# Rapport de projet
# Modélisation, optimisation, graphes, programmation linéaire
# **Dice Battle**

Merrouche, Aymen        Sidhoum, Imad

Année universitaire : 2019-2020

# Table des matières

# Chapitre 1

# Introduction

Dans ce Projet, nous allons étudier un jeu à somme nulle et nous essayerons de trouver une stratégie optimale pour ce jeu en utilisant les notions de programmation dynamique et de programmes linéaires.

Le principe de ce jeu est assez simple. Il consiste en un duel de lancés de dés entre deux joueurs, le but du jeu étant d'être le premier joueur à atteindre un certain nombre $N$. À chaque round du jeu, tout joueur doit choisir un nombre de dés à lancer entre 1 et $D$, si au moins un des $d$ dés tombe sur 1 le score du joueur durant cette manche est de 1, si aucun des $d$ dés ne tombe sur 1 son score est égal à la somme des dés. Nous étudierons deux variantes du jeu, une séquentielle où les joueurs jouent à tour de rôle et une autre simultanée.

# Chapitre 2

# Probabilités :

## 2.1 Évaluation de la probabilité de marquer un certain nombre de points lorsque l'on lance les dés :

Soit $D$ le nombre maximal de dés à lancer :
Un joueur choisi de lancer $d$ dés avec $1 \leq d \leq D$, soit $P(d,k)$ la probabilité qu'un joueur obtienne $k$ points en lançant $d$ dés, deux situation sont à considérer :

— Au moins un des $d$ dés tombe sur 1, et donc la joueur obtient un score $k$ égal à 1 point :

$$P(d,1) = P(\text{"}au\ moins\ un\ dé\ tombe\ sur\ 1\text{"}) = 1 - P(\text{"}aucun\ dé\ ne\ tombe\ sur\ 1\text{"}) = 1 - (\frac{5}{6})^d$$

— Aucun des $d$ dés ne tombe sur 1, et donc le joueur obtient un score $k$ égal à la somme des dés (avec $2d \leq k \leq 6d$ : car aucun des $d$ dés qu'il lance ne tombe sur 1 et donc il obtient au moins 2 points pour chaque dé, et au plus 6 points pour chaque dé) :

$P(d,k) = P(\text{"}aucun\ dé\ ne\ tombe\ sur\ 1\text{"} \cap \text{"}la\ somme\ des\ d\ dés\ vaut\ k\text{"}) = P(\text{"}aucun\ dé\ ne\ tombe\ sur\ 1\text{"}) \times$
$P(\text{"}la\ somme\ des\ d\ dés\ vaut\ k\text{"} \mid \text{"}aucun\ dé\ ne\ tombe\ sur\ 1\text{"}) = (\frac{5}{6})^d \times Q(d,k)$

Le terme $Q(d,k)$ peut être calculé selon la formule de récurrence :

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

$Q(d,k)$ est la probabilité que la somme des $d$ dés vaut $k$, sachant qu'aucun dé n'est tombé sur 1. Supposons qu'au premier lancé la joueur obtienne $j$ point, ce $j$ doit obligatoirement être $2 \leq j \leq 6$ puisque aucun dé ne tombe sur 1. On a donc 5 cas à considérer $(\{j = 2\}, .., \{j = 6\})$. $(k - j)$ représente le gain obtenu par le joueur au $d - 1$ lancés suivants. On applique la loi des probabilités totales :

$Q(d,k) = \sum_{j=2}^{6} [P(\text{"}obtenir\ j\ points\ au\ premier\ lancé\text{"}) \times P(\text{"}la\ somme\ des\ d\ dés\ vaut\ k\text{"} \mid$
$\text{"}aucun\ dé\ ne\ tombe\ sur\ 1\text{"} \mid \text{"}obtenir\ j\ points\ au\ premier\ lancé\text{"})]$
$= \sum_{j=2}^{6} [P(\text{"}obtenir\ j\ points\ au\ premier\ lancé\text{"}) \times P(\text{"}obtenir\ k{-}j\ points\ au\ d{-}1\ lancés\ de\ dé\ suivants\text{"} \mid$
$\text{"}obtenir\ j\ points\ au\ premier\ lancé\text{"})]$
$= \sum_{j=2}^{6} \frac{1}{5} \times Q(d-1,k-j) = \sum_{j=2}^{6} \frac{Q(d-1,k-j)}{5}$

Pour initialiser la récurrence :

- $Q(d, k) = 0$ si $k < 2d$ ou $k > 6d$.
- $Q(1, k) = P(k \mid k \neq 1) = \frac{1}{5}$.

