LES ENSEMBLES FLOUS AU JEU DE GO

Bruno BOUZY

LAFORIA-IBP, Université Paris VI
Tour 46 2ème étage
4, place Jussieu 75252 PARIS
Tél.: 44 27 70 10 - Fax: 44 27 70 00
e-mail : bouzy@laforia.ibp.fr

RESUME

Les transformations sur des ensembles flous sont adaptées en analyse d'image quand les régions ne peuvent pas être définies de manière nette et précise. Le jeu de Go, par sa nature visuelle, est dans ce cas. Cet article présente l'apport d'une formalisation de ce domaine avec des ensembles flous. Les meilleurs programmes de Go ont un niveau très moyen à cause de l'explosion combinatoire. Pour évaluer une position, ils utilisent tous des concepts de connexion, d'influence et de territoire. Ces concepts, contenant parfois déjà implicitement ou partiellement des notions floues, sont présentés dans une première partie. Ensuite, ils sont explicitement formalisés à l'aide de connexion floue et de morphologie mathématique floue. L'apport de ces outils flous est étudié au travers des résultats d'un programme de Go et au travers d'une comparaison avec ce que reconnaissent les joueurs humains.

1. Introduction

Les ensembles flous sont souvent une bonne représentation en analyse d'image pour des problèmes de segmentation, de classification, de fusion de données, quand les régions ou les classes ne peuvent pas être définies de manière nette et précise. Leur manipulation, en gardant le caractère flou, permet de traiter des données incertaines, redondantes ou complémentaires, en repoussant la décision binaire le plus tard possible. Ces manipulations s'expriment comme des transformations sur ou entre des ensembles flous. Des transformations ensemblistes [7], géométriques [5], topologiques [1] ou morphologiques [2] ont déjà été développées.

Le but de cet article est de montrer une illustration de quelques unes de ces transformations - morphologiques ou topologiques - appliquées au jeu de Go. Par exemple, sur la figure 1, les joueurs de Go reconnaissent des "groupes", des "territoires" et de "l'influence". Nous allons montrer comment les ensembles flous permettent de modéliser ces objets. Nous discuterons de leur pertinence de deux manières : utilisés au sein du programme de Go INDIGO, développé pendant notre thèse [4], et comparés aux objets reconnus par des joueurs humains.

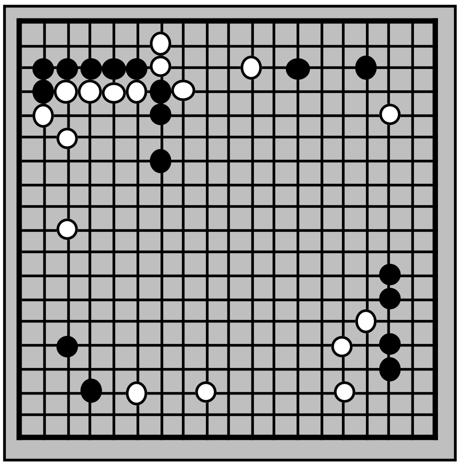

figure 1

Dans la première partie, nous présentons les aspects de la programmation du jeu de Go qui peuvent (mais ne le sont pas) être mis en relation avec la théorie des ensembles flous. Dans la seconde partie, nous explicitons ces relations en exhibant des fonctions d'appartenance d'ensembles flous, puis nous discutons de leur validité au sein d'un programme de Go et par comparaison avec les résultats des joueurs humains.

Cet article intéresse à la fois les théoriciens sur les ensembles flous, qui y verront une illustration de leur théorie au jeu de Go, et les programmeurs de Go, qui y trouveront des informations que nous espérons utiles pour améliorer leurs programmes.

2. L'état de l'art de la programmation du jeu de Go

Nous commençons cette partie par un bref rappel sur le jeu de Go et ses règles. Ensuite, nous présentons l'état de l'art de la programmation du jeu de Go au travers de trois concepts fondamentaux utilisés par les programmes de Go : la connexité, l'"influence" et le "territoire".

