Université Sorbonne

Master Androide 2nd Year

MADI

Rapport du projet MADI : Algorithmes pour la plannification dans le risque

Auteurs :
Yuhan WANG

Encadrant :
Patrice Perny

1 Introduction : Processus décisionnels Markoviens

Le processus décisionnel Markovien est une forme de représentation d'une interaction synchrone entre un agent et l'environnement dans lequel il se trouve. Il a été beaucoup utilisé pour résoudre le problème de décision dans l'incertitude comme la planification. Un PDM est composé de 4 parties :

$$PDM =< S, A, T, R >$$

S représente l'ensemble fini d'états, A un ensemble fini d'actions, $T : S + A \to L(S)$ la fonction de transition entre les états, et $R : S \times A \to \mathbb{R}$ représente la récompense immédiate après avoir pris l'action a dans l'état s. Dans le certain cas, la récompense immédiate s'agit aussi le prochain état s'.

Une politique est constitué par un ensemble des règles de décision :

$$d : S \to A$$

La facteur d'actualisation γ indique l'importance que l'agent pose sur les récompenses à long terme. Quand $\gamma = 0$, le problème de PDM est résolu par l'algorithme glouton.

Dans notre projet, le robot se trouve dans une grille rectangulaire crée de façon aléatoire et il veut atteindre la case but qui se trouve au sud-est. On vousdrait modéliser les PDMs pour résoudre les problèmes concernant de différents critères, implémenter et tester les algorithmes permettant de déterminer les politiques optimales.

1.1 Modélisation du problème

Dans cette partie, on va préciser comment modéliser le PDM dans le projet pour déterminer la politique demandée à résoudre de différents problèmes.

Chaque case dans la grille est désignée par de différents niveaux de gains ou de risques. Le risque se produit quand le robot traverse une case. Le robot peut choisir une action parmi les quatres élémentaires et il y a une certaine probabilité de déplacer vers son but ou ses voisins accessibles. L'action illégale dont le but n'est pas accessible va faire le robot rester sur place. Donc l'ensemble des états de PDM dans ce contexte est l'ensemble des cases accessibles dans la grille. L'état de sortie est la case du but qui se trouve au coin sud-est, on considère que c'est un état absorbant.

$$A = \{up, down, left, right\}$$

On définit maintenant la fonction de transition : Si cette action est illégale(déplacement vers une mûr ou à l'extérieur de la grille) où il n'y a aucune case accessible :

$$P^a_{s,s'} = 1,\ s' = s$$

Si le but x et ses deux voisines y et z sont tous accessibles :

$$P^a_{s,s'} = \begin{cases} p & s' = x \\ \frac{1-p}{2} & s' = y \text{ ou } s' = z \end{cases}$$

Si x et l'une de ses deux voisines sont accessible, alors :

$$P^a_{s,s'} = \begin{cases} \frac{1+p}{2} & s' = x \\ \frac{1-p}{2} & \text{s' la voisine accessible} \end{cases}$$

Et la conséquence R(récompense ou risque) est condifiée par une échelle de couleur, qui sera déterminé aléatoirement avec une probabilité d'occurence contrôlée.

$$R(s, a) = \sum_{s'} T(s, a, s') * R(s, a, s')$$

2 Moindre risque par itération de la valeur

On considère ici que ce sont les couleurs qui représentent le niveau de risque. Comme on veut minimiser l'espérence des risques encourus, il faut transformer les récompenses en valeurs négative. Et pour motiver l'agent pour atteindre la case but, il faut lui donner une conséquence significative pour la case du but(ici on utilise 1000).

x	vert	bleu	rouge	noir
c(x)	1	2	3	4

2.1 Equations de Bellman

Nous écrivons les équations de Bellman :

$$\forall s \in S, V(s) = max_{a \in A} R(s,a) + \gamma \sum_{s' \in S} T(s,a,s')V(s') \tag{1}$$

Comme dans notre grille, la récompense dépend non seulement de l'état actuel et l'action choisie, mais aussi l'état suivant, on considère que la récompense immédiate est l'espérance de toutes les récompenses possibles. Donc l'équation s'écrit comme :

$$\forall s \in S, V(s) = max_{a \in A} \{\mathbb{E}(R(s,a,s')) + \gamma \sum_{s' \in S} T(s,a,s') * V(s')\} \tag{2}$$

2.2 Expérimentation1

2.2.1 Essais numériques

Dans cette partie, on va effectuer des essais numériques en variant : la taille de grille, la probabilité de transition et le facteur d'actualisation γ. Pour chaque taille entre (10,10),(10,15) et (15,20), on tire 15 instances aléatoirement et calcule le temps moyens de résolution et le nombre moyen d'itérations.

