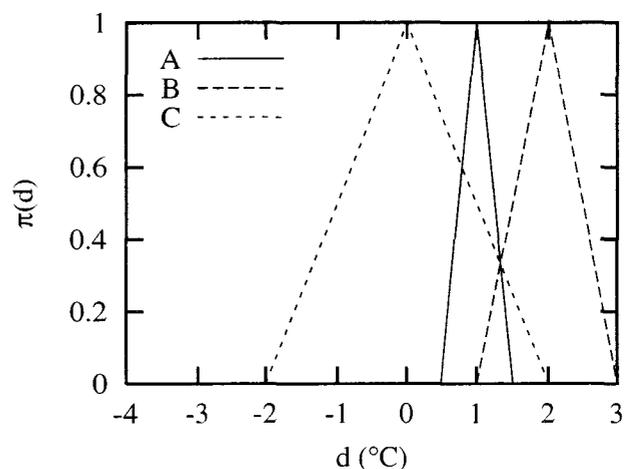


FIG. 7.5 – Distributions de test.

trois capteurs de température C_1 , C_2 et C_3 qui mesurent respectivement la température du liquide au fond, au milieu et au sommet de la cuve. Supposons que ces capteurs fournissent un nombre flou défini par sa fonction d'appartenance. A intervalles de temps réguliers, la valeur des trois capteurs est lue au même instant. Les trois nombres flous obtenus doivent être combinés pour obtenir la température de la cuve. Le mode de combinaison le plus adapté semble être le compromis, mais dans un but d'illustration, nous allons fusionner les nombres flous dans les trois modes classiques : la conjonction, le compromis et la disjonction. Nous évaluerons alors chacun des trois systèmes de fusion.

Dans cette simulation, les distributions (voir figure 7.5) sont supposées triangulaires sans perte de généralité. Chaque distribution est définie par son centre et par l'écartement de part et d'autre. La première, A , est centrée sur 1 avec un écartement de 0.5 et donc le support de A est $[0.5, 1.5]$. La seconde, B est centrée sur 2 avec un écartement de 1 et la troisième C est centrée en 0 avec un écartement de 2. La fusion a été réalisée dans les trois modes classiques : la conjonction (avec min), la disjonction (avec max) et le compromis (avec la moyenne arithmétique). Les résultats de la fusion dans les trois modes sont présentés sur la figure 7.9 page ci-contre et les résultats de l'évaluation sont présentés sur la figure 7.13.

Les courbes d'évaluation ont été obtenues en utilisant l'algorithme suivant : fixer une valeur δ de l'écartement de la distribution A dans l'intervalle $[0.5; 2.5]$. Il est alors possible de calculer l'évaluation du processus

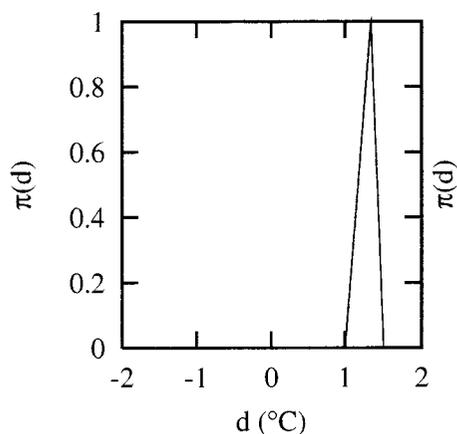


FIG. 7.6: Conjonction.

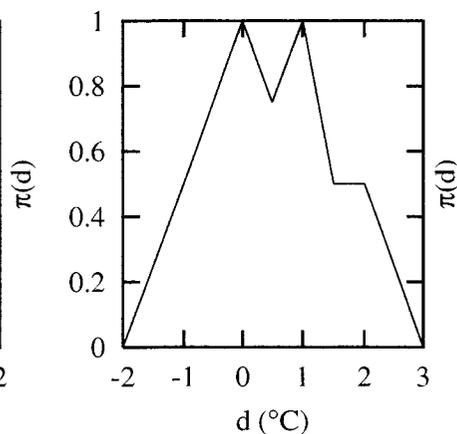


FIG. 7.7: Compromis.

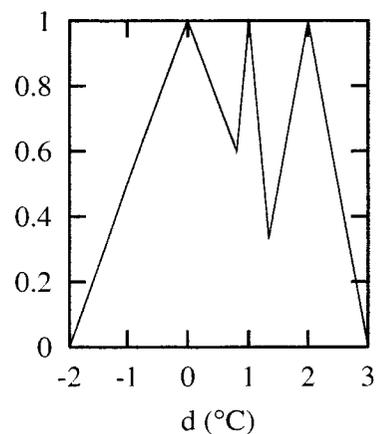


FIG. 7.8: Disjonction.

FIG. 7.9: Fusion des nombres flous.

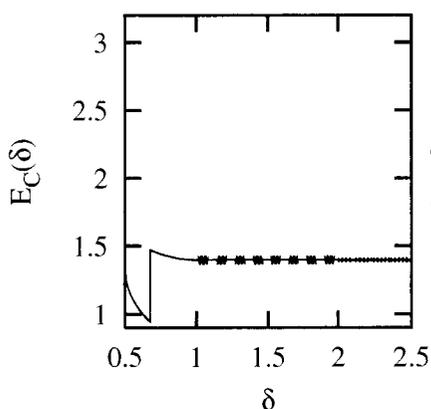


FIG. 7.10: Evaluation (min).

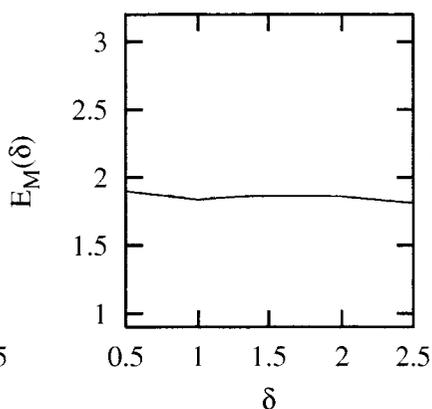


FIG. 7.11: Evaluation (moy).

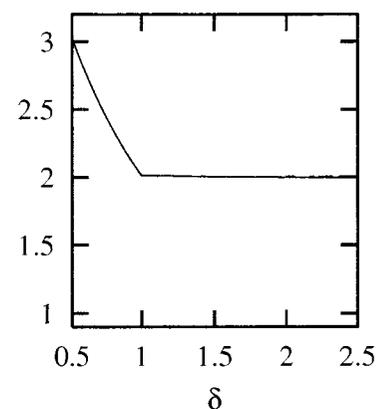


FIG. 7.12: Evaluation (max).

FIG. 7.13: Evaluation des trois modes.

de fusion sur ces trois distributions (noté e). Le point de coordonnées (δ, e) est alors placé sur une courbe et un nouveau δ est choisi. L'axe horizontal représente donc l'écartement de la distribution A et l'axe vertical la valeur du critère d'évaluation.

Nous constatons que lorsque la distribution A grandit (elle devient alors de plus en plus cohérente avec les autres sources) le travail introduit par les opérateurs max et min se stabilise. En ce qui concerne la moyenne, la valeur ajoutée décroît logiquement à partir de 2 et se stabilise aussi lorsque δ croît (ici à la valeur 2 qui n'a pu être représentée pour des raisons d'échelle).

7.3 Evaluation des systèmes de fusion dissymétrique

Les paragraphes suivant abordent l'évaluation des systèmes de fusion dissymétriques de conditionnement, pour deux informations de même type ([Zunino et al., 1999]).

