Le défi des trois dés

Par Enigmath, le 7 janvier 2026

Énoncé

Considérons un jeu auquel participent K joueurs. Chaque joueur reçoit trois dés à six faces. Chacun lance simultanément ses trois dés et obtient un score, défini comme la somme des trois résultats. Après ce premier lancer, chaque joueur doit décider :

— soit de conserver ce score et d'arrêter son tour,

— soit de relancer les trois dés pour tenter d'obtenir un meilleur score.

Si un joueur choisit de relancer, il perd irréversiblement son score précédent, même si celui-ci était plus élevé. Chaque joueur peut relancer au maximum deux fois, c'est-à-dire qu'il effectue entre un et trois lancers au total. Les scores intermédiaires sont gardés secrets jusqu'à la fin de la partie. Lorsque tous les joueurs ont terminé, les scores sont révélés. Le joueur qui possède le plus petit score est déclaré perdant. En cas d'égalité pour le plus faible score, le perdant est choisi au hasard parmi les ex æquo.

Questions :

1. 🌶️[2.1] 💻[2.9] Déterminez, en fonction de K, un profil de stratégies pures (c'est-à-dire non aléatoires) tel que chaque joueur adopte sa meilleure réponse compte tenu des choix des autres joueurs.
2. 🌶️[2.2] On considère un joueur isolé face à une coalition composée des $K-1$ autres joueurs, et on s'intéresse aux stratégies mixtes (le coup joué est sélectionné au hasard selon une certaine distribution de probabilité). Montrez que, sans perte de généralité, le problème peut se ramener à un jeu à somme nulle de taille 120×120 entre deux joueurs (le joueur isolé versus la coalition).
3. 🌶️[2.3] En généralisant à $\ell + 1$ lancers, montrez qu'un joueur isolé face à une coalition composée des $K-1$ autres joueurs peut se ramener à un jeu à somme nulle de dimension $\binom{14+\ell}{\ell} \times \binom{14+\ell}{\ell}$ entre deux joueurs.
4. 🌶️[1.1] 💻[3.3] On considère de nouveau le cas du jeu en concurrence pure (sans coalition). On suppose qu'après chaque relance, tous les joueurs observent le nombre L de joueurs encore en jeu (n'ayant pas arrêté). Pour trois lancers et pour tout $K \leq 100$, déterminez un profil de stratégies pures tel que chaque joueur adopte sa meilleure réponse compte tenu des décisions des autres.

Solution

1. On note p_s la probabilité qu'un lancer de trois dés donne un score total égal à s. Une *stratégie pure de seuil* est définie par deux entiers $t_1, t_2 \in \{4, \dots, 18\}$ tels que :
   — après le premier lancer donnant le score s, le joueur *conserve* son score si $s \geq t_1$, et *relance* sinon ;
   — si le joueur relance une première fois et obtient un nouveau score s', il conserve ce score si $s' \geq t_2$, et relance une dernière fois sinon.

On suppose dans un premier temps que $K \geq 2\,887\,502$. Pour la stratégie pure définie par les seuils $(4,4)$, on note X la variable aléatoire représentant le score final obtenu après application de cette stratégie.

Nous allons montrer que la stratégie pure $(4,4)$ constitue une *meilleure réponse* pour un joueur donné, en supposant que tous les adversaires jouent également la stratégie pure $(4,4)$, autrement dit que le profil où tous les joueurs jouent la stratégie $(4,4)$ forme un *équilibre de Nash*. La probabilité que le joueur considéré *perde*, sachant que son score final vaut $s \in \{3,\dots,18\}$, est :

$$\begin{aligned} f_X(s) &\triangleq \sum_{j=0}^{K-1} \binom{K-1}{j} \mathbb{P}(X=s)^j \mathbb{P}(X>s)^{K-1-j} \frac{1}{j+1} \\ &= \frac{\mathbb{P}(X\geq s)^K - \mathbb{P}(X>s)^K}{K\,\mathbb{P}(X=s)} = \frac{1}{K}\sum_{j=0}^{K-1} \mathbb{P}(X>s)^j \mathbb{P}(X\geq s)^{K-1-j}. \end{aligned}$$

