

Problèmes

Tous les énoncés qui suivent ont fait l'objet de projets réalisés au cours des dernières années par des étudiants principalement de l'IIE, mais aussi de l'Université Paris VI ou de l'ENSTA.

Problème 1 Programmation en Fril

a) Soit le programme :

((vole X) (oiseau X)) : (0.9 1)

((vole X) (pingouin X)) : (0 0)

((oiseau X) (pingouin X))

((oiseau X) (aigle X))

((pingouin hector))

((aigle edgar))

On pose la question : qs ((vole Hector)) quelle sera la réponse ?

On rajoute alors au début, avant les clauses, la "déclaration" : dempster vole

Quelle sera la réponse à qs ((vole X)) ?

b) Soit maintenant le programme Fril :

(haut [0.4 : 0, 0.7 : 1])

(assez_haut [0.1 : 0, 0.6 : 1])

(moyen [0.2 : 0, 0.5 : 1, 0.8 : 0])

(bas [0.4 : 1, 0.7 : 0])

(cher [30 : 0, 70 : 1])

(correct [20 : 0, 50 : 1, 60 : 1, 90 : 0])

(pascher [30 : 1, 80 : 0])

dempster choix

((choix X ok) (performance X bonne) (aspect X moderne)) : (0.9 1)

((choix X ok) (prix X cher)) : (0 0.05)

((choix X ok) (securite X bas)) : (0 0.2)

((performance X bonne) (moteur X ok) (securite X haut)) : (0.9 1)

((performance X bonne) (securite X haut)) : (0.6 1)

((aspect panhard moderne)) : (0.8 1)
((aspect levassor moderne)) : (0.9 1)
((securite panhard assez_haut)) : (1 1)
((securite levassor moyen)) : (1 1)
((moteur panhard ok)) : (0.7 1)
((moteur levassor ok)) : (0.9 1)
((prix panhard correct)) : (1 1)
((prix levassor pascher)) : (1 1)

Réponses à la question qs ((choix X Y)) ?

c) Trouver des exemple didactique (ni trop complexe, ni trop trivial) entre vingt et cinquante règles d'utilisation de Fril utilisant des ensembles flous. Le système doit faire intervenir des opérations arithmétique sur les trapèzes et aussi éventuellement manipuler les supports avec "supp_query".

Problème 2 MVL

Explorer le système MVL, en particulier donner un compte-rendu sur l'aspect probabiliste et trouver un exemple de démonstration non trivial.

Problème 3 Modus ponens généralisé

Pour la règle $X \text{ est } A \rightarrow Y \text{ est } B$, créer un outil visualisant B' par rapport à B lorsque A, B sont des triangles et X une valeur précise vérifiant plus ou moins A . L'utiliser pour produire une table de schémas suivant les différents opérateurs T de conjonction et π d'implication.

Problème 4 Opérateurs d'agrégation

Créer un petit logiciel où il soit possible de définir et représenter deux ensembles flous trapézoïdaux, et d'en déduire une agrégation des deux suivant les méthodes exposés au chapitre 3. Prévoir plusieurs relations possibles.

Problème 5 Systeme expert flou

Construire un système expert en chaînage-avant (analyse financière, évaluation du prix d'un appartement ...) dans lequel on puisse discuter des différents modes d'implication.

Problème 6 La machine à laver

On souhaite étudier la normalisation de différentes valeurs floues inférées par un système expert destiné à contrôler une machine à laver (degré de saleté = degré de transparence).

Les règles sont :

synthétique ou laine \rightarrow fragile	fragile \rightarrow essorage court
laine \rightarrow température froide	synthétique \rightarrow durée courte
laine \rightarrow durée très courte	saleté moyenne \rightarrow durée courte
coton \rightarrow durée longue	saleté grande \rightarrow durée longue
durée courte \rightarrow essorage court	saleté faible \rightarrow durée très courte
couleur vive \rightarrow RD élevé	saleté grande \rightarrow trempage
blanc \rightarrow température très chaude	RD élevé \rightarrow température froide
poids élevé \rightarrow durée élevée	RD moyen \rightarrow température tiède
RD faible \rightarrow température chaude	saleté très grande \rightarrow trempage et prélavage

On pourra prendre une normalisation différente pour les valeurs des différents arguments: le min pour la durée et la température, le max pour le risque de déteindre RD, ou encore l'intersection et éventuellement en signalant les contradictions pour les autres variables (programmation en lisp).

