

# Le Problème de la patrouille multi-agents

Encadré par :Mme Zahia GUESSOUM

Elaboré par : Lila IDRI, Dragan PAVKOVIC

# Plan



# 1

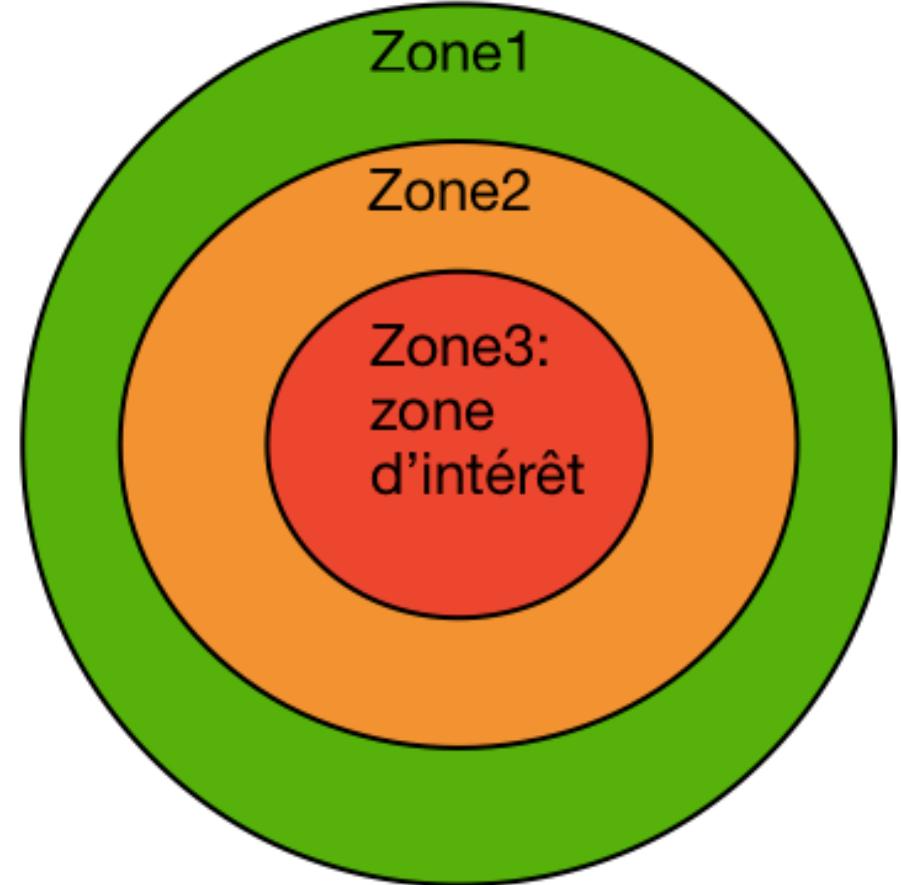
## Introduction

Problématique et solution proposée



# Introduction

- On a une zone de niveau de criticité élevé qu'il faut protéger contre toute attaque adverse. Pour ce faire, on dispose d'un ensemble d'agents cognitifs.
- On se limite à une seule zone d'intérêt et de trois niveaux de criticité.
- Nécessité de mettre en place un système permettant la protection de la zone d'intérêt.





# Introduction



## ► Solution:

Le développement d'un système multi-agents de la patrouille en environnement connu.

# 2

## Conception



## Conception

- Le problème de la patrouille à était largement étudier ces dernières années.
- Parmi les algorithmes utilisés dans les stratégie de patrouille: Voronoï et Lloyd.
  - Algorithme de Voronoï:
    - calcule une configuration optimale d'un ensemble de robots en environnement connu
    - découper le plan en cellules dites « régions de Voronoï » dont le nombre est égale au nombre d'agent du système



## Conception

### Algorithme de Voronoï

Pour chaque agent protecteur faire :

Affecter son numéro dans **indice-agent** (variable globale qui intervient dans recolor)

Affecter sa couleur dans **couleur-agent** (variable globale qui intervient dans recolor)

Affecter sa **perception** à 2

Affecter à **voisins** l'ensemble des agents protecteurs dans sa perception

Effectuer la procédure recolor dans sa perception (colorie les points de sa région de Voronoi perceptive, la partie de sa région de Voronoi située dans sa perception)

Affecter à **point** le patch de sa région de Voronoi perceptive le plus éloigné de l'agent.

Tant que **perception** est plus petite que  $2 * \text{distance}(\text{agent}, \text{point})$  faire  
doubler **perception**

Affecter à **voisins** l'ensemble des agents protecteurs dans sa perception

Effectuer la procédure recolor dans sa perception

Affecter à **point** le patch de sa région de Voronoi perceptive le plus éloigné de l'agent.

