

Projet IA - Problème de la patrouille - Approche EVAP

Timothé Rios - Nicolas Venot

mars 2021

Table des matières

1	Principe général	3
2	Implémentation	3
2.1	Les cases	3
2.2	Les agents	3
3	Algorithme	4
3.1	Algorithme des cases	4
3.2	Algorithme des agents	4
4	Étude des paramètres	5
4.1	Nombre d'agents	5
4.2	Probabilité d'aller tout droit	5
5	Conclusion	5
5.1	5
A	Récapitulatif des types et fonctions	6

Introduction

Ce projet a pour objectif de modéliser aussi précisément que possible le problème de patrouille en utilisant l'approche d'évaporation des phéromones. Cette approche, inspirée par l'étude des colonies d'insectes sociaux, a pour particularité d'attribuer à chaque zone géographique un taux de phéromones spécifique qui diminue avec le temps. Les agents ont alors pour fonction de maximiser autant que possible le taux de phéromones des zones autour d'eux, d'où la nécessité de patrouiller et la pertinence de ce modèle pour résoudre ce problème.

1 Principe général

Dans ce modèle, le temps est discrétisé et l'environnement est modélisé par une grille de cases. Chaque agent avance d'une case par pas de temps, sur la case à gauche, à droite, devant ou derrière lui qui a la plus petite quantité de phéromones. Chaque fois qu'un agent arrive sur une case, il dépose une quantité P_{max} de phéromones sur celle-ci.

À chaque pas de temps, toutes les cases perdent une certaine quantité de phéromones : si elles ont P_t phéromones à l'instant t , elles ont $P_{t+1} = P_t * (1 - p)$ à l'instant $t + 1$, avec p compris entre 0 et 1 exclus.

Une catégorie de cases ne peut pas recevoir de phéromones et ne peut pas être traversée par les agents. Cette catégorie permet de représenter des murs ou des obstacles dans ce modèle. Évidemment, les agents en contact avec des cases de type mur ne vont pas prendre en compte ces dernières dans leurs déplacements.

Enfin, si plusieurs cases atteignables par un agent au prochain pas de temps ont toutes deux le niveau de phéromones minimum parmi les voisins de cet agent, il choisit au hasard parmi celles-ci. Cependant, si une de ces cases est la case qui lui fait face, il a la probabilité p_{avant} de garder sa trajectoire. Ceci permet d'éviter des mouvements trop erratiques.

Afin de pouvoir étudier l'efficacité du modèle en fonction des valeurs de ses différents paramètres, nous utilisons trois paramètres d'observation. Le premier est l'inoccupation instantanée de la case (INI pour *Instantaneous Node Idleness*), un paramètre spécifique à chaque case qui correspond au nombre de pas depuis la dernière visite de cette case par un agent. Le deuxième est l'inoccupation instantanée du graphe (IGI pour *Instantaneous Graph Idleness*), qui correspond à la moyenne des INI de toutes les cases à chaque pas. Enfin, le dernier paramètre d'observation que nous utilisons est l'inoccupation instantanée maximum (IWI pour *Instantaneous Worst Idleness*), qui correspond à l'INI maximum à chaque pas.

2 Implémentation

2.1 Les cases

Chaque case se voit attribuer plusieurs paramètres.

Le premier est un booléen **mur** indiquant si la case est un mur ou non et le deuxième est un nombre **phéromone** indiquant le niveau de phéromones de la case. Le dernier paramètre est un entier **ini** qui recense l'INI de la case.

Durant une simulation, le **phéromone** de chaque case diminue à chaque pas de temps comme expliqué plus-haut : p , le coefficient d'évaporation, est un paramètre réglable de la simulation mais n'a qu'une influence esthétique. les cases ayant leur paramètre **mur** à **true** ne font pas diminuer la valeur de leur **phéromone** et ne font pas augmenter leur **ini**, qui ne sont de toute façon jamais pris en compte dans la simulation. la couleur des cases varie en fonction de leur état de mur ou non et de leur niveau de phéromones : les cases ayant le booléen **mur** à **true** sont bleues et les autres ont une couleur verte dont l'intensité varie proportionnellement à leur niveau de phéromones.

2.2 Les agents

À chaque pas de temps, les agents se déplacent sur la case mitoyenne (uniquement sur la même ligne ou sur la même colonne) ayant le paramètre **phéromone** le plus bas et l'incrémentent de un . Ce faisant, il remet aussi le paramètre **ini** de la case à **zéro**. Les agents ignorent les cases ayant leur paramètre **mur** à **true**. Tous les agents ont la même probabilité p_{avant} de privilégier la case devant eux si celle-ci fait partie des cases mitoyennes ayant la valeur minimum de **phéromone**.

