Sorbonne Université

MOGPL

Projet :
Dice battle

HUANG Qijia
HUANG Weiqing

IMA
Année 2019/2020

Table des matières

Introduction

Ce projet consiste de trouver un stratégie optimal du jeu Dice battle. Deux joueurs choisit á lancer entre 1 et D dés á chaque tour et le nombre de points marqué á 1 points si lún de dés au moins tombe sur 1. Dans le cas contraire, c'est le somme de dés. Le but est de atteintre au moins N points.

L'objectifs de ce projet sont la mise en pratique des algorithme ce qu'on a vu dans le cour(programmation linéaire, programmation dynamique etc). Dans le premier temps, on vas étudier la probabilité de marquer un certain nombre de points lorsque l'on lance les dés. Dans le deuxième temps,on vas étudier le cas séquentielle, c'est á dire que le joueur 1 lance d'abord le dés et ensuite joueur 2 lence le dés et ainsi suite. Dans le dernière temps, on vas étudier le cas stiultané, donc deux joueurs lancent le dés stimutanément.

1 Probabiliste

1.1 Question 1

Explique le réccurence :

$$Q(d,k) = \sum_{j=2}^{6} \frac{Q(d-1,k-j)}{5}$$

$Q(d,k)$ signifie la probabilité d'obtenir k points en jetant d dés sachant qu'aucun dé n'est tombé sur 1. Supposons que on a déjà lancé d-1 dés, donc on a $Q(d-1,k')$ oú k' est le point marqué en jetant $d-1$ dés. Pour le d ème dés, par définition, on a $\frac{1}{5}$ de chance tomber sur un des valeurs entre $\{2,3,4,5,6\}$. Donc la moyenne de k :

$$k = \frac{(k'+2)+(k'+3)+(k'+4)+(k'+5)+(k'+6)}{5}$$

On remplace k' par k en ajoutant une variable j, donc on a :

$$k = \sum_{j=2}^{6} \frac{(k-j)}{5}$$

Donc on a bien obtenu la récurrence.

Algorithm 1 Probabilite

Require: $d \geq 0$ **and** $k \geq 0$

function P(d : integer, k : k : integer)
    if $k = 0$ **then**
        return $1 - (\frac{5}{6})^d$
    if $((k \geq 2)$ **and** $(k \leq 2d-1))$ **or** $(k \geq 6d)$ **then**
        return 0
    return $(\frac{5}{6})^d \times$ Q(d,k)

function Q(d : integer, k : integer)
    if $k \leq 0$ **or** $k < 2d$ **or** $k > 6d$ **then**
        return 0
    if $d == 1$ **then**
        return $\frac{1}{5}$
    for $j = 2$ to 6 **do**
        $res \leftarrow res +$ Q(d-1,k-j)
    return $\frac{res}{5}$

2 Variant séquentielle

2.1 Stragégie aveugle

2.1.1 Question2

le nombre de dés $d^*(D)$ maximisant l'espérance du nombre de points peut être calculé par :

$$d^*(D) = \arg\max_d 4d(\frac{5}{6})^d + 1 - (\frac{5}{6})^d$$

2.1.2 Question 3

Montrer que $d^*(D)$ n'est pas toujour le choix optimal.

Soit un instance de jeu dont $D = 10, N = 100$. On considère que un moment le point marqué par joueur 1 est 98 et le points marqué par joueur 2 est j et c'est le tour de joueur 1. Donc joueur 1 a le choix entre 1 à 10 dés á lancer.Le nombre de dés $d^*(D)$ maximisant l'espérance du nombre de points est :

$$d^*(10) = \arg\max_d 4d(\frac{5}{6})^d + 1 - (\frac{5}{6})^d = 6$$

Par contre, le choix optimale est $d = 1$ car il minimise le probabilité de marquer juste 1 point dans ce tour(Joueur 1 ne peut pas gangner directement dans ce tour s'il marquera 1 point). Donc $d^*(D)$ n'est pas toujour un choix optimal.