Problème 7 Détermination de l'âge d'un enfant

Le but est de déterminer un âge ou de vérifier le développement par rapport à l'âge réel en répondant vaguement par "ne sait pas", "vrai", "plutôt vrai", "à moitié vrai", "plutôt faux", "faux", à des questions. Ces questions correspondent à des quadruplets de possibilités (m, a, b, M) dont la signification est la suivante : possibilité 1 entre a et b, possibilité réduite entre m et a et entre b et M de façon à obtenir une représentation trapézoïdale. Chaque règle s'applique avec un coefficient fonction de la valeur de vérité donnée. Le résultat est une distribution de possibilité sur [1, 36] que l'on peut visualiser (programmation en lisp).

sourire (1 1 1 2)	suivre un objet horizontalement (1 2 2 3)
suivre un objet (2 3 3 4)	tient sa tête (2 3 3 4)
gazouillis (1 3 4 5)	attraper un objet d'une main (3 3 4 5)
se retourne (1 2 3 6)	2 incisives en bas (3 6 8 10)
tenir entre le pouce et l'index (8 9 9 10)	dire une syllabe répétée (6 9 9 10)
passer un objet d'une main à l'autre (6 7 7 8)	reconnaissance des parents (6 8 10 10)
position assise (6 7 9 11)	monter un escalier (9 12 12 15)
debout avec soutien (8 9 11 12)	2 incisives en haut (3 8 10 12)
marcher à 4 pattes (6 6 12 12)	manger seul à la cuillère (15 18 24 24)
tient son biberon (10 12 18 18)	8 incisives (6 10 14 20)
faire au-revoir (10 12 15 18)	debout sans soutien (8 12 15 24)
marcher (9 12 15 24)	dire un mot signifiant (10 12 15 24)
connaître des verbes (12 13 17 18)	2 premières molaires (12 14 16 18)
empiler des cubes (12 18 20 24)	désigner les parties du corps (12 15 16 24)
canines (12 15 20 24)	chercher un objet caché (18 20 24 24)
assembler deux mots (18 18 24 28)	propreté le jour (15 24 30 30)
s'habiller (18 24 30 36)	propreté la nuit (18 30 36 36)
descendre un escalier (16 24 30 30)	dessin d'une tête (30 31 35 36)
acquisition du moi (30 31 35 36)	

Problème 8 Détermination de l'époque d'une peinture

Même problème avec la base de connaissance suivante :

- scène religieuse (700 1000 1550 1700)
- peinture sur bois (1000 1100 1450 1650)
- présence de dorure (1150 1200 1400 1475)
- portrait (1450 1450 1600 1850)
- perspective déformée (1000 1300 1400 1450)
- architecture classique (1400 1450 1600 1850)
- tableau solennel (1550 1600 1700 1830)
- peinture précise très détaillée (1500 1550 1850 1875)
- personnages statufiés (1750 1775 1840 1870)
- jeux d'ombres et de lumières (150 1600 1700 1800)
- personnages assis (1550 1620 1700 1750)
- représentation de société (1650 1700 1800 1850)
- les couleurs se mélangent (1720 1730 1990 1990)
- caractère historique (1750 1800 1850 1900)
- ombres de couleurs (1800 1850 1900 1950)
- scène réaliste de la vie quotidienne (1750 1800 1850 1900)
- grosse couche de peinture (1850 1870 1990 1990)
- beaucoup de points (1850 1880 1900 1920)
- personnages déformés (1890 1910 1950 1990)
- tableau abstrait (1900 1920 1990 1990)
- produit de consommation (1940 1950 1990 1990)
- aucune forme distinguable (1920 1960 1990 1990)
- mouvement des corps et des tissus (1575 1600 1875 1900)

On pourra sortir les résultats sous forme des 0,25 ou 0,5 ou 0,75 ou 1-coupes de l'ensemble flou obtenu.