Cette dernière expression montre que $f_X(s)$ est une fonction décroissante de s. Pour qu'une stratégie pure (t_1,t_2) soit une meilleure réponse, il faut que chaque seuil t_i vérifie :

$$t_i \in \arg\min_t \left(\sum_{s=t}^{18} p_s f_X(s) + \sum_{s=3}^{t-1} p_s E_i \right),$$

où les quantités E_1 et E_2 sont définies par :

$$E_2 = \sum_{s=3}^{18} p_s f_X(s), \qquad E_1 = \sum_{s=t_2}^{18} p_s f_X(s) + \sum_{s=3}^{t_2-1} p_s E_2.$$

Autrement dit, le seuil optimal peut se réécrire sous la forme :

$$t_i = \min\{\, s \mid f_X(s) \leq E_i \,\} = \max\{\, s \mid f_X(s-1) > E_i \,\},$$

ce maximum existant puisque $f_X(3) > E_i$ pour tout $i \in \{1,2\}$. Pour que la stratégie pure $(4,4)$ soit une meilleure réponse, il faut donc que $f_X(4) \leq E_i$ pour tout $i \in \{1,2\}$, i.e.,

$$f_X(4) \leq \sum_{s=3}^{18} p_s f_X(s), \quad f_X(4) \leq \sum_{s=4}^{18} p_s f_X(s) + p_3 \sum_{s=3}^{18} p_s f_X(s).$$

Ces deux conditions sont satisfaites si l'on montre que $f_X(4) \leq p_3^2 f_X(3)$. Or, on a :

$$\mathbb{P}(X\geq 4) = 1-p_3^3 \quad \text{et} \quad \mathbb{P}(X>4) = (1-p_3^3)\frac{1-p_3-p_4}{1-p_3}.$$

Ainsi :

$$f_X(4) = \frac{(1-p_3^3)^K \left(1 - \left(\frac{1-p_3-p_4}{1-p_3}\right)^K\right)}{K p_4 \frac{1-p_3^3}{1-p_3}}, \qquad f_X(3) = \frac{1-(1-p_3^3)^K}{K p_3^3}.$$

On en déduit :

$$\frac{f_X(4)}{p_3^2 f_X(3)} = \frac{(1-p_3^3)^K \left(1 - \left(\frac{1-p_3-p_4}{1-p_3}\right)^K\right) p_3}{p_4 \frac{1-p_3^3}{1-p_3}\left(1-(1-p_3^3)^K\right)} \leq \frac{(1-p_3^3)^K p_3}{p_4 \frac{1-p_3^3}{1-p_3}\left(1-(1-p_3^3)^K\right)}.$$

Cette quantité est inférieure à 1 dès que

$$(1-p_3^3)^{-K} \geq 1 + \frac{p_3}{p_4 \frac{1-p_3^3}{1-p_3}}, \quad \text{c'est-à-dire} \quad K \geq -\frac{\log\left(1 + \frac{p_3}{p_4 \frac{1-p_3^3}{1-p_3}}\right)}{\log(1-p_3^3)} \approx 2\,887\,501.82,$$

ce qui est bien vérifié par hypothèse.

Pour $K \leq 2\,887\,501$, nous utilisons le code Python suivant, permettant de déterminer un équilibre de Nash symétrique pour chaque K. Sur un ordinateur portable classique, le calcul complet nécessite environ 400 secondes. Les équilibres obtenus sont résumés dans le tableau ci-dessous. Dans le code, on cherche un point fixe d'une certaine application en itérant ses compositions successives, en prenant comme stratégie initiale $(4, 4)$. L'unicité de l'équilibre de Nash n'est pas garantie : plusieurs profils de seuils peuvent simultanément constituer des équilibres pour certaines valeurs de K. Cependant, le choix de la stratégie initiale $(4, 4)$ assure une continuité entre les deux régimes $K \leq 2\,887\,501$ et $K \geq 2\,887\,502$: dès que $(4, 4)$ devient un équilibre, c'est celui qui est effectivement sélectionné par l'algorithme.