Fin faire tant que.

Fin faire pour chaque agent.



## Conception

- Algorithme de Lloyd: calcule le point centroïde d'une région de Voronoï

Choisir  $p$

Définir  $V_p$

Tant que vrai faire

    définir  $g_p$

$p$  reçoit la position de  $g_p$

    définir  $V_p$

fin tant que

$P$  : un point germe

$V_p$ : La région Voronoï associé à  $P$

$g_p$ : le centroïde de la région Voronoï  $V_p$

$g_p = \arg\min p \in V_p \sum \|y - p\|^2 y \in V_p$



## Implémentation et démonstration

- Plusieurs langage de modélisation d'un SMA: jade, NetLogo,...
- NetLogo est un langage de programmation orienté agent.
- NetLogo propose 4 type d'agents:



Les agents patches



Les agents turtles



Les agents link

L'agent protector

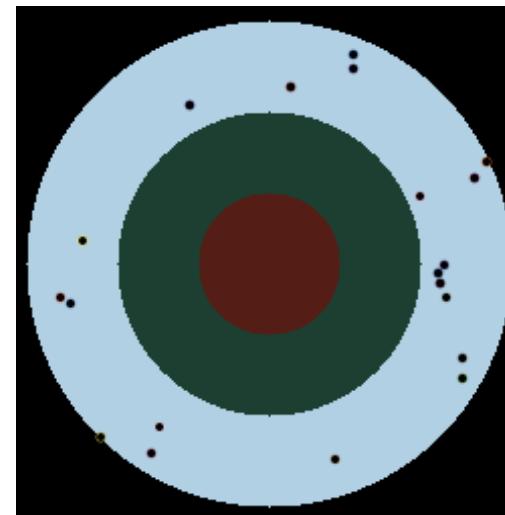
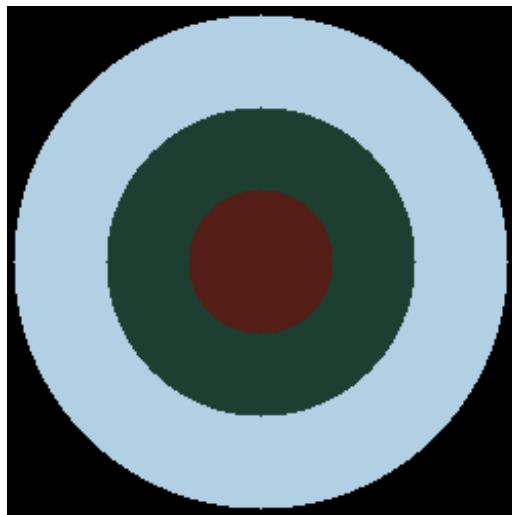
- Deux procédures principales: « setup » et « go » .



## Implémentation et démonstration

### 1. Implémentation de l'environnement en NetLogo

- L'environnement est représenté par des agents patches.
- Les agents du SMA sont représentés par les agents turtles.



**Figure:** Modélisation de l'environnement



## Réalisation

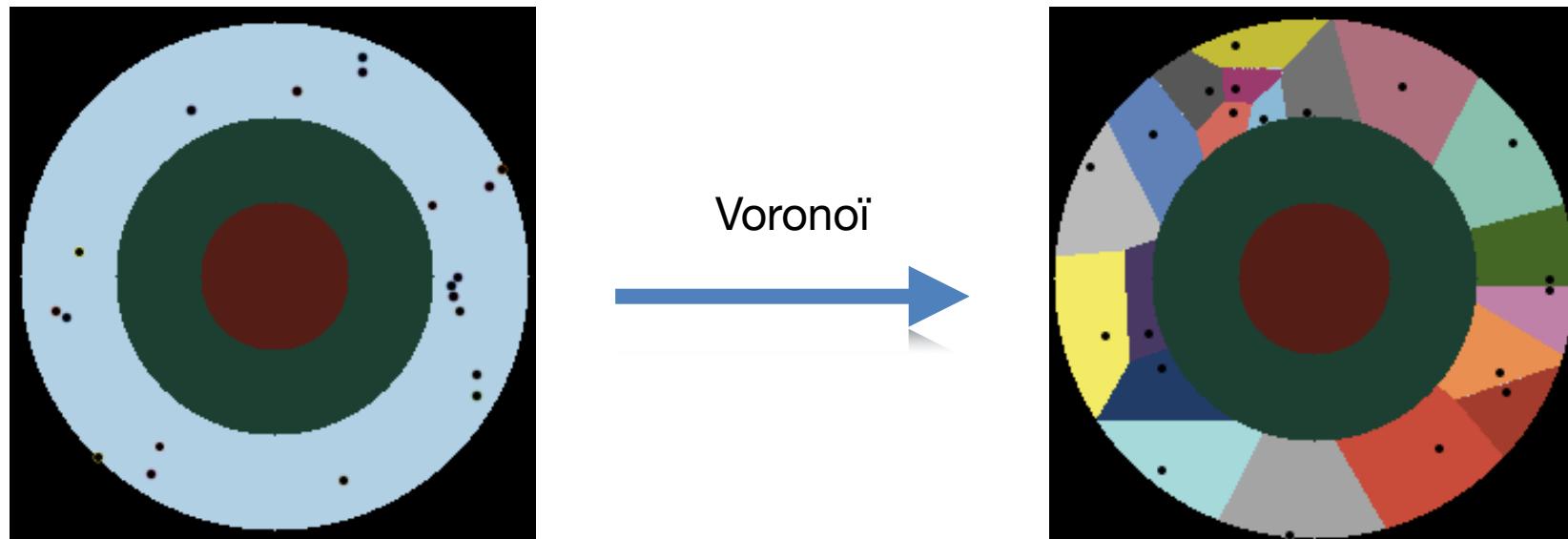
### 2. Couverture optimale d'une zone de protection