Problème 9 Aide à la décision dans l'achat d'une voiture

On qualifie les attributs d'une voiture de la façon suivante :

Cylindrée : pet. = (1100 1100 1300 1500) moy. = (1300 1500 1700 1900) grande = (1900 2000 2500)

Puissance : petite = (30 30 50 55) moyenne = (50 55 70 85)

grande = (70 85 120 140) en kW

Vitesse pet.= (120 140 160 175) moy.= (165 175 185 195)

grande = (185 195 210 220)

Prix petit = (45 50 70 80) moyen = (70 80 95 105)

grand = (95 105 120 140) en kF

Taille à définir en m.

Chaque voiture existante, pour ces 5 attributs reçoit un triplet de possibilités correspondant à chaque fois aux 3 qualificatifs. L'utilisateur va répondre pour ces 5 attributs, le qualificatif qu'il souhaite et l'importance qu'il y attache. (voir intégrale de Sugeno)

Problème 10 Un système hiérarchique de règles définissant une fonction

Les règles toutes du type "u est A → v est B", sont classées par ordre décroissant de priorités allant des plus précises au moins précises [Yager 92]. Si F(R) désigne l'ensemble des sous ensemble flous de R, chaque paquet de règles va réaliser une fonction de F(R) dans F(R). Soit h(E) le sup de la fonction d'appartenance de l'ensemble flou E (la hauteur), et faible(x) = 1-x un prédicat sur [0,1].

Soit u une entrée réelle, on pose V_0 l'ensemble vide donc $\text{faible}(V_0) = 1$.

Si V_{i-1} est la sortie du paquet i-1 réalisant une fonction f_{i-1} de F(R) dans F(R), alors f_i sera le résultat de ni règles modifiées par rapport à celles qui ont été données du type "u est A_{ij} et $h(V_i)$ est faible → v est B_{ij} "

$h(V_i)$ mesure en fait la façon dont on a pu atteindre l'étape i+1, il se peut donc qu'une conclusion précise fasse sortir avant toutes les étapes.

A chaque paquet, on fait la disjonction des conclusions à la Mamdani ce qui fait le résultat $B_j = \cup B_{ij}$ et on pose $V_i = V_{i-1} \cup B_j$. La sortie v sera la moyenne du dernier V_n .

Faire l'exemple suivant (en lisp) et la représentation de la fonction $u \rightarrow v$, si les règles sont :

- 1° paquet U est 5 → V est 13
 U est 75 → V est 180
 U est 85 → V est 100
- 2° paquet U est environ 10 → V est environ 20
 U est environ 30 → V est environ 50
 U est environ 60 → V est environ 90
 U est environ 80 → V est environ 120
 U est environ 100 → V est environ 150
- 3° paquet U est petit → V est environ 40
 U est moyen → V est environ 85
 U est grand → V est environ 130
- 4° paquet U est quelconque → V est 2*U

dans lequel "environ x" est le triangle isocèle centré sur x de demi-base 10 sur [0,100] et 20 sur [0, 200], petit, moyen, grand étant une famille de triangles de demi-base 50 sur [0, 100]. Programmer en lisp et donner d'autres exemples.

Problème 11 Système expert fonctionnant avec la logique des défauts

Dans un système déductif où on a des règles du type $A \rightarrow C$, et $A \text{ et } B \rightarrow \neg C$, on pourra dire que A entraîne C en général (par défaut d'une information B supplémentaire, auquel cas la première règle ne doit plus se déclencher). Ainsi si T et T' sont des ensembles de règles et d'axiomes avec T inclu dans T', on n'a pas nécessairement la théorie de T incluse dans celle de T' (logique non monotone).