2.1. Rappel sur le jeu de Go

Le jeu de Go est un jeu oriental agé de plus de 4000 ans. C'est un jeu à deux joueurs (Noir et Blanc) à information complète qui se joue sur un damier, appelé goban, où les joueurs posent des pierres noires et blanches à tour de rôle. Au départ le goban est vide. La partie s'arrête lorsque les deux joueurs ont contrôlé tout le goban. Celui qui contrôle le plus de cases, appelées intersections, gagne la partie. Pour contrôler une intersection, un joueur peut l'occuper ou faire en sorte que si l'adversaire y joue, il sera irrémédiablement capturé. On capture une chaîne de pierres en supprimant les intersections vides adjacentes de la chaîne, appelées les libertés. Une description formelle de la règle du jeu figure dans [4].

2.2. La programmation du jeu de Go

En dépit de l'effort consacré à la programmation du jeu de Go depuis les années soixante, les meilleurs logiciels jouant une partie complète ont le niveau d'un joueur très moyen : 10ème kyu[1]. Le grand nombre de coups possibles associés à une position - déclinant linérairement de 361 au début de la partie à 0 à la fin - et le grand nombre de coups d'une partie - 300 environ - constituent la première explication : la taille de l'ensemble des états, 10^{171}, et la puissance des machines actuelles rend impossible l'utilisation de la force brute utilisée aux Echecs[2]. Evidemment, cette explication est approximative car elle tendrait à dire que la programmation du jeu de Go sur des damiers suffisament petits (9-9) aboutirait à des résultats comparables à ceux obtenus aux Echecs. Ce qui n'est pas le cas! En fait, l'évaluation d'une position au Go est complexe. Personne n'a trouvé comment définir une fonction d'évaluation suffisamment juste, précise et calculable en un temps raisonnable. Nous allons montrer dans les paragraphes suivants quels sont les concepts utilisés par les programmes actuels pour évaluer une position : les groupes, les territoires et l'influence.

[1]Un novice est 30 ème kyu, un débutant confirmé est 20 ème kyu, un joueur fort est 1er kyu, puis 1er dan. Les meilleurs joueurs mondiaux sont 9ème dan.

[2]La taille de l'ensemble des états est évaluée à 10^{43} aux Echecs.

2.2.1. Préliminaire à propos des ensembles classiques

Soit E l'ensemble des intersections du goban. Chaque intersection i possède un ensemble d'intersections voisines que nous appelons v(i). Une intersection peut être vide, noire ou blanche. Nous appelons E_N (resp. E_B et E_V) l'ensemble des intersections noires (resp. blanches et vides).

2.2.2. Les ensembles connexes

La règle du jeu de Go associe les pierres voisines de même couleur en chaînes de pierres. Les chaînes sont les composantes connexes des ensembles E_B et E_N. La figure 2a montre les 13 chaînes blanches et les 9 chaînes noires présentes sur la position de la figure 1.

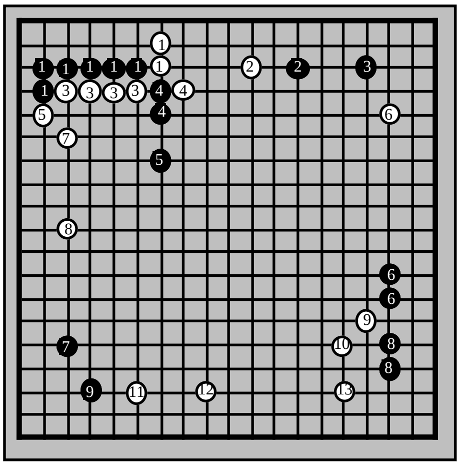

figure 2a

A partir des chaînes de pierres, les joueurs de Go construisent les "groupes" de pierres. Les critères utilisés par l'être humain sont mal connus : la connexion tactique, la proximité des pierres, leur coopération ? Il semblerait que ces critères interviennent à des degrés différents. Mais la validation de ces critères est trop complexe et sort du cadre de cet article. Toutefois, sur la figure 1, les joueurs humains reconnaissent les ensembles connexes, appelés "groupes" de pierres dans le jargon du Go, comme indiqué par la figure 2c.