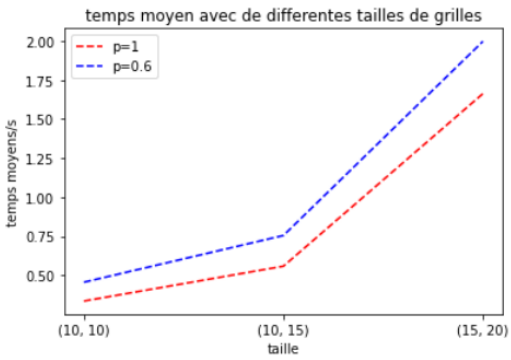


FIGURE 1 – Temps moyens

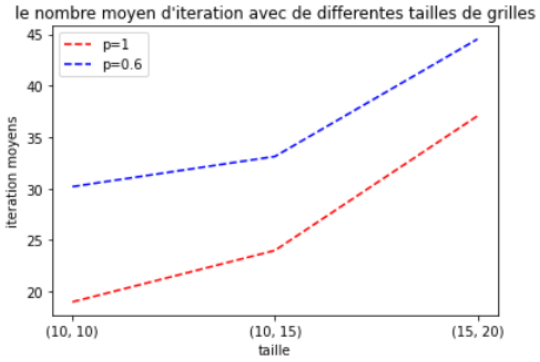


FIGURE 2 – Le nombre moyens d'itérations

On peut voir depuis les deux figures ci-dessus, le temps de résolutions et le nombre moyen d'itérations ont tous augmenté en augmentant la taille du problème et l'incertitude(la probabilité de transition).

Maintenant on va étudier l'impact du facteur d'actualisation γ en le variant sur le temps de résolution et sur le nombre moyen d'itérations.

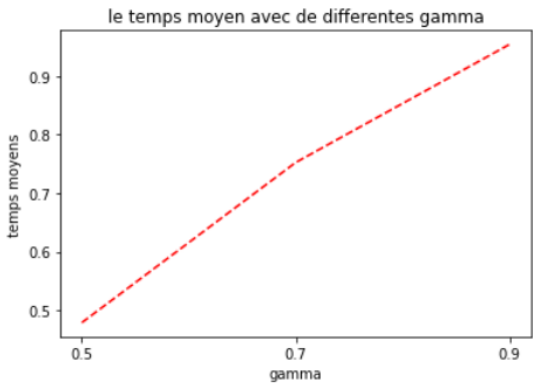


FIGURE 3 – Temps moyens

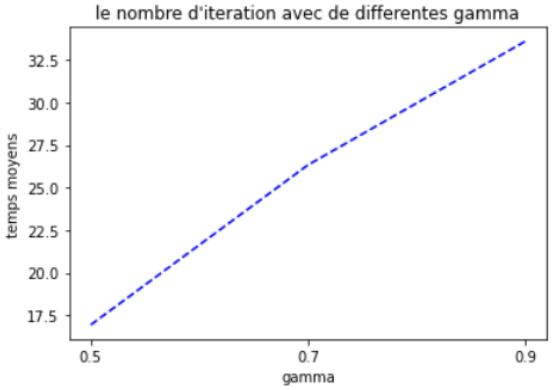


FIGURE 4 – Le nombre moyens d'itérations

Avec un facteur γ plus grand, la méthode de l'itération des valeurs prend plus de temps à se converger. Avec un γ moins important, le robot va se concentrer plutôt sur les récompenses à court terme sans considérer l'avenir.

2.2.2 Un exemple

Ici on va donner un exemple d'instance résolu et la visualisation de sa politique stationnaire optimale.