On souhaite marier les approches flou et défauts dans un petit système déductif limité au calcul propositionnel. Les propositions y reçoivent un degré de vérité v satisfaisant à $v(\neg P) = 1 - v(P)$.

Définitions :

Axiome : règle sans prémisse

Preuve de P : arbre de racine P dont les feuilles sont des axiomes et dont chaque noeud est la conclusion d'une règle dont les prémisses sont les fils du noeud.

Longueur d'une preuve : nombre d'arêtes du plus long chemin

Cardinal d'une preuve : nombre de variables propositionnelles ayant une occurrence dans l'arbre.

On considère un système de règles maintenant affectées de coefficients et la théorie engendrée grâce aux métarègles suivantes :

- 1) Si $A_1 \text{ et } A_2 \text{ et } A_3 \text{ et } \dots \rightarrow C$ est une règle de coefficient r avec chaque A_i connu avec le coefficient a_i , alors C est admis (provisoirement) avec le coefficient $r \cdot \min(a_1, a_2, a_3, \dots)$
- 2) Si P est obtenu par deux preuves de longueurs distinctes, P aura le coefficient correspondant à la preuve de plus courte longueur. (Ainsi on donne la priorité à des déductions plus immédiates et on évite le fait d'avoir à agréger par exemple un axiome avec son éventuelle démonstration, cas fâcheux où une connaissance initiale pourrait se trouver affaiblie.)
- 3) Si P est obtenu par deux preuves de mêmes longueur mais de cardinaux distincts, c'est la preuve de plus grand cardinal qui l'emporte (Ainsi des précisions supplémentaires entraînent une priorité sur des règles générales.)
- 4) Si P est obtenu par deux preuves de mêmes longueur et cardinal, alors on prend simplement la moyenne, mais comme cette opération n'est pas associative, il faudra tenir compte du nombre d'occurrences.
- 5) Si $\neg Q$ et $P \rightarrow Q$ alors $\neg P$ (contraposition).

Soit la base de connaissances suivante :

En général les oiseaux volent. Les pingouins sont des oiseaux qui ne volent pas, sauf ceux des Kerguelen qui volent légèrement. Hector et Victor sont des pingouins des kerguelen, mais Victor ne vole pas vraiment. Nestor est un autre pingouin.

a) Dérouler à la main l'enchaînement des déductions suivant la stratégie ci-dessus.

b) Dessiner et expliquer le graphe pour la théorie de $T_1 = \{A, E, A \rightarrow B, B \rightarrow C, E \rightarrow B\}$ ainsi que pour $T_2 = \{A, B, D, E, A \& B \rightarrow C, B \& C \rightarrow D, C \& E \rightarrow F, D \& F \rightarrow C\}$. L'ensemble de toutes les conclusions est-il toujours parfaitement défini ? Donner un algorithme pour le construire.

c) Trouver une représentation qui va permettre de mettre en oeuvre un tel moteur d'inférence et le programmer en lisp.

d) Trouver un exemple plus étoffé pour le tester et commenter les résultats.

Problème 12 Morphologie mathématique

Si E , ensemble flou de \mathbb{R}^2 , est caractérisé par sa fonction d'appartenance μ , et f est un élément structurant (une définition floue de voisinage) défini par exemple par $f(t) = \min(1, \max(0, 1 - t/a))$ (un cône) où a est de l'ordre de 1 à 5 pixels).

On rappelle que l'on définit la dilatation $\text{Dil}_\mu(x) = \sup_{y \in E} \{T(\mu_E(y), f(d(x, y)))\}$ où T est une t -norme et S sa conorme associée et l'érosion $\text{Ero}_\mu(x) = \inf_{y \in E} \{S(\mu_E(y), 1 - f(d(x, y)))\}$.

On veut une image floue mais non bruitée à partir d'une image floue.

Faire un petit logiciel de démonstration avec quelques images tests dans \mathbb{R}^2 , réalisant la dilatation de l'érosion floue en permettant un choix de a . On prendra la t -norme de Zadeh et la norme Hamming dans \mathbb{R}^2 pour des raisons de simplicité, ainsi, si $a = 1$, en tout point on teste ses 8 voisins, et si $a = 2$ ses 24 voisins.