**Implémentation : méthode P(D) , Cette fonction retourne la matrice des** $P(d, k)$

# Chapitre 3

# Variante séquentielle :

## 3.1 Stratégie aveugle :

Cette stratégie consiste à lancer un nombre de dés $d$ qui maximise notre espérance de gain (nombre de points $k$ marqués) :
Pour un $d$ fixé :

| Gain | $k = 1$ | $2d \leq k \leq 6d$ |
|---|---|---|
| P(Gain) | $1 - (\frac{5}{6})^d$ | $(\frac{5}{6})^d$ |

Et donc l'espérance du nombre de points obtenus en lançant $d$ dés est de :

$$EP(d) = 1 \times (1 - (\frac{5}{6})^d) + \frac{6d + 2d}{2} \times (\frac{5}{6})^d$$

Le nombre de dés maximisant le nombre de dés maximisant le nombre de points de points est donc $d^*(D) = \underset{d}{argmax}[1 - (\frac{5}{6})^d + 4d \times (\frac{5}{6})^d]$

### 3.1.1 Cette stratégie est-elle optimale ?

Cette stratégie n'est pas toujours la meilleure, il existe des situations où il vaut mieux jouer autrement, par exemple :
- $N = 100$
- $D = 10$

Le joueur 1 a accumulé 98 points et c'est à lui de joueur. Selon cette stratégie aveugle il décide de jeter $d^*(D) = 6$ pour $D = 10$ :
- Le joueur a une probabilité de $(\frac{5}{6})^6$ de gagner (ne pas obtenir 1, et donc au minimum $2 \times 6 = 12$ et $98 + 12 \geq 100$).
- Alors que si le joueur décide de ne pas suivre cette stratégie et de lancer un seul dé, la probabilité pour lui de gagner serait de $\frac{5}{6}$ (obtenir de 2 à 6 points).

Puisque $\frac{5}{6} > (\frac{5}{6})^6$, il n'a pas intérêt à jouer selon cette stratégie aveugle qui n'est donc pas optimale.

**Implémentation : méthode maxD(D) , qui renvoie la valeur de $d$ maximisant** $EP(d)$

## 3.2 Programmation dynamique :

Dans cette partie on s'intéresse à trouver une stratégie optimale pour la variante séquentielle de ce jeu en utilisant la programmation dynamique. On note par $(i, j)$ l'état du jeu où le premier joueur à accumulé $i$ points et où le deuxième joueur à accumulé $j$ points et c'est au premier joueur de jouer. Supposons que les deux joueurs jouent de façon optimale (même si le premier joueur à un avantage puisqu'il commence), on note par $EG(i, j)$ l'espérance de gain du joueur 1 dans cet état.

### 3.2.1 Formule de récurrence pour le calcul de $EG(i, j)$ l'espérance de gain du joueur 1 dans l'état $(i, j)$ :

Nous sommes dans l'état $(i, j)$ et c'est au joueur 1 de jouer : Supposons que le premier joueur décide de jouer $d_1$ dés et obtient un score de $k_1$, nous évoluons donc vers l'état $(i + k_1, j)$ et c'est au deuxième joueur de jouer, ce dernier joue aussi de façon optimale, il va donc jouer un $d_2$ qui va maximiser son espérance de gain et donc minimiser l'espérance de gain du premier joueur :

$$\underset{d_2}{min}[\sum_{k_2=1}^{6d_2} p(d_2, k_2) \times EG(i + k_1, j + k_2)]$$

Le premier joueur doit donc choisir le $d_1$ qui maximisera son espérance de gain après la réponse optimale du deuxième joueur, on en déduit la formule de récurrence :

$$EG_1(i, j) = \underset{d_1}{max}[\sum_{k_1=1}^{6d_1} p(d_1, k_1) \times \underset{d_2}{min}[\sum_{k_2=1}^{6d_2} p(d_2, k_2) \times EG(i + k_1, j + k_2)]]$$

$\underset{d_2}{min}[\sum_{k_2=1}^{6d_2} p(d_2, k_2) \times EG(i + k_1, j + k_2)]$ correspond à l'étape du jeu où le deuxième joueur à accumulé $j$ points, le premier à accumulé $i + k_1$ points et c'est au deuxième joueur de jouer ce qui représente l'espérance de gain du deuxième joueur dans l'état $(j, i + k_1)$, puisque le jeu est à somme nulle ça correspond à $-EG(j, i + k_1)$, la formule de récurrence peut donc s'écrie aussi sous la forme :

$$EG_1(i, j) = \underset{d_1}{max}[\sum_{k_1=1}^{6d_1} p(d_1, k_1) \times -EG(j, i + k_1)]$$

Initialisation de la récurrence :
Le nombre de points d'un joueur ne peut jamais dépasser $N - 1 + 6D$, ce qui correspond au cas où le joueur a accumulé $N - 1$ points il joue $D$ dés et obtient le score maximal de $6D$.

