

# RECHERCHE DE CONSENSUS EN ROBOTIQUE EN ESSAIM

LY Jean-Baptiste

9 juin 2020

encadré par BREDECHE Nicolas et MAUDET Nicolas

Figure 1 – L'arène réelle

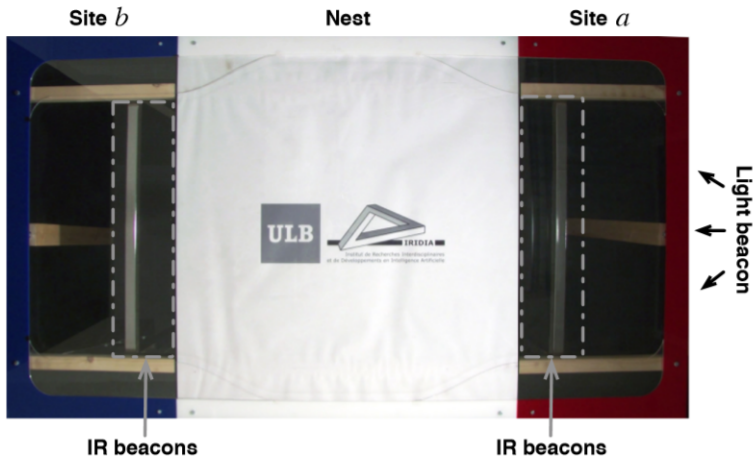


Figure 2 – L'arène sur le simulateur Kilombo

lx time: 0.0 kilo\_ticks: 105 FPS: 25.4

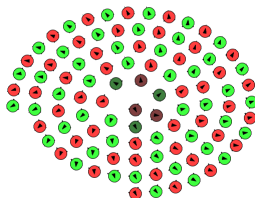
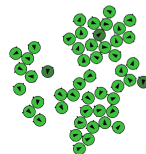


Figure 3 – L'arène sur le simulateur Kilombo

131x time: 827.7 kilo\_ticks: 25697 FPS: 24.0



# Objectifs

- Algorithmes dédiés ✓

# Objectifs

- Algorithmes dédiés ✓
- Algorithmes d'apprentissage

# Objectifs

- Algorithmes dédiés ✓
- Algorithmes d'apprentissage
- Amélioration du système de tracking visuel de robots

## 1 Cas du regroupement autour de la ressource de plus grande valeur

- Majority rule
- Voter model
- Expérimentations et résultats
  - Sites statiques
  - Sites dynamiques



## 1 Cas du regroupement autour de la ressource de plus grande valeur

- Majority rule
- Voter model
- Expérimentations et résultats
  - Sites statiques
  - Sites dynamiques

## 2 Cas où l'essaim doit distribuer ses forces au pro-rata de la valeur de chaque ressource

- Premier algorithme
- Second algorithme

## 1 Cas du regroupement autour de la ressource de plus grande valeur

- Majority rule
- Voter model
- Expérimentations et résultats
  - Sites statiques
  - Sites dynamiques

## 2 Cas où l'essaim doit distribuer ses forces au pro-rata de la valeur de chaque ressource

- Premier algorithme
- Second algorithme

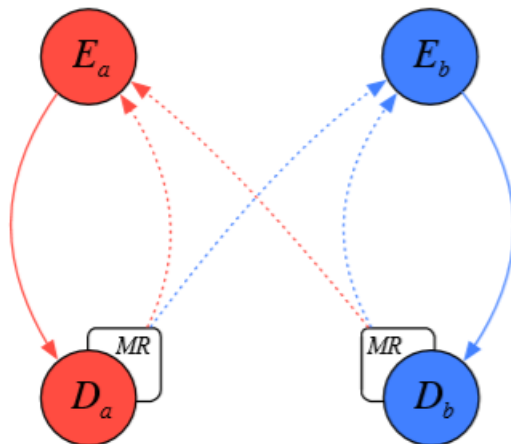
## 3 Conclusion

# Table of Contents

- 1** Cas du regroupement autour de la ressource de plus grande valeur
  - Majority rule
  - Voter model
  - Expérimentations et résultats
    - Sites statiques
    - Sites dynamiques
- 2 Cas où l'essaim doit distribuer ses forces au pro-rata de la valeur de chaque ressource
  - Premier algorithme
  - Second algorithme
- 3 Conclusion