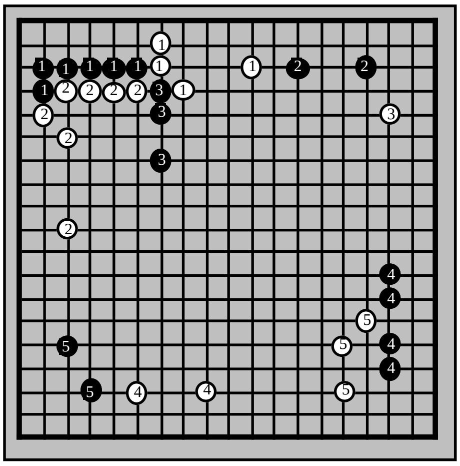

figure 2c

Dans les programmes de Go, les groupes de pierres sont définis par des connexions tactiques. La figure 2b montre les groupes ainsi définis : 8 groupes blancs et 6 groupes noirs. En pratique, cette définition donne les meilleurs résultats dans un programme de Go, même si cela ne correspond pas à ce qu'un joueur humain reconnaît. Nous verrons dans la deuxième partie de cet article pourquoi cette différence existe.

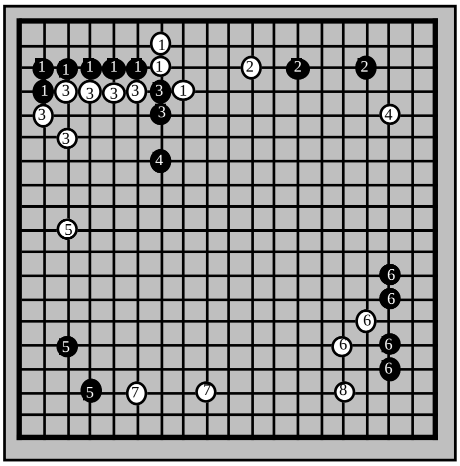

figure 2b

2.2.3. Le modèle de Zobrist pour l'influence

Zobrist a montré l'interêt des modèles de diffusion de potentiel au début des années 70. Cela constitue une première utilisation des ensembles flous sans en porter le nom. Zobrist définit une fonction d'influence dans [8] de la manière suivante. Il affecte des valeurs numériques sur les intersections : +50 sur les éléments de E_N, -50 sur les éléments de E_B et 0 sur ceux de E_V. Puis chaque intersection positive (resp. négative) émet +1 (resp. -1) sur chacune de ses voisines. Ces nombres sont additionnés et cette procédure répétée quatre fois. La figure 3 montre l'influence noire (resp. blanche) avec des carrés gris foncés (resp. clairs) qui correspond à la position de la figure a.

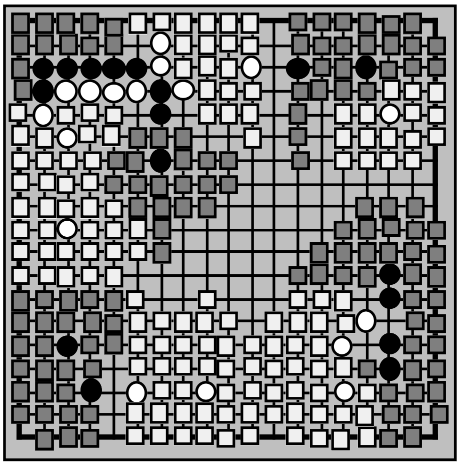

figure 3

La littérature sur la programmation du jeu de Go ne contient pas d'autre référence significative sur ce point. On peut penser que les programmes de Go utilisent toujours ce type de technique de diffusion de potentiel pour modéliser l'influence [3] et même peut-être le territoire. Les joueurs humains interrogés sur ce type de positions sont d'accord avec cette interprétation du goban du point de vue de l'influence.