— $EG(i, j) = 1 \ \ \forall i \in \{N, N - 1 + 6D\} \ \ \forall j \in \{0, N - 1 + 6D\}$ c'est au joueur 1 de jouer donc c'est le joueur 2 qui à produit cet état, au moment où le joueur 2 a joué le joueur 1 avait déjà gagné.

— $EG(i, j) = -1 \ \ \forall i \in \{0, N - 1\} \ \ \forall j \in \{N, N - 1 + 6D\}$ le deuxième joueur a cumulé $N$ points et c'est au premier joueur de jouer et il n'a pas encore gagné, donc le joueur 2 gagne.

— $EG(N - 1, j) = 1 \ \ \forall j \in \{0, N - 1\}$ c'est au joueur 1 de jouer donc c'est le joueur 2 qui à produit cet état et il n'a pas encore gagné, le joueur 1 va marquer au moins 1 points ce qui lui suffit pour gagner (On peut ne pas initialiser ces cases elles seront calculées par la formule de récurrence)

### 3.2.2 Adaptation de l'algorithme pour retourner une stratégie optimale :

Il suffit de sauvegarder à chaque étape du remplissage de la matrice $EG$ le $d$ optimale, par exemple si on est dans l'étape de calcul de $EG(i, j)$ sauvegarder le $d_1$ qui maximise $\sum_{k_1=1}^{6d_1} p(d_1, k_1) \times -EG(j, i + k_1)$, c'est la stratégie optimale à jouer si on est dans l'état $(i, j)$.

### 3.2.3 Explication de la difficulté algorithmique supplémentaire si le score était de 0 au lieu de 1 si au moins un dé tombe sur 1 :

Dans le cas où un joueur peut obtenir un score nul le calcul récursif de l'espérance $EG(i, j)$ va dépendre de cette valeur elle même. Et donc $EG(i, j)$ va être un terme qui va entrer dans le calcul de $EG(i, j)$ lui même ce qui provoquera un bouclage.

## 3.3 Mise en œuvre :

### 3.3.1 Évaluations expérimentales :

On dessine les courbes des gains cumulées des joueurs pour 100 parties en alternant les stratégies et en variant les valeurs de $D$ et de $N$.

**Stratégie optimale vs Stratégie aveugle :**

**Le joueur optimale est en bleu.**

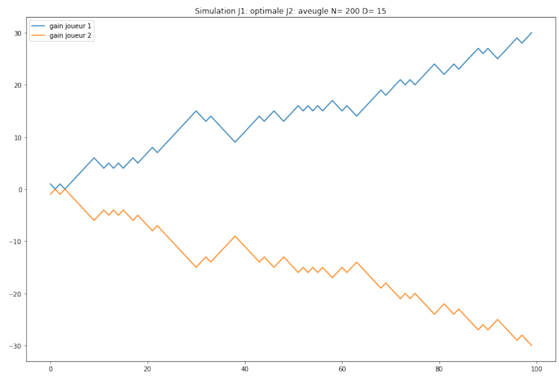



FIGURE 3.1 – $N = 200 \mid D = 15$

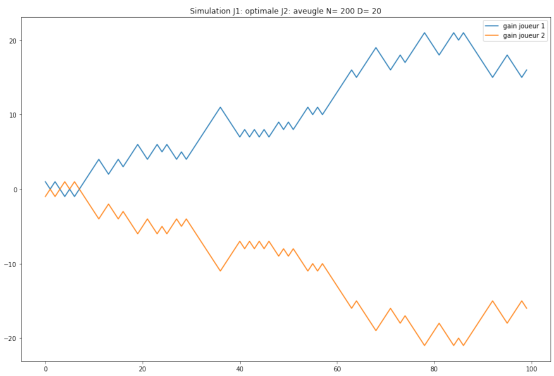



FIGURE 3.2 – $N = 200 \mid D = 20$

**Analyse et commentaires :** Le joueur optimale joue en premier, il est logique qu'il possède un large avantage par rapport au joueur jouant selon la stratégie aveugle.