7.3.1 La contraction

Cette opération diminue la spécificité de l'information *a priori*. Elle consiste à "oublier" dans l'information *a priori* l'information présente dans la nouvelle évidence. Le résultat sera donc moins spécifique que l'information *a priori*. L'évaluation tiendra compte de la diminution de la spécificité en utilisant (7.2). Le principe du changement minimal régissant cette opération, il est intéressant de compléter l'évaluation par une estimation des changements réalisés pendant l'expansion. L'augmentation de la spécificité est mesurée à l'aide de

la variation de la spécificité, l'estimation des changements réalisés est effectuée en utilisant la distance de HAMMING entre le résultat et la distribution *a priori* soustrait de la variation des cardinaux.

$$E_- = [S_p(\pi_A) - S_p(\pi_B^-)] + d_{Ham}(\pi_B^-, \pi_A) - [|\pi_B^-| - |\pi_A|] \quad (7.14)$$

Lorsque $\pi_A = \pi_B$, l'évaluation E_- n'est pas 0 : l'opérateur a introduit un "travail" car il y a eu oubli d'une information. L'évaluation croît alors proportionnellement au "travail" introduit. L'évaluation est égale à 0 lorsque $\pi_A = \pi_B^-$ c'est-à-dire lorsque l'information à oublier n'était pas présente dans A .

7.3.2 L'expansion

Cette opération augmente la spécificité de l'information *a priori* à l'aide d'une nouvelle évidence. La priorité est donnée à la nouvelle information et l'opération ne transforme pas l'information *a priori* si l'entrée est déjà connue. L'évaluation tiendra compte de l'augmentation de la spécificité lors du processus en utilisant (7.2). Le principe du changement minimal régissant cette opération, il est intéressant de compléter l'évaluation par une estimation des changements réalisés pendant l'expansion. L'augmentation de la spécificité est mesurée à l'aide de la variation de la spécificité, l'estimation des changements réalisés est effectuée en utilisant la distance de HAMMING entre le résultat et la distribution *a priori* sommée à la variation des cardinaux :

$$E_+ = [S_p(\pi_B^+) - S_p(\pi_A)] + d_{Ham}(\pi_B^+, \pi_A) + [|\pi_A| - |\pi_B^+|] \quad (7.15)$$

Lorsque $\pi_A = \pi_B^+$, l'évaluation E_+ est 0 : l'opérateur n'a introduit aucun "travail" car la distribution n'a pas changé. L'évaluation croît alors proportionnellement au "travail" introduit. Lorsqu'une expansion conduit à un état absurde (aucun élément de l'univers n'est possible), la spécificité n'est pas définie. La limite de (7.15) est alors infinie. Si l'expansion résulte en l'ignorance totale, la spécificité n'est pas définie et le cardinal est infini. L'évaluation est alors infinie.

7.3.3 La révision "classique"

Il s'agit d'une révision pour laquelle le résultat n'est ni plus spécifique ni moins spécifique que le résultat qui serait fourni par une expansion de l'information *a priori*. La priorité est donnée à la nouvelle information. L'évaluation tiendra compte de la variation de la spécificité lors du processus en utilisant (7.2). L'opération dépend du degré de concordance entre les sources dont une approximation est donnée par l'étendue de l'intersection des distributions π_A et π_B^+ . Le principe du changement minimal régissant cette opération, il est intéressant de compléter l'évaluation par une estimation des changements réalisés pendant l'expansion en utilisant une distance.

$$E_* = [S_p(\pi_B^*) - S_p(\pi_A)] + d_{Ham}(\pi_B^*, \pi_A) + [|\pi_B^* \cap (\pi_B \cap \pi_A)| - |\pi_A|] \quad (7.16)$$

Lorsque $\pi_A = \pi_B^*$, l'évaluation E_* est 0 : l'opérateur n'a introduit aucun "travail" car la distribution n'a pas changé. L'évaluation croît alors proportionnellement au "travail" introduit.

7.3.4 Exemple

Nous appliquons cette évaluation à l'exemple de fusion dissymétrique présenté au paragraphe 5.3. Nous nous intéressons à l'évolution du travail introduit tout au long du processus de fusion lors de la rencontre d'un obstacle. La figure 7.14 montre les résultats de l'évaluation. L'axe horizontal représente les étapes et l'axe vertical la valeur de l'évaluation pour chaque étape. Pour les étapes où la fusion était une union simple, l'évaluation n'a pas été calculée. L'expansion de l'ignorance totale (étape 1) introduit une grande valeur ajoutée. Entre les étapes 5 et 6 d'une part et 9 et 10 d'autre part, les distributions ne sont pas modifiées. L'opérateur n'introduit donc aucun travail. Sur la figure 7.14, les distributions des étapes 2, 4 et 7 ont été obtenues par simple union entre l'information *a priori* et la nouvelle évidence. Il ne s'agit donc pas d'un processus de fusion dissymétrique et les distributions correspondant à ces étapes ne peuvent pas être évaluées de la même manière que les autres étapes.

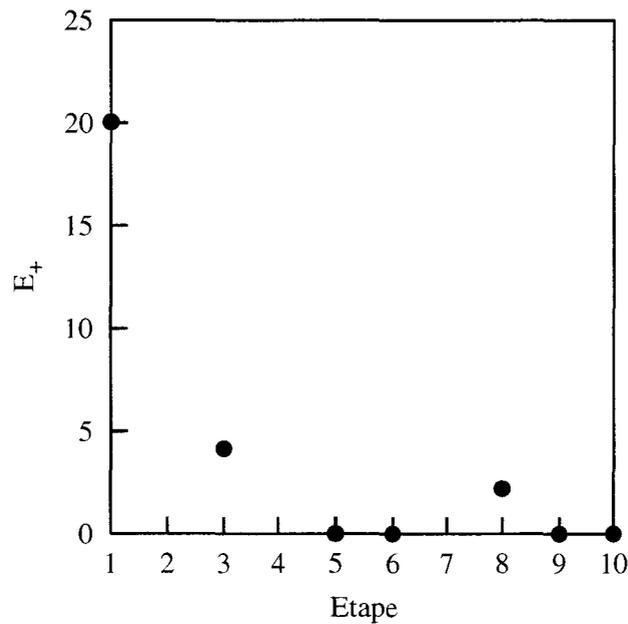


FIG. 7.14: L'évaluation.

L'évaluation des systèmes de fusion permet de connaître leurs performances mais peut aussi faciliter le choix de l'opérateur de fusion pour une application donnée. Nous présentons maintenant un algorithme destiné à faciliter le choix de l'opérateur de fusion.

7.4 Quel opérateur choisir ?

Face à une application donnée, il n'est pas toujours facile de choisir l'opérateur de fusion le plus adapté. Après avoir étudié de nombreux opérateurs de fusion dans le formalisme de la théorie des possibilités, nous tentons ici de proposer un guide succinct de choix de l'opérateur. Pour cela, nous procédons en trois étapes :

1. Choisir quelques types d'application classiques : l'estimation de paramètres, la surveillance de capteurs, l'affinement d'une information dans le temps ... Ces applications sont choisies car elles sont récurrentes dans le domaine de l'automatique,
2. Simuler logiciellemment l'application avec plusieurs opérateurs de fusion et évaluer ses performances,
3. Analyser les résultats et fournir un guide aidant au choix de l'opérateur.

Etant donné le grand nombre d'essais effectués et le volume des résultats collectés, il est impossible de présenter tous les détails de la deuxième étape ici. Seules les types d'applications classiques et les conclusions (phases 1 et 3) seront présentées.

Dans un premier paragraphe, nous décrivons les situations-type retenues. La nature de ces situations-type conduit à diviser le guide de prescription en deux sous-parties : les applications de fusion symétrique telles que l'estimation de paramètres et les applications dissymétriques telles que l'affinement d'une information dans le temps. Le guide d'emploi des opérateurs de fusion est alors présenté sous forme de tableau.