2.2.4. La morphologie mathématique classique

Dans notre thèse [4], nous avons montré comment il est possible de distinguer le territoire de l'influence en utilisant la fermeture morphologique plutôt que la dilatation morphologique [6]. Sur la figure 4, nous montrons un ensemble de pierres noires de départ (à gauche), sur lequel on applique l'opérateur de dilatation morphologique pour obtenir l'ensemble dilaté (au milieu), ou bien l'opérateur de fermeture morphologique pour obtenir l'ensemble fermé (à droite). Le territoire correspondant au groupe noir du diagramme de gauche est donc l'ensemble des intersections indiquées par des ronds gris foncés sur le diagramme de droite. L'influence correspondant au groupe noir du diagramme de gauche est l'ensemble des intersections indiquées par des ronds gris foncés sur le diagramme du milieu.

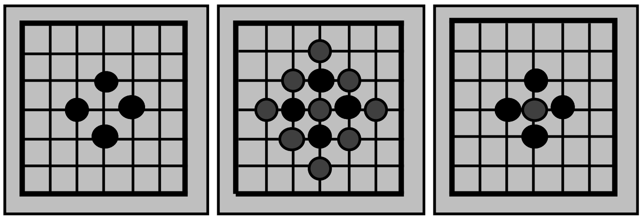

figure 4

Plus formellement et avec les définitions précédentes, si A est un ensemble d'intersections, on a :

D(A) = {y dans E | il existe x dans A : y appartient à v(x)} est l'ensemble dilaté de A.

E(A) = {y dans A | pour tout x hors de A, y n'appartient pas à v(x)} est l'ensemble érodé de A.

F(A) = E(D(A)) est la fermeture de A.

Ce qui coïncide avec les définitions classiques de morphologie mathématique de [6].

I(A) = D(A)-A est l'influence de A.

T(A) = F(A)-A est le territoire de A.

Ce qui coïncide avec le jargon des joueurs de Go.

Pour reconnaître des territoires de plus grande taille, nous avons défini une famille d'opérateurs morphologiques X qui dilatent n fois puis érodent m fois :

$$X(m,n) = E^m \circ D^n$$

Les figures 5a, 5b et 5c représentent un "pattern" fréquent dans les parties de Go, à savoir deux pierres de même couleur, sur la 3ème ligne, distantes de trois intersections. Les rectangles gris foncés de la figure 5a montrent le territoire reconnu avec l'opérateur X(3,3).

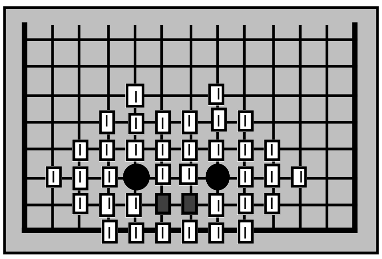

figure 5a

Les rectangles gris foncés de la figure 5b montrent le territoire reconnu avec l'opérateur X(2,3).

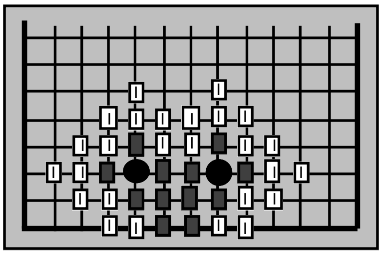

figure 5b

Les rectangles gris foncés de la figure 5c montrent le territoire reconnu par un joueur humain.

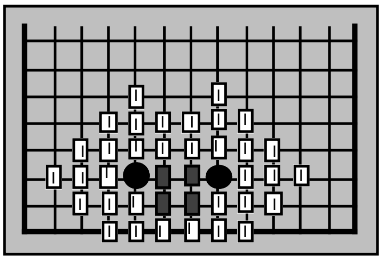

figure 5c

Le lecteur constatera que X(3,3) reconnaît un territoire trop petit (figure 5a) et que X(2,3) reconnaît un territoire trop grand (figure 5b) par rapport à ce que reconnaît un joueur humain (figure 5c). Le lecteur comprendra donc que la morphologie mathématique symbolique est un outil utile pour reconnaitre les territoires au Go, mais qu'il est trop grossier pour donner des résultats aussi fins que ceux des joueurs humains. Nous verrons dans la prochaine partie comment une fuzzification permet de reconnaître des territoires flous correspondant précisément à ceux reconnus par les joueurs humains.

