# Les équilibres de Nash : étude des stratégies gagnantes

*TIPE MPSI/MP : Jeux et sports*

*2023-2024*

Adélaïde Broucas n° SCEI : 46165


# Sommaire :

- Concepts mathématiques
- Mise en place du contexte
- Problématique
- Étude statistique
- Optimisation des tirs aux buts
- Analyse et interprétation
- Annexes

# Concepts mathématiques

# Exemple de jeu matriciel (1)

- Deux joueurs
- Deux matrices de gain
  - $m \text{ x } n\ , (m, n)\ \epsilon (\mathbb{N}^*)^2$
  - notées $A_\mathrm{e}$ et $A_\mathrm{p}$

| **Élève \| Professeur** | **Venir en cours** | **Ne pas venir** |
|---|---|---|
| **Venir en cours** | 3 | -1 |
| **Ne pas venir** | -2 | 1 |

*Tableau de gain de l'élève :*

| **Élève \| Professeur** | **Venir en cours** | **Ne pas venir** |
|---|---|---|
| **Venir en cours** | 3 | 0 |
| **Ne pas venir** | 1 | 0 |

*Tableau de gain du professeur :*

- Matrices de gain associées :

$$A_e = \begin{pmatrix} 3 & -1 \\ -2 & 1 \end{pmatrix}$$ 👨‍🎓

$$A_p = \begin{pmatrix} 3 & 0 \\ -1 & 0 \end{pmatrix}$$ 👨‍🏫

# Exemple de jeu matriciel (2)

- Stratégie $\sigma$ : vecteur colonne de $m$ lignes avec des coefficients positifs dont la somme vaut 1
- Stratégie de l'élève : $\sigma_e = \binom{x_e}{y_e}$
- Stratégie du professeur $\sigma_p = \binom{x_p}{y_p}$
- $x_i = \text{probabilité de venir}$
- $y_i = \text{probabilité de ne pas venir}$

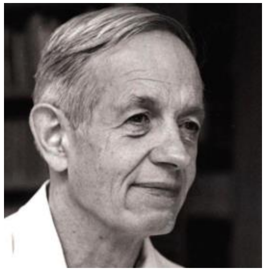
**Définition des fonctions de gain :**

$$u_i(\sigma_1, \sigma_2) = \sigma_1^T A_i \sigma_2 \quad \mathrm{i} \in \{1,2\}$$

- Matrices de gain associées :
  - $A_e = \begin{pmatrix} 3 & -1 \\ -2 & 1 \end{pmatrix}$ 
  - $A_p = \begin{pmatrix} 3 & 0 \\ -1 & 0 \end{pmatrix}$ 

- $u_e = (x_e\ y_e) \begin{pmatrix} 3 & -1 \\ -2 & 1 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix}$
- $u_p = (x_e\ y_e) \begin{pmatrix} 3 & 0 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} x_p \\ y_p \end{pmatrix}$

# Théorème fondamental



*John Nash*

**Théorème de Nash :**

*Tout jeu matriciel admet au moins un équilibre de Nash.*

- Un couple de stratégies $(\sigma_1^*, \sigma_2^*)$ est un équilibre de Nash pour le jeu matriciel représenté par $A_1$ et $A_{2,}$ si et seulement si :
  - $\forall\ \sigma_1\ stratégie\ pour\ le\ joueur\ 1, u_1(\sigma_1, \sigma_2^*) \leq u_1(\sigma_1^*, \sigma_2^*)$
  - $\forall\ \sigma_2\ stratégie\ pour\ le\ joueur\ 2, u_2(\sigma_1^*, \sigma_2) \leq u_2(\sigma_1^*, \sigma_2^*)$

# Caractérisation des équilibres de Nash

- **Équilibres purs :**
  - Le joueur choisit toujours la même action.
- **Équilibres mixtes :**
  - Le joueur choisit aléatoirement entre différentes actions en suivant une distribution de probabilité.