7.4.1 Les applications-type retenues

L'analyse des résultats de la recherche en automatique appliquée permet de constater que deux grands domaines⁴ font l'objet d'études poussées :

- ▷ L'acquisition des informations par les capteurs. Ce domaine de recherche groupe tout ce qui permet de fournir au module de décision une information. Cela inclus aussi la surveillance des capteurs car fournir

4. Il existe bien entendu de nombreux autres domaines de recherche en automatique appliquée mais nous ne retenons ici que les plus récurrents et les plus classiques.

une information pertinente est difficilement dissociable de la détection et la prévention des pannes de capteurs,

- ▷ Le calcul de la commande du système pour réguler celui-ci. Il s'agit de déterminer le modèle du système et de calculer la commande à appliquer.

Nous avons présenté dans cet ouvrage quelques exemples de la première catégorie : la détermination de la température dans la cuve dans ce chapitre ou l'estimation de la largeur d'un obstacle entravant la route. Le second domaine n'a pas été abordé. Du point de vue de la fusion, calculer la commande d'un système peut-être vu comme calculer la commande fournie par plusieurs sous-modèles locaux suivis d'une fusion des commandes. Cet aspect du contrôle/commande auquel [Delmotte, 1998] est entièrement consacré ne sera pas abordé ici et nous nous concentrerons sur le premier domaine.

L'acquisition d'information à l'aide de capteurs soulève plusieurs problèmes. La fusion de données propose des solutions pour certains d'entre eux :

1. Utiliser un groupe de capteurs pour fournir une information plus sûre, dans le sens où l'on compense la possible défaillance d'un capteur par un autre,
2. Surveiller (de manière naïve) un groupe de capteurs,
3. Reconstituer un environnement à l'aide d'informations complémentaires comme cela est fait en cartographie en utilisant les vues aériennes, les vues infrarouges . . . ,
4. Déterminer précisément la position d'un point dans l'espace à l'aide de capteurs placés en des endroits différents,
5. Affiner un paramètre dont la perception est de plus en plus précise au fur et à mesure que le temps passe.

La dernière application est de type "fusion dissymétrique". Nous proposons maintenant une aide au choix de l'opérateur pour chacune des situations précédentes.

7.4.2 Les applications exploitant la redondance

Pour choisir l'opérateur dans ce type d'applications, il est tout d'abord nécessaire de différencier l'univers dans lequel les informations sont modélisées. S'il s'agit d'un univers discret, les opérateurs logiques utilisant les T-Normes de type min ou max sont recommandés car plus adaptés que d'autres T-Normes comme le produit plus adaptées quant à elles à des référentiel non discrets.

La détermination d'un paramètre est l'application type de la fusion de données. Lorsque le mode de combinaison est connu, les opérateurs CIVB peuvent être choisis pour la rapidité avec laquelle ils peuvent être calculés. Il est cependant souvent préférable d'effectuer une combinaison à l'aide d'un opérateur OWA (voir paragraphe 4.1) plutôt qu'une combinaison en mode compromis. Lorsque le mode de combinaison n'est pas connu et que deux sources sont mises en jeu, les opérateurs adaptatifs décrits au paragraphe 2.3.1 fournissent d'excellents résultats. Lorsque plus de deux sources sont à combiner, il est possible d'utiliser les opérateurs à classement de sources de DUBOIS et PRADE décrits au paragraphe 3.2 ou de considérer les opérateurs "hybrides" de fusion qui permettent d'écarter les sources éventuellement erronées. Enfin, dans l'absolu, les opérateurs utilisant le principe d'extension fournissent de bons résultats, mais nécessitent un temps de calcul plus qu'incompatible avec la plupart des applications.

La surveillance de capteurs peut être réalisée de manière naïve à l'aide d'un opérateur de fusion. Dans ce type d'application, nous cherchons à déterminer l'éventuelle panne d'un capteur. Lorsque la panne est détectée, nous cherchons quel capteur est en panne et enfin, nous tentons d'utiliser les autres capteurs pour fournir une mesure plausible au module de décision. Les méthodes de fusion de plus de 2 sources "hybrides" qui effectuent un classement des sources sont adaptées à ce type d'application. Une fois le classement fait, il est facile de regarder quels sont les groupes en fort désaccord avec les autres et donc d'identifier le capteur. Par la suite, la fusion peut se poursuivre en supprimant les sources incorrectes. Cependant, ce processus ne fonctionne que si les sources sont totalement redondantes et en supposant qu'une au moins soit correcte.

La reconstruction considère souvent peu de sources d'informations, mais elles sont de natures différentes. C'est le cas en cartographie où une vue aérienne fait apparaître les routes et d'autres types de prise de vue

révèlent d'autres détails comme le type des cultures, les forêts . . . Il est alors intéressant de reconstruire la scène en utilisant ces informations de natures différentes après les avoir représentées dans un univers commun. Dans ce cas, les opérateurs adaptatifs et particulièrement les opérateurs adaptatifs géométriques comme décrite au paragraphe 2.3.1 fournissent d'excellents résultats.

Le positionnement d'un objet dans l'espace est une opération qui s'effectue très bien à l'aide d'intégrales floues, mais aussi avec des opérateurs de fusion fondés sur le principe d'extension (voir paragraphe 4.2). En effet, les mesures effectuées sur le terrain sont souvent nombreuses (c'est-à-dire qu'il y a plus de deux sources à combiner) et la position de l'objet est souvent déduite des mesures à l'aide d'une relation mathématique (la géométrie dans les triangles est souvent à la base de l'élaboration de ces relations mathématiques). La transposition du corps des réels vers les sous-ensembles flous s'effectue alors très facilement à l'aide des opérateurs basés sur le principe d'extension.

Le tableau 7.1 résume les opérateurs pour ces cinq applications.

Situation	Opérateurs adaptés
Affiner	<ul style="list-style-type: none"> ▷ Mode de combinaison connu <ul style="list-style-type: none"> • Opérateurs CICB (voir paragraphe 1.4.2.1) • Opérateurs OWA (voir paragraphe 4.1) ▷ Mode de combinaison inconnu <ul style="list-style-type: none"> • Si deux sources <ul style="list-style-type: none"> • Règle adaptative DUBOIS et PRADE et dérivées (paragraphe 2.3) • Si plus deux sources <ul style="list-style-type: none"> → Méthodes à classement de sources (voir paragraphe 3.2) → Méthodes "hybrides" (voir paragraphe 3.4) → Méthodes fondées sur le principe d'extension
Surveiller	▷ Méthodes à classement de sources "hybrides" (voir paragraphe 3.4)
Reconstruire	▷ Opérateurs adaptatifs (si 2 sources)
Positionner	▷ Opérateurs à classement de sources (voir paragraphe 3.2)
	▷ Intégrales floues ou opérateurs fondés sur le principe d'extension (paragraphe 4.2)

TAB. 7.1 – Les opérateurs recommandés dans les situations de fusion symétrique classiques.

7.4.3 Les applications affinant une information existante

L'application n° 6 est une opération de conditionnement agissant sur des informations factuelles. Les techniques de révision sont – de fait – adaptées. Il reste alors à choisir parmi expansion, contraction et révision. Le choix est fait en considérant l'univers dans lequel les informations sont représentées. S'il s'agit d'un univers continu, les règles d'expansion et de contraction sont difficiles à interpréter et doivent être évitées. Il sera préférable d'utiliser la règle de révision de DUBOIS et PRADE (5.24) construite avec une T-Norme de type produit (le min étant là encore adapté aux univers discrets et introduisant des discontinuités s'il est utilisé dans des univers).