2.3. Résumé de la première partie

Finalement, la programmation du jeu de Go utilise différentes techniques pour reconnaître les groupes, l'influence et le territoire. La connexité est tactique, sans notion de flou. L'influence, modélisée par Zobrist, ressemble déjà aux ensembles flous sans en porter le nom. Les territoires sont reconnus grossièrement avec la morphologie mathématique symbolique.

3. L'apport d'une formalisation dans le cadre des ensembles flous

Dans cette partie, nous montrons quels avantages et inconvénients apporte une formalisation à l'aide d'ensembles flous pour chacun des trois concepts fondamentaux au jeu de Go : groupe, influence et territoire. D'abord, nous faisons cadrer le modèle de Zobrist sur la théorie des ensembles flous. Nous discutons de l'utilité de la connexion floue pour modéliser les groupes au Go. Ensuite, nous montrons comment la morphologie mathématique floue affine la définition des territoires au Go, donnée dans la partie précédente.

3.1. Le modèle de Zobrist est un modèle de dilatation floue

Un ensemble flou F est défini par une fonction d'appartenance μ, qui associe à tout élément x de l'ensemble de considéré G, une valeur réelle dans l'intervalle [0,1]. μ remplace la notion de fonction caractéristique dans la théorie classique des ensembles. Dans la partie précédente nous avons rappelé comment Zobrist affecte des valeurs numériques z(x) à chaque intersection x de G. Il est possible de définir simplement l'ensemble I_N (resp. I_B) avec la fonction d'appartenance $\mu_{i,n}$ (resp. $\mu_{i,b}$) suivante. Soit zm_n (resp zm_b) la plus grande (resp. petite) valeur prise sur E_V. Il suffit de poser $\mu_{i,n}(x) = z(x)/zm_n$ (resp. $\mu_{i,b}(x) = z(x)/zm_b$) pour que $\mu_{i,n}$ (resp. $\mu_{i,b}$) soit la fonction d'appartenance d'un ensemble flou I_N (resp. I_B). Cette formalisation ne change rien au fond du problème mais elle permet de faire rentrer le modèle de Zobrist comme illustration de la théorie des ensembles flous.

3.2. La connexité floue

Nous avons vu que les programmes de Go utilisent des connexions tactiques pour identifier les groupes de pierres. Nous nous sommes demandés si cette définition des groupes pouvait être améliorée avec des ensembles flous. En effet, la finesse de la reconnaissance des groupes par les joueurs humains laisse penser que ceux-ci utilisent d'autres mécanismes que la seule connexion tactique. Dans le programme INDIGO, nous avons essayé de définir les groupes comme étant les composantes connexes des ensembles flous I_N et I_B. Nous avons utilisé des degrés de connexité définis par [5]. Pour deux points x et y d'un ensemble flou caractérisé par une fonction d'appartenance μ, le degré de connexité entre x et y est défini par :

$$c_\mu(x,y) = \max[\min(\mu(z), \text{ pour } z \text{ dans } l), \text{ pour } l \text{ dans } L(x,y)]$$

où L(x,y) est l'ensemble des chemins reliant x et y au sens de la connexité classique définie sur G[1]. Il est alors possible de définir la composante connexe floue d'un point x par la fonction d'appartenance π_x suivante : $\pi_x(y) = c_\mu(x,y)$. Les propriétés du degré de connexité sont décrites dans [1].

Utiliser ce type de connexité dans un programme de Go, est actuellement inadapté. En effet, nous avons fait jouer une version de INDIGO utilisant la connexion floue contre une version de INDIGO utilisant la connexion tactique. La version "tactique" était beaucoup plus forte que la version "floue". La version "tactique" coupait en pièces les groupes de la version "floue". Toutefois, cela ne signifie pas qu'une version floue doive être abandonnée. En effet, le paradoxe est le suivant : notre modélisation des groupes avec des ensembles flous correspond aux groupes reconnus par les humains (!) mais donne de moins bons résultats dans un programme de Go (dont l'objectif est de gagner contre d'autres programmes de Go).

