

# RAPPORT DE MINI-PROJET D'IARO

3i025, Licence d'Informatique L3

Projet Multi-robots "MULTIROBOT WARS"

REALISER PAR:  
DJEDDAL Hanane.  
TOUZARI Leticia.

2018/2019

## INTRODUCTION

La robotique est par essence multidisciplinaire, traitant dès leur origine des problèmes couplant mécanique, automatique, électronique, informatique, etc. sans omettre ceux relatifs au domaine d'application lui-même.

Cette complexité induit de nombreuses exigences, notamment sur l'informatique censée gérer le fonctionnement du robot et supporter ses capacités d'action, d'adaptation, de décision, etc., cette « intelligence » que lui confère son contrôle. C'est donc de l'architecture logicielle de contrôle, c'est-à-dire la manière dont est conçu et développé le logiciel chargé du contrôle du robot

Dans le cadre de ce projet, on propose différentes architectures de comportements et de sélections ayant pour but de parcourir une grille de jeu et posséder le maximum de cases.

## I. Architectures

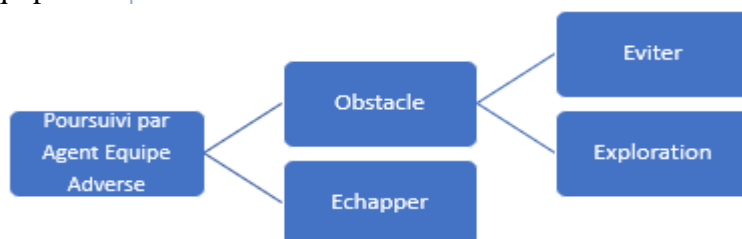
Pour chaque agent une architecture de subsomption, composée de comportements et d'un processus de sélection est définie comme suit :

### 1.1 Architecture 1 :

La principale tâche de l'agent est de faire un parcours en évitant les obstacles pour cela on a défini deux comportements avec priorités :

Priorité 1 : Échapper à un agent de l'équipe adverse dans le cas où il est poursuivi en faisant un demi-tour et cela en surveillant les senseurs de derrière dès qu'ils captent la présence d'un agent adversaire plusieurs fois de suite

Priorité 2 : Parcours de la grille en évitant tout obstacle : murs, agents des deux équipes.



### 1.2 Architecture 2 :

Cette architecture a pour but de permettre à un agent de poursuivre des agents de l'équipe adverse, les priorités des comportements sont comme suit :

Priorité 1 : Échapper à un agent de l'équipe adverse dans le cas où il est poursuivi en faisant un demi-tour.

Priorité 2 : Éviter les murs et les agents de son équipe

Priorité 3 : Éviter les agents (adversaire notamment) lorsque l'agent est bloqué (l'agent se libère s'il y a un blocage avec l'agent adverse qu'il poursuivait) tout en évitant les murs.

Priorité 4 : Suivre les agents de l'équipe adverse



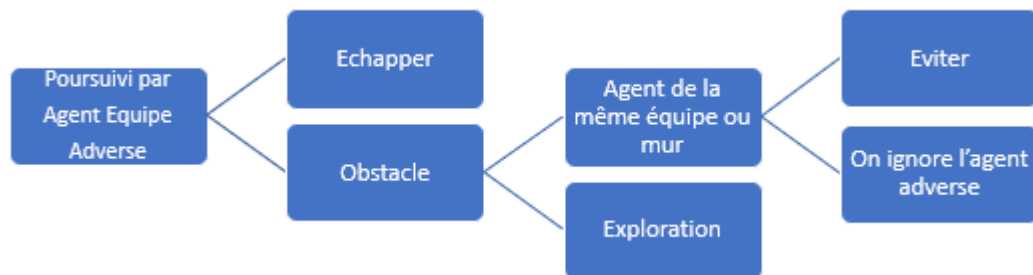
### 1.3 Architecture 3 :

L'agent fait un parcours en évitant les murs et ignorant les agents

Priorité 1 : Echapper à un agent de l'équipe adverse dans le cas où il est poursuivi en faisant un demi-tour.

Priorité 2 : Éviter les murs et les agents de son équipe

Priorité 3 : Parcours de la grille en évitant tout obstacle : murs, agents des deux équipes.



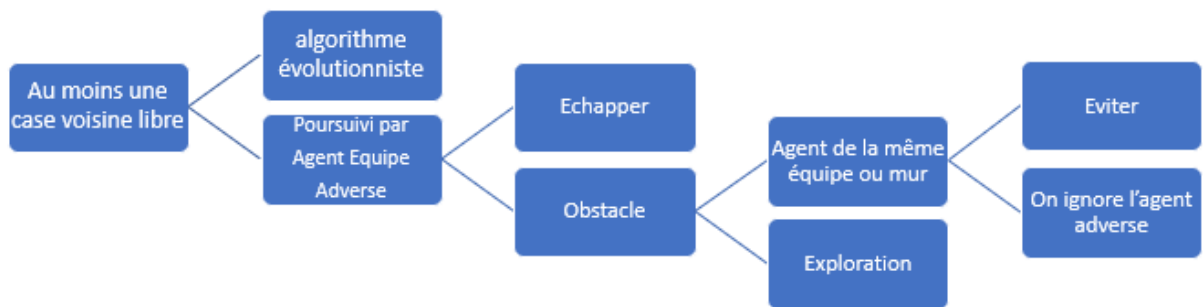
### 1.1 Architecture 4 :

Dans cette approche, on utilise un algorithme évolutionniste. L'agent commence par générer une population des facteurs aléatoire. A chaque itération, il construit des enfants à partir de la population précédente en utilisant la méthode bit flip.

La sélection d'un enfant se fait selon une fonction fitness qui cherche à maximiser le nombre des cases vides ou de l'adversaire à visiter.

Dans le cas où l'agent est entouré par des cases déjà visitées par son propre équipe (fitness =0) on favorise l'exploration, en utilisant une des stratégies déjà développées.

L'enfant sélectionné est sauvegardé dans l'état de l'agent et est utilisé pour construire la génération suivante.



## CONCLUSION

Avec la croissance de la complexité des architectures logicielles robotiques et l'explosion de la diversité toujours plus grande des applications et des missions, la conception et le développement d'architectures logicielles performantes et correctes devient un enjeu majeur. Il n'y a à ce jour aucune approche communément admise pour capitaliser et mutualiser les connaissances et les bonnes pratiques et pour réutiliser les briques logicielles développées. L'état des lieux que nous avons dressé témoigne de la diversité des propositions.