**Stratégie aveugle vs Stratégie optimale :**

**Le joueur optimale est en orange.**

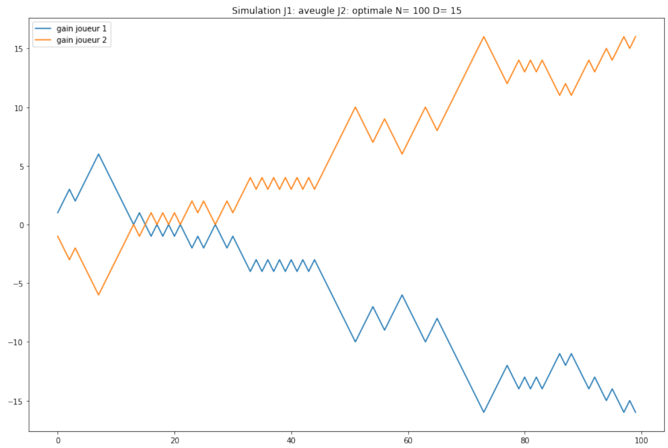



FIGURE 3.3 – $N = 100|D = 15$

**Analyse et commentaires :** Même si le joueur optimale joue en deuxième, il conserve un large avantage par rapport au second joueur jouant selon la stratégie aveugle.

**Stratégie optimale vs Stratégie aléatoire :**

**Le joueur optimale est en bleu.**

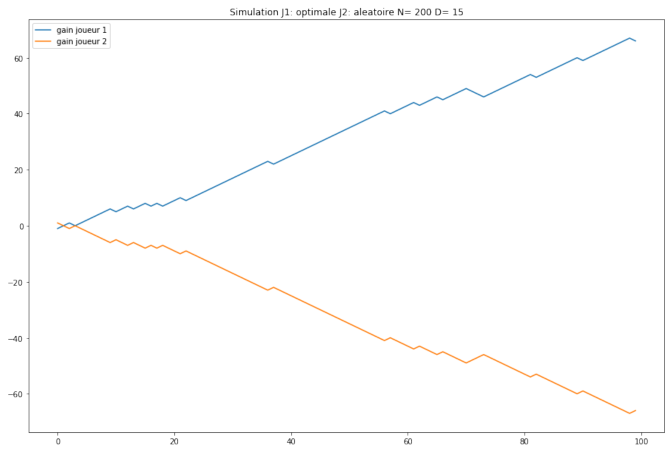



FIGURE 3.4 – $N = 200|D = 15$

**Analyse et commentaires :** Le joueur optimale commence, le joueur jouant selon la stratégie aléatoire n'a aucune chance.

**Stratégie optimale vs Stratégie gourmande(consiste à jouer $D$) :**

**Le joueur optimale est en bleu.**

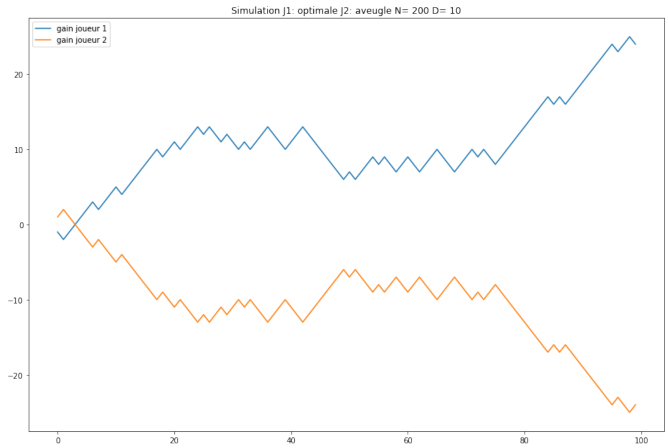



FIGURE 3.5 – $N = 200|D = 10$

**Analyse et commentaires :** Le joueur optimale commence, il possède un large avantage face au joueur jouant selon la stratégie gourmande.

# Chapitre 4

# Variante Simultané

## 4.1 Jeu en un coup :

Dans cette partie, on suppose que les deux joueurs lancent simultanément et une seule fois les dés. Le joueur qui gagne est celui qui obtient le plus grand nombre de points, les deux joueurs sont ex æquo s'ils obtiennent le même score.

### 4.1.1 Calcul de l'espérance de gain du premier joueur en fonction des probabilités $P(d,k)$ :

Soit $D$ le nombre maximal de dés à lancer.
On note $EG_1(d_1,d_2)$ l'espérance de gain du premier joueur s'il lance $d_1$ dés et que le deuxième joueur lance $d_2$ dés :

$EG_1(d_1,d_2) = 1 \times P(\text{"}joueur\ 1\ gagne\text{"}) + 0 \times P(\text{"}ex\ aequo\text{"}) - 1 \times P(\text{"}joueur\ 2\ gagne\text{"}) = P(\text{"}joueur\ 1\ obtienne\ k_1\ ,\ le\ joueur\ 2\ obtienne\ k_2 < k_1\text{"}) - P(\text{"}joueur\ 2\ obtienne\ k_2\ ,\ le\ joueur\ 1\ obtienne\ k_1 < k_2\text{"}) = \sum_{k_1=1}^{6\times d_1} \sum_{k_2=1}^{k_1-1} P(d_1,k_1) \times P(d_2,k_2) - \sum_{k_2=1}^{6\times d_2} \sum_{k_1=1}^{k_2-1} P(d_2,k_2) \times P(d_1,k_1)$