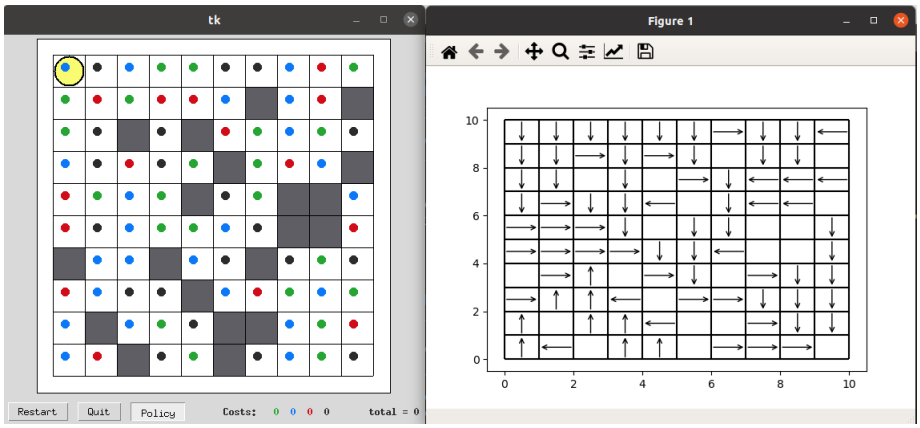


FIGURE 5 – Un exemple de grille résolue et sa politique stationnaire

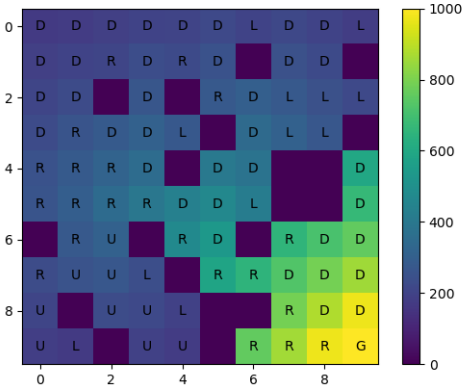


FIGURE 6 – visualisation avec valeurs de l'état

2.3 Fonction coût $c^q(x)$

2.3.1 Impact de γ

En remplaçant la fonction coût par sa puissance de q, notre première observation est avec le même facteur d'actualisation gamma=0.9, la performance du robot est moins bonne dans la région de départ. Au lieu d'essayer d'atteindre le but case et obtenir une récompense significative, il préfère de ne pas bouger pour éviter d'obtenir les récompense négatives en traversant des cases. C'est à dire le regard du robot n'est pas suffisamment long pour qu'il puisse considérer que la récompense sigificative est beaucouplus importante que les récompenses négatives immédiates. Ici si on voudrait corriger le comportement du robot et obtenir la bonne stratégie, il faut aussi augmenter le facteur d'actualisation.

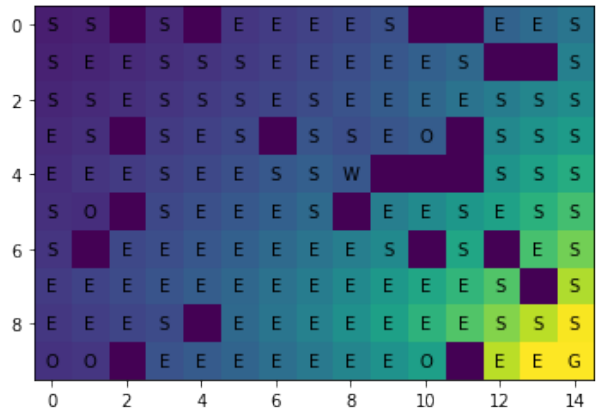


FIGURE 7 – politique avec $\gamma = 0.9, q = 1$

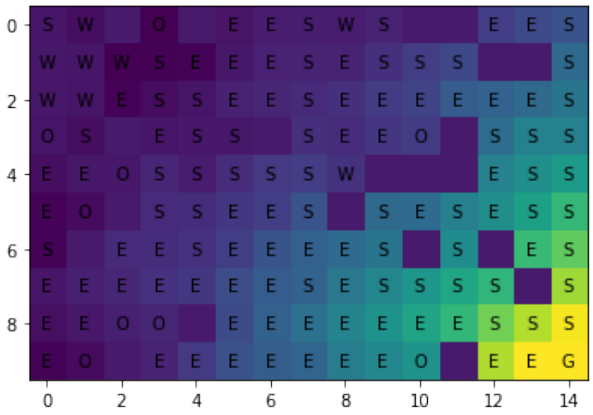


FIGURE 8 – politique avec $\gamma = 0.9, q = 3$

Les deux figures visualisent les politiques retournées en utilisant la méthode d'itération des valeurs avec de différentes valeurs de q. La valeur des états $V_t(s)$ sont représentée par la couleur. Depuis les figures, on peut observer que quand $q = 3$, les valeurs des états dans la région autours du départ sont négatives, beaucoup moins importantes que celles de $q = 1$. Le robot est pessimiste quand il se trouve dans cette région car il ne voit pas l'intérêt de bouger vers la case but. Au contraire, dans la région auprès de la case finale, la politique devient raisonnable : le robot voit la récompense significative de la case finale et avec le but de maximiser l'espérance des récompenses, il va chercher une politique pour atteindre la case finale.

