Utilisation d'une carte "cognitive" pour le problème de la sélection de l'action dans un cadre multi-motivationnel

S. Leprêtre & P. Gaussier & A. Revel , ENSEA ETIS
6 Av du Ponceau, 95014 Cergy Pontoise Cedex
e-mail : lepretre@ensea.fr

1 Introduction

Notre but est de développer des architectures neuronales permettant à un robot autonome de "survivre" dans un environnement a priori inconnu et/ou imprévisible (approche *animat* [6]). L'Animat doit être capable de sélectionner à chaque instant la ou les actions lui permettant de maintenir un certain nombre de variables essentielles à l'intérieur d'une zone de confort prédéterminée. On sait depuis Tolman [8] que les mammifères (les rats par ex.) sont capables d'apprendre une carte cognitive (carte "topologique") de leur environnement même en l'absence de toute récompense (apprentissage latent). Ce type de carte est ensuite utilisé par l'animal pour rejoindre rapidement un but (nourriture, eau...). Dans cet article, nous allons d'abord décrire un réseau de neurones permettant l'apprentissage et l'utilisation d'une carte cognitive pour la navigation visuelle dans un environnement complexe (environnement ouvert). Une simulation basée sur un algorithme de navigation déjà testé sur un robot réel sera ensuite présentée. Nous montrerons que notre algorithme peut aussi être utilisé pour permettre à notre animat de choisir entre plusieurs buts, de gérer la disparition et l'apparition de buts et finalement de résoudre de manière relativement satisfaisante le dilemme exploration/exploitation des ressources.

2 Architecture du système

L'architecture globale du système comprend un mécanisme très simple d'évitement d'obstacles (inspiré des travaux de Braitenberg [2]) associé à une carte cognitive non cartésienne permettant de mémoriser la position des lieux appris et leurs relations topologiques (fig 1 et 2).

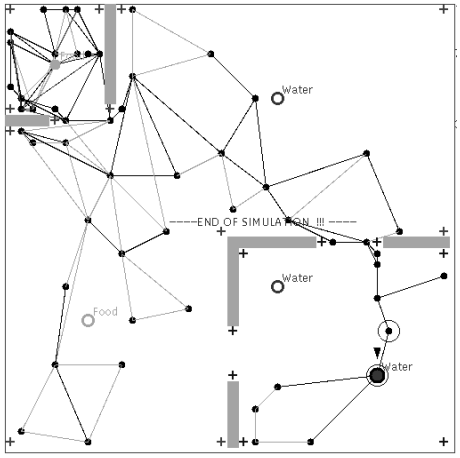

Figure 1: Exemple d'environnement de test. L'animat (le triangle) détecte les obstacles grâce à la vision et/ou à un capteur de proximité. Il ne trouve les sources de nourriture et d'eau qu'en leur passant juste dessus. Les cercles pleins représentent les lieux appris. Les liens entre ces noeuds sont associés à des transitions apprises lorsque l'animat chemine du voisinage d'un noeud au noeud suivant(sous-buts).

L'apprentissage de la carte s'effectue en continu. Les liaisons entre chacun des noeuds du graphe sont renforcées (renforcement de type hebbien) pour des situations successivement reconnues. Soit w le poids d'un lien entre une entrée $\overline{E}$ (intégrée dans le temps) et la sortie S :

$$\frac{dw}{dt} = -\lambda.w + (A + \frac{dR}{dt}).(1 - w).\overline{E}.S \qquad (1)$$

Dans nos simulations λ est très faible et permet l'oubli dans le temps des poids des liens peu exploités. Le terme $\frac{dR}{dt}$ correspond

(négatif ou positif) qui apparait lorsque l'animat entre ou sort d'une zone "difficile" (terrain accidenté par ex.). Il permet de moduler la force du poids créé et par la même, la perception des distances entre les lieux appris ($A = 1$ dans nos simulations).
La proximité temporelle étant équivalente à une proximité spatiale, le système se crée une représentation topologique de l'environnement (fig 1). La planification exploite cette information spatiale pour permettre à l'animat de rejoindre un but particulier. Pour cela, au niveau but (fig 2), les motivations activent directement les neurones buts à atteindre (l'intensité étant proportionnelle à la motivation). Ces activités sont alors diffusées de neurones en neurones par l'intermédiaire des liaisons synaptiques. On utilise l'algorithme de diffusion des motivations des buts suivant : [7]