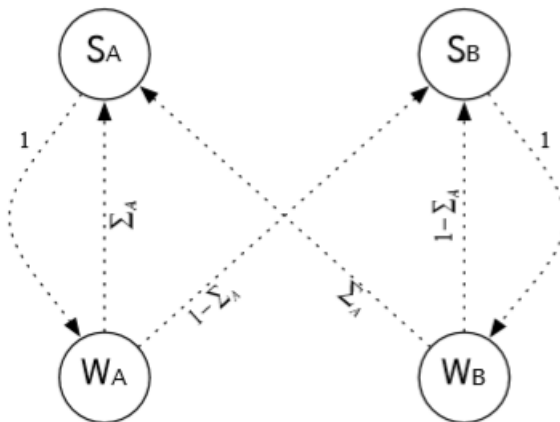
## Majority rule

Figure 4 – Probabilistic finite-state machine



## Voter model

Figure 5 – Probabilistic finite-state machine



## Caractéristiques

- Les qualités des ressources ne changent pas au cours du temps
- Maintenir le consensus

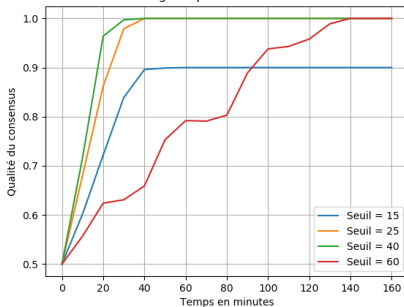
# Majority rule : Expérimentations et résultats

## Deux ressources

- $q_a = 1, q_b = 2$
- Seuils : 15, 25, 40, 60

Figure 6 –

Evolution du nombre d'agents pour la meilleure ressource sur 10 runs :

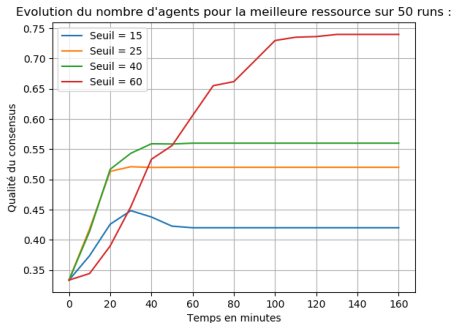


# Majority rule : Expérimentations et résultats

## Trois ressources

- $q_a = 1, q_b = 2, q_c = 3$
- Seuils : 15, 25, 40, 60

Figure 7 –





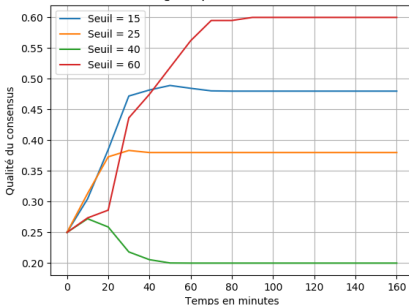
# Majority rule : Expérimentations et résultats

## Quatre ressources

- $q_a = 1, q_b = 2, q_c = 3, q_d = 4$
- Seuils : 15, 25, 40, 60

Figure 8 –

Evolution du nombre d'agents pour la meilleure ressource sur 50 runs :



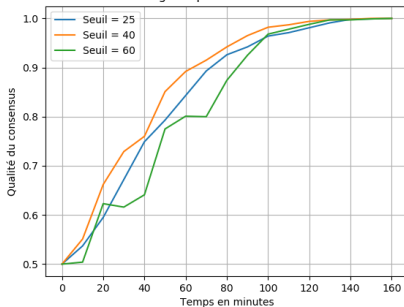
# Voter model : Expérimentations et résultats

## Deux ressources

- $q_a = 1, q_b = 2$
- Seuils : 25, 40, 60

Figure 9 –

Evolution du nombre d'agents pour la meilleure ressource sur 10 runs :



## Voter model : Expérimentations et résultats

### Plus de deux ressources

- $q_a = 1, q_b = 2, q_c = 3, q_d = 4$
- Seuils : 25, 40, 60

## Caractéristiques

- Les qualités des ressources changent au cours du temps
- Changer de consensus au cours du temps

## Les agents dits "têtu" ou "explorateurs"

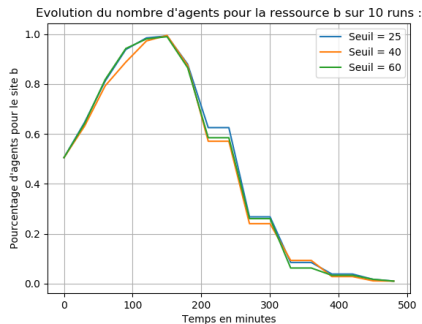
- Agents qui ne changent pas ou plus d'opinion
- Un agent têtu initialisé par ressource

# Sites dynamiques : Expérimentations et résultats

## Deux ressources

- $q_a = 1, q_b = 2$
- Seuils : 25, 40, 60

Figure 10 –



# Table of Contents

- 1 Cas du regroupement autour de la ressource de plus grande valeur
  - Majority rule
  - Voter model
  - Expérimentations et résultats
    - Sites statiques
    - Sites dynamiques
- 2 Cas où l'essaim doit distribuer ses forces au pro-rata de la valeur de chaque ressource
  - Premier algorithme
  - Second algorithme
- 3 Conclusion