**Fonctions de gain :**

- $u_e = (x_e\ y_e)\begin{pmatrix} 3 & -1 \\ -2 & 1 \end{pmatrix}\begin{pmatrix} x_p \\ y_p \end{pmatrix}$
- $u_p = (x_e\ y_e)\begin{pmatrix} 3 & 0 \\ -1 & 0 \end{pmatrix}\begin{pmatrix} x_p \\ y_p \end{pmatrix}$

**Deux équilibres purs :**

$$\left(\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}\right) \text{ et } \left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}\right)$$

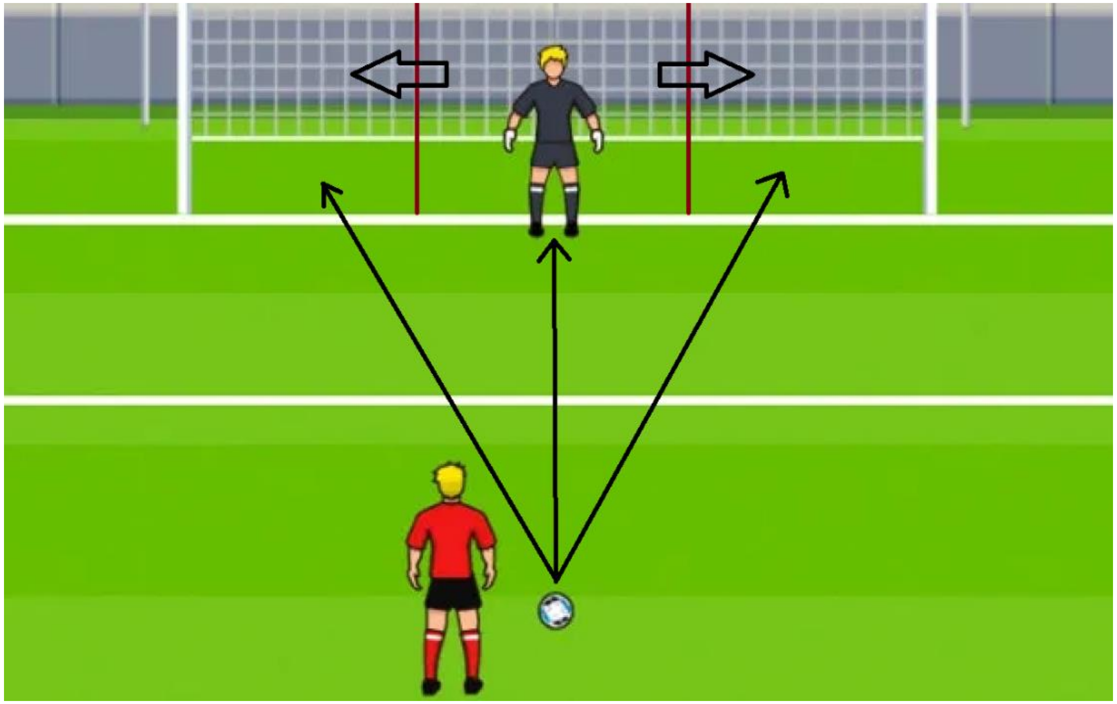
**Un équilibre mixte :**

$$\left(\begin{pmatrix} \frac{1}{4} \\ \frac{3}{4} \end{pmatrix}, \begin{pmatrix} \frac{2}{7} \\ \frac{5}{7} \end{pmatrix}\right)$$

# Mise en place du contexte

# Pénaltys au football

- Un jeu non coopératif
- 3 choix indépendants :
    - Droite
    - Centre
    - Gauche



© jeuxjeuxjeux.fr

# Problématique

Comment un tireur ou un gardien peut augmenter significativement ses chances de marquer ou d'arrêter la balle lors d’un tir au but en se basant sur les équilibres de Nash et l’étude des statistiques ?