**Exemple pour $D = 3$ :**

$$EG_1(d_1,d_2) = \begin{bmatrix} 0. & -0.375 & -0.22685185 \\ 0.375 & 0. & -0.19881687 \\ 0.22685185 & 0.19881687 & 0. \end{bmatrix}$$

### 4.1.2 Stratégie optimale du joueur 2 face à une stratégie mixte du joueur 1 :

Le joueur 1 joue selon une stratégie mixte $P_1 = [p_1(1), p_1(2), ..., p_1(D)]$, avec $p_1(d)$ la probabilité que le joueur 1 lance $d$ dés ($1 \leq d \leq D$ et $\sum_{d=1}^{D} p_1(d) = 1$), si le joueur 2 connaît ce vecteur, quel serait sa réponse optimale ?
L'espérance de gain du joueur 2 en jouant $d_2$ dés :

$$EG_2(d_2) = \sum_{d_1=1}^{D} EG_2(d_1,d_2) \times p_1(d_1)$$

$EG_2(d_1, d_2) = -EG_1(d_1, d_2)$, car c'est un jeu à somme nulle, on a donc :

$$EG_2(d_2) = \sum_{d_1=1}^{D} -EG_1(d_1, d_2) \times p_1(d_1)$$

Le joueur 2 doit maximiser son espérance de gain i.e. trouver le $d_2$ qui maximise $EG_2(d_2)$, sa réponse optimale est la stratégie pure $d_2^{opt}$ telle que :

$$d_2^{opt} = \underset{d_2}{argmax}[\sum_{d_1=1}^{D} -EG_1(d_1, d_2) \times p_1(d_1)] = \underset{d_2}{argmin}[\sum_{d_1=1}^{D} EG_1(d_1, d_2) \times p_1(d_1)]$$

### 4.1.3 Stratégie mixte optimale du joueur 1, si le joueur 2 la connaît et y répond optimalement :

Le joueur 1 joue selon $P_1 = [p_1(1), p_1(1), ..., p_1(D)]$, le joueur 2 y répond optimalement en jouant $d_2^{opt}$. L'espérance du joueur 1 avec la stratégie $P_1$ est donc :

$$EG_1(P_1) = \sum_{d_1=1}^{D} p_1(d_1) \times EG_1(d_1, d_2^{opt})$$

Or $d_2^{opt} = \underset{d_2}{argmin}[\sum_{d_1=1}^{D} EG_1(d_1, d_2) \times p_1(d_1)]$
d'où :

$$EG_1(P_1) = \underset{d_2}{min}[\sum_{d_1=1}^{D} p_1(d_1) \times EG_1(d_1, d_2)]$$

Le stratégie optimale du jouer 1 $P_1^*$est celle qui maximise son espérance de gain :

$$P_1^* = \underset{P_1}{argmax}[\underset{d_2}{min}[\sum_{d_1=1}^{D} p_1(d_1) \times EG_1(d_1, d_2)]]$$

Le programme linéaire correspondant est :

$$\begin{cases} \max\limits_{P_1} \quad \min\limits_{d_2} \displaystyle\sum_{d_1=1}^{D} EG_1(d_1, d_2) \times P_1^{d_1} \\ \text{s.c.} \qquad \displaystyle\sum_{d_1=1}^{D} P_1^{d_1} = 1 \\ \qquad\qquad P_1^{d_1} \geq 0, \ \forall d_1 \in \{1, .., D\}. \end{cases} \tag{4.1}$$

On pose $z = \min\limits_{d_2} \sum_{d_1=1}^{D} EG_1(d_1, d_2) \times P_1^{d_1}$, il en découle que :

$$z \leq \sum_{d_1=1}^{D} EG_1(d_1, d_2) \times P_1^{d_1} \quad \forall d_2 \in \{1, .., D\}$$

Le programme linéaire devient :

