Rapport RLD TME2

Christopher Heim et Thomas Floquet

I/ Comparaison entre les algorithmes de Policy et Value Iteration

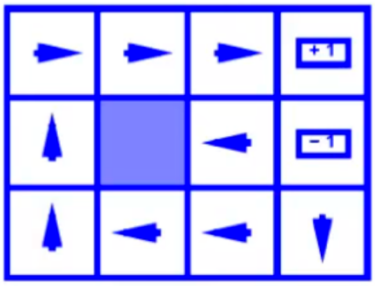
La première remarque est que la Value Iteration et la Policy Iteration convergent vers la même politique optimale, ce qui est en accord avec ce qui a été vu en cours (la Politcy Iteration converge de manière certaine vers la politique stationnaire optimale π en un temps fini, et la Value Iteration converge asymptotiquement vers π).
Pour le plan 8, l'algorithme de Value Iteration met 1,08 seconde pour converger tandis que l'algorithme de Policy Iteration met 41,6 secondes, ce qui confirme là encore ce qui a été vu en cours : la Policy Iteration est beaucoup plus coûteuse en calculs car à chaque itération on fait une évaluation complète de la politique.
Cette différence s'accentue logiquement avec la taille des graphes : pour un plus petit plan comme le plan 1, la différence n'est plus que de 0,30 secondes passant de 0,15 à 0,45 seconde.

II/ Résultats sur les différents plans et impact du paramètre *pénalité*

Dans ce TME, on appellera pénalité le coût de se déplacer sur une case vide. On prend par défaut une pénalité de -0,01.
Globalement, si on réduit le discount (par exemple de 0,999 à 0,6), alors la policy optimale consistera généralement à aller le plus vite possible vers la case verte plutôt qu'à se cogner contre un mur en boucle ou à éviter une pénalité, car plus le discount est proche de 0, alors plus le chemin est long, moins les récompenses éloignées ont de valeur.
La valeur par défaut choisie pour le discount est 0,999.

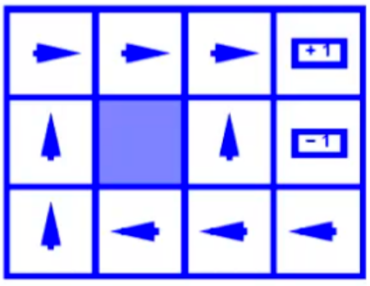
Plan 0 :

Avec plan 0, on retrouve les exemples du cours :



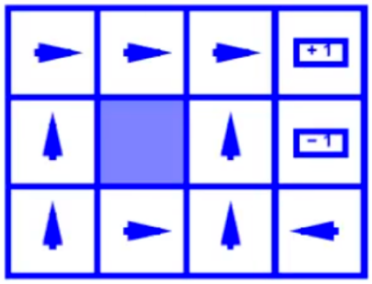
R(s) = -0.01

Quand la pénalité vaut -0,01, on observe qu'il vaut mieux toujours aller vers le bas pour ne pas risquer de prendre une pénalité de -1. Cependant, il arrive parfois que même en décidant d'aller vers le bas le joueur aille finalement à gauche avec une probabilité de 0,1, et alors il vaut mieux essayer de faire le tour complet pour rejoindre la case verte (+1) en haut à droite sans risquer de tomber dans la case rouge (-1).



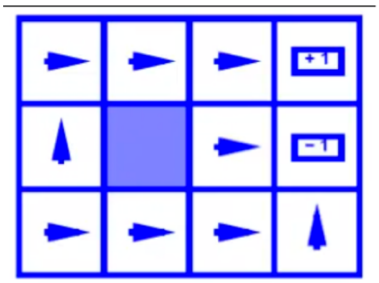
R(s) = -0.03

Avec une pénalité à -0,03, il est alors plus intéressant de chercher à atteindre la case verte (+1) qu'à continuer à toujours aller vers le bas, mais la politique optimale est toujours de faire le grand tour afin de ne pas risquer de tomber dans la case rouge (-1).
Quand on baisse le discount de 0,999 à 0,6, la politique optimale n'est plus de faire le grand tour mais de prendre le chemin le plus court pour rejoindre le carré vert, car à chaque pas le carré vert perd de sa valeur.



R(s) = -0.4

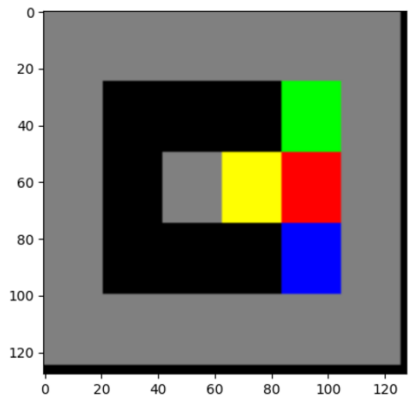
Avec une pénalité à -0,4, on observe que le joueur ne cherche plus à faire le grand tour mais prend le risque de tomber dans la case rouge en rejoignant la case verte par le chemin le plus court. Et même si le joueur se trompe et arrive sur la troisième case en bas en partant de la droite, la pénalité est si élevée qu'il reste plus rentable de faire demi-tour plutôt que de chercher à faire le grand tour.