Lorsque l'univers est discret, le choix est réalisé en fonction du besoin. Si la dernière information est compatible avec l'information *a priori*, il est alors possible de réaliser une expansion ou une révision (baie sur une T-Norme de type min). Si la nouvelle information doit être oubliée, dans l'information *a priori*, il est alors nécessaire d'effectuer une contraction. Le tableau 7.2 résume le choix des opérateurs pour cette application.

7.5 Quelques exemples de fusion adaptative plus complexes

Nous présentons ici quelques exemples plus complexes de fusion adaptative avec la règle de DUBOIS et PRADE (voir 2.30). Dans tout ce document (tout comme dans la grande majorité de la littérature) la fusion est effectuée sur des distributions unimodales. Nous choisissons ici quelques distributions multimodales.

Situation	Opérateurs adaptés
Affiner	▷ Univers discret <ul style="list-style-type: none"> • Nouvelle information et information <i>a priori</i> compatibles <ul style="list-style-type: none"> → Expansion (5.5) → Révision (5.24) bâtie sur min • Nouvelle information et information <i>a priori</i> incompatibles <ul style="list-style-type: none"> → Contraction (5.7)
	▷ Univers continu <ul style="list-style-type: none"> • Révision (5.24) avec une T-Norme de type produit

TAB. 7.2 – Les opérateurs recommandés dans les situations de fusion dissymétrique classiques.

Tout d'abord, la figure 7.15 illustre la fusion adaptative de deux distributions multimodales dont l'un des modes est en accord parfait et l'autre, en conflit. Le résultat est compatible avec l'intuition : le mode en accord entre les deux distributions n'a pas changé alors que le mode en conflit possède maintenant un degré de possibilité plus faible.

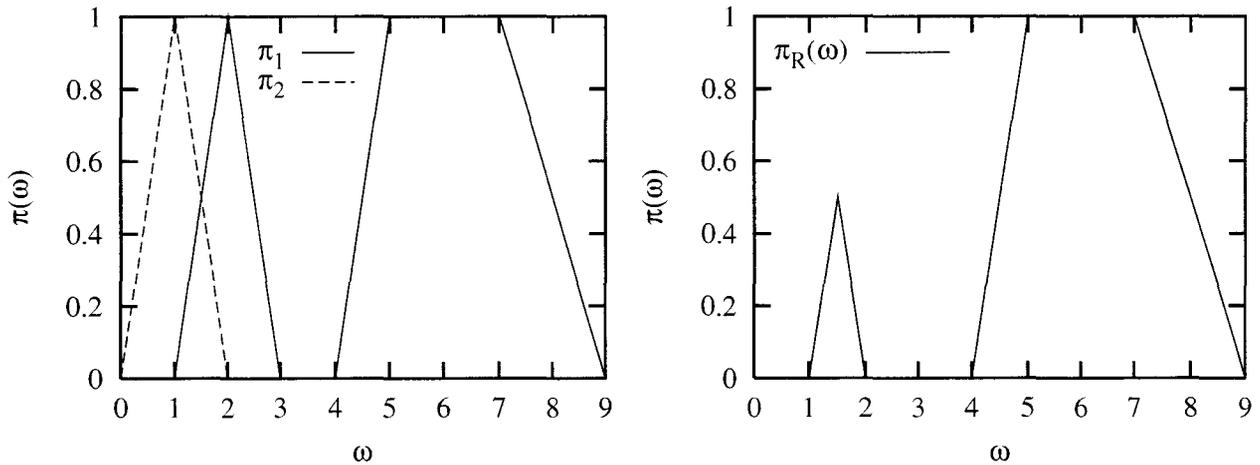


FIG. 7.15 – Distributions de test et résultat de la fusion (ex 1).

Nous nous intéressons maintenant à la fusion deux distributions multimodales dont un mode est en fort conflit (au voisinage de l'abscisse 1) et le second en conflit moyen (au voisinage de l'abscisse 5.5). Les distributions et le résultat de la fusion sont visibles sur la figure 7.16. Nous vérifions que l'opérateur associe bien le degré de possibilité le plus grand au point où l'intersection est la plus haute. Intuitivement, l'intersection entre les deux autres modes (au voisinage de l'abscisse 1) devrait elle aussi être privilégiée ce qui n'est pas le cas. L'opérateur adaptatif de DUBOIS et PRADE semble dans ce cas fournir un résultat contraire à l'intuition.

Considérons maintenant la fusion adaptative entre une distribution monomodale simple (par exemple un triangle) et une distribution multimodale. Dans cet exemple (voir figure 7.17) les deux distributions s'accordent relativement bien. Le résultat semble compatible avec l'intuition : les deux zones d'intersection ont été privilégiées dans le résultat.

Dans le dernier exemple, nous nous intéressons à la fusion entre une distribution monomodale (un triangle) et une distribution multimodale. La figure 7.18 présente les distributions de test et le résultat. Dans le résultat, au voisinage de l'abscisse 4.5, il est possible de se demander pourquoi le degré de possibilité diminue juste avant d'augmenter, surtout que ce phénomène se produit alors que les deux distributions ont une zone en commun. Là encore, le résultat de l'opérateur est discutable.

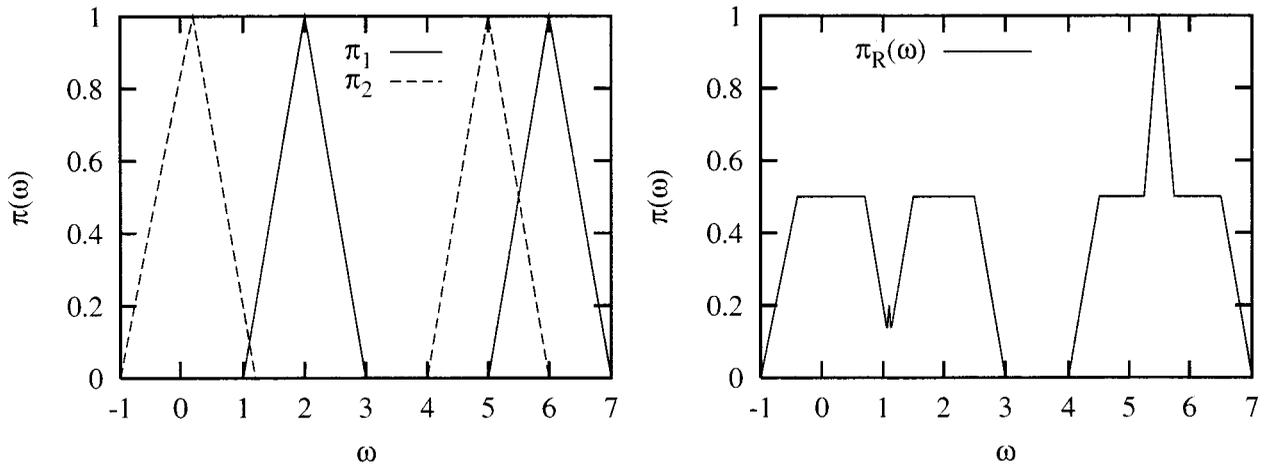


FIG. 7.16 – Distributions de test et résultat de la fusion (ex 2).

