

## TP3 : Déminage

### Systemes multi-agents

#### I) Description basique du robot

**Question 1 :** *Quelle architecture vous paraît la plus à même de traiter ce problème ?*

Le problème posé dans ce TP est similaire à l'exemple du "Mars Explorer" donné en cours. La seule différence est que les agents ne ramènent pas les objets (ici mines) trouvés au vaisseau, mais les désamorce sur place.

L'architecture qui semblerait la plus adaptée serait celle appelée "subsumption" qui est une architecture par niveaux. En effet, dans notre cas, les comportements des agents sont simples et sont pour la plupart indépendants : ils n'interagissent pas entre eux. De plus, les agents n'ont pas d'état interne et ne communiquent pas directement entre eux. Il est donc possible de classer les différents comportements des agents par niveaux de priorité.

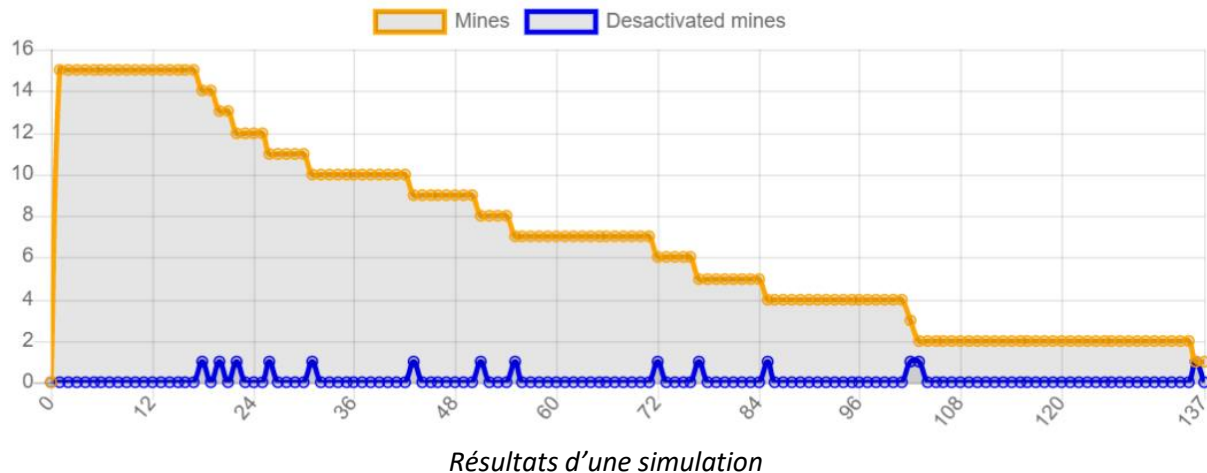
**Question 2 :** *Il est capital que les robots ne rentrent pas en collision les uns avec les autres. Cela prime sur chacun de leurs déplacements. Proposer un ordre de priorité pour les comportements décrits ci-dessus qui respecte cette contrainte et permette aux robots d'accomplir leur mission (NB, il est possible que plusieurs comportements doivent être fusionnés)*

On peut organiser les comportements des agents par ordre décroissant de priorité comme suit :

1. Obstacle, autre agent ou bord de l'environnement → éviter une collision en changeant d'angle
2. Sables mouvants → ralentir (vitesse/2)
3. Se trouve sur une mine → désamorcer la mine
4. Détecte une mine → se déplacer vers la mine
5. Explore en changeant d'angle aléatoirement et en se déplaçant en ligne droite

Implémentez l'ordre que vous proposé précédemment. Ajoutez une métrique représentant le cumul des mines désamorçées à chaque tour. Enregistrez ce graphe et joignez-le à votre TP. Lancez la simulation une dizaine de fois et donnez le temps moyen de désamorçage de toutes les mines.