3 Algorithmes

3.1 Algorithme des cases

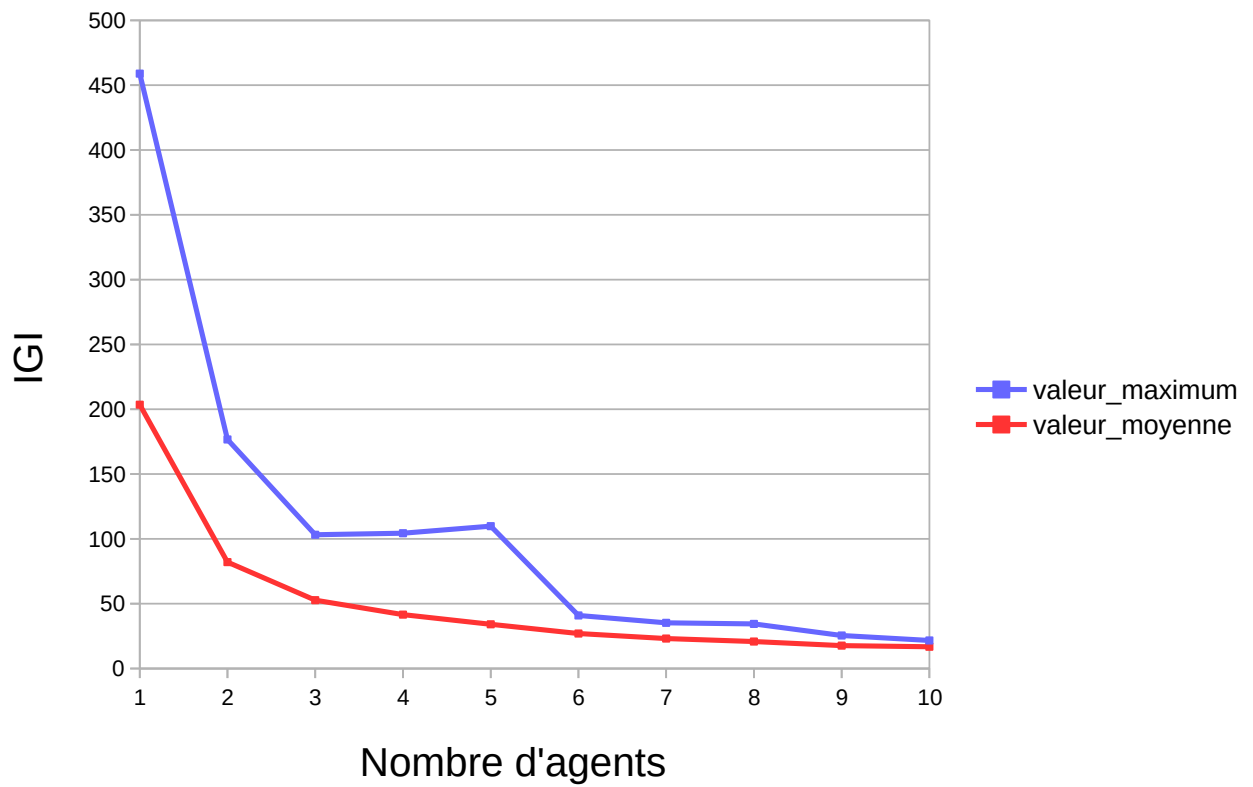
```
for chaque case  $c$  do
  if  $c$  n'est pas un mur then
     $c.phéromone \leftarrow c.phéromone * (1 - p)$ 
     $c.ini \leftarrow c.ini + 1$ 
  end if
end for
```

3.2 Algorithme des agents

```
for chaque agent  $a$  do
   $voisins\_min \leftarrow \{\}$ 
   $min \leftarrow$  minimum du phéromone des voisins
  for chaque case voisine  $v$  de  $a$  do
    if  $v$  n'est pas un mur et  $v.phéromone = min$  then
       $voisins\_min \leftarrow voisins\_min + v$ 
    end if
  end for
   $case\_devant \leftarrow$  case en face de l'agent
  if  $case\_devant$  est dans  $voisins\_min$  et  $random(1) < p$  then
     $case\_prochaine \leftarrow case\_devant$ 
  else
     $case\_prochaine \leftarrow random(voisins\_min)$ 
  end if
  l'agent va sur  $case\_prochaine$ 
end for
```

4 Étude des paramètres

4.1 Nombre d'agents



4.2 Probabilité d'aller tout droit

5 Conclusion

5.1

A Récapitulatif des types et fonctions