3I005
TME 2–4 : Projet Exploration / Exploitation

Ariana Carnielli
Yasmine Ikhelif

Table des matières

1 Introduction

Ce mini-projet s'intéresse à la problématique de l'exploitation *vs* exploration, qui consiste à choisir, parmi une quantité de ressources limitées, combien de ces ressources vont être utilisées pour explorer le problème, afin d'avoir plus de chances de trouver la meilleure solution possible, et combien seront utilisées pour exploiter cette meilleure solution trouvée. Si beaucoup de ressources sont dépensées pour l'exploration, on aura plus de chances de trouver une solution proche de l'optimale, mais moins de ressources disponibles pour son exploitation. De l'autre côté, arrêter l'exploration trop tôt peut conduire au choix d'une action sous-optimale pendant la phase d'exploitation, ce qui conduit à un gain plus petit à long terme.

Dans ce mini-projet, on illustre la problématique de l'exploitation *vs* exploration dans trois situations. La première, décrite dans la Section 2, s'intéresse à l'exemple des bandits-manchots. Il s'agit de considérer une machine à sous à N leviers, chacun ayant une probabilité de victoire différente inconnue du joueur. L'objectif est de maximiser le gain, ce qui nécessite un bon équilibre entre l'exploration des différents leviers afin d'avoir une bonne estimée de la probabilité de gain de chacun et l'exploitation du meilleur levier. L'algorithme principal permettant cet équilibre est l'algorithme UCB, dont les détails sont donnés dans la Section 2.

La deuxième situation considérée, décrite dans la Section 3, est celle du jeu de morpion, où trois stratégies de jeu sont implémentées. La première consiste dans une stratégie purement aléatoire, la deuxième est une stratégie de Monte Carlo, qui explore les

actions possibles de façon aléatoire et choisit la meilleure et la dernière, une stratégie de Monte Carlo Tree Search, qui utilise l'algorithme UCB pour optimiser l'exploration des actions possibles.

Finalement, ces mêmes algorithmes, ayant été codés de façon généraliste, sont appliqués à un jeu légèrement plus complexe que morpion, le jeu puissance 4. Les détails de l'implémentation et les résultats obtenus sont donnés dans la Section 4.

2 Bandits-manchots

2.1 Description

On considère une machine à sous avec N leviers, numérotés par les entiers de 0 à $N-1$. Chaque levier, lorsqu'il est actionné, peut donner une récompense de 0 ou 1 de façon aléatoire. On suppose que tous les leviers sont indépendants, que deux actionnements différents du même levier sont aussi indépendantes, et que la récompense du levier i suit une loi de Bernoulli de paramètre μ^i constant en temps.

On dispose de T parties pour jouer à la machine à sous. À chaque partie $t \in \{0,\dots,T-1\}$, on choisit une *action* $a_t \in \{0,\dots,N-1\}$, qui représente le levier choisi pour cette partie, et on accumule le gain obtenu avec ce levier, noté par r_t. Ainsi, r_t est une variable aléatoire suivant une loi de Bernoulli de paramètre μ^{a_t}. L'objectif est de maximiser le gain total G_T obtenu au bout de T parties, $G_T = \sum_{t=0}^{T-1} r_t$. Comme G_T est une variable aléatoire, on maximise son espérance, qui vaut

$$\mathbb{E}(G_T) = \mathbb{E}\left(\sum_{t=0}^{T-1} r_t\right) = \sum_{t=0}^{T-1} \mathbb{E}(r_t) = \sum_{t=0}^{T-1} \mu^{a_t}.$$

Ainsi, si les $\mu^0,\dots,\mu^{N-1}$ étaient connus, la stratégie maximisant son espérance serait de choisir

$$a_t = \underset{i\in\{0,\dots,N-1\}}{\operatorname{argmax}}\ \mu^i.$$

Comme les μ^i ne sont pas connus, on est confronté à un problème du type exploitation *vs* exploration, où l'exploration consiste à tester les leviers afin d'estimer leurs paramètres μ^i et l'exploitation consiste à jouer le levier avec le plus grand paramètre μ^i estimé.

3 Morpion

4 Puissance 4

5 Conclusion

Ce rapport a permis, dans la Section 2, de mettre en évidence la problématique de l'exploitation *vs* exploration à travers l'exemple des bandits-manchots, qui illustre bien l'intérêt d'un algorithme équilibrant exploration et exploitation comme l'algorithme

UCB. On a également pu remarquer, dans les Sections 3 et 4, que l'utilisation de cet algorithme dans la stratégie Monte Carlo Tree Search permet d'obtenir des joueurs très performants.