Nous avons ajouté une métrique permettant de compter le nombre de mines désamorçées par tour (en bleu dans le graphe ci-dessous). Nous lançons une simulation en ne modifiant pas les paramètres de simulation par défaut. Nous obtenons le graphe suivant :



robots : 7, obstacles : 5, sables mouvants : 5, vitesse des robots : 15, mines : 15

Après avoir lancé 10 simulation, on obtient les résultats suivants :

| Simulation | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Nb run     | 174 | 194 | 139 | 370 | 178 | 236 | 149 | 149 | 120 | 229 |

Nombre d'itérations avant le désamorçage de toutes les mines

Les mines sont en moyenne toutes désamorçées au bout de 194 itérations.

**Question 3 :** Quels principes des agents réactifs sont ici respectés ? Lesquels ne le sont pas ? Justifiez.

Les principes des agents réactifs qui sont respectés sont :

- Simplicité : les agents n'ont pas d'état interne. Ils agissent uniquement en fonction de ce qu'ils perçoivent. Leur intelligence repose seulement sur le déplacement et la perception.
- Pas de modèle : les agents n'ont pas une vision globale de leur environnement. Ils n'ont accès qu'à des données brutes issues de leurs capteurs (ici dont la portée vaut le rayon d'action des agents). Ces capteurs permettent aux agents d'identifier des objets et de les localiser uniquement lorsque ceux-ci sont situés dans leur champ de vision.
- Modularité : chaque tâche des agents (éviter les collisions, explorer, désamorcer une mine...) est associée à un comportement spécifique (changer de direction, avancer en ligne droite, supprimer la mine...). Aucune planification n'est faite, la tâche est exécutée dès qu'un stimuli (perception) la déclenche.

A contrario, les principes des agents réactifs qui ne sont pas complètement respectés sont :

- Incarnation : les agents ne sont pas représentés par des corps mais seulement de manière ponctuelle. Cette perte d'information ne permet pas de traiter l'évitement de collisions de manière complète et précise.
- Emergence de complexité : les comportements simples des agents et leurs interactions avec l'environnement ne permettent pas de créer un système complexe. Une collaboration entre agents au moyen d'une communication indirecte pourrait être envisagée pour complexifier le système.

## II) Communication indirecte

**Question 4** : *Commentez ce dernier point au regard des principes de l'architecture réactive.*

Le fait que les agents communiquent entre eux n'est pas en contradiction avec les principes de l'architecture réactive, puisque cette communication a lieu de manière indirecte. A la suite d'une perception (détection d'une mine) ou d'une expérience (passage dans les sables mouvants), les agents déposent une balise avec une indication pour prévenir les autres agents de ce qu'ils viennent de percevoir. Les agents qui perçoivent une balise, lisent l'information qu'elle contient et agissent en conséquence. Une tâche est donc toujours associée à un comportement spécifique des agents. Nous ajoutons simplement de nouvelles tâches à remplir par les agents et de nouveaux comportements leur correspondant.

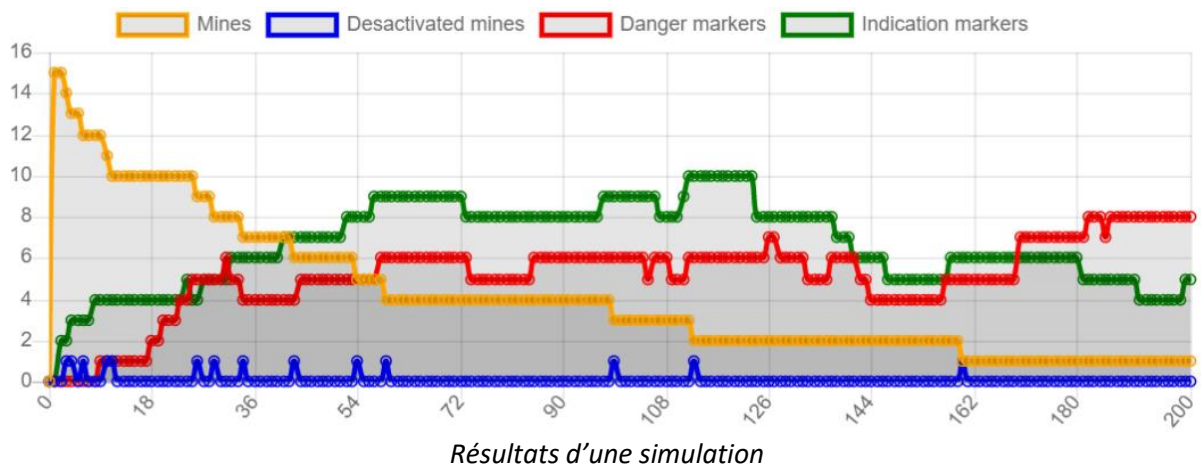
**Question 5** : *Proposez une nouvelle manière d'organiser ces comportements et justifiez.*