# Étude statistique

# Statistiques (1)

- Visionnage d’enregistrements vidéo des pénaltys tirés par Lionel Messi
- 214 pénaltys répertoriés
- ~ 5 heures de vidéos

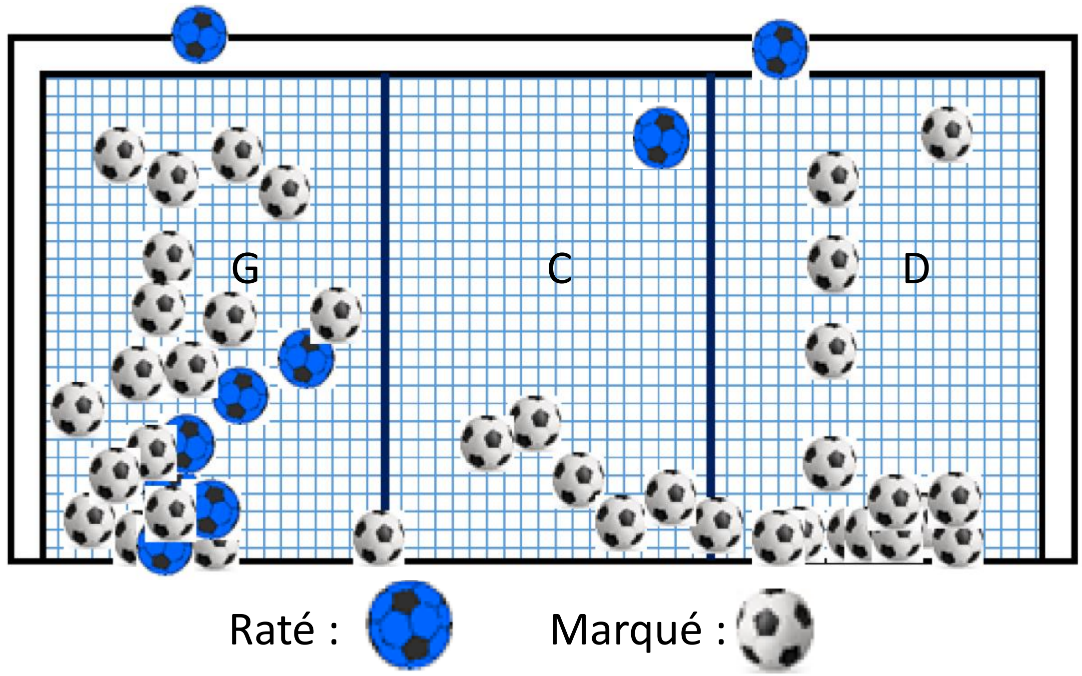
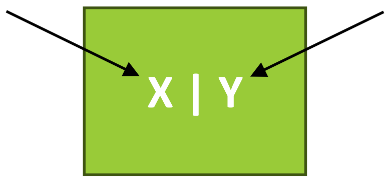



*Schéma de la répartition des pénaltys tirés par Lionel Messi (46 tirs sélectionnés)*

*Exemple de vidéo utilisée pour l’échantillonnage :*
https://www.youtube.com/watch?v=-vCGalBno9o

# Statistiques (2)

Position occupée par le tireur




Position occupée par le gardien

| Pénaltys | G \| G | G \| C | G \| D | C \| C | C \| D | C \| G | D \| G | D \| C | D \| D | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Réussites | 32 | 4 | 34 | 20 | 0 | 18 | 24 | 12 | 14 | 158 |
| Échecs | 26 | 1 | 3 | 2 | 3 | 1 | 2 | 2 | 1 | 56 |
| Total | 58 | 5 | 37 | 22 | 3 | 19 | 26 | 14 | 30 | 214 |
| Pourcentage de succès | 55 % | 80 % | 92 % | 91 % | 0 % | 95 % | 92 % | 86 % | 47 % | 74 % |