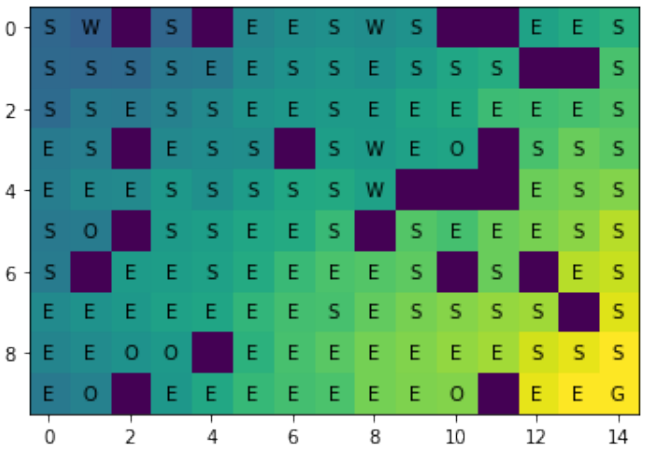


FIGURE 9 – politique avec $\gamma = 0.99, q = 3$

En augmentant la facteur γ, le robot procède une vision à terme plus long. Influencées par la récompense de case finale, les valeurs des états sont tous augmentées, le robot est donc plus motivé à atteindre le but.

2.3.2 Impact de q

En fixant la valeur de $\gamma = 0.99$, probabiité du mouvement $p = 0.8$, on va étudier sur une instance la variation de la solution lorsque q augmente.

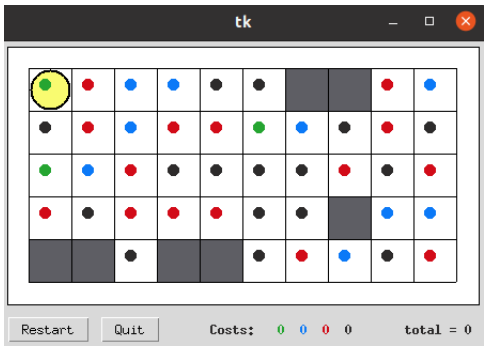


FIGURE 10 – une instance de grille

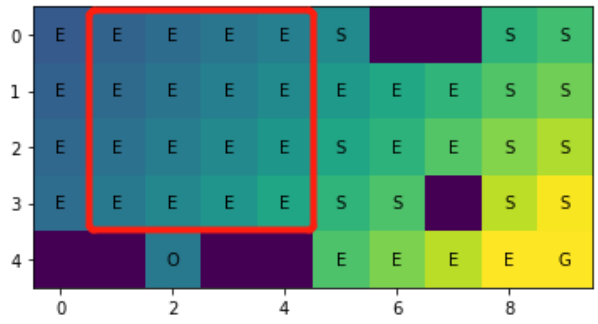


FIGURE 11 – politique avec $\gamma = 0.9, q = 1$

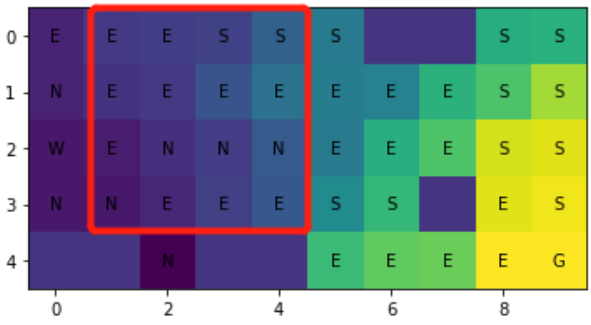


FIGURE 12 – politique avec $\gamma = 0.99, q = 4$

On peut trouver que avec $q = 4$, la stratégie a eu un changement considérable dans le zone rouge où les cases rouges et noirs sont beaucoup apparues. Le robot a montré une tendance d'éviter traverser les cases noires. Donc on peut supposer que, quand il existe un écart significatif entre les risques de différentes couleurs.

