Ecole Polytechnique, Cycle Ingénieur (2A)
MAP471A - Problem solving en mathématiques appliquées
Enseignants : Leila Bassou [(mail)](mailto:leila.bassou@polytechnique.edu) (mailto:leila.bassou@polytechnique.edu) Lucas Gerin (mail) (mailto:lucas.gerin@polytechnique.edu) Teddy Pichard (mail) (mailto:teddy.pichard@polytechnique.edu) Nicole Spillane (mail) (mailto:nicole.spillane@polytechnique.edu)

Introduction à l'arrêt optimal

```python
# css style
from IPython.core.display import HTML
def css_styling():
    styles = open("./style/custom2.css").read()
    return HTML(styles)
css_styling()
```

```python
# load the libraries
import matplotlib.pyplot as plt # 2D plotting library
import numpy as np              # package for scientific computing
import random
%matplotlib inline
```

Table des matières

Introduction

L'arrêt optimal est le problème que l'on rencontre dans le contexte suivant :

- On dispose de variables aléatoires $X_1, \ldots, X_n, \ldots$ dont la loi jointe est connue
- Pour tout n il existe une fonction de gain connue et déterministe $G_n : \mathbb{R}^n \to \mathbb{R}$.
- A chaque n, on observe X_n et on doit prendre la décision :
  - d'arrêter et on gagne alors $G_n(X_1, \ldots, X_n)$.
  - de continuer.

En pratique la plupart des problèmes sont à **horizon fini** : le processus s'arrête pour un certain N (connu d'avance), et alors on est obligé de s'arrêter et de gagner G_N.

Temps d'arrêt

Dans ce contexte, le *temps d'arrêt* τ associé à une stratégie est la variable aléatoire à valeurs dans $\{1, \ldots, N\}$ qui donne l'instant où l'on s'arrête. Par hypothèse on a que pour tout l'événement

$$\{\tau = n\}$$

est une fonction de $\{X_1, \ldots, X_n\}$ (ça ne peut pas dépendre du futur!).

On cherche la stratégie qui maximise

$$\mathbb{E}[G_\tau(X_1, \ldots, X_\tau)].$$

Arrêt optimal pour le dé

On considère le problème d'arrêt optimal suivant. Soit $N \geq 1$ fixé, vous avez le droit de lancer un dé équilibré jusqu'à N fois de suite. Lorsque vous décidez de vous arrêter vous gagnez la somme indiquée sur le dé et le jeu s'arrête (si vous avez lancé le dé N fois vous gagnez la valeur du dernier tirage).

On a donc un jeu à horizon fini et

$$G(x_1, \ldots, x_n) = x_n.$$

et $X_1, X_2, \ldots, X_N$ sont les résultats des N dés, qui sont donc des variables iid uniformes sur $\{1, 2, \ldots, 6\}$.

Le but de cette partie est d'implémenter une stratégie particulière (dont on peut montrer qu'elle est optimale) et d'évaluer son efficacité.

Pour une stratégie fixée et un entier $1 \leq n \leq N$ on note $\mathcal{G}(n, N) \in \{1, 2, \ldots, 6\}$ la variable aléatoire donnée par le gain d'un joueur qui s'apprête à lancer le dé n. Ce qui nous intéresse est donc $\mathcal{G}(1, N)$: le gain du joueur qui commence à jouer.

On pose $g(n, N) = \mathbb{E}[\mathcal{G}(n, N)]$.

On considère la stratégie suivante :

- Si $n = N$ on doit lancer le dernier dé et on gagne $\mathcal{G}(N, N) = X_N$
- Sinon
    - si c'est avantageux : $X_n > g(n+1, N)$ on s'arrête ;
    - si $X_n \leq g(n+1, N)$ on continue.