Problème 13 Classification floue

On prendra l'algorithme de Bezdek modifiant les centres de gravité des différentes classes.

Le programme devra visualiser les itérations de la façon suivante : chaque point de l'ensemble sera représenté par un carré de 10 sur 10 pixels (pour cela ils ne doivent pas être trop voisins). S'il y a par exemple 3 classes fixées, (rouge, verte et jaune) chaque exemple sera représenté par un rectangle contenant (régulièrement répartis) des pixels de chaque couleur au prorata de ses 3 appartenances. On pourra par ailleurs consulter la table des points avec ces taux d'appartenance aux différentes classes, à la fin des itérations. Prévoir deux ou trois bases d'exemples avec 2, 3, 4 classes.

Problème 14 Régulateur cardiaque

On veut modifier de minute en minute les doses de deux substances A et R destinées à accélérer ou retarder le rythme cardiaque (mesuré par B en battements par minutes) en temps réel dans un hôpital. Si B_0 est le rythme fixé, on note $E = B - B_0$ et on fixe les règles par une liste comportant un poids, la conclusion et les deux prémisses :

(0,5 A PB E NB dE PS) (0,8 A PM E NM dE ZE) (1 A PS E NS dE NS)
 (1 A ZE E ZE dE NM) (1 R ZE E ZE dE PM) (1 R PS E PS dE PS)
 (0,8 R PM E PM dE ZE) (0,5 R PB E PB dE NS)

Si chaque prédicat est représenté par une liste de 21 valeurs (les possibilités sur $[-1, 1]$) programmer (en lisp) la fonction B_0 (cible), B_1, B_2 (deux mesures consécutives) \rightarrow couple A, R des doses réelles, en passant par des fonctions ϕ, ψ, χ ramenant à $[-1, 1]$ à définir.

Problème 15 Evitement d'obstacles

On suppose une voiture équipée de 8 capteurs de distances aux obstacles se trouvant sur 8 directions réparties régulièrement AV (avant), AR, GA (gauche), DR, VG (avant gauche), VD, RG, RD ("angles morts").

Ecrire des règles destinées à contrôler la vitesse v et sa rotation α , du type :

(ΔAV est NB) et (v est PB) \rightarrow (Δv est NB)

Que doit-il se passer lors d'une contradiction avec (ΔAR est NB) \rightarrow (Δv est PB) ?

Effectuer une simulation avec simplement trois capteurs dans une fenêtre occupée par différents obstacles aléatoirement distribués (entre 10 et 100), une sorte de labyrinthe.

Règles de Simiand-Noury, pour un pas maximal $pm = 10$, un angle de braquage maximal $am = 60^\circ$ une accélération comprise entre -3 et 2 pixels ainsi que 3 prédicats ZE (1 avant 5, 0 après 15), PS (support $[5, 25]$, noyau $\{15\}$) et PB (nul avant 15, 1 après 25), les règles portent sur les distances gauche g , frontale f , et droite d .

4 règles modifiant le pas :

f ZE \rightarrow dp NB f PB \rightarrow dp PB d ZE \rightarrow dp NS g ZE \rightarrow dp NS

Si $d < g$ les 4 règles :

$g ZE \rightarrow NB$ $g PS \rightarrow da NS$ $(f \text{ ou } d ZE) \rightarrow da PB$ $(f \text{ ou } d PS) \rightarrow da PS$

Si $d < g$, les 4 règles analogues.