- *N_{i0} est le neurone but qui s'active en cas de motivation. x_{i0} est son activité.*
- $x_{i0} \leftarrow 1$ et, $x_i \leftarrow 0\ \forall i \neq i_0$
- *TANTQUE le réseau n'est pas stable, FAIRE:* $\forall i, x_j \leftarrow \max(W_{ij}.x_i)$

Chacune des liaisons a une valeur inférieure à 1 (CNS pour assurer la convergence de l'algorithme). Sur la figure 2 on présente un exemple de diffusion à partir d'une motivation A . L'algorithme permet de trouver le plus court chemin dans un graphe en remontant le gradient d'intensité des noeuds jusqu'au but. Il est équivalent à celui de Bellman-Ford [1][7].

3 Navigation avec des amers

Toutes nos simulations sont basées sur un modèle de localisation visuelle qui donne de très bon résultats sur un robot réel [4]. Ce mécanisme de reconnaissance des lieux tente de rendre compte de l'activité des "cellules de lieux" mises en évidence par les neurobiologistes dans l'hippocampe des rats [5]. L'activité de ces cellules dépend de l'orientation des amers visuels entourant la zone d'expérimentation [3].

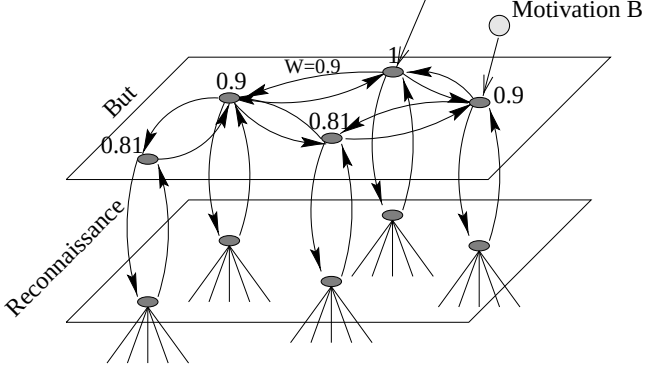

Figure 2: Architecture globale du système. Le niveau reconnaissance permet d'identifier des situations quand le robot arrive à proximité d'un lieu mémorisé. Ces situations sont directement reliées au niveau but qui va permettre de planifier un chemin d'attracteurs en attracteurs jusqu'à l'objectif. Lorsqu'une motivation A active un but, un mécanisme de rétropropagation de l'information se met en oeuvre vers tous les noeuds du graphe de planification (tous les poids vallent 0.9).

L'activité d'une de nos "cellules de lieu" s'écrit :

$$act_n = 1 - \frac{\sum_{i=1}^{N_n^a} v_{n,i}^a . f\left(\left|\theta_{n,i}^a - \theta_i\right|, v_i\right)}{\pi N_n^a}$$
$$avec \quad f(d, v_i) = \begin{cases} d \ \ si \ \ v_i = 1 \\ \pi \ \ si \ \ v_i = 0 \end{cases} \qquad (2)$$

N_n^a est le nombre d'amers visibles du lieu appris n (ou cellule n). $v_{n,i}^a$ et , v_i vallent 1 lorsque l'amer i est visible, pour respectivement, la position n apprise et la position courante (0 sinon). $\theta_{n,i}^a$ représente l'angle sous lequel l'amer i est vu depuis le lieu n. L'équation 2, donne une activité act_n croissante qui tend vers 1 lorsque les angles θ_i à une position, se rapprochent des $\theta_{n,i}^a$ mémorisés. En choisissant de remonter les lignes de gradient de cette activité, il est possible de rejoindre le but, et cela, quelque soit la position de départ de notre robot (fig 3). Des bassins d'attraction se trouvent ainsi créés autour de chacun des lieux appris. Dans le cas simple où 2 buts (2 lieux) ont été appris, le robot tombe toujours dans le bassin d'attraction du lieu le plus proche de lui. En modulant l'activité d'une de ces cellules de lieux par un signal lié à une motivation particulière (boire, manger...), le système peut être forcé à rejoindre un but plutôt qu'un autre (la taille des bassins d'attraction est modifiée).