R(s) = -2.0

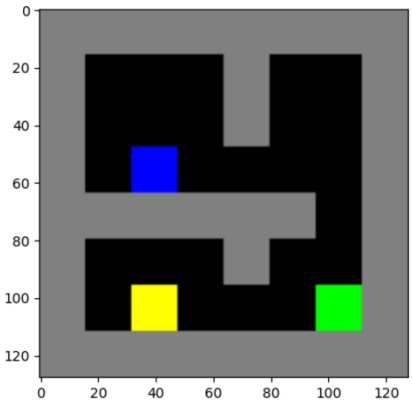
Enfin, avec une pénalité de -2, la pénalité est si élevée qu'il est plus rentable de tout de suite se suicider dans la case rouge.

<u>Plan 1 :</u>



La pénalité étant encore très faible, la stratégie optimale est toujours de continuer à aller vers le bas, et lorsque le joueur se trompe et se retrouve sur la case de gauche, alors la stratégie optimale est de récupérer la récompense (+1, carré jaune) puis d'aller vers la gauche afin de ne pas prendre le risque de tomber sur le carré rouge (-1), et ce jusqu'à ce que le joueur se trompe et se retrouve soit tout en haut soit tout en bas. Alors, il ne reste plus qu'à rejoindre le carré vert. Ainsi, on ne prend pas le risque de tomber dans le carré rouge, et la pénalité est si faible qu'on peut se permettre de faire beaucoup d'actions avant de rejoindre le carré vert.

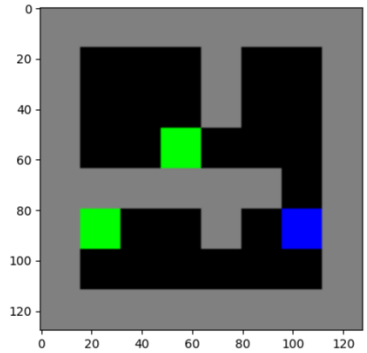
Plan 2 :



Etant donné que la pénalité est très faible, la solution consiste à faire un détour pour aller récupérer la récompense (carré jaune, +1) avant de rejoindre le carré vert.

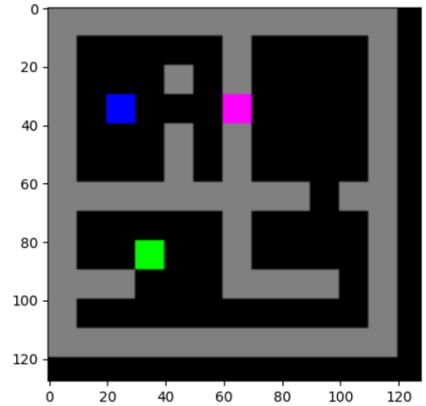
Mais quand on augmente la pénalité, par exemple à -0,4, le détour pour aller chercher la récompense devient trop coûteux et il est alors plus rentable de directement rejoindre le carré vert.

Plan 3 :



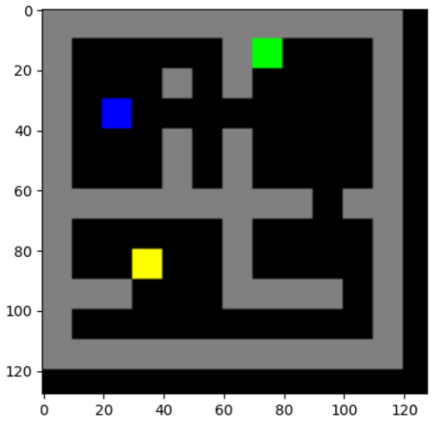
La politique optimale est de rejoindre le carré vert le plus proche, donc celui du haut.
Mais si jamais le joueur se trompe lors du premier mouvement et se retrouve à gauche de son point d'origine, alors il est plus rentable de rejoindre le carré vert du bas car c'est lui qui devient le plus proche.

Plan 4 :



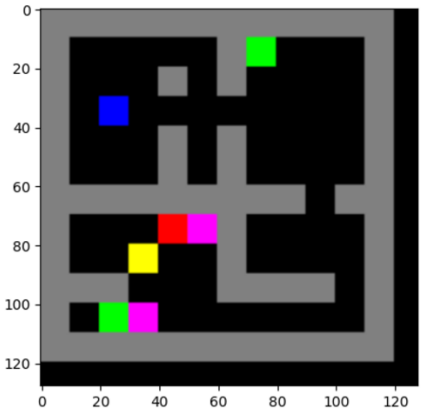
On est obligé de passer par le piège non mortel (-1) pour rejoindre le carré vert, ce qui est la politique optimale si la pénalité est faible. Mais si on augmente la pénalité (par exemple à -1), alors l'algorithme ne converge plus, car la solution optimale est de tourner indéfiniment à l'origine pour éviter le carré magenta.