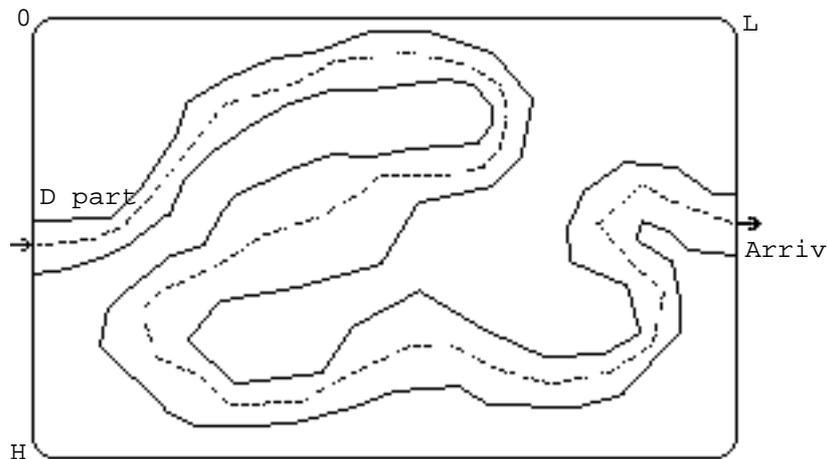
Règles de Nguyen Kiet - Queulin, pour un pas maximal $pm = 5$, un angle de braquage maximal $am = 30^\circ$, $ds = 1$, $dm = 100$ une accélération comprise entre -5 et 5 pixels ainsi que 3 prédicats N, Z, P ramenés à $[-100, 100]$, deux règles pour éviter l'obstacle frontal :

$f Z, pas P \rightarrow dp N, da N, f Z, d Z, g Z \rightarrow da P$, une pour accélérer : $f P \rightarrow dp P$, et 4 règles pour tourner : $d Z, g P \rightarrow da N, g Z, d P \rightarrow da P, g P, f Z \rightarrow da N, d P, f Z \rightarrow da P$

Problème 16 Simulation d'un suivi de route

En traçant sur l'écran un couloir de largeur variable, coudé avec des angles aigus, on cherche à contrôler le trajet d'un robot depuis l'entrée à gauche jusqu'à la sortie à droite.

Le robot doit avancer à vitesse constante (ou en modifiant à la fois sa direction et sa vitesse) en suivant les coudes et en restant si possible au milieu du couloir, il ne dispose pas de capteur frontal, mais seulement de deux capteurs mesurant les distances l et r perpendiculairement (ou plus généralement à un angle \pm ouv de sa direction), aux murs du tunnel.



Ecrire un système de règles floues $(l, \Delta l, r, \Delta r) \rightarrow \Delta\alpha, \Delta v$, avec 3 ou 5 prédicats dans $\{NB, NM, NS, ZE, PS, PM, PB\}$, contrôlant l'angle de la direction et la variation de pas à chaque instant. Visualiser l'avancer du robot.

On peut chercher à ce que le robot conserve sa droite, ou le milieu, ou encore à minimiser les variations de directions en conservant une certaine régularité de trajectoire à la façon de la navigation fluviale.

Problème 17 Simulation d'un robot suiveur

Dans une boucle infinie, un mobile A se déplace grâce aux flèches du clavier, un robot B cherche à le suivre en fixant son cap sur A en modifiant sa vitesse, et en maintenant une distance constante avec A.

Ecrire des règles de conduite destinées à définir une fonction :

- a) $(d, \Delta d) \rightarrow \Delta v$ pour une poursuite rectiligne.
- b) $(d, \alpha, \Delta\alpha, \Delta d) \rightarrow \Delta v$ pour le problème plan
- c) On laisse de côté le problème d'un missile dans l'espace.

Problème 20 Suivi d'une ligne blanche

Tracer la trajectoire oscillante d'un véhicule avec d sa distance algébrique à la ligne mesurée perpendiculairement à sa direction, et non perpendiculairement à la ligne d dans $[-dm, dm]$ et Dd dans $[-vm, vm]$ en entrée et a dans $[-A, A]$ en sortie (règles données par Kamada-Yoshida, programmation en C).

Problème 21 Même sujet

Mais on étudiera de façon systématique l'influence des facteurs d'échelle, du nombre de prédicats (3, 5, 7) de leur base $2p$, de la conjonction (min, produit, Lukasiewicz..) et surtout des règles en retirant ou modifiant chacune d'entr'elles partant par exemple de 5 prédicats et 25 règles.