## Description

## Caractéristique

- La distribution spatiale entre les deux ressources devra être à l'image de la valeur de chacune des ressources

2 ressources	$theorique_a$	$theorique_b$
$q_a = 1 ; q_b = 2$	33	67
$q_a = 1 ; q_b = 3$	25	75
$q_a = 1 ; q_b = 4$	20	80

3 ressources	$theorique_a$	$theorique_b$	$theorique_c$
$q_a = 1 ; q_b = 2 ; q_c = 3$	17	33	50
$q_a = 1 ; q_b = 3 ; q_c = 6$	10	30	60

4 ressources	$th_a$	$th_b$	$th_c$	$th_d$
$q_a = 1 ; q_b = 2 ; q_c = 3 ; q_d = 4$	10	20	30	40
$q_a = 1 ; q_b = 3 ; q_c = 5 ; q_d = 7$	6	19	31	44



## Problèmes

- Obtenir un consensus plus tôt
- Donner du poids pour les ressources de moindre qualité

# Premier algorithme

Figure 11 –

---

**Algorithm 1:** Premier algorithme

---

```

while  $n \neq \text{nbreAgents}$  do
  forall agent disséminateur do
    while  $\text{changementsOpinions} \neq \text{smax}$  do
      Dissémination
      Voter model
      indiceC = indice de la ressource choisie par le Voter model
       $\text{changementsOpinions} = \text{changementsOpinions} + 1$ 
       $\text{avisCourant}[\text{indiceC}] = \text{avisCourant}[\text{indiceC}] + 1$ 
      Exploration
    L'agent devient un agent explorateur en faveur de la ressource prise aléatoirement
    parmi ses anciennes opinions.
     $n = n + 1$ 
  return consensus

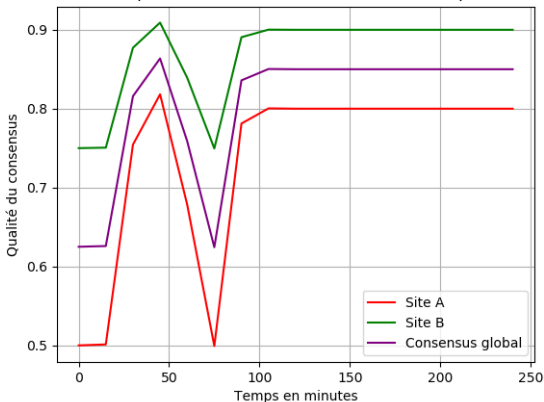
```

---

# Premier algorithme : expérimentations et résultats

Figure 12 –  $q_a = 1$ ,  $q_b = 2$

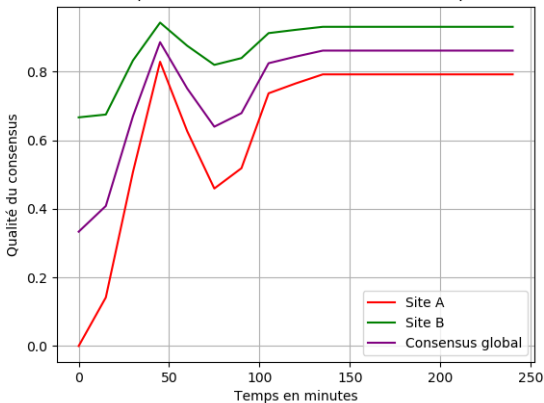
Evolution de la qualité du consensus en fonction du temps sur 50 runs :



# Premier algorithme : expérimentations et résultats

Figure 13 –  $q_a = 1$ ,  $q_b = 3$

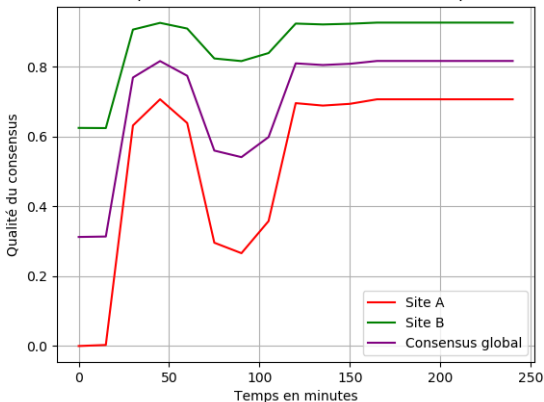
Evolution de la qualité du consensus en fonction du temps sur 50 runs :



# Premier algorithme : expérimentations et résultats

Figure 14 –  $q_a = 1$ ,  $q_b = 4$

Evolution de la qualité du consensus en fonction du temps sur 50 runs :