On peut organiser les comportements des agents par ordre décroissant de priorité comme suit :

1. Obstacle, autre agent ou bord de l'environnement → éviter une collision en changeant d'angle
2. Sables mouvants → ralentir ( $vitesse/2$ )
3. Sort des sables mouvants → déposer une balise DANGER
4. Se trouve sur une mine → désamorcer la mine
5. Se trouve sur une balise → ramasser la balise et changer d'angle en fonction de l'information contenue dans la balise
6. Détecte une mine → déposer une balise INDICATION et se déplacer vers la mine
7. Détecte une balise → se déplacer vers cette balise
8. Explore en changeant d'angle aléatoirement et en se déplaçant en ligne droite

**Question 6 :** Relancez 10 simulations et incluez un graphe dans le rendu du TP. Quel est le temps moyen de désamorçage des mines ? Commentez.

Après avoir ajouté le fait que les agents puissent déposer et lire des balises, on obtient les graphes suivants :



*robots : 7, obstacles : 5, sables mouvants : 5, vitesse des robots : 15, mines : 15*

On remarque sur le graphe précédent que la fréquence du nombre de mines désamorcées est plus élevée au début de la simulation, puisqu'il y a un plus grand nombre de mines et donc une probabilité plus élevée qu'un agent en perçoive une en explorant. On voit que dans le même temps le nombre de balises déposées augmente. Puis, ce nombre diminue, ce qui traduit le fait que les agents ont ramassé ces balises INDICATION, mais il semble en rester un nombre encore suffisant pour pouvoir informer d'autres agents.

Les 2/3 des mines sont désamorcées durant les 50 premiers tours sur 200 de la simulation. En comparaison avec le graphe que nous avons obtenu avec l'ancienne implémentation (sans communication indirecte), les performances ont été améliorées puisque les 2/3 des mines étaient désamorcées au bout de 80 itérations sur 137.

Après avoir lancé 10 simulations, on obtient les résultats suivants :

| Simulation | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Nb run     | 121 | 129 | 254 | 244 | 263 | 217 | 409 | 276 | 277 | 242 |

