

TP : Upper Confidence Bound

Florian VANSTEENE

03/07/2017

1 Commande pour exécuter le programme :

Les paramètres sont :
— nombre de manchots
— nombre de run
— nombre d'itérations
— valeur de K

Par exemple :

```
$ python Main.py 10 10 10000 2
```

2 Comment se comporte UCB en fonction de K ?

Le K permet de jouer sur le compromis exploration/exploitation, en effet on remarque que plus K est grand, plus on explore, à l'inverse plus il est petit plus on exploite.

Par exemple en prenant $K = 2$, on remarque que la courbe du UCB est au dessus des autres et reste stable :

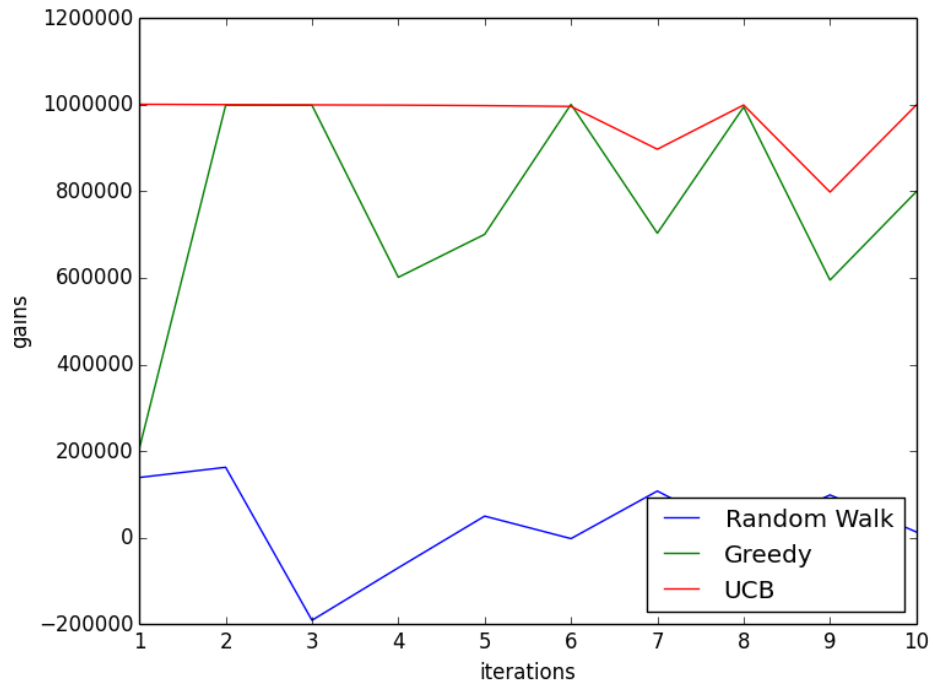


FIGURE 1 – 10 run de 10000 itérations, 20 machines et $K=2$

Au contraire, si on prend $K = 10000$, on remarque que la courbe du UCB est semblable à celle du Random Walk, on peut en déduire qu'on explore trop et on exploite pas assez ce qui nous rapproche de l'aléatoire :

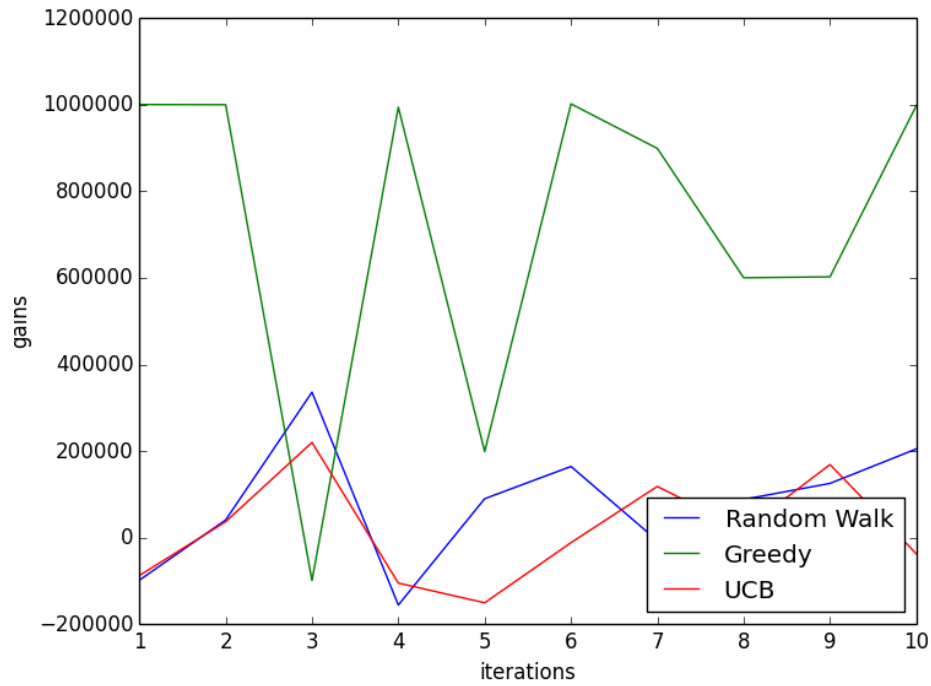


FIGURE 2 – 10 run de 10000 itérations, 20 machines et $K=10000$

3 Dans quel(s) cas la stratégie gloutonne est la meilleure ?

La stratégie gloutonne est la meilleure lorsque que la variance est faible, en effet on choisira la machine donnant le plus de gain au premier essai et on restera sur celle-ci. Si la variance est faible, le gain sera constant ou diminuera peu :