*Tableau de la répartition des tirs de Lionel Messi au penalty*

# Optimisation des tirs aux buts

# Matrices de gain obtenues

- ***Notation*** : $A_T$ la matrice de gain du tireur (resp. $A_G$ celle du gardien)

| Pénaltys | G \| G | G \| C | G \| D | C \| C | C \| D | C \| G | D \| G | D \| C | D \| D | Total |
|---|---|---|---|---|---|---|---|---|---|---|
| Pourcentage de succès | 55 % | 80 % | 92 % | 91 % | 0 % | 95 % | 92 % | 86 % | 47 % | 74 % |

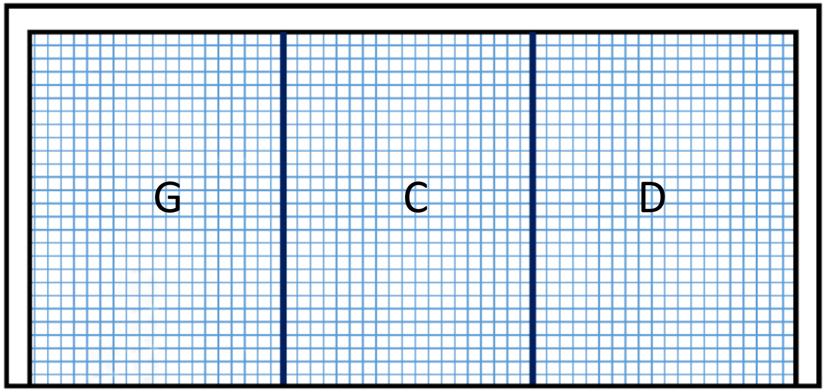
- $A_T = \begin{pmatrix} 0{,}55 & 0{,}8 & 0{,}92 \\ 0{,}91 & 0 & 0{,}95 \\ 0{,}92 & 0{,}87 & 0{,}47 \end{pmatrix}$

- $A_G = U_3 - A_T = \begin{pmatrix} 0{,}45 & 0{,}2 & 0{,}08 \\ 0{,}09 & 1 & 0{,}05 \\ 0{,}08 & 0{,}13 & 0, 53 \end{pmatrix}$

# Stratégies des joueurs

- Stratégie du tireur $\sigma_T = \begin{pmatrix} x_T \\ y_T \\ z_T \end{pmatrix}$

- Stratégie du gardien $\sigma_G = \begin{pmatrix} x_G \\ y_G \\ z_G \end{pmatrix}$




- $x_i$ = probabilité de choisir gauche “ $G$ ”
- $y_i$ = probabilité de choisir centre “ $C$ ”
- $z_i$ = probabilité de choisir droite “ $D$ ”

# Identification des fonctions de gain

| Fonction de gain du tireur | Fonction de gain du tireur |
|---|---|
| • $u_T(\sigma_T, \sigma_G) = \sigma_T^T A_T \sigma_G$ <br><br> $u_T = (x_T\ y_T\ z_T)\begin{pmatrix} 0{,}55 & 0{,}8 & 0{,}92 \\ 0{,}91 & 0 & 0{,}95 \\ 0{,}92 & 0{,}87 & 0{,}47 \end{pmatrix}\begin{pmatrix} x_G \\ y_G \\ z_G \end{pmatrix}$ | • $u_G(\sigma_T, \sigma_G) = \sigma_T^T A_G \sigma_G$ <br><br> $u_G = (x_T\ y_T\ z_T)\begin{pmatrix} 0{,}45 & 0{,}2 & 0{,}08 \\ 0{,}09 & 1 & 0{,}05 \\ 0{,}08 & 0{,}13 & 0{,}53 \end{pmatrix}\begin{pmatrix} x_G \\ y_G \\ z_G \end{pmatrix}$ |

On pose $z_T = 1 - x_T - y_T$ et $z_G = 1 - x_G - y_G$

# Recherche de l'équilibre de Nash (1)

- Calcul des dérivées partielles de $u_T$ et $u_G$ :
  - $\frac{\partial u_T}{\partial x_T}(\sigma_T, \sigma_{\mathrm{G}}^*)$= - 0,82 $x_G^*$ - 0,49$y_G^*$ + 0,45
  - $\frac{\partial u_T}{\partial y_T}(\sigma_T, \sigma_{\mathrm{G}}^*)$= - 0,49 $x_G^*$ - 1,35 $y_G^*$ + 0,48