On a donc par construction

$$\mathcal{G}(n, N) = X_n \mathbf{1}_{X_n > g(n+1,N)} + \mathcal{G}(n+1, N) \mathbf{1}_{X_n \leq g(n+1,N)} \qquad (\star)$$

Remark. Cette stratégie est en fait l'application à notre problème particulier de la stratégie appelée [Enveloppe de Snell (https://fr.wikipedia.org/wiki/Enveloppe_de_Snell)](https://fr.wikipedia.org/wiki/Enveloppe_de_Snell). On peut effectivement montrer que c'est la stratégie qui maximise l'espérance.

Do it yourself.

1. Ecrire une fonction `EsperanceMaxDe(v)` qui prend en entrée un réel $v \in [0, 6]$ et renvoie
$$\mathbb{E}[\max(X_1, v)].$$
(Pour vérifier votre code : vous devez trouver `EsperanceMaxDe(1)=3.5` et `EsperanceMaxDe(4)=4.5` .)
2. En passant à l'espérance dans l'équation $(\star)$, trouver une relation de récurrence entre $g(n, N)$ et $g(n+1, N)$. En utilisant `EsperanceMaxDe()` , coder une fonction `GainOptimal(n,N)` qui calcule $g(n, N)$.
(Pour vérifier votre code : pour tout N on doit trouver $g(N-1, N) = 4,25$.)

Answers.

On souhaite comparer numériquement l'algorithme optimal avec 2 benchmark :

Algorithme Naïf :

- Si $X_n = 6$ on s'arrête, sinon on relance le dé.

Oracle :

- On suppose que l'on connaît à l'avance les résultats $X_1, \dots, X_N$. On s'arrête quand on tombe sur $\max_{1 \le n \le N} X_n$.

Do it yourself.

1. Pour l'algorithme naïf et l'oracle, déterminer le gain moyen.
Indication : pour l'oracle, on rappelle que si M est une variable aléatoire entière positive alors
$$\mathbb{E}[M] = \sum_{k \ge 1} \mathbb{P}(M \ge k) = \sum_{k \ge 1} \Big(1 - \mathbb{P}(M \le k-1)\Big).$$
2. Tracer sur le même graphique les gains pour l'algorithme optimal, l'algorithme naïf et l'oracle pour $n = 1, \dots, N$ avec $N = 20$.

Answers.

1.

Do it yourself. On considère à nouveau l'algorithme optimal. Tracer pour N fixé la courbe qui donne pour chaque $n \leq N$ le résultat au-dessus duquel il faut arrêter de lancer le dé.

Simulations

On souhaite maintenant utiliser des simulations pour afficher des réalisations de la variable aléatoire $\mathcal{G}(0, N)$.

Do it yourself.

1. Modifier le code précédent pour simuler des réalisations de la variable aléatoire $\mathcal{G}(0, N)$.
2. Afficher le résultat de 1000 simulations de $\mathcal{G}(0, 8)$ dans un histogramme.

Answers.

Parking optimal

On s'intéresse à un autre problème d'arrêt optimal : le problème du parking sur $\mathbb{N}_*$.

On suppose que les places $\{1, 2, 3, \dots\}$ sont occupées indépendamment avec probabilité p par des voitures. Notre objectif est de se garer le plus proche possible de K (un entier fixé et connu) de sorte que le coût pour se garer à la place i est

$$F(|i - K|),$$

où F est une fonction croissante positive.

Le problème est que l'on ne peut observer les places occupées que une à une. On cherche donc le temps d'arrêt τ qui minimise

$$\mathbb{E}[F(|\tau - K|)].$$

Pour une stratégie fixée et une position $1 \leq i$ on note

$$\mathcal{P}(i, K)$$

la variable aléatoire donnée par la position à laquelle je vais me garer juste avant d'observer si i est libre ou pas. On note $G(i, K)$ le coût que je vais subir :

$$G(i, K) = \mathbb{E}\left[F(|\mathcal{P}(i, K) - K|)\right],$$

$$g(i, K) = \mathbb{E}\left[G(i, K)\right].$$

On considère la stratégie suivante :

- Si $i \geq K$ on se gare dès que possible
- Si $i < K$ on se gare en i si
  - la place est vide
  - ET c'est avantageux : $F(|i-K|) \leq g(i+1, K)$.