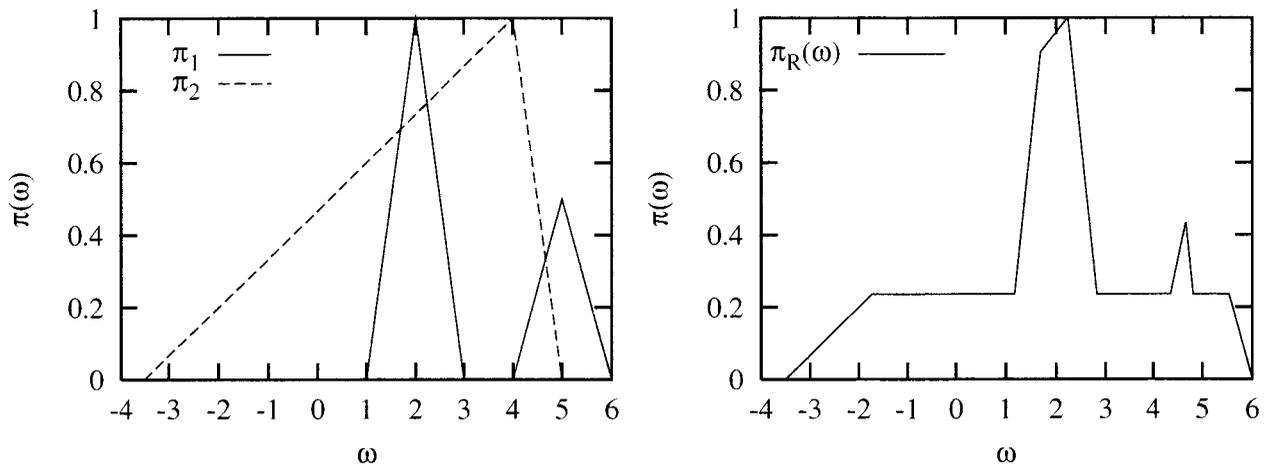


FIG. 7.17 – Distributions de test et résultat de la fusion (ex 3).

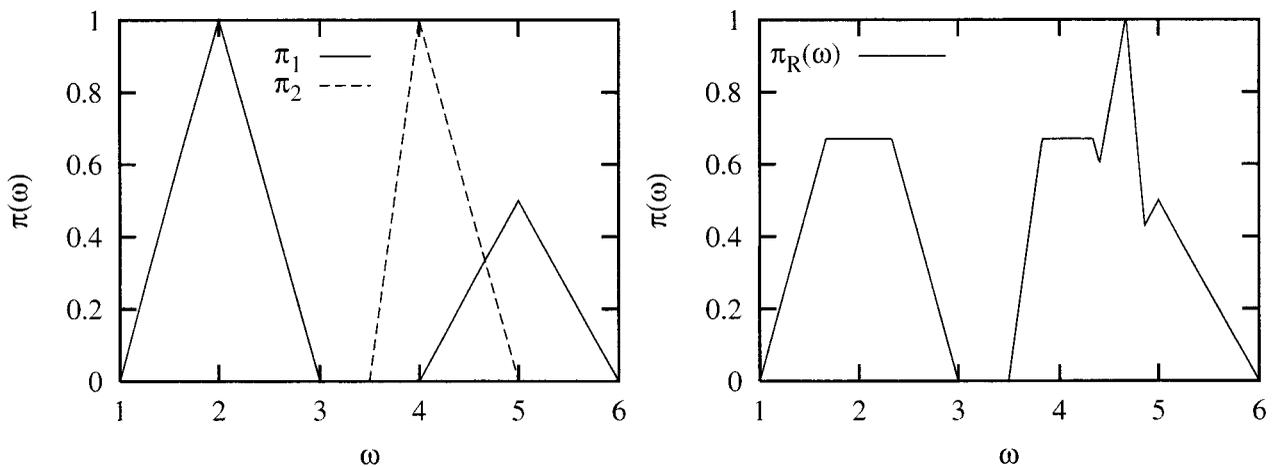


FIG. 7.18 – Distributions de test et résultat de la fusion (ex 4).

7.6 Conclusion

Nous avons présenté une définition possible de la notion de “valeur ajoutée” dans le cadre de la fusion symétrique et dissymétrique d’informations. Cette valeur ajoutée mesure le quantité de “travail” réalisé par

l'opérateur pour fournir le résultat étant donné les sources en entrée. Une mesure de la "valeur ajoutée" peut être utile pour vérifier que le processus de fusion est bien nécessaire (c'est-à-dire introduit un niveau de modifications minimum). S'il s'avère que le module de fusion n'a pas l'efficacité escomptée, il est alors envisageable de le remplacer par une autre technique plus efficace dans le contexte considéré. La mesure de la valeur ajoutée permet aussi de faciliter le choix de l'opérateur. Parmi tous les opérateurs possédant les propriétés requises, il est possible de choisir celui qui introduit le plus de travail. La mesure de la valeur ajoutée peut aussi permettre de régler les paramètres internes d'un opérateur de fusion : le pas de discrétisation, les éventuels seuils des sommes symétriques . . .

Quantifier la valeur ajoutée de l'opérateur de fusion peut aussi être vu comme l'estimation de la redondance entre les sources. Cette information peut être utilisée par le processus de prise de décision pour moduler ses ordres. Par exemple, un robot mobile pourra exploiter l'information de valeur ajoutée pour décider ou non de se repositionner dans l'espace pour obtenir une perception du monde qui l'entoure sous un autre angle.

Conclusion

DANS ce mémoire, nous nous sommes intéressés aux opérateurs de fusion dans le cadre de la théorie des possibilités. Notre but était tout d'abord de présenter un grand nombre d'opérateurs de ce formalisme et de rappeler leurs propriétés les plus classiques. Cette présentation a conduit à une classification des opérateurs en fonction du type de la redondance exploitée (redondance de sources ou redondance temporelle) tout d'abord puis ces deux familles ont été subdivisées en fonction du nombre de sources qu'il est possible de combiner et enfin, une dernière subdivision a permis de distinguer quatre grands comportements : la conjonction, le compromis, la disjonction et l'adaptativité. Enfin, il est aussi possible de distinguer les opérateurs logiques effectuant des combinaisons à l'aide d'opérateurs ensemblistes (intersection et union) et les opérateurs arithmétiques qui combinent linéairement les degrés de possibilités.

Nous avons mis l'accent sur les dangers du choix du mode de combinaison *a priori* et détaillé les opérateurs adaptatifs qui tentent de déterminer correctement le bon mode de combinaison à appliquer en utilisant la notion d'indice de conflit. Nous avons montré que la hauteur maximale de l'intersection de deux sources utilisée dans la règle adaptative de DUBOIS et PRADE est un indicateur de conflit au résultat parfois discutable. Ceci nous a conduit à présenter les autres indices de conflits existants comme les indices de similarité utilisés dans le cadre de l'analyse de données ou les distances entre distributions de possibilités. Nous avons alors proposé deux nouveaux indices de conflit : l'indice de conflit horizontal et l'indice de conflit flou.

L'indice de conflit horizontal utilise un algorithme de complexité 1 pour calculer le conflit entre deux distributions trapézoïdales ou triangulaires. La linéarité de ce nouvel indice a été démontrée mais il reste à l'explorer plus précisément pour découvrir d'éventuelles autres propriétés.

Il est parfois des applications dans lesquelles le conflit peut aussi être considéré comme une information imparfaite et donc représentée par un ensemble flou ou une distribution de possibilité. Nous avons effleuré la notion d'indice de conflit flou en le définissant comme un nombre flou particulier. Il serait intéressant d'en rechercher les éventuelles propriétés. Par ailleurs, il existe aussi des techniques permettant de définir un concept flou à partir d'une information nette. Ces voies n'ont pas été explorées et semblent prometteuses. Enfin, il existe certainement de nombreuses méthodes pour introduire la notion d'indice de conflit flou dans une règle adaptative. Il y a là encore matière à de nouveaux travaux.

La règle de DUBOIS et PRADE est composée de l'union d'un terme conjonctif et d'un terme disjonctif. Nous nous sommes interrogés sur l'absence d'un terme de compromis et avons proposé deux nouvelles règles de fusion adaptative intégrant un terme de compromis. Il resterait à étudier leurs propriétés, notamment sur des exemples complexes comme par exemple des distributions multimodales. Cependant, l'intégration du compromis ne semble pas être une piste sérieuse. En effet, si ce raisonnement est poussé à l'extrême, il serait possible d'envisager la combinaison avec tout type d'opérateur, ce qui ne semble pas très réaliste.