Plan 5 :



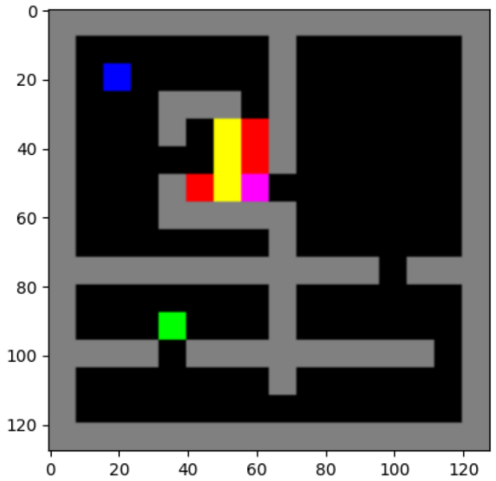
Même principe : si la pénalité est faible, alors il est plus rentable de faire le détour pour aller chercher la récompense, sinon il est plus rentable de directement rejoindre le carré vert.

Plan 6 :



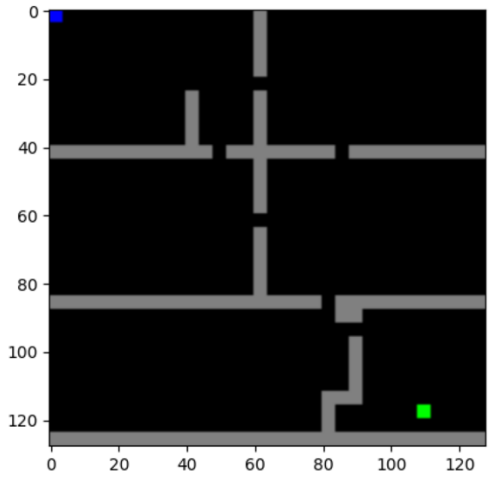
Si la pénalité est très faible (par exemple 0,01), alors la politique optimale est d'aller chercher la récompense puis de remonter chercher le carré vert en haut pour éviter le piège non mortel. Mais si elle est un peu plus élevée (par exemple -0,02), alors il n'est pas rentable d'aller chercher la récompense et il vaut mieux rejoindre le carré vert le plus proche.

Plan 7 :



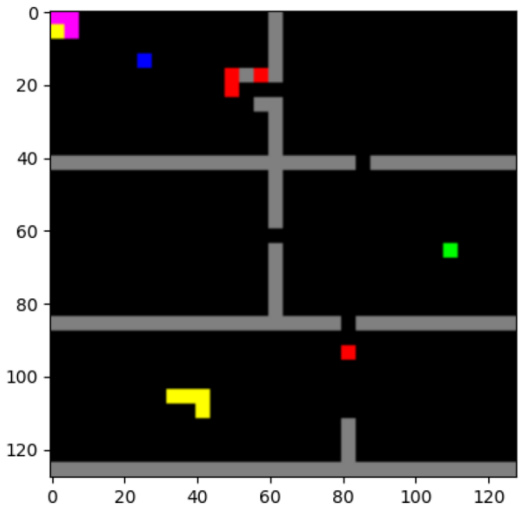
Plan très intéressant : si la pénalité est faible, le joueur prend son temps pour récupérer toutes les récompenses en minimisant le risque de tomber sur un piège mortel (cases rouges), donc le joueur vise souvent des murs ou fait des allers-retours entre cases vides et attend qu'il se trompe pour atterrir sur une récompense (cases jaunes). En effet, si le joueur prend les décisions de manière à récupérer les cases jaunes le plus rapidement possible, alors il risque de se tromper à un moment et d'atterrir sur une case jaune.
Mais si on augmente la pénalité, alors le joueur prend davantage de risques et tombe souvent sur des pièges mortels.

Plan 8 :



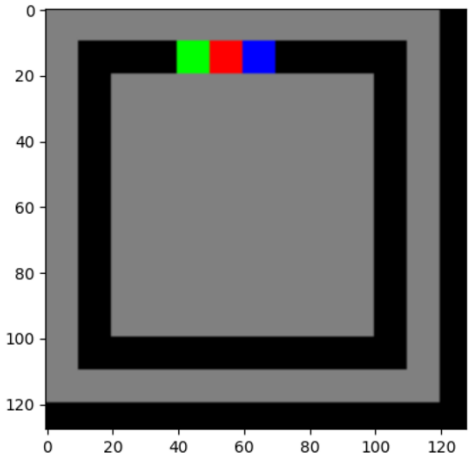
Le plus rapide est de passer par le bas mais le joueur peut se tromper et aller voir la droite et alors il peut être plus intéressant d'aller vers la droite que de faire demi-tour

Plan 9 :



On obtient le message d'erreur suivant en essayant de calculer le MDP : *RecursionError: maximum recursion depth exceeded while getting the repr of an object*.
Le MDP est trop lourd car la carte est très grande et il y a trop d'états possibles en fonction des pièges et des récompenses récupérées.

Plan 10 :



Comme on pouvait si attendre, si la pénalité est très faible, la politique optimale est de faire tout le tour pour rejoindre le carré vert, mais si la pénalité est un peu plus élevée, alors il vaut mieux se suicider directement.