2.4 Trajectoire moins de cases noires

Selon les observations obtenues depuis la dernière question, l'écart significative entre les risques condifiés par de couleurs peut modifier la stratégie prise par le robot. Pour trouver une trajectoire qui traverse moins de cases noires, en cas d'égalité, traverse moins de cases rouges...etc, on peut formuler le MDP de la manière suivante et le résoudre en utilisant l'itération de valeurs :

x	vert	bleu	rouge	noir
c(x)	1	10	100	1000

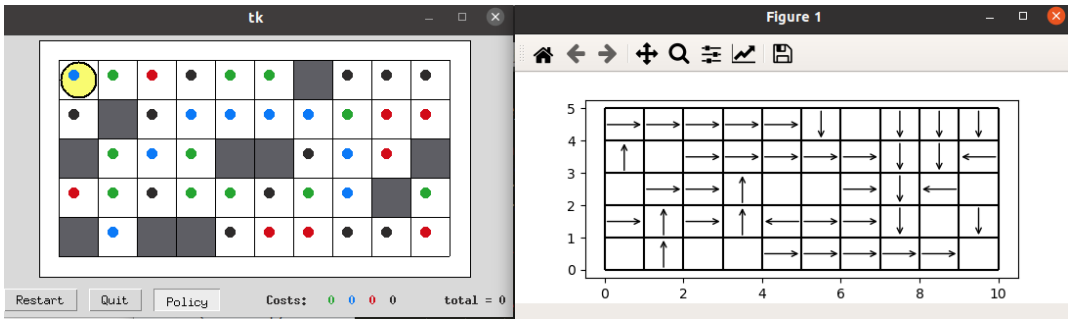


FIGURE 13 – Instance1-taille(5,10)

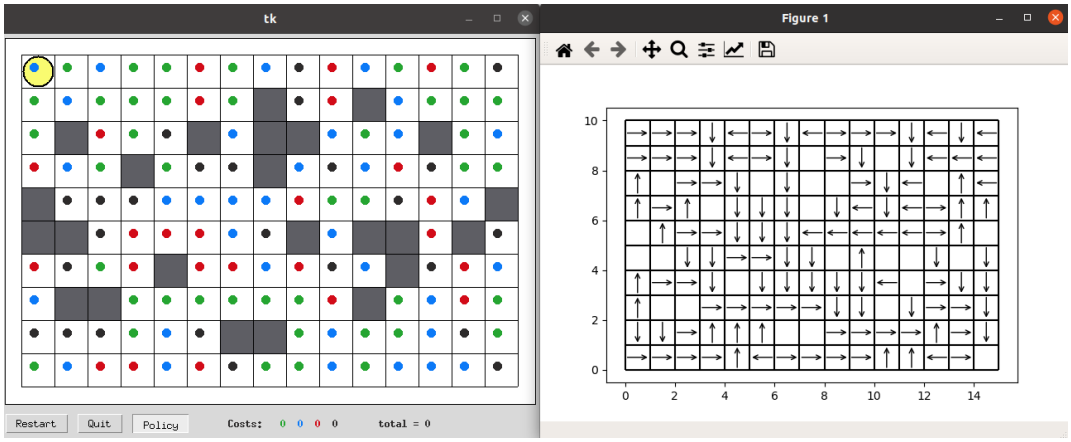


FIGURE 14 – Instance1-taille(10,15)

3 Programmation linéaire

Le problème en 2a de minimiser l'espérence de coût est un problème mono-objectif, ici on va formuler le programme dual pour résoudre le même problème en utilisant le programme linéaire.

3.1 Politique mixte optimale

$$\begin{cases} max \ \sum_{s\in S}\sum_{a\in A} x_{sa}\sum_{s'\in S} T(s,a,s')*R(s') \\ s.t. \sum_{a\in A} x_{sa} - \gamma * \sum_{s'\in S}\sum_{a\in A} T(s',a,s)*x_{s'a} = \mu(s), \forall s \in S \\ \forall s \in S, a \in A, x_{sa} \geq 0 \end{cases}$$

3.2 Politique pure optimale

Pour forcer des politiques purs, il faut ajouter dans le programme linéaire ci-dessus, les contraintes suivantes :

$$\begin{cases} \sum_{a\in A} d_{sa} \leq 1 \forall s \in S \\ (1-\gamma)x_{sa} \leq d_{sa}, \ \forall s \in S, \forall a \in A \\ d_{sa} \in \{0,1\}, \ \forall s \in S, \forall a \in A \end{cases}$$

3.3 Essais numériques

Dans cette partie, on va etudier l'impact posée par la probabilité du mouvement sur le temps de résolutions et les valeurs de la politique optimale pour le cas d'une politique mixte et pour le cas d'une politique pure optimale. On garde trois décimales pour le temps de résolution moyens et le nombre entier pour la valeur de politique.