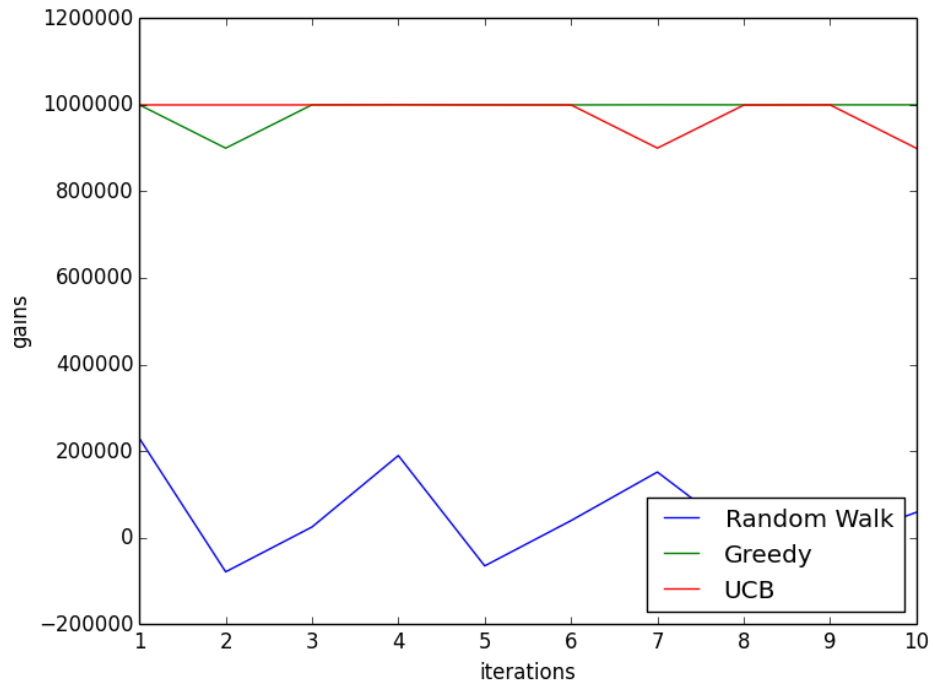


FIGURE 3 – 10 run de 10000 itérations, 20 machines et une variance entre 0 et 1

Elle sera aussi meilleure sur le court terme c'est à dire si le nombre d'itération est faible. En sélectionnant la machine avec le meilleur gain au départ, on peu espérer gagner quelques fois supplémentaire :

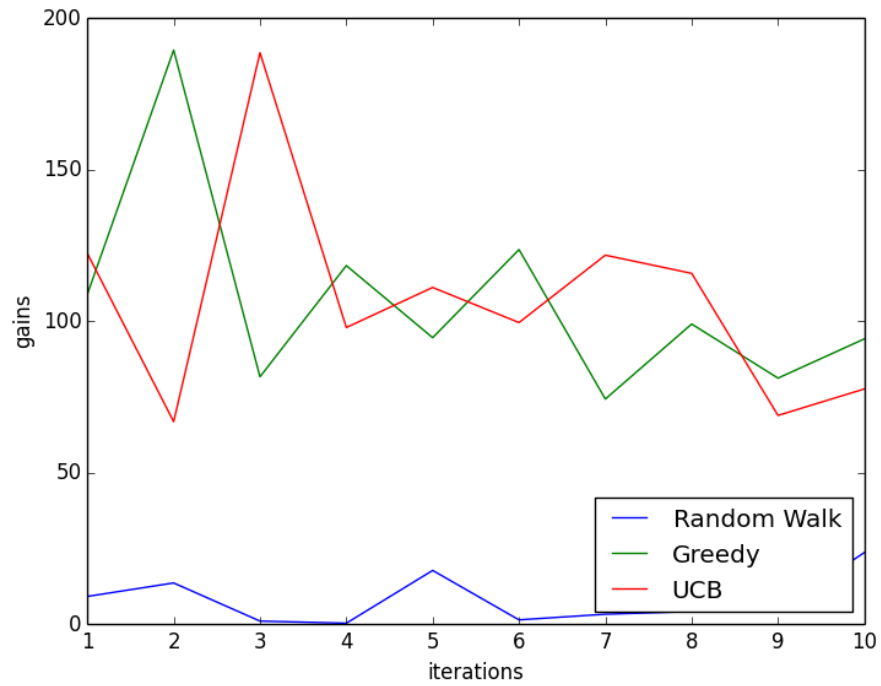


FIGURE 4 – 10 run de 2 itérations, 20 machines

4 Dans quel(s) cas la stratégie UCB est la meilleure ?

La stratégie UCB sera meilleure lorsque le nombre d'itérations est élevé, en effet le temps d'apprentissage est plus important et l'algorithme permet de changer de machine en fonction du gain :

