

## Projet Deep learning (No free lunch) Groupe 2

Erwan Morcet, Maxime Boidin

### Données obtenues

Algos	Monte Carlo off-policy			Monte Carlo on-policy		
Jeux	grid world small	grid world large	Sokoban	grid world small	grid world large	Sokoban
Tps d'exécution	22,4928973s	NA	26,2784549s	0,4370933s	NA	0,7092018s
nombre de simulation	500	1200	500	500	1200	500
Nombre de mouvements	100	100	100	100	100	100
Autres paramètres	iteration : 200	iteration : 50	iteration : 100	NA	NA	NA

Algos	Qlearning			Policy Iteration		
Jeux	grid world small	grid world large	Sokoban	grid world small	grid world large	Sokoban
Tps d'execution	0,0167684s	2,7580465s	1,3526107s	NA	NA	NA
nombre de simulation	500	1200	1200	150	1170	NA
Nombre de mouvements	100	100	100	NA	NA	NA
autres paramètres	alpha 0.5 / gamma 0.1	alpha 0.5 / gamma 0.1	alpha 0.5 / gamma 0.5	gamma : 0.6	gamma >= 0.2	NA

Algos	Sarsa		
Jeux	grid world small	grid world large	Sokoban
Tps d'exécution	0,007065s	2,12933s	0,9551858s
nombre de simulation	500	1200	700
Nombre de mouvements	100	100	100
Autres paramètres	alpha 0.5 / gamma 0.1	alpha 0.5 / gamma 0.1	alpha 0.5 / gamma 0.5

### Observation

L'implémentation de Qlearning et Sarsa est bien plus simple que MonteCarlo pour des temps d'exécution souvent plus rapide dans le cadre de gridworld et sokoban, on peut les rendre d'autant plus rapide en ajustant le nombre d'itération, et les paramètres alpha et gamma.

Malheureusement il est difficile de comparer le policy iteration avec le reste il n'a pas été implémenté de la même façon, (structure, game state ect).

Les données obtenues sont assez minces, il est surement possible de faire de plus nombreux tests, ici j'ai seulement ajusté les valeurs au mieux pour que les algos valident le jeu le plus souvent possible.

## Conclusion

On a pu observer que lorsque l'on passait du gridworld au Sokoban qui nous rajoute une contrainte (Les caisses), que le temps de calcul des algorithmes ont drastiquement augmentés.

On peut donc en conclure qu'utiliser ces algorithmes pour trouver des solutions à des jeux plus compliqués pourraient être une mauvaise idée. Cependant, l'utilisation de ces algorithmes afin de trouver des exploitations de bugs, optimiser certains passages et potentiellement générer des niveaux de jeux simples comme le sokoban ou gridworld, pourrait être bénéfique à tout équipe de développeurs.