2.2 Programmation dynamic

2.2.1 Question 4

La formule de récurrence pour calculer l'espérance de gain $EG(i,j)$ en un état (i,j) est :

$$EG(i,j) = -\min_{1\leq d\leq D} \sum_{k=1}^{6d} E(j, i+k)$$

L'initialisation de récurrence :

$$EG(i,j) = \begin{cases} 1 & \text{si } i \geq N \\ -1 & \text{si } j \geq N \end{cases}$$

2.2.2 Question 5

Le stratégie optimale s'agit de trouver un nombre de dés d qui permet de maximiser l'espérance de gain du joueur1 lorsque le joueur1 a marqué i points et le joueur2 a marqué 2 points. Par la question précedent, on a obtenir dans le cas optimal, l'espérance maximale du joueur 1.Donc il suffit de trouver tel valeur d qui proveque ce résultat :

$$d^*_{i,j} = -\arg\min_{1\leq d\leq D} \sum_{k=1}^{6d} E(j, i+k)$$

2.2.3 Question 6

Si on supposait que le nombre de points marqué est 0 si on obtient au moins 1 alors notre algorithme peut jamais s'arêter(car la récurrence commencer par $(0,0)$ et s'arrête soit sur $(100, j)$ soit sur $(i, 100)$, comme k peut être 0, il peut s'exécuter infinitivement). Il a donc une difficulté supplémentaire sur la terminaison.