Si plus de deux sources doivent être combinées les opérateurs adaptatifs ne peuvent plus être utilisés et l'utilisateur pourra se tourner vers les méthodes qui groupent préalablement les sources concordantes dans une même classe. Ces techniques exploitent l'idée que combiner des sources en accord est toujours plus facile (et conduit à de meilleurs résultats) que de combiner des sources en désaccord. En effet, lorsque les sources sont en accord, il "suffit" de s'intéresser aux zones communes. Nous avons proposé une méthode de fusion avec classification préalable dans laquelle une source appartenait à une classe et une seule. Les résultats obtenus sur un exemple simple (les distributions étaient triangulaires) montrent que cette technique est prometteuse. Il serait intéressant de développer la technique présentée ici et de la tester sur des distributions plus complexes.

D'autres méthodes de classifications peuvent être envisagées et notamment celles dans lesquelles la notion d'appartenance à une classe est floue (comme par exemple les méthodes de type FCM).

Le choix du processus de fusion étant fait, il est nécessaire d'implémenter ce dernier sur ordinateur. Cette implantation pose de nombreux problèmes qu'ils soient classiques (comme la détermination du pas de discrétisation) ou plus spécifiques aux techniques possibilistes comme la représentation efficace des distributions de possibilités en mémoire ou la rapidité de calcul du résultat. Nous nous sommes intéressés – dans la librairie de fusion – à la complexité des algorithmes et à l'optimisation de leur implantation tout en favorisant la réutilisabilité et la facilité d'ajout de nouveaux procédés de fusion. Cet outil a été utilisé avec succès au sein du laboratoire mais de nombreux opérateurs manquent. Il serait intéressant d'y ajouter la fusion par intégrales floues, les opérateurs **MICA** . . . Par ailleurs, la librairie ne peut manipuler que des distributions à référentiel numérique. Il serait judicieux d'y ajouter la possibilité de manipuler des distributions à référentiel symbolique. La librairie a été prévue pour fonctionner dans un environnement "multi-thread". Il resterait à passer les tests classiques pour s'en assurer. Enfin, les sources d'informations étant souvent fédérées par un réseau local, il semble prometteur de rendre les objets de la librairie "brokables" pour permettre un échange de données transparents entre différents ordinateurs.

Enfin, après avoir effectué une combinaison, l'intérêt d'évaluer le "travail" fourni par l'opérateur de fusion a été montré. Pour cela, nous définissons plus précisément le but de la combinaison dans chaque mode de base : conjonction, disjonction et compromis. L'évaluation consiste alors à calculer le "travail" introduit par l'opérateur en observant l'intensité avec laquelle il a atteint son but ; par comparaison des informations en entrée et du résultat. Dans une application, l'évaluation est animée par au moins deux buts : faciliter le choix de l'opérateur et le réglage d'éventuels paramètres propres à l'opérateur. L'évaluation des opérateurs dans quelques applications concrètes a permis de proposer un algorithme d'aide au choix de la technique de fusion. Cependant, seule l'évaluation d'opérateurs de fusion "atomiques" a été abordée et il resterait à s'intéresser à l'évaluation de systèmes de fusions complexes. Par ailleurs, les indices d'évaluation sont empiriques et l'analyse de leurs propriétés est nécessaire. Par ailleurs, il est aussi possible de choisir une autre voie pour effectuer l'évaluation. Le gain pourrait alors être défini comme la possibilité de prendre une meilleure décision avec le résultat de la fusion qu'avec les entrées prises séparément les unes des autres.

Index

– Symboles –

α -coupe 97
LEWIS' imaging 89

– A –

analyse de données 25
attribut 1

– B –

bibliothèques logicielles 4, 103
bisymétrie 17

– C –

cancellation (erreurs de) 104
cardinal 124
CCrisp (classe DFOL) 116
CDiscret (classe DFOL) 116
CDistribution (classe DFOL) 116
CFusionAlg (classe DFOL) 112
CIntervalle (classe DFOL) 107, 114
CIVB 8
CMembershipPoint (classe DFOL) 116
CNNFusionAdaptative (modèle DFOL) 112
CNNMultiFusionAdaptative (modèle DFOL) 112
CNNQuelconque (classe DFOL) 116
CNNQuelconque (classe DFOL) 119
combinaison partielle 76
combiner 1
commutativité généralisée 64
compatibilité 76
 d'ordre n 76
compensation 67
complexité 32
compromis 3, 7, 121, 122, 128
conditionnement 4, 87–89, 98
conjonction 3, 7, 121, 122, 128
contraction 88, 110, 121, 123, 133
convexité de ZADEH 41
CQuelconque (classe DFOL) 116
CRevisionAlg (classe DFOL) 112
CSegment (classe DFOL) 107, 111, 114
CSimilarite (classe DFOL) 114

– D –

DFOL 103

disjonction 3, 7, 121, 122, 128
distance 125
 de HAMMING 126
 maximum 34
distributions
 discrètes 105
 quelconques 105
dualité 16
décomposabilité 67

– E –

ensemble le plus flou 125
entropie de SHANNON 125
expansion 88, 110, 121, 123, 133

– F –

fiabilité d'une source 4, 60
focalisation 88
fusion
 d'un grand nombre de sources 49
 dissymétrique 4, 87, 103, 121
 en cascade 4
 modulaire 50
 symétrique 87, 103, 121

– G –

gain d'un opérateur de fusion 121

– H –

HauteurH (classe DFOL) 114
hypothèse
 du mode fermé 9
 du mode ouvert 10

– I –

idempotence 65
indicateur
 de (X, Y) 26
 de conflit 25, 31, 32
 de conflit flou 36
 de conflit plus pessimiste 32
 de conflit robuste 34
 de flou 125
 de similarité 26
indice

- de conflit 3
- de similarité 3, 25
- information
 - factuelle 88
 - générique 88
- instanciation 111
- intervalle 108
- intégrale floue 19, 79
 - de CHOQUET 67, 80, 82
 - de SUGENO 80
 - à T-Conormes 81
- L -
- linéarité de H_Z 36
- M -
- MATLAB 4, 103
- mesure 1
- mesure floue 77, 79
 - de SUGENO 79
 - de WIENER-SHANON 79
- mise-à-jour 4, 87, 89
- modèle 110
- monde
 - dynamique 89
 - statique 88
- mélanger 1
- méthodes de fusion
 - mesures de confiance 2
 - probabiliste 2
- N -
- non interactives (variables) 71
- non spécificité 124
- normalisation 10, 15
- noyau 35
- O -
- opérateur
 - "linéaire de DUBOIS et PRADE" 43
 - "ou exclusif" 60
 - "ou-exclusif" 53
 - "quantifié" 53
 - "simples" 3
 - "union/compromis" adaptatif 46
 - "et" logique 54
 - "non et" logique 54
 - "non ou" logique 54
 - "ou exclusif" 53
 - "ou exclusif" logique 54
- adaptatif 3, 15, 25, 121
- archimédien 12
- associatif 11
- auto-dual 11
- CD 14
- CICB 8, 14, 75
- CIVB 8, 14
- commutatif 10
- conjonctif 13
- continu 12
- de composition interne 10
- de compromis 13
- disjonctif 13
- homogène 12
- idempotent 11, 17
- intégrale floue 79
- loi du tiers exclu 13
- monotone 12
- moyenne 17
- médiane 17
- nilpotent 12
- non contradictoire 13
- OWA 3, 4, 19, 63–65, 69, 75, 79, 85
 - préservant la plausibilité maximale 13
 - préservant le zéro 13
 - quasi-associatif 11
 - robuste 12
 - simple 121
 - somme symétrique 19, 50
 - strictement monotone 12
- T-Conorme 50, 64, 65, 79, 81
- T-Norme 30, 50, 64, 65, 75, 118
- WOWA 68, 85
 - à classement de sources 52
- opération $-\Delta$ 81
- P -
- possibilité conditionnelle 4, 92
- principe d'extension 70
- probabiliste 96
- programmation par objets 4
- projection 4, 87, 89
- R -
- red black tree 72, 105
- révision 87, 88, 98, 110, 133
 - "classique" 88
 - "simple" 88, 121
- S -
- sAdaptiveDuboisPrade (structure DFOL)
 - 111
- segment 107, 114
- seuil de saturation 42
- similarité 26, 76