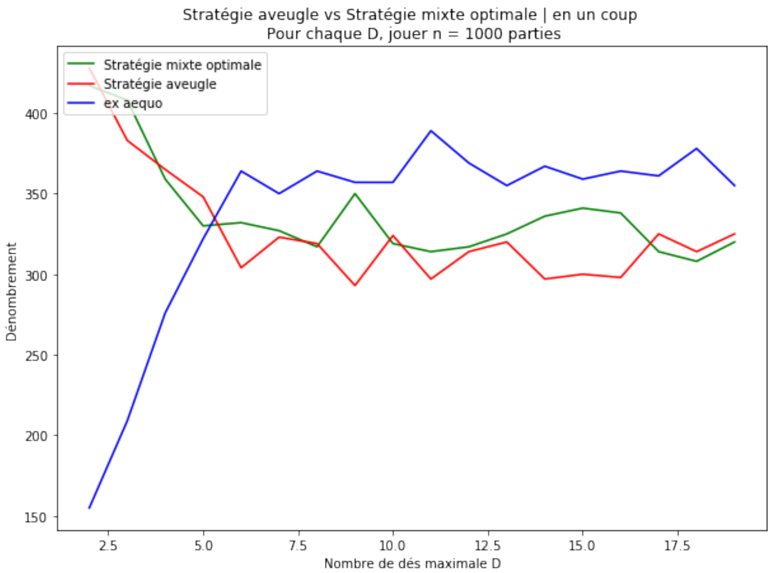
$$\begin{cases} \min\limits_{P_1} & z \\ \text{s.c.} & z - \sum\limits_{d_1=1}^{D} EG_1(d_1,d_2) \times P_1^{d_1} \leq 0 \quad \forall d_2 \in \{1,..,D\}. \\ & \sum\limits_{d_1=1}^{D} P_1^{d_1} = 1 \\ & z \in R, P_1^{d_1} \geq 0,\ \forall d_1 \in \{1,..,D\}. \end{cases} \tag{4.2}$$

**Implémentation : méthode un_coup_mixte_optimale(D) , qui résout le programme linéaire précédant à l'aide d'un solveur et renvoie le vecteur $P_1^*$.**

**Exemples pour différents D** :

$D = 2,\ \ P_1^* = [0, 1]$
$D = 3,\ \ P_1^* = [0, 0, 1]$
$D = 4,\ \ P_1^* = [0, 0, 0, 1]$
$D = 5,\ \ P_1^* = [0, 0.175, 0.052, 0, 0.771]$
$D = 6,\ \ P_1^* = [0, 0.175, 0.052, 0, 0.771, 0]$
$D = 7,\ \ P_1^* = [0, 0.175, 0.052, 0, 0.771, 0, 0]$

### 4.1.4 Comparaison expérimentale entre cette stratégie mixte optimale et la stratégie aveugle déterministe :




**Analyse et commentaires :** Les courbes correspondent aux résultats précédents :

— Avant $D = 4$ : les deux joueurs jouent identiquement et donc les gains sont équivalents.

— Pour $D = 5$ : le joueur 1 joue sa stratégie mixte et le joueur aveugle y répond optimalement $d_2 = 5$.
— À partir de $D = 6$ : le joueur 1 présente un avantage, effectivement la meilleure réponse du joueur 2 serait de jouer $d_2^{optimale} = 5$, or selon sa stratégie aveugle il va jouer $d_2^{aveugle} = 6$ ce qui lui donne une espérance de gain négative mais très proche de 0 expliquant ainsi le nombre important de matchs nuls et l'avantage pas très important du joueur 1.

## 4.2 Cas général :

### 4.2.1 L'espérance de gain du joueur 1 $EG_1(i, j)$ lorsque $i \geq N$ ou $j \geq N$ :

— $EG_1(i, j) = 1$ *si* $i \geq N$ *et* $i > j$ cas où le premier joueur à gagné, dans le cas où les deux joueurs atteignent un score supérieur à $N$ lors du même round le joueur 1 gagne s'il cumule un score supérieur à celui du joueur 2.
— $EG_1(i, j) = -1$ *si* $j \geq N$ *et* $j > i$ cas où le deuxième joueur à gagné, dans le cas où les deux joueurs atteignent un score supérieur à $N$ lors du même round le joueur 2 gagne s'il cumule un score supérieur à celui du joueur 1.
— $EG_1(i, j) = 0$ *si* $i = j$ *et* $i \geq N$ égalité entre les deux joueurs (ils atteignent un score supérieur ou égal à N lors du même round).

### 4.2.2 Expression de l'espérance de gain $E_1^{d_1,d_2}(i, j)$ du joueur 1 lorsqu'il lance $d_1$ dés et que le deuxième joueur lance $d_2$ dés :

Nous supposons que nous connaissons $EG_1(k, l)$ *pour* $k > i$ *et* $l > j$. On est dans l'état $(i, j)$ avec $i < N$ et $j < N$ :

$$E_1^{d_1,d_2}(i, j) = \sum_{k_1=1}^{6d_1} P(d_1, k_1) \times [\sum_{k_2=1}^{6d_2} P(d_2, k_2) \times EG_1(i + k_1, j + k_2)]$$

Un état du jeu sera donc représenté par une matrice qui donne pour chaque $d_1$ et $d_2$, $E_1^{d_1,d_2}(i, j)$.