  - $\frac{\partial u_G}{\partial x_G}(\sigma_{\mathrm{T}}^*, \sigma_G)$= 0,82 $x_T^*$ + 0,49 $y_T^*$ - 0,45
  - $\frac{\partial u_G}{\partial y_G}(\sigma_{\mathrm{T}}^*, \sigma_G)$= 0,52 $x_T^*$ + 1,35 $y_T^*$ - 0,4

# Recherche de l'équilibre de Nash (2)

- Recherche des points critiques – Résolution des systèmes suivants :
  - $\begin{cases} \dfrac{\partial u_T}{\partial x_T} = 0 \\ \dfrac{\partial u_T}{\partial y_T} = 0 \end{cases} \Leftrightarrow \begin{cases} x_G^* = 0{,}420 \\ y_G^* = 0{,}203 \\ z_G^* = 0{,}377 \end{cases}$
  - $\begin{cases} \dfrac{\partial u_G}{\partial x_G} = 0 \\ \dfrac{\partial u_G}{\partial y_G} = 0 \end{cases} \Leftrightarrow \begin{cases} x_T^* = 0{,}483 \\ y_T^* = 0{,}110 \\ z_T^* = 0{,}407 \end{cases}$
- Point critique obtenu en $\sigma_T^* = \begin{pmatrix} x_T^* \\ y_T^* \\ z_T^* \end{pmatrix} et\ \sigma_G^* = \begin{pmatrix} x_G^* \\ y_G^* \\ z_G^* \end{pmatrix}$

# Existence de cet équilibre

**Théorème d'existence d'un équilibre de Nash mixte :**

Dans tout jeu fini (*i.e.* un jeu avec un nombre fini de joueurs, chacun ayant un nombre fini de stratégies), il existe au moins un équilibre de Nash en stratégies mixtes.

- Un équilibre de Nash est un maximum.
- Le point critique observé en $\sigma_T^*$ , $et$ $\sigma_G^*$ est donc un maximum.
- C'est donc un équilibre de Nash mixte.

# Existence d'équilibres purs

**Proposition :**

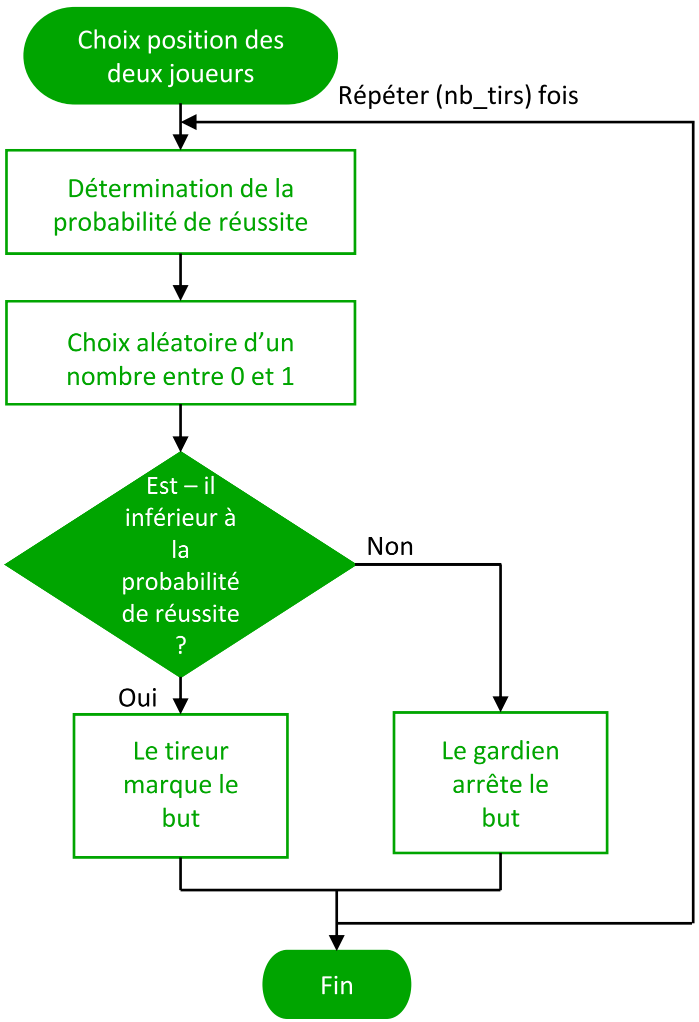
Soit deux matrices opposées $A_1,\ A_2\ \epsilon M_3(\mathbb{R}),\ A_1 = \left(a_{ij}\right)$ possède un point d'équilibre pur (ou point-selle), si et seulement si :