3.3.1 $p = 1, p = 0.6, \gamma = 0.99$

On trouve que , γ pose une influence importante pour la recherche d'une politique pure optimale. Si sa valeur est inférieure à 0.9, ça va prendre du temps énorme pour que le programme linéaire puisse trouver la politique optimale pure. Donc dans l'expérimentation, on a fixé $\gamma = 0.99$ pour des essais de la politique pure.

policy	p=1	p=0.6
politique mixte	0.185	0.169
politique pure	0.21	0.34

TABLE 1 – temps de résolution moyens en variant p

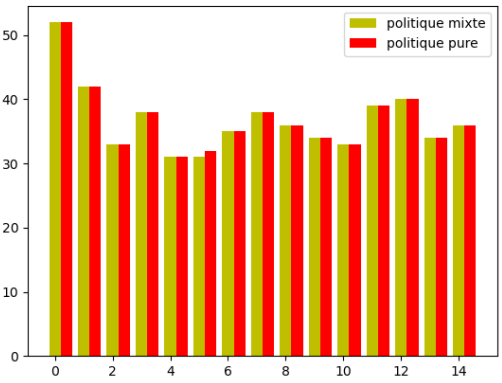


FIGURE 15 – p=1

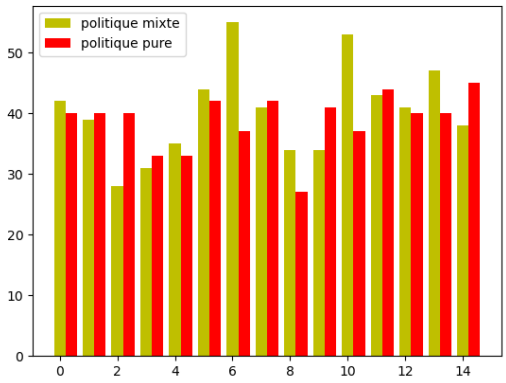


FIGURE 16 – p=0.6

Depuis les résultats obtenus, c'est toujours la politque mixte qui prend moins de temps de résolution. C'est raisonnable car pour obtenir une politqiue pure, on a ajouté des contraintes dans le programme linéaire, qui ralentit le temps de résolution. On présente ici les valeurs de la politique pour le cas mixte et le cas pure optimale des 15 instances tirées aléatoirements et on trouve que pour ce problème mono-objectif(minimiser l'espérence du risque), vu la figure13 produite par $p = 1$ les deux politiques sont aussi performants l'un que l'autre. La différence dans la figure14 est dûe à l'incertitude du mouvement.

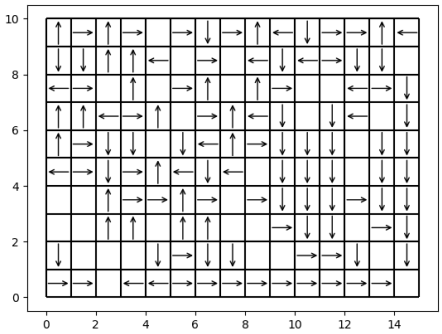
3.3.2 $\gamma = 0.9, 0.7, 0.5$

Comme on a expliqué dans la dernière question, dans cette partie on n'étudie l'impact de γ que sur les politiques mixtes.