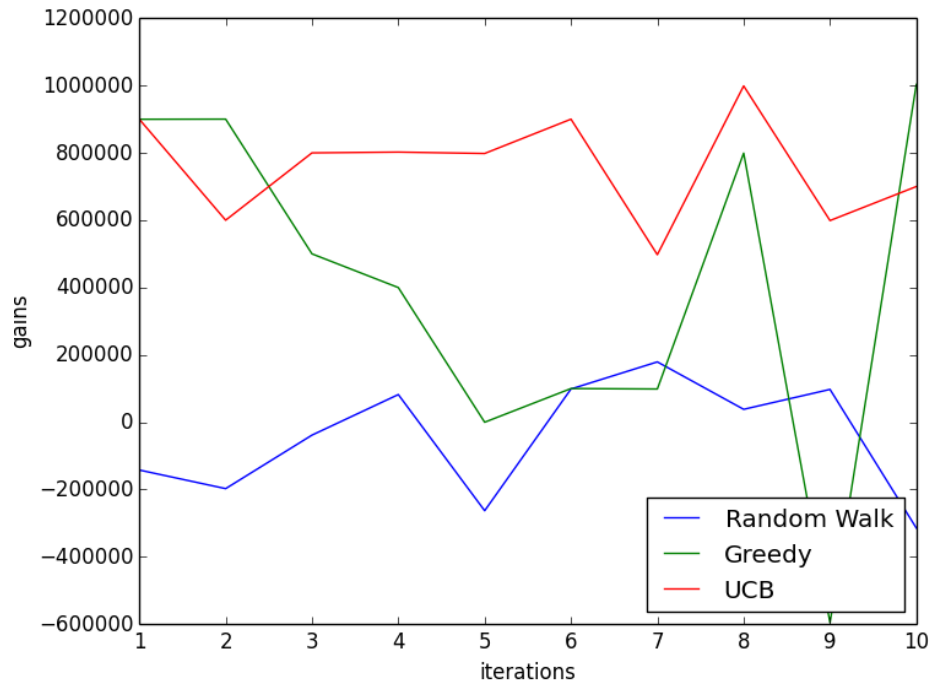


FIGURE 5 – 10 run de 100000 itérations, 5 machines

Cette stratégie est encore plus efficace lorsque le nombre de machines est important, on remarque alors que le gain reste stable entre les différentes exécutions de l'algorithme :

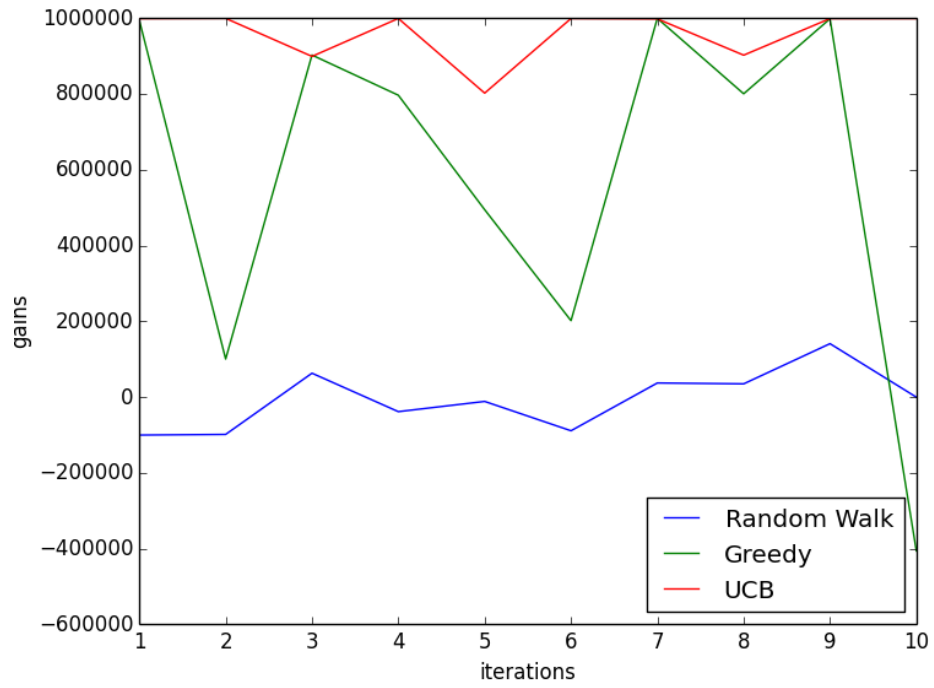


FIGURE 6 – 10 run de 100000 itérations, 50 machines