3.3. La morphologie mathématique floue

Dans ce paragraphe, nous montrons comment les limitations de la reconnaissance des territoires avec la fermeture morphologique symbolique sont dépassées grâce à une fuzzification qui permet alors de reconnaître les territoires du Go à partir de fermetures morphologiques floues.

3.3.1. Description de notre algorithme

3.3.1.1. Initialisation

Notre algorithme affecte +128 (resp. -128) aux intersections noires (resp. blanches) et 0 aux intersections vides et il effectue autant de dilatations et d'érosions qu'il est demandé. Pour reconnaître des territoires à échelle N, il faut N dilatations et 1+N(N-1) érosions[2].

---

[1] La connexité de la règle du jeu qui détermine les chaînes de pierres (cf. ci-dessus).

[2] Le lecteur comprendra que, si E est le nombre d'érosions et D le nombre de dilatations, la relation E=D(D-1)+1 doit être vérifiée pour que l'opérateur X(E,D) soit neutre sur une pierre seule, placée au centre d'un goban.

3.3.1.2. Dilatation

Pour chaque intersection du goban, si une intersection est positive (resp. négative) ou nulle et n'est pas voisine d'une intersection négative (resp. positive), il additionne (resp. soustrait) le nombre de voisines positives (resp. négative) à l'intersection.

3.3.1.3. Erosion

Pour chaque intersection du goban, si une intersection est positive (resp. négative), il soustrait (resp. additionne) le nombre de voisines négatives (resp. positives) ou nulles à l'intersection en vérifiant qu'elle reste positive (resp. négative) ou nulle.

3.3.2. Exemple

Reprenons la position des figures 5a, 5b et 5c de la première partie (la ligne du bas est un bord du goban) :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	128	0	0	128	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Après trois dilatations :

0	0	0	1	0	0	1	0	0	0
0	0	2	2	2	2	2	2	0	0
0	2	4	6	5	5	6	4	2	0
1	2	6	136	7	7	136	6	2	1
0	2	4	6	5	5	6	4	2	0
0	0	2	2	2	2	2	2	0	0

Après sept érosions supplémentaires :

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	128	6	6	128	0	0	0
0	0	0	0	4	4	0	0	0	0
0	0	0	0	0	0	0	0	0	0

Le lecteur comparera cette figure à la figure 5c et comprendra que la reconnaissance des territoires est correcte.

Notre algorithme affecte une valeur numérique t(x) à chaque intersection x de E. Il est possible de définir l'ensemble T_N (resp. T_B) avec la fonction d'appartenance $\mu_{t,n}$ (resp. $\mu_{t,b}$) suivante. Soit tm_n (resp tm_b) la plus grande (resp. petite) valeur prise sur E_V. Il suffit de poser $\mu_{t,n}(x) = t(x)/tm_n$ (resp. $\mu_{t,b}(x) = t(x)/tm_b$) pour que $\mu_{t,n}$ (resp. $\mu_{t,b}$) soit la fonction d'appartenance d'un ensemble flou T_N (resp. T_B).

Les carrés gris foncés (resp. clairs) de la figure 6 montrent les supports des territoires noirs (resp. blancs) reconnus par l'opérateur flou X(21,5) appliqué sur la position de la figure 1.

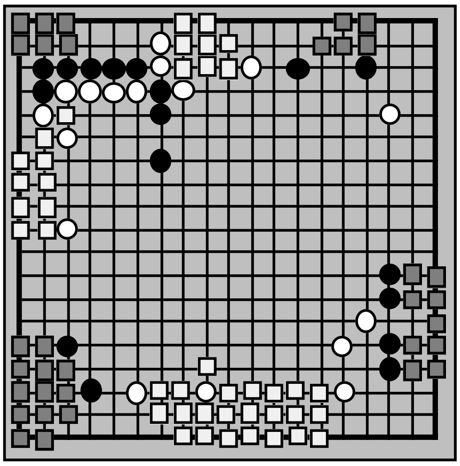

figure 6

Ces territoires correspondent étonnament bien aux territoires reconnus par les joueurs humains. Dans un programme de Go, l'utilisation des territoires doit être faite avec précautions. En effet, un territoire est un ensemble dans lequel toute intrusion de l'adversaire se termine logiquement par sa mort. Donc, les territoires reconnus par le programme doivent correspondre aux capacités du programme pour tuer un intrus potentiel. Le programme INDIGO utilise des territoires reconnus par l'opérateur X(7,3) uniquement. S'il utilisait les opérateurs X(13,4) et X(21,5), il ne serait pas capable de défendre les territoires qu'il reconnaitrait.