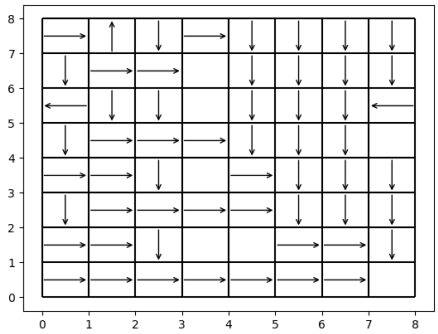
	$\gamma = 0.9$	$\gamma = 0.7$	$\gamma = 0.5$
temps	0.153	0.151	0.127

TABLE 2 – Impact de γ sur politiques mixtes

Alors que la valeur γ se dimunue, le temps de résolution du programme linéaire se rapetisse et les politiques retournées sont moins correctes. Au lieu d'aller à la case du but, le robot se tourne dans une certaine région ou prend les actions vers les mûrs. C'est parce qu'on considère que, si le robot touche la mûr, il va rester sur place et obtenir le risque codifié par la case actuel. Selon le calcul de la suite géométrique et l'impact posé par γ, ce n'est pas étonnant que que le robot choisit l'action vers la mûr car c'est quand-même 'moins risqué' pour lui. Cette phénomène s'aggrave quand la taille de grille s'augmente. Si on voudrait fixer ce problème, on doit ajouter une sanction significative quand le robot touche la mûr(par exemple une récompense de -1000).



politique mixte (10,15)



politique mixte (8,8)

FIGURE 17 – politique obtenue avec $\gamma = 0.5$

4 Recherche d'une trajectoire équilibrée

Dans cette partie, on considère que la récompense de chaque case est affecté par le chiffre.

4.1 PDM

Pour minimiser l'espérance de la consommation totale de ressources, au lieu d'utiliser le risque condifié par la couleur, il faut utiliser le chiffre de chaque case distribuée à la création de la grille comme le coût en le traverçant. Les ressources sont ajoutées dans de différentes critères selon la couleur.

4.2 MOMDP

Pour déterminer une politique qui consomme les différentes ressources de manière équilibrée, on modélise un PDM multi-objectif(MOMDP). On ajoute une variable z qui est supérieur à la consommation de chaque ressource(procédure max), et on minimise le z (procédure min) pour contrôler le pire cas possible.

$$\text{MOMDP} = \begin{cases} \max z \\ z \geq \sum_{s\in S} \sum_{a\in A} x_{sa} \sum_{s'\in S} T(s,a,s') * R_i(s'),\ \forall i = 0,1,2,3 \\ \sum_{a\in A} x_{sa} - \gamma * \sum_{s'\in S} \sum_{a\in A} T(s',a,s) * x_{s'a} = \mu(s),\ \forall s \in S \end{cases}$$

Avec multi-objectif, la méthode de l'itération de valeur n'arrive pas, ou c'est très difficile à se converger.

4.3 PDM vs MOMDP

Dans cette partie, on va comparer le vecteur coût qu'on obtient pour la politique calculée avec MOMDP et celle avec PDM sur deux instances d'exemples.

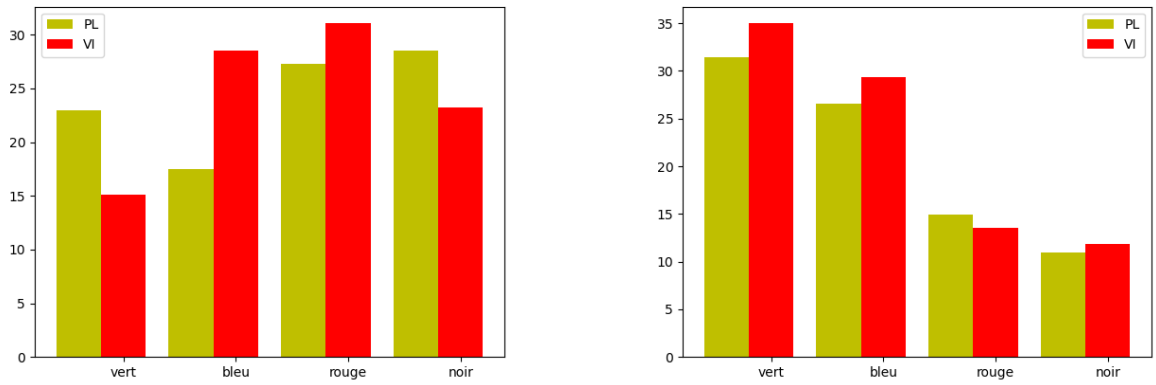


FIGURE 18 – Deux instances d'exemples

Comme ce qu'il a montré dans les figures ci-dessus, on trouve que les deux stratégies sont aussi performants pour dans l'aspect du coût total, mais la stratégie de minmax retourne les solutions sont plus équilibrées, parfois pour atteindre le but d'équilibration, son coût va être légèrement plus grand que celui de PDM.