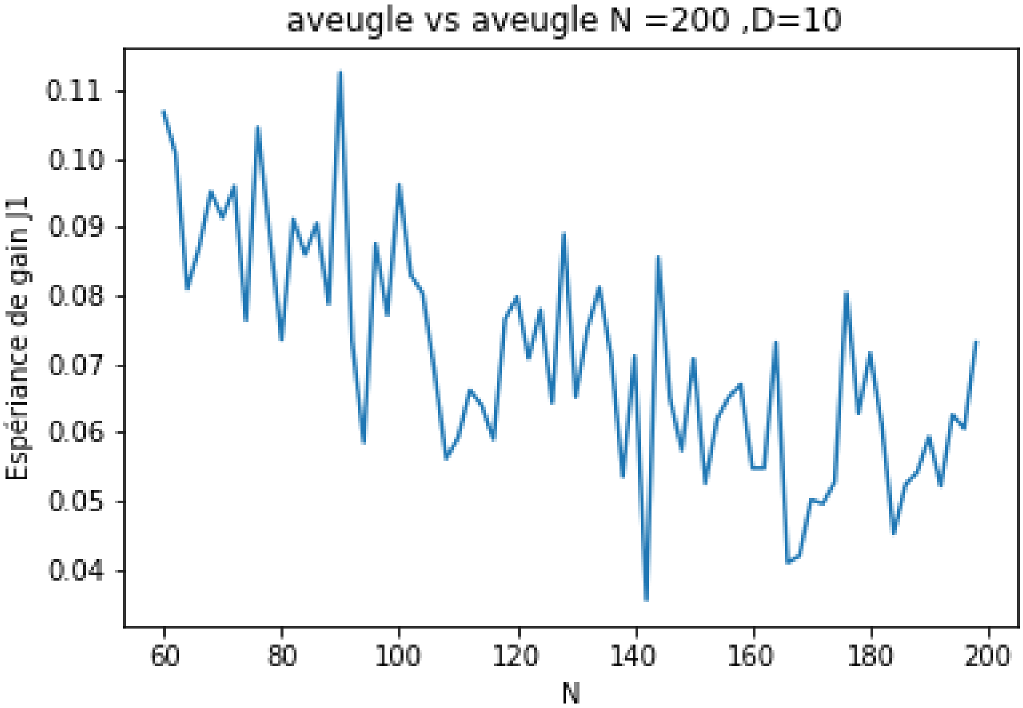
2.3 Mise en oeuvre

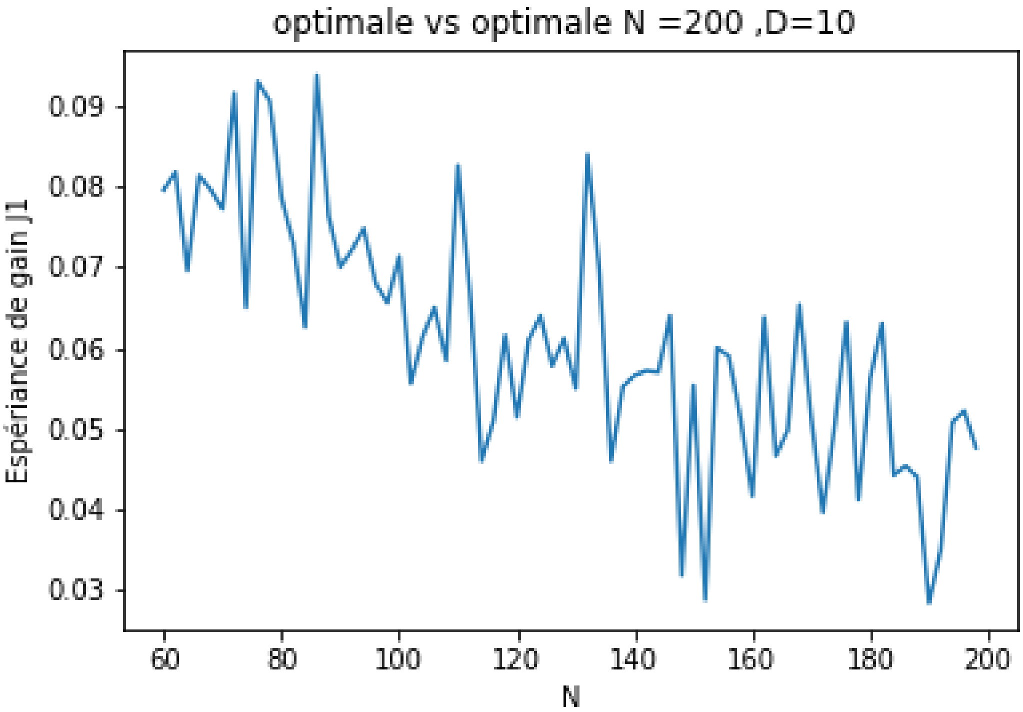
2.3.1 Question 7

Algorithm 2 Stratégie

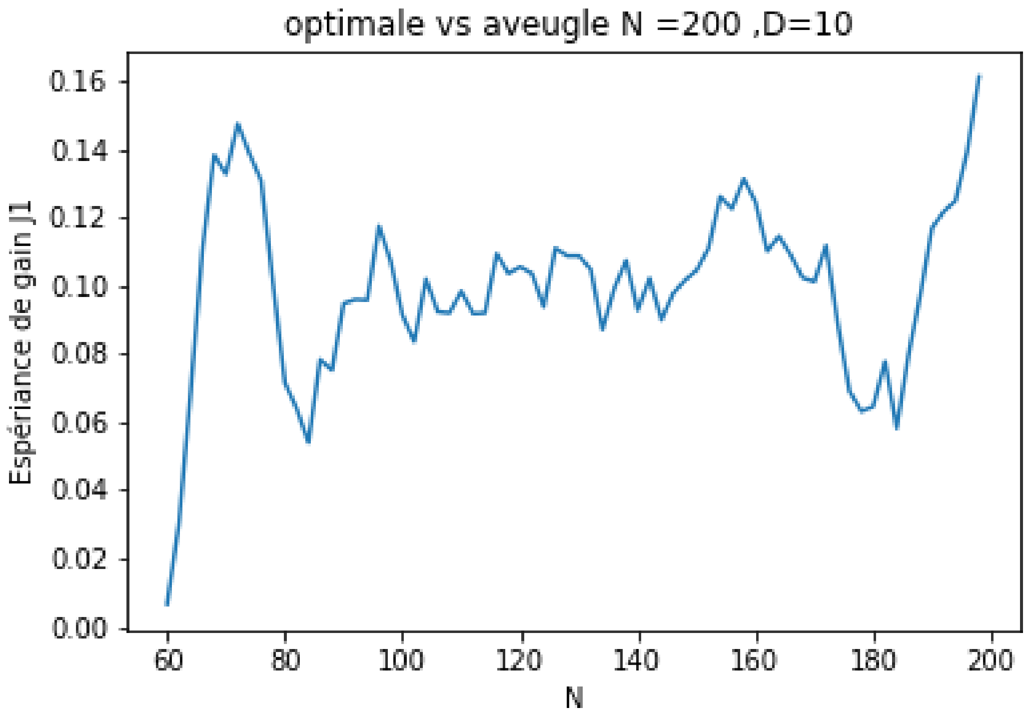
function STRATÉGIE AVEUGLE(i : integer, j :integer D : integer N :integer)
    res : D integer tableau filled with 0
    for $d = 1$ to $D + 1$ **do**
        $res[d] \leftarrow 4d(\frac{5}{6})^d + 1 - (\frac{5}{6})^d$
    return argmax res
function STRATÉGIE OPTIMAL(i : integer, j :integer D : integer N :integer)
    E : $N \times N \times 2$ integer matrix filled with -2. The first value of E[i][j] is the esperance and the second value is the quantity of dice.
    if $i \geq N$ **then**
        return 1,0
    if $j \geq N$ **then**
        return 0,0
    if $E[i][j][0]! = -2$ **then**
        return $E[i][j]$
    else
        $res_{max} \leftarrow -\infty$
        $d_{optimal} \leftarrow 0$
        for $d = 1$ to $D + 1$ **do**
            $cum \leftarrow = 0$
            for $k = 1$ to $6d + 1$ **do**
                if $i + k \leq N$ **then**
                    $cum \leftarrow cum +$ P(d-1,k-j) $\times -1$
                else
                    if $E[j][i + k][0] == -2$ **then**
                        $E[j][i + k] \leftarrow$ STRATÉGIE OPTUMAL(j,i+k,D,N)
                    $cum \leftarrow cum +$ P(d,k) $\times E[j][i + k][0]$
            if $res_{max} \leq -cum$ **then**
                $res_{max} \leftarrow -cum$
                $d_{optimale} \leftarrow d$
    $E[i][j] \leftarrow res_{max}, d_{optimal}$
    return $E[i][j]$