$$\max_{1\le j\le 3}\left(\min_{1\le i\le 3} a_{ij}\right) = \min_{1\le i\le 3}\left(\max_{1\le j\le 3} a_{ij}\right)$$

- $A_T = \begin{pmatrix} \boxed{0{,}55} & 0{,}8 & 0{,}92 \\ 0{,}91 & 0 & 0{,}95 \\ 0{,}92 & \boxed{0{,}87} & 0{,}47 \end{pmatrix}$
- $A_G{}' = A_G - U_3 = \begin{pmatrix} -0{,}55 & -0{,}8 & -0{,}92 \\ -0{,}91 & 0 & -0{,}95 \\ -0{,}92 & -0{,}87 & -0{,}47 \end{pmatrix}$
- $\max_{1\le j\le 3}\left(\min_{1\le i\le 3} a_{ij}\right) = 0{,}55$
- $\min_{1\le i\le 3}\left(\max_{1\le j\le 3} a_{ij}\right) = 0{,}87$
- $\max_{1\le j\le 3}\left(\min_{1\le i\le 3} a_{ij}\right) \neq \min_{1\le i\le 3}\left(\max_{1\le j\le 3} a_{ij}\right)$
- **Conclusion :** la propriété n'est pas vérifiée

# Analyse et interprétation

# Simulation python




```python
import numpy as np
positions = ["G", "C", "D"]

def choix_pondere(probabilites):
    return np.random.choice(positions, p=probabilites)

probabilites_tireur = [1/3, 1/3, 1/3]
probabilites_gardien = [1/3, 1/3, 1/3]

probabilites_reussite = {
    'G': {'G': 0.55, 'C': 0.8, 'D': 0.92},
    'C': {'G': 0.91, 'C': 0, 'D': 0.95},
    'D': {'G': 0.92, 'C': 0.86, 'D': 0.47}}

def resultat_tir(tireur_choix, gardien_choix):
    prob_reussite = probabilites_reussite[tireur_choix][gardien_choix]
    if np.random.random() < prob_reussite:
        return "but"
    else:
        return "arrêt"

def simulation_seance(nb_tirs):
    resultat = {"buts": 0, "arrêts": 0}
    for _ in range(nb_tirs):
        tireur_choix = choix_pondere(probabilites_tireur)
        gardien_choix = choix_pondere(probabilites_gardien)
        resultat_tir_actuel = resultat_tir(tireur_choix, gardien_choix)
        if resultat_tir_actuel == "but":
            resultat["buts"] += 1
        else:
            resultat["arrêts"] += 1
    return resultat
```

# Différentes simulations sur 10 000 tirs

*Choix aléatoire des positions pour les deux joueurs :*

```
#Simulation 1
probabilites_tireur_ale = [1/3, 1/3, 1/3]
probabilites_gardien_ale = [1/3, 1/3, 1/3]
Le pourcentage de but marqué est :  70.464 %
Le pourcentage de tirs arrêtés est de :  29.536 %
```

*Choix optimal pour le gardien et aléatoire pour le tireur :*

```
#Simulation 2
probabilites_tireur_ale = [1/3, 1/3, 1/3]
probabilites_gardien_opti = [0.420, 0.203, 0.377]
Le pourcentage de but marqué est :  73.75 %
Le pourcentage de tirs arrêtés est de :  26.25 %
```