Intervalle de K	**Seuils d'équilibre** (t_1, t_2)
$2 > K \geq 0$	non défini
$3 > K \geq 2$	$(13, 12)$
$4 > K \geq 3$	$(12, 12)$
$8 > K \geq 4$	$(12, 11)$
$11 > K \geq 8$	$(11, 11)$
$18 > K \geq 11$	$(11, 10)$
$34 > K \geq 18$	$(10, 10)$
$51 > K \geq 34$	$(10, 9)$
$120 > K \geq 51$	$(9, 9)$
$190 > K \geq 120$	$(9, 8)$
$546 > K \geq 190$	$(8, 8)$
$946 > K \geq 546$	$(8, 7)$
$3\,265 > K \geq 946$	$(7, 7)$
$6\,509 > K \geq 3\,265$	$(7, 6)$
$31\,019 > K \geq 6\,509$	$(6, 6)$
$77\,495 > K \geq 31\,019$	$(6, 5)$
$713\,156 > K \geq 77\,495$	$(5, 5)$
$2\,852\,409 > K \geq 713\,156$	$(5, 4)$
$\infty > K \geq 2\,852\,409$	$(4, 4)$

python Télécharger le code

```python
import numpy as np, time

def conditional_losing_probability(o, K):
    p, mask = o.tail**K, o.pdf > 0
    p[mask] = ((o.pdf+o.tail)**K - o.tail**K)[mask] / (o.pdf[mask]*K)
    return p

class Strategy:
    p = np.bincount([i + j + k for i in range(1,7) for j in range(1,7) for k
         in range(1,7)]) / 216
    def __init__(self, oponents=None, K=None, t1=None, t2=None):
        self.n = len(self.p)
        self.t1, self.t2 = self.BR(oponents, K) if t1 is None else (t1,t2)
        self.pdf = self.pdf1(self.t1, self.pdf1(self.t2))
        self.tail = 1 - np.cumsum(self.pdf)
    def pdf1(self, t, p=None):
        if p is None:
            p = self.p
        p_redraw, out = 0, np.zeros(self.n)
        for i, pi in enumerate(self.p):
            if i >= t:
```

```
                out[i] += pi
            else:
                p_redraw += pi
        return out + p_redraw * p
    def BR(self, oponents, K):
        clp = conditional_losing_probability(oponents, K)
        t2 = np.where(np.sum(clp * self.p) >= clp)[0][0]
        p2 = self.pdf1(t2)
        t1 = np.where(np.sum(clp * p2) >= clp)[0][0]
        return (t1, t2)

start_time = time.time()
record = f"Not defined for %d>K>=0"
previous_thresholds = (0,0)
for K in range(2, 2887502):
    S = Strategy(t1=4, t2=4)
    rS = Strategy(oponents=S, K=K)
    while (S.t1, S.t2) != (rS.t1, rS.t2):
        S = rS
        rS = Strategy(oponents=S, K=K)
    if (S.t1, S.t2) != previous_thresholds:
        print(record % K)
        record = f"thresholds={(S.t1, S.t2)} for %.0f>K>={K}"
    previous_thresholds = (S.t1, S.t2)
print(record % np.inf)
print(f"({time.time() - start_time} secondes)")
```