2.3.2 Question8

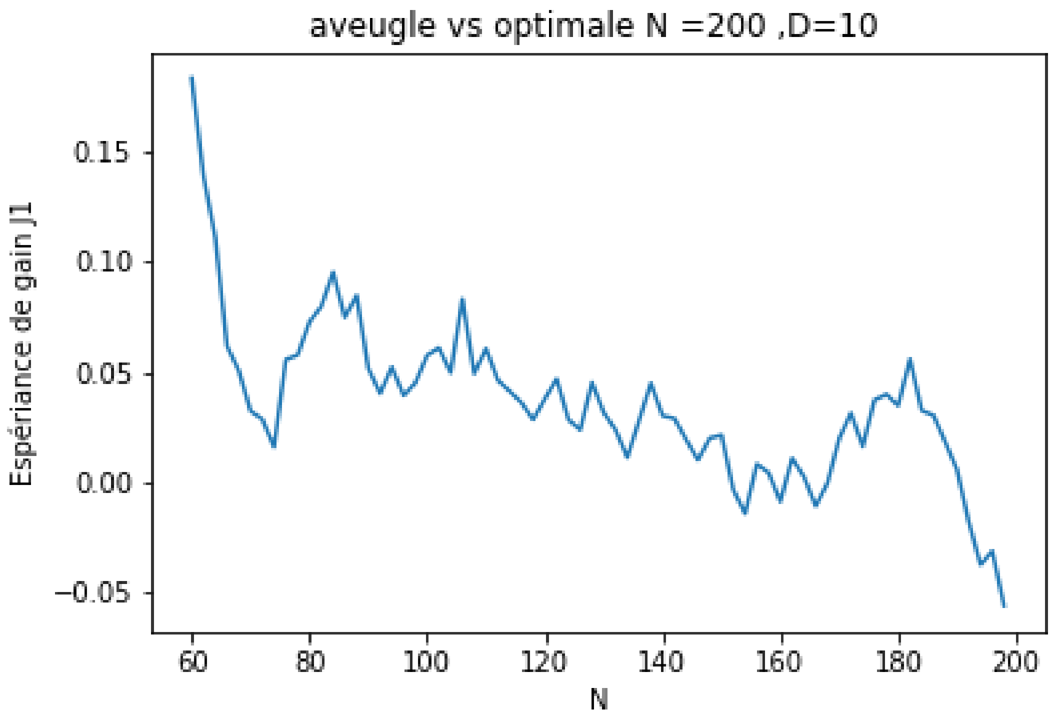




On voit que comme le stratégie utilisé pour les deux joueur est le même,plus N est grand,plus l'espérenrance de joueur1 proche de 0



Lorsque le joueur 1 choisit le stratégie optimal contre le joueur 2 en choisisant le stratégie aveugle, l'espérance de gain de joueur 1 est proche de 0.11. Le joueur 1 a l'avantage.



Lorsque le joueur 1 choisit le stratégie aveugle contre le joueur 2 en choisisant le stratégie aveugle,plus N est grand l'espérance de gain de joueur 1 est plus petite. Le joueur 2 a l'avantage quand N est assez grand car joueur 1 jouer en premier.

2.3.3 Question9

Algorithm 3 Stratégie

1: **function** STRATÉGIE ALÉATOIRE(i : integer, j :integer D : integer N :integer)
2: r : random vlue between 0 and 1
3: **return** $\lceil r \times D \rceil$

3 Variante simultané

3.0.1 Question 10

$$EG_1(d_1, d_2) = \sum_{k1=1}^{6d_1} \sum_{k2=1}^{6d_2} p(d_1, k_1) p(d_2, k_2) \frac{\sqrt{(k_1 - k_2)^2}}{k_1 - k_2}$$

la matrice des gains pour D = 3 :

J_1 - J_2	d_2=1	d_2=2	d_2=3
d_1=1	0	-0.375	-0.22685185
d_1=2	0.375	0	-0.19881687
d_1=3	0.22685185	0.19881687	0

3.0.2 Question 11

si le J_2 connait [$p_1(1)$, $p_1(2)$, $p_1(2)$,...,$p_1(D)$]

si J_2 joue j : son expériance de gain est E(j) = -$\sum_{i}^{D} EG_1(i, j)$

$$\begin{cases} p_2(\arg\max_i(E(j))) = 1 \\ p_2(j) = 0, j \neq \arg\max_i(E(j)) \end{cases}$$

3.0.3 Question 12

J_1 doit choisir sa stradégie [$p_1(1)$, $p_1(2)$, $p_1(2)$,...,$p_1(D)$] de manière à minimiser la maximum de l'expérience de gain de J_2. Soit α la maximum de l'expérience de gain de J_2

$$\begin{cases} \min \alpha \\ sc \begin{cases} \alpha \geq -\sum\limits_{i}^{D} EG_1(i,1)p_1(i) \\ \alpha \geq -\sum\limits_{i}^{D} EG_1(i,2)p_1(i) \\ ... \\ \alpha \geq -\sum\limits_{i}^{D} EG_1(i,D)p_1(i) \\ \sum\limits_{i}^{D} p_1(i) = 1 \\ p_1(1), p_1(2), p_1(3), ..., p_1(D) \geq 0 \end{cases} \end{cases}$$