## Second algorithme

Figure 15 –

---

**Algorithm 2:** Algorithme "amélioré"

---

```

while  $n \neq \text{nbreAgents}$  do
  forall agent disséminateur do
    while  $\text{changementsOpinions} \neq \text{smax}$  do
       $\text{changementsOpinions} = \text{changementsOpinions} + 1$ 
       $\text{avisCourant}[\text{indiceC}] = \text{avisCourant}[\text{indiceC}] + 1$ 
      if  $\text{changementsOpinions} = \text{itCP}$  then
         $\text{avisCourant}[\text{indiceC}] = \text{avisCourant}[\text{indiceC}] + \text{poids}$ 
      if  $\text{changementsOpinions} = \text{itCS}$  then
         $\text{smax} = \text{smin}/\text{qualiteC}$ 
        if  $\text{smax} < \text{changementsOpinions}$  then
           $\text{changementsOpinions} = \text{smax}$ 
     $n+ = 1$ 

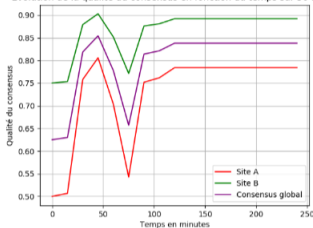
```

---

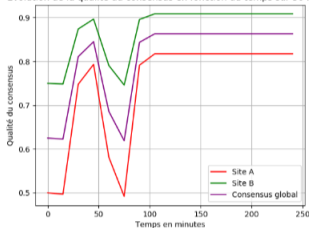
# Second algorithme : expérimentations et résultats

Figure 16 –  $q_a = 1$ ,  $q_b = 2$

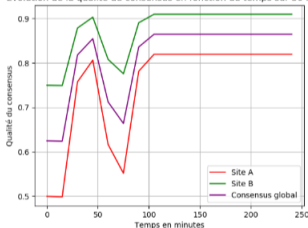
Evolution de la qualité du consensus en fonction du temps sur 50 runs :



Evolution de la qualité du consensus en fonction du temps sur 50 runs :



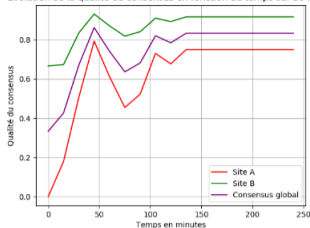
Evolution de la qualité du consensus en fonction du temps sur 50 runs :



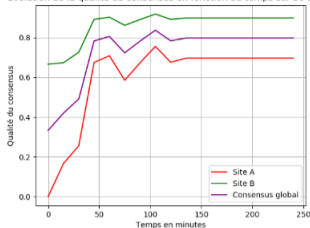
# Second algorithme : expérimentations et résultats

Figure 17 –  $q_a = 1$ ,  $q_b = 3$

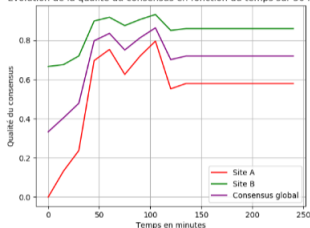
Evolution de la qualité du consensus en fonction du temps sur 50 runs :



Evolution de la qualité du consensus en fonction du temps sur 50 runs :



Evolution de la qualité du consensus en fonction du temps sur 50 runs :

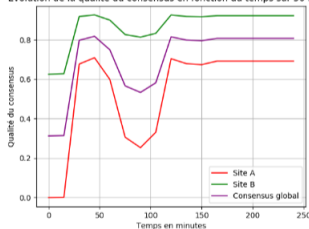




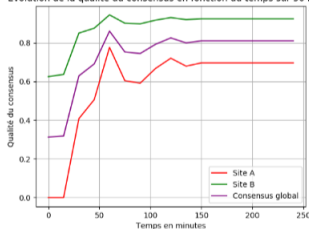
# Second algorithme : expérimentations et résultats

Figure 18 –  $q_a = 1$ ,  $q_b = 4$

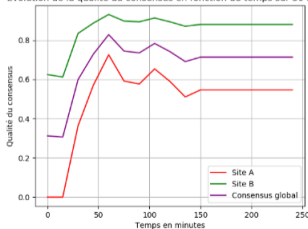
Evolution de la qualité du consensus en fonction du temps sur 50 runs :



Evolution de la qualité du consensus en fonction du temps sur 50 runs :



Evolution de la qualité du consensus en fonction du temps sur 50 runs :



# Table of Contents

- 1 Cas du regroupement autour de la ressource de plus grande valeur
  - Majority rule
  - Voter model
  - Expérimentations et résultats
    - Sites statiques
    - Sites dynamiques
- 2 Cas où l'essaim doit distribuer ses forces au pro-rata de la valeur de chaque ressource
  - Premier algorithme
  - Second algorithme
- 3 Conclusion

## *Conclusion*

*Merci*