

07/10/2021

**Sujet TP 2 SMA**  
**A rendre au plus tard le 07/12/2021**

Soit un environnement représenté par une grille (matrice  $N \times M$ ) sur lequel on dispose initialement aléatoirement, des objets identifiés par des lettres A ou B en quantités respectives  $n_A$ ,  $n_B$  et un nombre  $nb_{Agents}$  d'agents aussi répartis aléatoirement. On souhaite programmer à l'aide de règles de comportements simples, le comportement de tri collectif, dont voici les principes de base :

- Un agent se déplace aléatoirement d'un pas (exprimé en nombre  $i$  ( $i \geq 1$ ) de cases, dans l'environnement dans les 8 directions autour de la case. le nombre  $i$  est un paramètre que l'on fixe selon l'étendue de l'environnement et le nombre d'agents disponibles.
- La prise ou le dépôt d'un objet dans l'environnement sont conditionnés respectivement par une probabilité de prise et une probabilité de dépôt qui s'exprime respectivement par :  $P_{prise} = (k^+ / (k^+ + f))^2$  et  $P_{dépôt} = (f / (k^+ + f))^2$  avec :  $k^+$  et  $k^-$  des constantes et  $f$  représentant la proportion d'objet de même type A ou B dans l'environnement immédiat (voisinage de l'agent). Le voisinage de l'agent est défini par les cases atteignables en 1 pas de temps par l'agent dans les 8 directions.

Pour calculer  $f$ , l'agent a une mémoire à court terme des objets déjà rencontrés sur les  $t$  derniers pas. On représente cette mémoire par une chaîne de  $t$  (taille de la mémoire) caractères correspondants au  $t$  emplacements déjà visités. Le caractère stocké correspondant au label de l'objet (A, B ou 0 si la case ne contient pas d'objet). Exemple : la chaîne 00A0ABBOOA dans le cas  $t=10$ . Dans ce cas  $f_A = 3/10$ ,  $f_B = 2/10$ . <sup>1</sup>(Pour plus de détails cf. [1])

**Réalisation version 1**

1- Programmer ce système en considérant les valeurs suivantes dans un premier temps : Une grille  $50 \times 50$ ,  $i=1$ , 20 agents,  $k_+=0,1$ ,  $k_-=0,3$ , 200 objets de chaque type A ou B.

**NB** : Pour rappel, le programme d'un agent consiste en une boucle perception(), action(). La perception, permet à l'agent de récupérer les informations de l'environnement et l'action, dans le cas du sujet est soit : le déplacement (aléatoire) d'un pas, la prise (suivant probabilité) d'un objet ou le dépôt (probabiliste) d'un objet.

2- Une variante intéressante de ce modèle consiste à introduire un pourcentage d'erreur  $e$  dans la reconnaissance des objets. Dans ce dernier cas, la proportion d'objets d'un type donné (ex A) dans la mémoire sera calculée en intégrant le taux d'erreur sur les objets de l'autre type (ici B). Par exemple pour la chaîne : 00A0ABBOOA on aura :

---

<sup>1</sup> J.L Denebourg & al., The dynamics of Collective sorting Robot-Like Ant and Ant-Like Robot

$f_A = (3+2e)/10$ , autrement dit, le nombre d'objets de type A additionné au taux d'erreur multiplié par le nombre d'objets de type B divisé par la taille de la mémoire considérée (t ici égale à 10).

**Note :** le cas de l'erreur est dans la perception (plus l'erreur est grande, plus il n'y a pas de discrimination de cluster)

### **Réalisation Version 2**

On souhaite ajouter un nouveau comportement collectif aux robots. Pour cela on ajoute  $n_C$  objets de type C, qui nécessitent la collaboration de 2 robots pour être portés. Dans ce cas, quand un robot tombe sur un objet C dans son environnement, il émet un signal (appel à l'aide) qui se code par une information propagée dans son voisinage (8 cases autour) sur une distance de diffusion du signal  $d_s$  (en nombre de cases : ex. les 8 cases autour de l'agent sur 1 distance de  $d_s=2$  : pour un agent se trouvant à l'emplacement  $(i,j)$ , on aura toutes les cases se trouvant à  $i+k, j+l$ , avec  $0 \leq k \leq 2$ , et  $0 \leq l \leq 2$ ). Il faut noter que l'intensité du signal se réduit au fur et à mesure qu'on s'éloigne de la case où se trouve l'agent d'un taux  $(1/d_s)$ . Exemple si l'intensité du signal est égale à  $S$ , l'atténuation du signal est :  $S-S/d_s$ , puis  $(S-S/d_s)-S/d_s$ , etc. On propose de gérer le signal comme une sorte de phéromone qui s'évapore d'un taux  $r$ , à définir (valeur faible) et que le robot qui appelle à l'aide peut renouveler son appel (renforcer l'intensité du signal) ou abandonner la tâche au bout d'un certain temps, pour éviter des situations d'inter-blocage (les robots sont bloqués à attendre de l'aide, alors qu'il pourraient débloquer la situation en s'aidant mutuellement l'un après l'autre).

Proposer une modélisation de ces aspects et une implémentation pour permettre le Tri (sans l'erreur de perception, comme dans la question 1 de la version 1) et en considérant le cas de figure de l'ajout du comportement coopératif pour porter les objets de type C.

### **Rendu selon échéancier :**

Version 1 : le 07/11/2021 (envoi d'un lien GIT vers un répertoire contenant les sources, une démo (vidéo ou autre), et un mini rapport (5 pages max) analysant le travail.

Version 2 : le 07/12/2021 (envoi d'un lien GIT vers un répertoire contenant les sources, une démo (vidéo ou autre), et un mini rapport (8 pages max) décrivant les choix de modélisation analysant le travail.