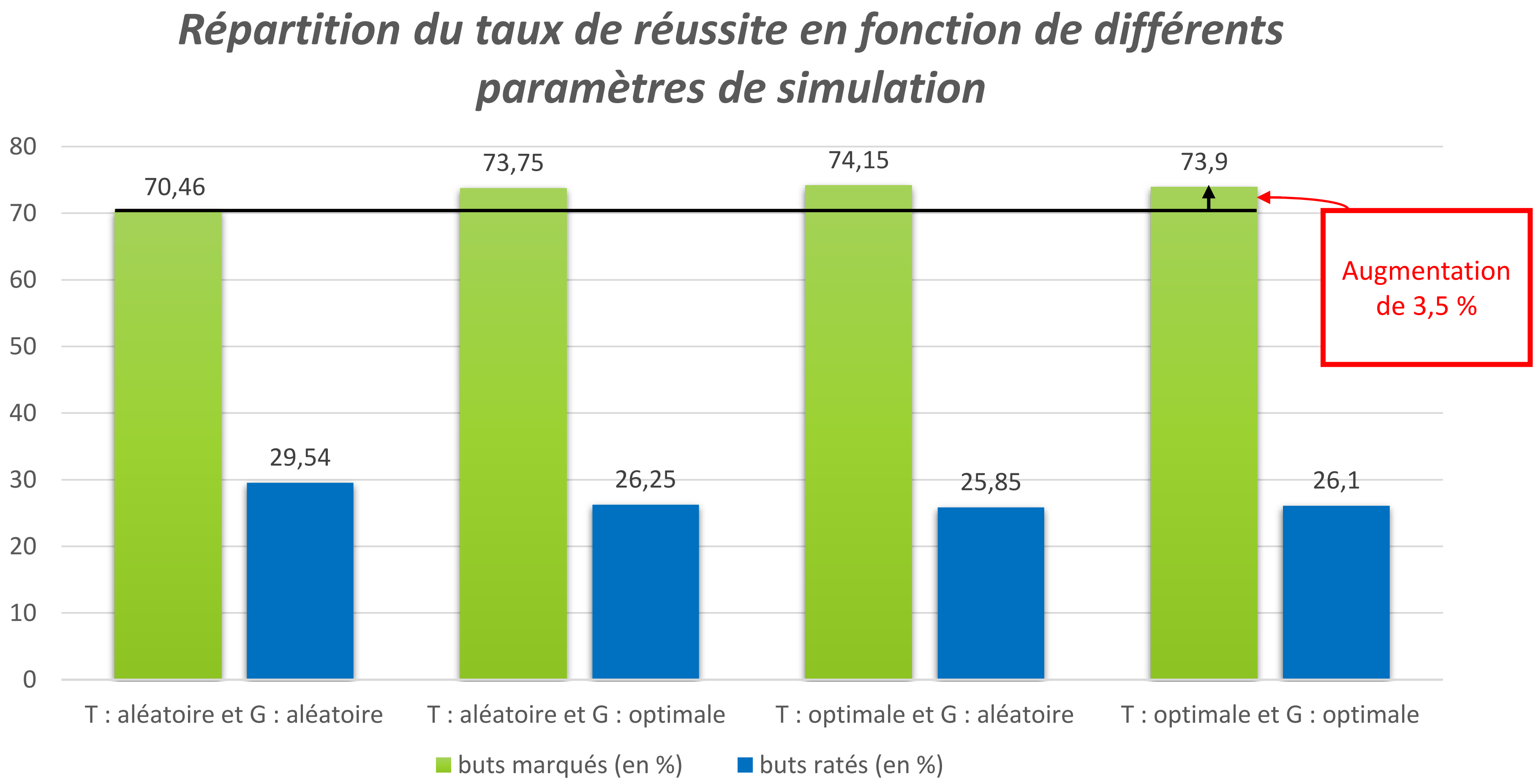
*Choix optimal pour le tireur et aléatoire pour le gardien :*

```
#Simulation 3
probabilites_tireur_opti = [0.483, 0.110, 0.407]
probabilites_gardien_ale = [1/3, 1/3, 1/3]
Le pourcentage de but marqué est :  74.149 %
Le pourcentage de tirs arrêtés est de :  25.851 %
```