de BRAVAIS-PAARSON	29
de BRAVAIS-PAARSON modifié	29
de DICE	28
de HARMAN	28
de JACCARD	28
de KENDALL	27
de KULEZYNSKI	28
de KULEZYNSKI modifié	28
de OCHIAI	29
de ROGER et TAMINO	28
de RUSSEL-RA	27
de SOKAL-MICHENER	27
de SOKAL-SNEAT	29
de YULE	27
de YULE modifié	29
spécificité	124
STL	105
système atomique	121
système de T-Conormes	81

- T -

T-Conorme	
associativité	16
commutativité	16
coïncidence intersection/union	16
loi de DE MORGAN	16
élément neutre	16
T-Norme	3, 52, 92
associativité	16
commutativité	16
coïncidence intersection/union	16
loi de DE MORGAN	16
paramétrée	17
élément neutre	16
table de hachage	104
template	110

Bibliographie

- [Abidi et Gonzalez, 1992] ABIDI, M. A. et GONZALEZ, R. C. (1992). « *Data Fusion in Robotics and Machine Intelligence* », Chapitre 3. Academic Press, Inc.
- [Bandemer et Näther, 1992] BANDEMER, H. et NÄTHER, W. (1992). « *Fuzzy data analysis* », pages 121–176. Kluwer Academic Publishers, Dordrecht.
- [Bloch, 1996] BLOCH, I. (1996). « Information Combination Operators for Data Fusion: A Comparative Review with Classification ». *IEEE SMC*, 26(1):447–458.
- [Bouchon-Meunier et al., 1996] BOUCHON-MEUNIER, B., RIFQI, M., et BOTHOREL, M. (1996). « Towards general measures of comparison of objects ». *Fuzzy Sets and Systems*, 84:143–153.
- [Chen et al., 1995] CHEN, S.-M., YEH, M.-S., et HSIAO, P.-Y. (1995). « A comparison of similarity measures of fuzzy values ». *Fuzzy sets and systems*, 72:79–89.
- [Chi et al., 1996] CHI, Z., YAN, H., et PHAM, T. (1996). « *Fuzzy Algorithms with Applications to Image Processing and pattern Recognition* », Chapitre 4, pages 85–106. World Scientific.
- [Chin, 1994] CHIN, L. (1994). « Application of neural networks in target tracking data fusion ». Dans *IEEE AES*, volume 30(1), pages 281–286.
- [Cliff, 1997] CLIFF, J. (1997). « *Uncertainty Analysis in Engineering and the Sciences* », Chapitre 3, pages 37–52. Bilal Ayyub and Madan Gupta, Kluwer.
- [de Cooman, 1997] de COOMAN, G. (1997). « Possibility Theory II: Conditional Possibility ». *Int. Journal of General Systems*, pages 325–351.
- [De Luca et Termini, 1972] DE LUCA, A. et TERMINI, S. (1972). « A definition of non-probabilistic entropy in the setting of fuzzy sets theory ». Dans *Information and Control*, volume 20, pages 301–312.
- [Delmotte, 1998] DELMOTTE, F. (1998). « *Analyse multi-modèle* ». PhD thesis, Université des Sciences et Techniques de Lille, Ecole Centrale, Lille.
- [Delplanque et al., 1997] DELPLANQUE, M., DESODT-JOLLY, A., et JOLLY, D. (1997). « Fusion dissymétrique d'informations incomplètes pour la classification d'objets sous-marins ». *Traitement de signal*, 14:511–522.
- [Deveughele, 1993] DEVEUGHELE, S. (1993). « *Etude d'une Méthode de Combinaison Adaptative d'Informations Incertaines dans un cadre Possibiliste* ». PhD thesis, Université de Technologie de Compiègne, Compiègne.
- [Deveughele et Dubuisson, 1996] DEVEUGHELE, S. et DUBUISSON, B. (1996). « The Influence of a Conflict Index in the Frame of the Adaptative Combinaison ». Dans *CESA'96 IMACS Symp. Robotics and Cybernetics*, pages 98–103, Lille.
- [Dong et Wong, 1987] DONG, W. M. et WONG, F. (1987). « Fuzzy Weighted Averages an Implementation of the Extension Principle ». *Fuzzy Sets and Systems*, 21:183–199.
- [Dubois et Kalfsbeek, 1990] DUBOIS, D. et KALFSBEEK, H. (1990). « Elicitation, Assessment and Pooling of Expert Judgements using Possibility Theory ». Dans *World Congres on Cybernetics and Systems*, volume 11, pages 360–367, New York.
- [Dubois et Prade, 1988] DUBOIS, D. et PRADE, H. (1988). *Théorie des Possibilités*. Masson.
- [Dubois et Prade, 1991] DUBOIS, D. et PRADE, H. (1991). Updating with belief functions, ordinal conditional functions and possibility measures. Dans P.P. BONISSONE, M. Henrion, L. K. et LEMMER, J., éditeurs, *Uncertainty in Artificial Intelligence*, volume 6, pages 311–329. Elsevier Science Publisher B.V.