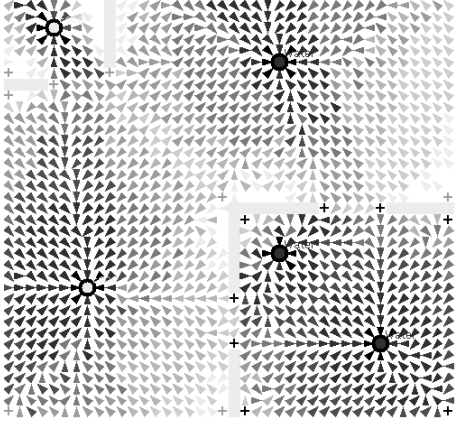

Figure 3: Attracteurs liés aux buts appris (les ronds) dans un exemple d'environnement. Les croix représentent des amers visibles en l'absence d'obstacles.

Lorsque le but n'est pas directement accessible, il est possible d'utiliser notre mécanisme de planification pour se diriger vers le lieu le plus actif dans le voisinage du lieu occupé. En remontant de manière réactive le gradient d'activité sur la carte cognitive (planification réactive) l'animat sélectionne de proche en proche les sous-buts à atteindre (fig 2). En même temps, il exploite l'information de gradient liée au niveau de reconnaissance du sous-but le plus activé pour s'en rapprocher. Les éventuels obstacles sont évités sans nécessité d'information de planification trop détaillée (l'évitement d'obstacles est prioritaire par rapport au mécanisme d'attraction vers un sous-but).

4 Résultats expérimentaux

Dans un environnement contenant plusieurs pièces ou obstacles, notre architecture neuronale permet à un animat d'apprendre à passer d'une pièce à une autre afin de trouver puis rejoindre un objectif (fig 1). Dans nos simulations (environnement continu), notre animat démarre de son nid et est tout le temps en déplacement (seule la direction de son déplacement est contrôlée). Il explore son environnement lorsqu'il n'a ni trop faim ni trop soif. Mais il peut retourner rapidement vers une source de nourriture ou d'eau déjà trouvée dès que le niveau de ses variables essentielles (ses besoins) le rend nécessaire. Au naissances sur l'environnement en apprenant un certain nombre de lieux et en créant (ou modifiant) les connexions entre les cellules associées à ces lieux. Un lieu est appris lorsque l'animat passe sur un lieu important pour sa survie : typiquement il peut rencontrer des points de nourriture et des points d'eau (futurs buts). Pour lui permettre, par la suite, de rejoindre ces buts, d'autres endroits sont aussi appris (les sous-buts). Ils correspondent aux situations suivantes :

- la fin d'un évitement d'obstacle. Par exemple, l'animat mémorise les passages entre deux pièces.
- des lieux globalement mals reconnus ($max(act_n) < val$, $val = 0.85$ dans nos simulations). Plus val est grand, plus il y aura de lieux appris. On obtient un pavage spatial relativement régulier de sous-buts dans les zones où les mêmes amers sont visibles.

L'ensemble des buts et sous-buts forment des attracteurs qui s'activent lorsque l'animat veut satisfaire un ou plusieurs de ses besoins. En vue de simuler le comportement de faim (n niveau de nutrition) et de soif (h niveau d'hydratation) d'un *animat*, nous avons employé les équations différentielles suivantes :

$$\begin{array}{l} \frac{dn}{dt} = -\alpha.n + Q_n + \mu.\delta h.n \\ \frac{dh}{dt} = -\alpha^{'}.h + Q_h + \mu^{'}.\delta n.h \\ avec : \\ Q_n = \beta.(n_{max} - n).\delta n - \gamma.(n - n_{min}).(1 - \delta n) \\ Q_h = \beta^{'}.(h_{max} - h).\delta h - \gamma^{'}.(h - h_{min}).(1 - \delta h) \end{array} \quad (3)$$

Elles régissent les motivations "aller se nourrir" et "aller boire" de l'agent. Dès que l'animat se trouve sur un lieu de ravitaillement, le niveau de la ressource correspondante est immédiatement augmenté d'une quantité (δn et δh valent 1, sinon 0). Les termes avec μ et $\mu^{'}$ prennent en compte, quantitativement, de légères intéractions du type "manger donne soif" et "boire diminue la faim". En outre, nous avons imposé au système que le niveau des variables essentielles soit maintenu entre 2 seuils définissant une zone de confort. Ainsi dès que le niveau d'une des variables

tion croît et active les buts associés (apparition d'attracteurs). Dans le cas inverse, au dessus du seuil haut, il y a inhibition des buts associés qui deviennent repousseurs.