- Pour cela, on a utilisé deux algorithmes vu précédemment: Voronoï et Lloyd.
  - Algorithme Voronoï: pour trouver la réparation des agents sur la zone à couvrir



## Réalisation

### 2. Couverture optimale d'une zone de protection



**Figure:** Algorithme de Voronoï sur 20 agents



## Réalisation

### 2. Couverture optimale d'une zone de protection

- Problème: les agents se positionnent sur les points germes de leurs régions Voronoï qui ne correspondent pas aux centroïdes.
- Solution: faire tourner l'algorithme de Lloyd.



## Réalisation

### 2. Couverture optimale d'une zone de protection

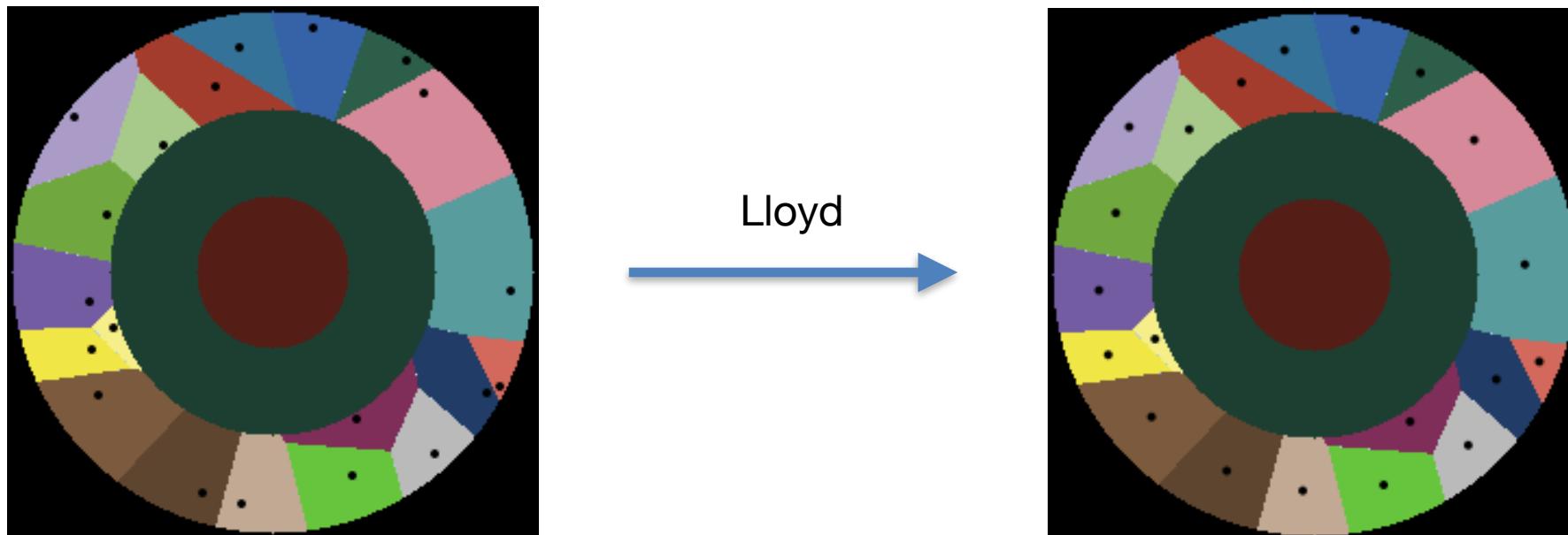


Figure: Algorithme Lloyd sur 20 agents