On retrouve à nouveau le paradoxe découvert précédemment pour les groupes que l'on peut résumer ainsi : la reconnaissance floue des objets présents sur un goban coïncide avec celle des joueurs humains mais l'utilisation de ces objets par un programme de Go échoue partiellement (pour les groupes ou les gros territoires).

4. Conclusion

Nous avons discuté de la reconnaissance floue des groupes, territoires et influences par rapport à l'échelle humaine et de leur utilisation dans le programme de Go INDIGO. Nous rappelons synthétiquement ces discussions :

Pour les groupes, les ensembles connexes flous correspondent à ce qu'un joueur humain reconnaît mais ils ne sont pas utilisables dans un programme de Go dont l'objectif est de battre d'autres programmes de Go. Dans INDIGO, nous avons préféré définir les groupes avec des connexions tactiques plûtot que floues.

Pour l'influence, les ensembles dilatés flous correspondent parfaitement à ce qu'un joueur humain reconnaît mais le modèle de Zobrist, qui existe depuis 1969, marche aussi. Il contient de la modélisation floue sans en porter le nom. Notre apport a seulement consisté à intégrer ce modèle dans la modélisation floue.

Pour les territoires, les ensembles fermés flous correspondent parfaitement à ce qu'un joueur humain reconnaît mais leur utilisation dans un programme de Go doit se limiter aux petits territoires. Notre apport est significatif et car aucune description de territoire par fermeture morphologique floue n'existait avant.

Le lecteur se demandera sûrement pourquoi une meilleure compréhension d'une situation grâce aux ensembles flous ne permet pas d'atteindre un programme meilleur que les “traditionnels”. La réponse tient au fait que la reconnaissance floue des objets est une approximation qui simule correctement la reconnaissance faite par des joueurs humains mais que celle-ci est découplée de l'utilisation de ces objets. Pour obtenir un programme basé sur les ensembles flous meilleur que les programmes traditionnels, il resterait à trouver un mécanisme d'utilisation de ces objets, parfaitement couplé avec le processus de reconnaissance flou.

Nous espérons que les chercheurs en logique floue et les concepteurs de programmes de Go, successeurs de Zadeh ou de Zobrist depuis trente ans, auront trouvé, une illustration de leur théorie pour les premiers, et des idées intéressantes pour les seconds. Nous sommes sommes évidemment ouverts aux remarques et suggestions que pourraient faire l'une ou l'autre de ces deux communautés de chercheurs.

REFERENCES

[1] I. Bloch - Connexité floue et morphologie mathématique - Rapport interne Télécom Paris, 92D001 - Janvier 1992

[2] I. Bloch, H. Maître - Ensembles flous et morphologie mathématique - Rapport interne Télécom Paris, 92D007 - Mars 1992

[3] M. Boon - Overzicht van de ontwikkeling van een Go spelend programma - Afstudeer scriptie informatica onder begeleiding van prof. J. Bergstra - 1991

[4] B. Bouzy - Modélisation cognitive du joueur de Go - Thèse de l'Université Paris 6, LAFORIA-IBP - Janvier 1995

[5] A. Rosenfeld - The Fuzzy Geometry of Image Subsets - Pattern Recognition Letters 2, pp. 311-317 - 1984

[6] J. Serra - Image Analysis and Mathematical Morphology - Academic Press - London - 1982

[7] L.A. Zadeh - Fuzzy Sets - Inform. and Control 8, pp. 338-353 - 1965

[8] A. Zobrist, A model of visual organization for the game of Go, Proceedings AFIPS 34 103-112, 1969