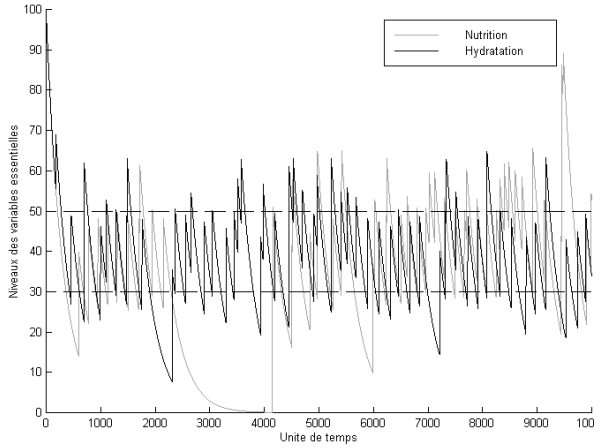

Figure 4: Evolution des 2 variables essentielles au cours d'une experience (réserves en eau et nourriture).

La figure 4 présente l'évolution des 2 variables essentielles au cours d'une expérience. Les pics indiquent les moments où l'animat ingère de la nourriture ou de l'eau. On peut constater que le système réussit relativement bien à maintenir ses deux variables internes dans la zone de confort. Un cycle d'aller et venues s'installe entre les 2 types de sources. Ces cycles ne s'inversent que lorsque le robot profitant d'une de ses périodes d'exploration découvre par hasard un nouveau but. Les différentes expériences avec ou sans obstacles montrent qu'après avoir trouvé des sources de nourriture et d'eau l'animat navigue correctement entre ces différents lieux (les niveaux des variables restent relativement longtemps dans la zone de confort).

5 Conclusion

Nous avons testé notre système sur d'autres types d'environnement de manière à mettre en lumière d'éventuels problèmes de choix. Par exemple, le robot peut trouver de la nourriture et de l'eau dans une branche d'un labyrinthe et seulement de l'eau dans une branche plus courte. Bien que notre animat ne sache pas explicitement "additionner" des motivations afin de favoriser des mouvements dans la direction des deux sources, nous avons constaté qu'après un certain temps, notre animat finissait par renforcer le poids des connexions représentant le meilleur chemin. Le renforcement des poids sur la carte cognitive est complètement similaire au renforcement des traces de phéromones utilisées par les fourmis pour marquer leur chemin. On pourrait ainsi voir notre modèle de carte cognitive et de planification comme une simple internalisation d'un mécanisme de navigation identique à celui de simples insectes. Nos prochains travaux concerneront la capacité à catégoriser les plans d'actions effectués de manière à pouvoir les hiérarchiser.

Remerciements: Ce projet est financé par un projet GIS sur la "Comparaison d'architectures de contrôle adaptatives pour le problème de la sélection de l'action", projet en collaboration avec AnimatLab, LISC, RFIA.

Bibliographie

[1] R.E. Bellman. On a routing problem. *Quarterly of Applied Mathematics*, (16):87–90, 1958.

[2] V. Braitenberg. *Vehicles : Experiments in Synthetic Psychology.* Bradford Books, Cambridge, 1984.

[3] C. Collison D. S. Olton and M. A. Werz. *Neural Connections, Mental Computation*, chapter Computations the hippocampus might perform, pages 225–284. MIT Press, 1989.

[4] P. Gaussier, C. Joulain, S. Zrehen, J.P. Banquet, and A. Revel. Visual navigation in an open environement without map. In *International Conference on Intelligent Robots and Systems - IROS'97*, Grenoble, France, September 1997. IEEE/RSJ.

[5] J.O'Keefe and N. Nadel. *The hippocampus as a cognitive map.* Clarendon Press, Oxford, 1978.

[6] J.A. Meyer and S.W. Wilson. From animals to animats. In MIT Press, editor, *First International Conference on Simulation of Adaptive Behavior.* Bardford Books, 1991.

[7] Arnaud REVEL. *Contrôle d'un robot mobile autonome par approche neuro-mimétique.* Doctorat de traitement de l'image et du signal, Université de Cergy-Pontoise, Novembre 1997.

[8] E.C. Tolman. Cognitive maps in rats and men. *The Psychological Review*, 55(4), 1948.