### 4.2.3 Explication de l'obtention par programmation dynamique du calcul des $EG_1(i, j)$ :

- On a pour chaque état $(i, j)$ du jeu tel que $i < N$ et $j < N$ la matrice des gains du joueur 1 en fonction des valeurs de $d$ possibles : $E_1^{d_1,d_2}(i, j)$.
- Le joueur 1 peut donc appliquer le programme linéaire de la section prétendante pour calculer sa stratégie mixte optimale correspondante à cet état $P_1^{ij}$.
- Reste à trouver la réponse optimale du jouer 2 face à cette stratégie :

$$d_2^{i,j} = \underset{d_2}{argmax}[\sum_{d_1=1}^{D} -E_1^{d_1,d_2}(i, j) \times p_1^{ij}(d_1)] = \underset{d_2}{argmin}[\sum_{d_1=1}^{D} E_1^{d_1,d_2}(i, j) \times p_1^{ij}(d_1)]$$

On en déduit la formule suivante pour $i < N$ et $j < N$ :
$EG_1(i, j) = \sum_{d_1=1}^{D} E_1^{d_1,d_2^{ij}}(i, j) \times p_1^{ij}(d_1)$
$= \underset{d_2}{min}[\sum_{d_1=1}^{D} E_1^{d_1,d_2}(i, j) \times p_1^{ij}(d_1)]$

$= \underset{P_1}{max}[\underset{d_2}{min}[\sum_{d_1=1}^{D} E_1^{d_1,d_2}(i,j) \times p_1(d_1)]]$
$= \underset{P_1}{max}[\underset{d_2}{min}[\sum_{d_1=1}^{D} [\sum_{k_1=1}^{6d_1} P(d_1,k_1) \times \sum_{k_2=1}^{6d_2} P(d_2,k_2) \times EG_1(i+k_1, j+k_2)] \times p_1(d_1)]]$
$EG_1(i,j)$ n'est donc fonction que des $EG_1(k,l)$ avec $k > i$ et $l > j$.

### 4.2.4 Adaptation de l'algorithme pour retourner une stratégie optimale :

Il suffit de sauvegarder à chaque étape de remplissage de la matrice $EG_1$ le $P_1^{ij}$ optimal, c'est la stratégie mixte optimale à jouer si on est dans l'état $(i,j)$.

### 4.2.5 Évaluations expérimentales :

On dessine les courbes des gains cumulées des joueurs pour 100 parties en alternant les stratégies et en variant les valeurs de $D$ et de $N$.