*Nombre d'itérations avant le désamorçage de toutes les mines*

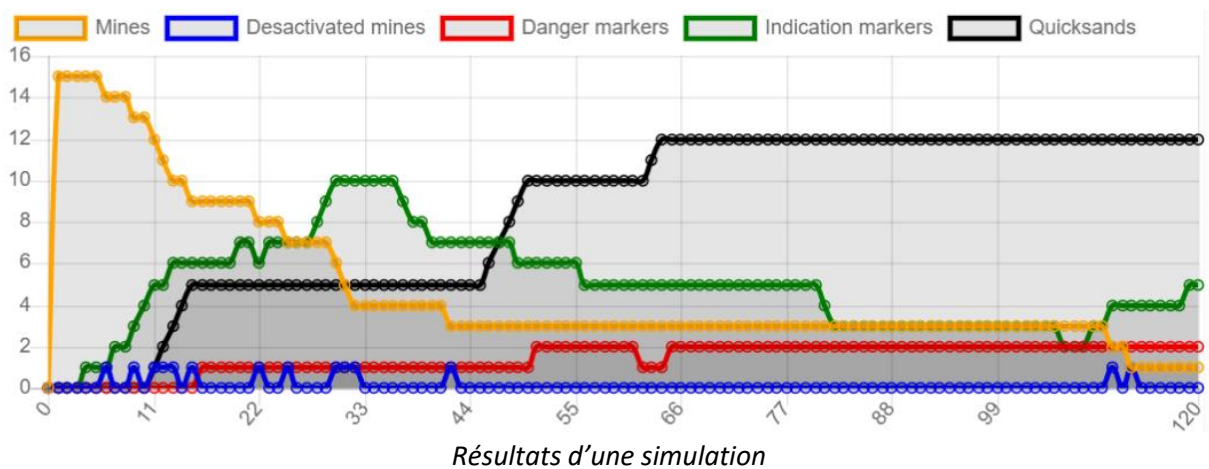
Lorsque les agents communiquent entre eux, les mines sont en moyenne toutes désamorcées au bout de 243 itérations.

On remarque que le temps moyen de désamorçage de toutes les mines a augmenté depuis que l'on a ajouté la possibilité aux agents de communiquer indirectement. Cela peut être expliqué par le fait que les agents perdent du temps en allant récupérer des balises. C'est pour cela que nous avons choisi de limiter le nombre de balises déposées par un agent : il n'en dépose qu'une seule lorsqu'il sort des sables mouvants et lorsqu'il détecte une mine. Pour ce dernier cas, il n'est effectivement pas nécessaire que l'agent dépose une balise tout au long de son trajet vers la mine qu'il a détectée puisque les autres agents ne doivent pas suivre sa trajectoire (comme dans l'exemple « Mars explorer ») mais au contraire s'en éloigner.

De plus, les informations données par les balises sont assez peu précises : l'indication ne porte que sur une direction un peu grossière que doit prendre l'agent (demi-tour ou quart de tour par rapport à la direction de la mine). Les informations données par les balises ne permettent donc pas d'aiguiller précisément un agent. Le rapport entre le temps perdu par un agent pour pouvoir récupérer la balise et le temps que l'indication contenue dans la balise lui fait gagner n'est donc pas très satisfaisant.

**Question 7 :** Ajoutez un reporter permettant de suivre le nombre de tours passés dans les sables mouvants. Comment les balises DANGER influencent-elles le temps moyen passé dans les sables mouvants ?

Après avoir ajouté cet indicateur, on obtient les graphes suivants :