Le but de ce travail est de décrire de façon rationnelle l'influence de certains facteurs sur les résultats d'un contrôleur flou, en regroupant des informations faites à partir d'une simulation graphique.

On dispose d'un programme C réalisant la trajectoire d'un robot devant suivre une sinusoïde, et d'une interface délivrant pour chaque expérience, un tracé légendé par les valeurs des paramètres utilisés, ainsi qu'une table fournissant le nombre d'occurrences des règles déclenchées.

Le robot mesure à chaque pas sa distance algébrique à la courbe, d mesurée suivant un angle \pm ouv par rapport à sa trajectoire. Sa variation de direction sera contrôlée suivant les deux entrées d et dd (variation de distance). L'angle de braquage étant dans l'intervalle $[-A, A]$, d dans $[-B, B]$, dd dans $[-C, C]$, l'avance ds , la portée maximale des capteurs dm , nsi le nombre de sommets de la courbe inscrits dans la fenêtre considérée, et dim la dimension de la fenêtre. On définit par ailleurs les critères de performance appréciés par la somme rot des valeurs absolues de rotations et la distance parcourue sd que l'on cherche à minimiser.

Travail demandé :

1° Partant des meilleurs résultats donnés dans le rapport de C.Deneuille, pour un jeu de règles complet avec 5 prédicats on désire visualiser le nuage de points correspondant aux meilleures expériences (au sens de rot + sd) et obtenir une loi par corrélation linéaire pour les fonctions empiriques $ds / dim \rightarrow A$, $ds / B \rightarrow A$, $nsi \rightarrow A$, on pourra chercher d'autres relations, attendu que dm , ouv ont moins d'importance que B et C qui en ont eux-même moins que A ou ds . Le rapport B/C est-il toujours du même ordre ?

2° Faire l'examen pour les 25 conclusions de les retirer ou de les modifier. Il serait bon de définir une méthodologie pour ce problème afin d'avoir automatiquement cette recherche. Modification dans le poids des règles?

3° Reprendre cette recherche des "bonnes règles" en faisant varier p la demi-base des triangles avec 5, 7 ou 9 prédicats analogues, et k pour avoir $2k+1$ prédicats triangulaires non isocèles formant une famille binaire dont les sommets $0 = x_0 < x_1 < x_2 < \dots < x_k = 1$ tels que $x_i = i(i+1) / k(k+1)$.

Problème 22 Le chat et la souris

Un chat doit poursuivre une souris en allant m fois plus vite qu'elle mais avec un braquage maximal m fois plus petit. Chacun d'entre eux est contrôlé par des règles floues, le chat cherchant à rejoindre la souris va modifier sa direction suivant des paramètres d'entrées à définir et la souris va chercher à fuir en conservant une direction la plus perpendiculaire possible à celle du chat. Créer un logiciel permettant de modifier les prédicats grâce aux paramètres p et h s'ils sont définis par des triangles centrés sur m , de base $[m-p, m+p]$ et de hauteur $h \geq 1$, tronqués par 1.

Les deux entrées peuvent être les angles tc et ts que font respectivement les directions du chat et de la souris avec l'axe CS qu'ils forment, ces angles étant ramenés à $[-4\pi/5, 4\pi/5]$. Quant aux règles, on pourra essayer pour la souris : tc NB et ts PS alors ∂ dir PB, tc ZE et ts NB alors ∂ dir NS, tc ZE et ts NS alors ∂ dir NB, et leurs symétriques. Outre ce qui est donné au chapitre 5, voici deux autres exemples de systèmes de règles pour la souris :