**Stratégie optimale simultanée vs Stratégie optimale simultanée :**

**Le joueur optimale simultanée est en bleu.**

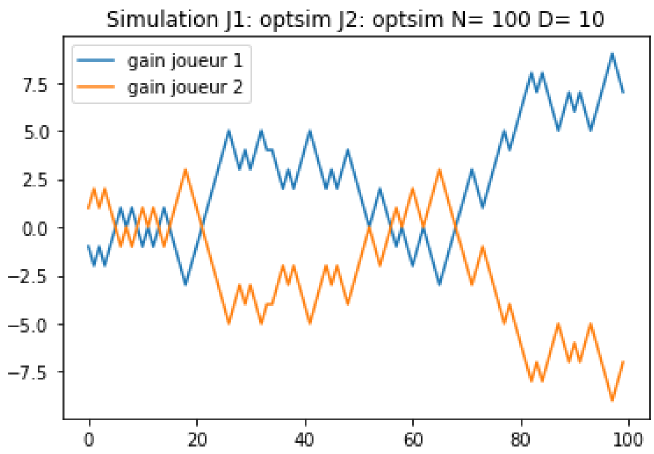



FIGURE 4.1 – $N = 100|D = 10$

**Analyse et commentaires :** Les joueurs jouent simultanément et selon la même stratégie leur gains sont équivalents.

$$EG(\text{"}joueur\ optimale\ simultanée\text{"}) = 0.06$$

$$EG(\text{"}joueur\ optimale\ simultanée\text{"}) = -0.06$$

**Stratégie optimale simultanée vs Stratégie aléatoire :**

**Le joueur optimale simultanée est en bleu.**

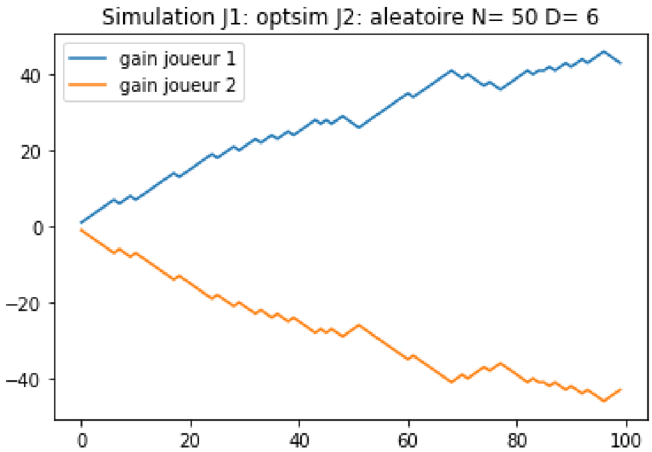



FIGURE 4.2 – $N = 50|D = 6$

**Analyse et commentaires :** Les joueurs jouent simultanément, le joueur aléatoire n'a aucune chance face au joueur optimale simultanée.

$$EG(\text{"}joueur\ optimale\ simultan\acute{e}e\text{"}) = 0.43$$

$$EG(\text{"}joueur\ al\acute{e}atoire\text{"}) = -0.43$$

**Stratégie optimale simultanée vs Stratégie aveugle :**

**Le joueur optimale est en bleu.**

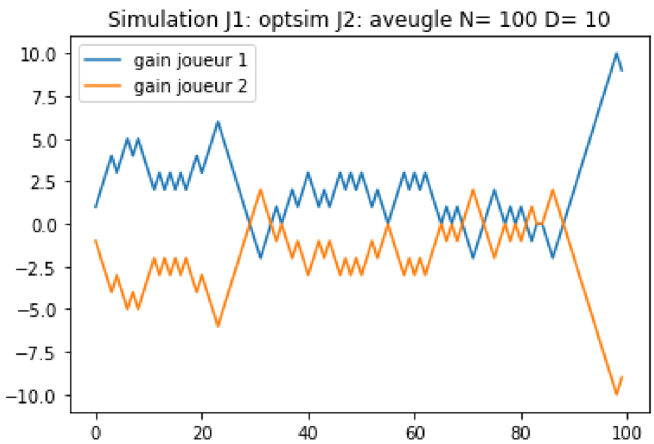



FIGURE 4.3 – $N = 100|D = 10$

**Analyse et commentaires :** Les joueurs jouent simultanément, le joueur optimale simultanée possède un léger avantage sur le joueur aveugle. La stratégie aveugle étant très efficace.

$$EG(\text{"}joueur\ optimale\ simultan\acute{e}e\text{"}) = 0.09$$

$$EG(\text{"}joueur\ aveugle\text{"}) = -0.09$$

**Stratégie optimale simultanée vs Stratégie optimale séquentielle :**

**Le joueur optimale est en bleu.**

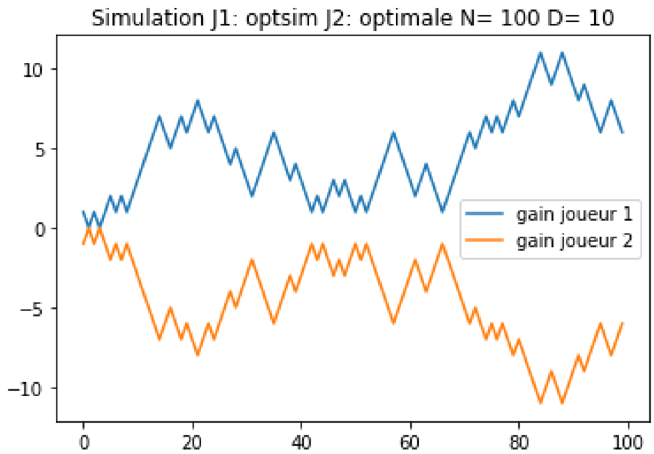



FIGURE 4.4 – $N = 100|D = 10$

**Analyse et commentaires :** Les joueurs jouent simultanément, le joueur optimale simultanée possède un léger avantage sur le joueur optimale séquentielle. La stratégie optimale de la variante séquentielle reste efficace dans la variante simultanée.

$$EG(\text{"}joueur\ optimale\ simultan\acute{e}e\text{"}) = 0.06$$

$$EG(\text{"}joueur\ optimale\ s\acute{e}quentielle\text{"}) = -0.06$$

# Chapitre 5

# Conclusion :

L'utilisation des notions de programmation dynamique et des programmes linéaires s'avère efficace dans l'élaboration de stratégies optimales pour des jeux à somme nulle. Grâce aux outils statistiques et informatiques à notre portée, nous avons été capables de réaliser une étude théorique d'un jeu à somme nulle dans différentes situations et selon différents critères, et de vérifier par la suite ces résultats théoriques en implémentant ces algorithmes.