Do it yourself. 1. Pour la stratégie ci-dessus, déterminer $\mathbb{P}(\mathcal{P}(i,K)=j)$ dans le cas $i \geq K$. En déduire $g(K,K)$. 2. Pour $i \geq 1$ on note $Z_i = 0$ (resp. $Z_i = 1$) si la place i est libre (resp.occupée). Pour $i < K$ écrire une relation de récurrence entre $G(i,K), G(i+1,K), Z_i$ et $g(i+1,K)$.

Answers.

1.
2.

Do it yourself.

1. Ecrire une fonction qui calcule $c(i,K,p)$. (On prendra $F(x)=|x|$.)
2. Pour $K=20$ tracer une courbe qui permette de déterminer à partir de quand doit-on se garer à la première place disponible.

Problème des secrétaires

On modélise maintenant un autre problème d'arrêt optimal.

On considère une suite de variables aléatoires $X_1, \dots, X_N$ i.i.d. de loi inconnue (*), on observe successivement les X_i et on cherche le temps d'arrêt qui maximise

$$\mathbb{P}(X_\tau = \max\{X_i\}).$$

Pour $1 \leq k \leq N$ fixé on considère la stratégie suivante :

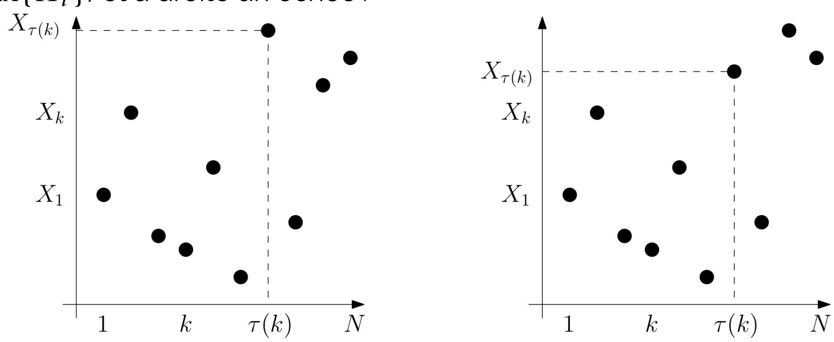
- On observe $X_1, \dots, X_k$ sans s'arrêter, et on stocke en mémoire la plus grande valeur $M_k = \max_{1 \leq i \leq k}\{X_i\}$.
- Ensuite pour $j > k$ on s'arrête dès que $X_j > M_k$. (Si cet événement n'arrive jamais alors on s'arrête en N.)

Autrement dit on introduit le temps d'arrêt

$$\tau(k) = \min\{j > k, X_j > \max_{1\leq i\leq k}\{X_i\}\},$$

avec la convention $\min\emptyset = N$.

Voici un schéma qui illustre les notations. A gauche un exemple de succès de la stratégie (on a $X_{\tau(k)} = \max\{X_i\}$) et à droite un échec :



Le but est de calculer

$$p_N(k) = \mathbb{P}(\text{ La stratégie est gagnante }) = \mathbb{P}(X_{\tau(k)} = \max\{X_i\})$$

(*) On suppose pour simplifier que la loi est à densité, ce qui assure que tous les Z_i sont distincts.

Do it yourself.

1. Pour $j > k$ calculer $\mathbb{P}(\tau(k) = j; X_{\tau(k)} = \max\{X_i\})$.
2. En déduire une expression de $p_N(k)$.
3. Pour différentes valeurs de N, tracer $k \mapsto p_N(k)$.

Answers.

1.
2.

Sur les courbes de $p_N(k)$ que vous avez obtenues vous devez observer que $p_N(k)$ semble toujours maximale pour k de l'ordre de $c \times N$ où c est une certaine constante dans $(0, 1)$. (Pour observer cela le mieux est de tracer $c \mapsto p_N(\lfloor cN \rfloor)$.

Do it yourself. **Question théorique.** En analysant votre expression de $p_N(k)$ déterminer la valeur du c optimal.

Answers.