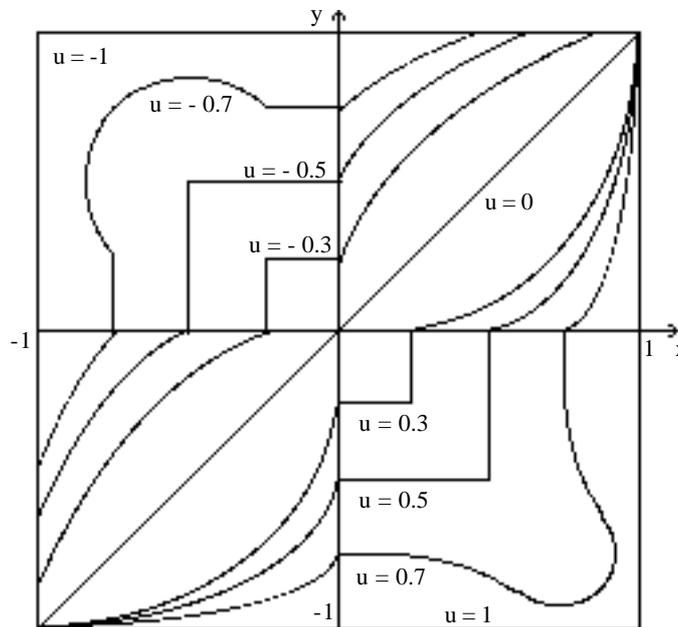
"et" et "ou", les règles en débutant avec un tableau complet et en rompant la symétrie du tableau et en les rayant une par une.

Exemple de contrôleur $(x, y) \in [-1, 1]^2 \rightarrow u \in [-1, 1]$ donné par [Bühler 94].

On présente sur la partie haute gauche le résultat u obtenu par max-min et centre de gravité avec une base de 9 règles. Sur la partie droite basse, les mêmes lignes si 5 règles opèrent seulement.

P	N	N	Z
Z	N	Z	P
N	Z	P	P
y / x	N	Z	P

P	N		
Z	N	P	
N	P		
y / x	N	Z	P



Problème 25 Prototypes de commandes linguistiques approchées

Réaliser un logiciel acceptant des ordres de visualisation approchés. Les commandes seront soit des phrases, soit une suite de commandes à cliquer parmi un certain nombre de mots, de telle sorte que l'ordre donné soit du style "tracer / déplacer / supprimer / agrandir / réduire / déformer, (un peu / beaucoup), un / le, (petit / moyen / grand), rectangle / ellipse, (tres / peu, exentré), debout / couché, rouge / bleu / jaune ..., ((très, peu...) au dessus / sous / à droite, proche / loin... du GN) / en haut / bas, à droite / gauche". (GN = groupe nominal). On pourra simplifier l'exentricité comme une notion commune $1 - b/a$ aux rectangles et ellipses (0 pour le cercle et le carré).

Formaliser la syntaxe de ces phrases afin de n'accepter que celles qui ont un sens.

Donner une représentation floue des prédicats unaires "petit"... et binaires "loin de" ..., des modificateurs "très", "peu" ... Tout est relatif à la taille de la fenêtre de démonstration.

Le programme doit de plus pouvoir traiter une donnée par défaut et modifier l'interprétation des ordres donnés de telle façon que chaque figure soit entièrement incluse dans l'écran et qu'elles soient deux à deux disjointes. Il est permis d'introduire d'autres spécificités.

Problème 26 Réalisation d'un contrôleur neuro-flou

Devant un système présentant un tel caractère expérimental, il faut des outils permettant de conclure rapidement sur les résultats, aussi va-t-on utiliser une visualisation, entre 5 et 10 lignes de niveaux pour la fonction principale $x, y \rightarrow u$ comme pour les "et".

Programmer l'ensemble pour avoir un réseau global R, après expérimentation sur une fonction $x, y \rightarrow u$ obtenue au seuil près sur quelques exemples, (on pourra prendre plusieurs corps de règles initiaux).

Préparer une interface présentant clairement pour un corps de règles initial, au bout d'un certain nombre de rétropropagations, le graphique des prédicats modifiés, les différentes fonctions "et" par leurs lignes de niveaux sur $[-1,1]^2$, l'allure de la fonction principale et l'état des règles par une table des w_i .