## Réalisation

### 3. Le mécanisme de repli

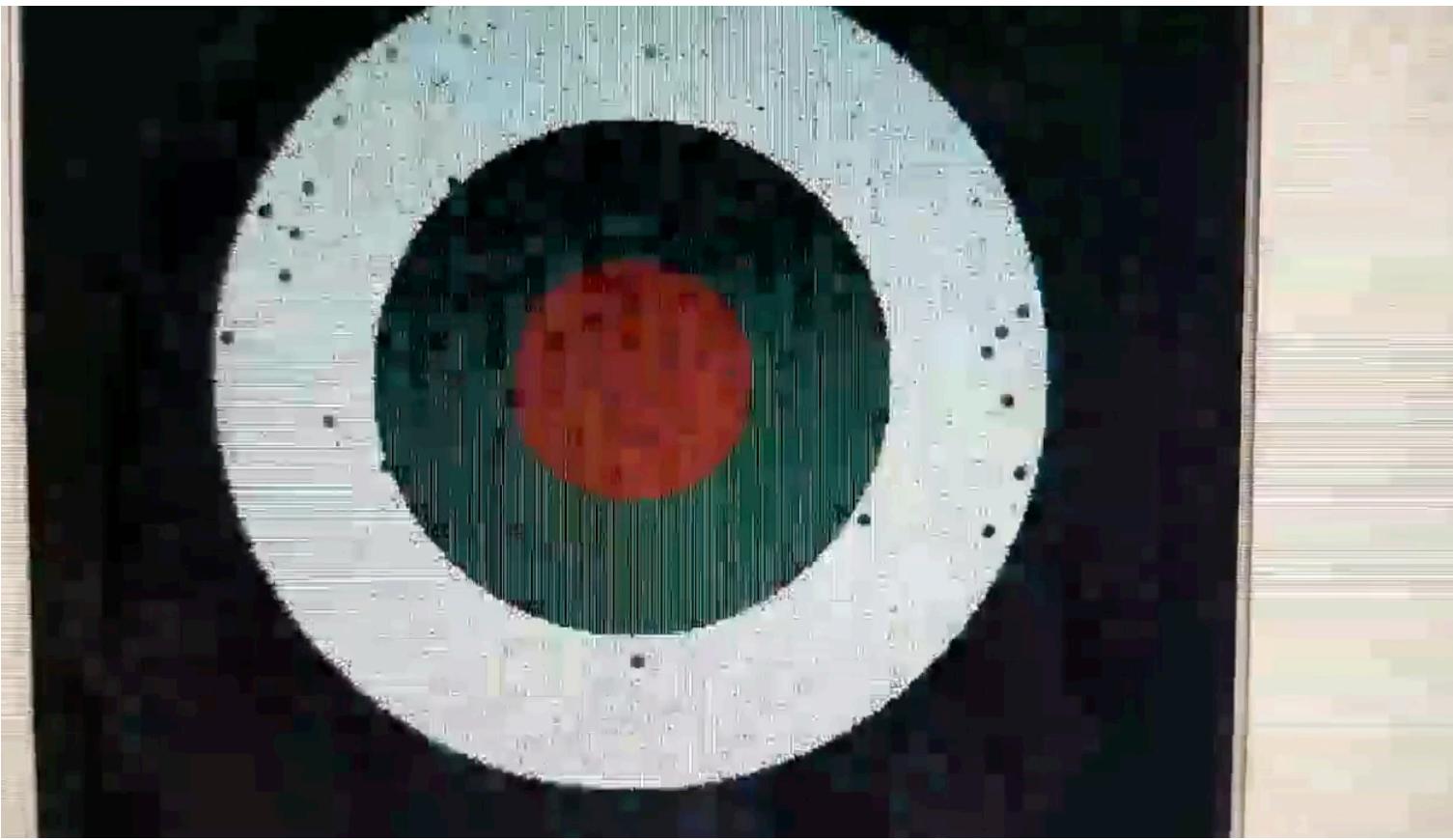
- Mécanisme déclenché suite à:
  - une sortie d'un agent sans qu'il y est attaqué.
  - une sortie d'un agent lorsqu'il y a attaque adverse.



## Réalisation

### 3. Le mécanisme de repli

- Sortie d'un agent sans qu'il y est attaque.





## Réalisation

### 3. Le mécanisme de repli

- Sortie d'un agent lorsqu'il y a attaque adverse.



# Conclusion

## Objectif atteint

- ✓ Développement d'un SMA pour résoudre notre version du Problème de la patrouille

## Nouvelle connaissance

- ✓ Le paradigme agent



## Travail en équipe

- ✓ Organiser le travail
- ✓ Faire preuve de patience



Merci de votre  
attention !