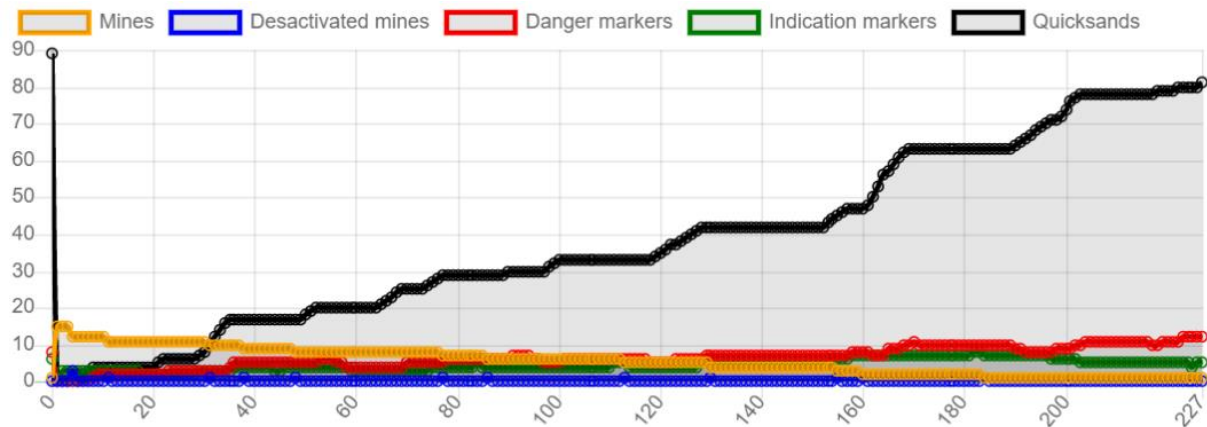


robots : 7, obstacles : 5, sables mouvants : 5, vitesse des robots : 15, mines : 15

On peut voir que le nombre de tours que les agents passent dans les sables mouvants est constant à partir de la 65<sup>ème</sup> itération. Cela signifie qu'aucun autre agent n'a traversé de sables mouvants à partir de ce tour là. On peut expliquer cela par les balises DANGER déposées par les agents une fois sortis des sables mouvants. Ces balises ont permis d'avertir les autres agents en leur indiquant de faire demi-tour.

On peut également noter que le cumul du nombre de tours passés dans les sables mouvants pourrait être amené à augmenter à nouveau si on continue la simulation. En effet, les agents ramassent les balises une fois qu'ils ont pris connaissance des informations qu'elles renferment. Ces informations ne sont donc plus disponibles pour les autres agents qui vont alors avoir plus de chance de traverser des sables mouvants. Une solution pourrait être de modifier le comportement des agents ayant lu une balise de type DANGER : ils ne devraient plus ramasser la balise puisque ce type de « danger » est permanent (les positions des sables mouvants ne changent pas).

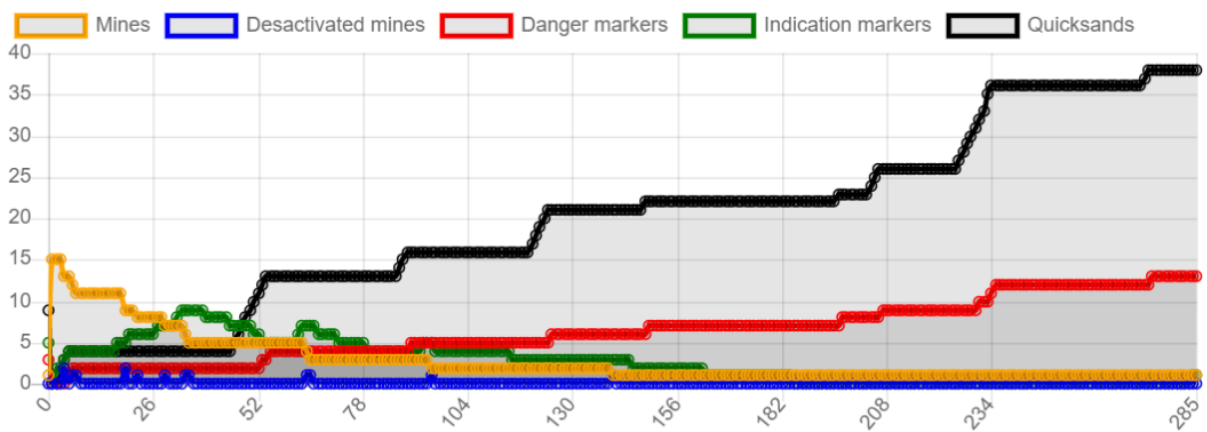
On implémente donc cette alternative dans laquelle les balises DANGER ne sont plus ramassées une fois qu'elles ont été lues. On augmente également le nombre de sables mouvants pour augmenter la probabilité qu'un robot traverse des sables mouvants et ainsi mieux évaluer l'influence des balises DANGER.