*Choix optimal pour les deux joueurs :*

```
#Simulation 4
probabilites_tireur_opti = [0.483, 0.110, 0.407]
probabilites_gardien_opti = [0.420, 0.203, 0.377]
Le pourcentage de but marqué est :  73.892 %
Le pourcentage de tirs arrêtés est de :  26.108 %
```

# Interprétation des résultats



# Conclusion et limites

Augmentation des chances de marquer de 3%

Données rares donc peu précises

Affiner le système en ciblant des zones de tirs plus précises


# Annexes

# Annexes

Simulation des résultats d’une séance de tirs aux buts :

```python
import numpy as np
positions = ["L", "C", "R"]
probabilites_tireur = [1/3, 1/3, 1/3]
probabilites_gardien = [1/3, 1/3, 1/3]

def choix(probabilites):
    return np.random.choice(positions, p=probabilites)

def resultat_tir(tireur_choix, gardien_choix):
    if tireur_choix == gardien_choix:
        return "arrêt"
    else:
        return "but"

def simulation_seance(nb_tirs):
    resultat = {"buts": 0, "arrêts": 0}

    for _ in range(nb_tirs):
        tireur_choix = choix(probabilites_tireur)
        gardien_choix = choix(probabilites_gardien)


        resultat_tir_actuel = resultat_tir(tireur_choix, gardien_choix)

        if resultat_tir_actuel=="but":
            resultat["buts"]+=1
        else:
            resultat["arrêts"]+=1

    return resultat
```

# Annexes

Simulation des tirs aux buts en utilisant les statistiques des joueurs :

```
import numpy as np
positions = ["G", "C", "D"]

def choix_pondere(probabilites):
    return np.random.choice(positions, p=probabilites)

probabilites_tireur = [1/3, 1/3, 1/3]
probabilites_gardien = [1/3, 1/3, 1/3]

probabilites_reussite = {
    'G': {'G': 0.55, 'C': 0.8, 'D': 0.92},
    'C': {'G': 0.91, 'C': 0, 'D': 0.95},
    'D': {'G': 0.92, 'C': 0.86, 'D': 0.47}}

def resultat_tir(tireur_choix, gardien_choix):
    prob_reussite = probabilites_reussite[tireur_choix][gardien_choix]
    if np.random.random() < prob_reussite:
        return "but"
    else:
        return "arrêt"

def simulation_seance(nb_tirs):
    resultat = {"buts": 0, "arrêts": 0}
    for _ in range(nb_tirs):
        tireur_choix = choix_pondere(probabilites_tireur)
        gardien_choix = choix_pondere(probabilites_gardien)
        resultat_tir_actuel = resultat_tir(tireur_choix, gardien_choix)
        if resultat_tir_actuel == "but":
            resultat["buts"] += 1
        else:
            resultat["arrêts"] += 1
    return resultat
```

# Annexes

Calcul des pourcentages de réussite sur 10000 tirs :

```
def moyenne_pourcentage (nb_tirs) :
    s_buts=0
    s_arrets=0
    for i in range(100):
        resultat_final = simulation_seance(nb_tirs)
        buts , arrets = resultat_final['buts'],resultat_final['arrêts']
        s_buts += buts
        s_arrets += arrets

    moy_but_pourcentage = s_buts/1000  #on répète 100 fois 1000 tirs
                                       #on obtient la moyenne en /100
                                       #puis la moyenne en pourcentage en /10 ce qui revient à /1000

    moy_arrets_pourcentage= s_arrets/1000
    return(moy_but_pourcentage,moy_arrets_pourcentage)
```