- [Dubois et Prade, 1992] DUBOIS, D. et PRADE, H. (1992). « Combination of Fuzzy Information in the Framework of Possibility Theory ». Dans *Data Fusion in Robotics and Machine Intelligence*, Abidi, Mongi A. and Gonzalez, Rafael C., pages 481–505. Academic Press, Inc.
- [Dubois et Prade, 1994a] DUBOIS, D. et PRADE, H. (1994a). « La Fusion d'Informations Imprécises ». *Traitement de signal*, 11:447–458.
- [Dubois et Prade, 1994b] DUBOIS, D. et PRADE, H. (1994b). « A survey of belief revision and updating rules in various uncertainty models ». *Int. Journal of Intelligent Systems*, pages 61–100.
- [Fishburn, 1967] FISHBURN, P. C. (1967). « Methods of Estimating Additive Utilities ». *Management Science*, 13(7):435–453.
- [Fodor et al., 1995] FODOR, J., MARICHAL, J., et ROUBENS, M. (1995). « Characterization of the Ordered Weighted Averaging Operators ». *IEEE Transactions on Fuzzy Systems*, 3(2):236–240.
- [Fortemps et Roubens, 1996] FORTEMPS, P. et ROUBENS, M. (1996). « Ranking and Defuzzification Methods based on Area Compensation ». *Submit on Fuzzy sets and Systems*.
- [Gärdenfors, 1988] GÄRDENFORS, P. (1988). *Knowledge in Flux: Modeling the Dynamics of Epistemic States*. MIT Press, Cambridge.
- [Gärdenfors, 1991] GÄRDENFORS, P. (1991). « *Belief Revision* », pages 1–28. Cambridge University Press, Lund University.
- [Grabisch et al., 1995] GRABISCH, M., NGUYEN, H. T., et WALKER, E. A. (1995). *Fundamentals of Uncertainty Calculi with Applications to Fuzzy Inference*, volume 30. Kluwer Academic Publisher.
- [Hisdal, 1978a] HISDAL, E. (1978a). « Conditional possibilities – Independance and non-interactivity ». Dans *Fuzzy Sets and Systems*, volume 1, pages 283–297.
- [Hisdal, 1978b] HISDAL, E. (1978b). « Conditional Possibilities Independence and Noninteraction ». *Fuzzy Sets and Systems*, pages 283–297.
- [Jeffrey, 1965] JEFFREY, R. (1965). *The Logic of Decision*. McGraw-Hill, New York.
- [Kailath, 1967] KAILATH, T. (1967). « The Bhattacharyya distance measure in signal detection ». *IEEE Trans. Commun. Tech.*, 1:607–637.
- [Katsuno et Mendelzon, 1992] KATSUNO, H. et MENDELZON, A. (1992). On the difference between updating knowledge base and revising it. Dans P.GÄRDENFORS, éditeur, *Belief revision*, pages 183–203. Cambridge University Press, Cambridge.
- [Kaufmann, 1977] KAUFMANN, A. (1977). « *Introduction à la théorie des ensembles flous* », volume 4, Chapitre 92, pages 39–58. Masson.
- [Kewley, 1993] KEWLEY, D. (1993). « A Model for Evaluating Data Fusion Systems ». *IEEE – 1058-6393/93*, pages 273–277.
- [Klir et Folger, 1988] KLIR, G. et FOLGER, T. (1988). « *Fuzzy Sets Uncertainty and Information* », Chapitre 4. Prentice Hall.
- [Lerman, 1970] LERMAN (1970). *La classification automatique*. Gautier Villars Paris.
- [Lewis, 1976] LEWIS, D. (1976). « Probabilities of conditionals and conditionnal probabilities ». *The Philosophical Review*, pages 297–315.
- [Oussalah, 1998] OUSSALAH, M. (1998). « *Fusion de données par la théorie des possibilités. Application à la localisation d'un robot mobile* ». PhD thesis, Université d'Evry Val D'Essone, Every Val D'Essonne.
- [Schubert, 1994] SCHUBERT, J. (1994). « *Cluster-based specification techniques in Dempster-Shafer theory for an evidential intelligence analysis of multiple target tracks* ». PhD thesis, Department of numerical analysis and computing science, Royal Institute of Technology, Stockholm.
- [Sedgewick, 1991] SEDGEWICK, R. (1991). « *Algorithmes en Langage C* », Chapitre 6. InterEditions.
- [Sims et Wang, 1996] SIMS, J. R. et WANG, Z. (1996). « Fuzzy Measures and Fuzzy Integrals : an overview ». *International Journal on General Systems*, 17:157–189.
- [Smets et Keenes, 1990] SMETS, P. et KEENES, R. (1990). « The Transferable Belief Model ». Rapport Technique, Université Libre de Bruxelles.

- [Sobrevilla et al., 1996] SOBREVILLA, P., DOMÈNEC, P., et MONTSÉNY, E. (1996). « Segmentation of Textured Aerial Images using OWA Operators ». Dans *IPMU'96*, pages 1513–1518.
- [Strauss et Aimé, 1996] STRAUSS, O. et AIMÉ, E. (1996). « Filtering data by using three error theories together ». Dans *EUFIT'96*, pages 1640–1644, Aachen, Germany.
- [Torra, 1997] TORRA, V. (1997). « The Weighted OWA Operator ». *International Journal of Intelligent Systems*, 12:153–166.
- [Vienne et al., 1996] VIENNE, F., ZUNINO, G., DESODT, A., et JOLLY, D. (1996). « Application of Fuzzy Logic to the Fusion of Information in the Framework of Man/Machine Interface ». Dans *CESA'96 IMACS Symp. Robotics and Cybernetics*, pages 416–421, Lille.
- [Vincke, 1967] VINCKE, P. (1967). « Modélisation des Préférences et Théorie de l'Utilité : Résultats existants et voies de recherche ». *Revue Belge de recherche opérationnelle, de Statistiques et d'Informatique*, 16(4):1–15.
- [Waltz et Llinas, 1990] WALTZ, E. et LLINAS, J. (1990). *Multisensor Data Fusion*. Artech House.
- [Wang, 1997] WANG, W.-J. (1997). « New similarity measures on fuzzy sets and on elements ». *Fuzzy sets and systems*, 85:305–1309.
- [Wang et al., 1995] WANG, X., DE BAETS, B., et KERRE, E. (1995). « A comparative study of similarity measures ». *Fuzzy sets and systems*, 73:259–268.
- [Yager, 1983] YAGER, R. (1983). « An introduction to applications of possibility theory ». *Human Systems Management*, 3:246–269.
- [Yager, 1988] YAGER, R. R. (1988). « On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decisionmaking ». Dans *IEEE SMC*, volume 18(1), pages 183–190.
- [Yager, 1998] YAGER, R. R. (1998). « Fusion of Ordinal Information using Weighted Median Aggregation ». *International Journal of Approximate Reasoning*, 18:35–52.
- [Yager et Kelman, 1996] YAGER, R. R. et KELMAN, A. (1996). « Fusion of Fuzzy Information With Considerations for Compatibility, Partial Aggregation, and Reinforcement ». *International Journal of Approximate Reasoning*, pages 93–122.
- [Zadeh, 1977] ZADEH, L. (1977). Theory of fuzzy sets. Dans BELZER, H. . K., éditeur, *Encyclopedia of Computer Science and Technology*. Marcel Dekker.
- [Zunino et al., 1997a] ZUNINO, G., DESODT, A., et JOLLY, D. (1997a). « La fusion de données imparfaites ». Dans *Automatique-Génie Informatique-Image AGI'97*.
- [Zunino et al., 1998a] ZUNINO, G., DESODT, A., et JOLLY, D. (1998a). « Data Fusion with Multiple Sources ». Dans *Information Processing and Management of Uncertainty in knowledge-based systems, IPMU'98*, volume 2, pages 1585–1591.
- [Zunino et al., 1997b] ZUNINO, G., JOLLY-DESODT, A., et JOLLY, D. (1997b). « Data Fusion : A New Adaptative Combination Rule ». Dans *ICSC Symp. on Soft Computing SOCO'97*, pages 196–202, Nîmes.
- [Zunino et al., 1997c] ZUNINO, G., JOLLY-DESODT, A., et JOLLY, D. (1997c). « DFOL : A Data Fusion Object Library ». Dans *ICSC Symp. on Soft Computing SOCO'97*, pages 189–195, Nîmes.
- [Zunino et al., 1997d] ZUNINO, G., JOLLY-DESODT, A., et JOLLY, D. (1997d). « Indication of Conflict in Data Fusion ». Dans *IFAC Symp. on Intelligent Components and Instruments for Control Applications SICICA'97*, pages 393–398, Annecy.
- [Zunino et al., 1998b] ZUNINO, G., JOLLY-DESODT, A., et JOLLY, D. (1998b). « Influence of T-Norm Choice in Possibilistic Dissymmetric Data Fusion Rule ». Dans *6th European Congress on Intelligent Techniques & Soft Computing EUFIT'98*, volume 2, pages 1267–1271, Aachen, Allemagne.
- [Zunino et al., 1999] ZUNINO, G., JOLLY-DESODT, A., et JOLLY, D. (1999). « Data Fusion System Evaluation in Possibility Theory ». *EUFIT'99*.
- [Zwich et al., 1987] ZWICH, R., CARLSTEIN, E., et BUDESCU, D. V. (1987). « Measures of Similarity Among Fuzzy Concepts : A Comparative Analysis ». *International Journal of Approximate Reasoning*, 1:221–242.