Résultats d'une simulation

robots : 7, obstacles : 5, sables mouvants : 7, vitesse des robots : 15, mines : 15

Ramassage des balises DANGER : oui



Résultats d'une simulation

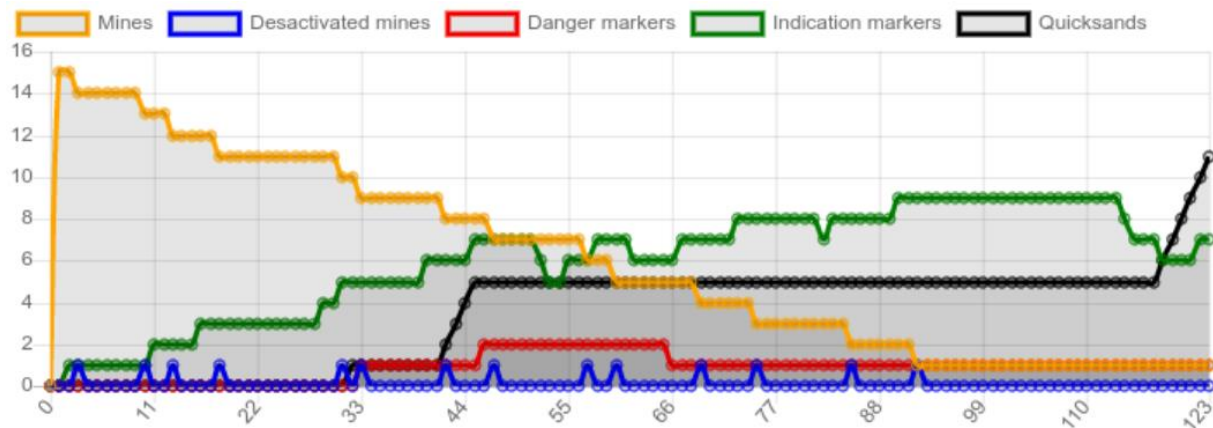
robots : 7, obstacles : 5, sables mouvants : 7, vitesse des robots : 15, mines : 15

Ramassage des balises DANGER : non

On remarque que, dans le cas où les robots ne ramassent pas les balises DANGER, le nombre cumulé de passage des robots dans les sables mouvants augmente moins vite qu'auparavant. En effet, au bout de 230 itérations, environ 35 robots ont traversé des sables mouvants alors que dans l'ancienne version (avec ramassage des balises DANGER) environ 80 robots en ont traversé. Le fait de ne pas ramasser les balises DANGER semble donc nettement impacter le nombre de passages dans les sables mouvants. Cela permet ainsi aux robots de se déplacer en moyenne plus rapidement et donc de mieux explorer l'espace dans une même période de temps. De cette manière, on augmente la probabilité que les agents perçoivent une mine dans un temps donné. Les robots ont ainsi plus de chance de désamorcer toutes les mines plus rapidement.

**Question 8 (bonus) :** Ajouter le mécanisme suivant : on suppose désormais que les agents sont capables de transmettre un signal leur permettant de savoir où se trouvent les autres. Lorsqu'un agent modifie son angle aléatoirement, faites en sorte qu'il se tourne de manière à maximiser l'angle entre sa nouvelle direction et la direction envers chacun de ses 2 plus proches voisins. Relancez 10 expérimentations. Qu'observez-vous ?

On implémente ce mécanisme et on obtient les résultats suivants :



Résultats d'une simulation avec la maximisation d'angle  
robots : 7, obstacles : 5, sables mouvants : 5, vitesse des robots : 15, mines : 15

Après avoir lancé 10 simulations, on obtient les résultats suivants :

| Simulation | 1   | 2   | 3   | 4   | 5   | 6   | 7   | 8   | 9   | 10  |
|------------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| Nb run     | 179 | 141 | 308 | 264 | 210 | 277 | 282 | 226 | 375 | 214 |

Nombre d'itérations avant le désamorçage de toutes les mines

La moyenne sur ces 10 simulations obtenue est de 225, la médiane de 206 et l'écart type de 70. Il s'agit donc d'une légère amélioration.