2. La coalition est représentée par les $K-1$ premiers joueurs, et le joueur « seul » est le joueur K, dont on examine la probabilité de perdre. On note X_i la variable aléatoire représentant le score final obtenu par le joueur $i \in \{1, \dots, K-1\}$. La probabilité que le joueur seul perde, sachant que son score final vaut $s \in \{3, \dots, 18\}$, est :

$$\sum_{S \subseteq \{1,\dots,K-1\}} \prod_{i \in S} \mathbb{P}(X_i = s) \prod_{i \notin S} \mathbb{P}(X_i > s) \frac{1}{|S|+1}.$$

Si l'on introduit une variable aléatoire $U \sim \mathcal{U}(0,1)$, alors $\mathbb{E}(U^k) = \frac{1}{k+1}$ pour tout $k \in \mathbb{N}$. La probabilité précédente peut donc se réécrire sous la forme :

$$\mathbb{E}\left(\sum_{S \subseteq \{1,\dots,K-1\}} \prod_{i \in S} \mathbb{P}(X_i = s) \prod_{i \notin S} \mathbb{P}(X_i > s) U^{|S|} \right) = \mathbb{E}\left(\prod_{i=1}^{K-1} \left(\mathbb{P}(X_i = s) U + \mathbb{P}(X_i > s) \right) \right).$$

Par l'inégalité arithmético-géométrique, on obtient la borne supérieure suivante :

$$\mathbb{E}\left(\frac{1}{K-1} \sum_{i=1}^{K-1} \left(\mathbb{P}(X_i = s) U + \mathbb{P}(X_i > s) \right)^{K-1} \right) = \frac{1}{K-1} \sum_{i=1}^{K-1} f_{X_i}(s).$$

Ainsi, du point de vue de la coalition, il est toujours au moins aussi avantageux de jouer la *stratégie mixte symétrique* dans laquelle chaque joueur de la coalition adopte les mêmes seuils, choisis uniformément parmi les paires $\{(t_{i,1}, t_{i,2}) \mid i \in 1, \dots, K-1\}$, plutôt qu'une *stratégie pure hétérogène* où chaque joueur i aurait ses propres seuils $(t_{i,1}, t_{i,2})$. En conclusion, lorsqu'on énumère les stratégies pures envisageables pour la coalition, on peut, sans perte de généralité, se restreindre aux *profils de coalition symétriques*.

On note X_τ la variable aléatoire correspondant au score final obtenu lorsqu'un joueur applique la stratégie $\tau = (\tau_1, \tau_2)$. On peut alors se restreindre à un jeu à somme nulle à deux joueurs : le joueur « seul » contre la coalition. Lorsque la coalition adopte la stratégie pure $t = (t_1, t_2)$ (i.e., la stratégie pure symétrique $(t_1, t_2), \dots, (t_1, t_2)$) et que le joueur seul applique la stratégie pure $\hat{t} = (\hat{t}_1, \hat{t}_2)$, la valeur du jeu est donnée par $\mathbb{E}\left(f_{X_t}(X_{\hat{t}})\right)$. On remarque que, conformément à l'intuition, cette quantité croît lorsque la loi de la coalition devient plus favorable, c'est-à-dire

lorsque $\mathbb{P}(X_t \geq \cdot)$ augmente. De même, cette quantité décroît lorsque la loi du joueur seul devient plus favorable, c'est-à-dire lorsque $\mathbb{P}(X_{\hat{t}} \geq \cdot)$ augmente. On peut donc éliminer une stratégie pure $\tau = (\tau_1, \tau_2)$ envisagée par l'un ou l'autre des deux camps dès que la fonction de survie associée $\mathbb{P}(X_\tau \geq \cdot)$ est majorée en tout point par une combinaison convexe de deux autres fonctions de survie :

$$\mathbb{P}(X_\tau \geq \cdot) \leq \alpha \mathbb{P}(X_{\tau'} \geq \cdot) + (1-\alpha)\mathbb{P}(X_{\tau''} \geq \cdot), \qquad \text{avec } \alpha \in [0,1].$$

On note $\tau \prec [\tau', \tau'']$ si c'est le cas, et on dit que la stratégie τ est dominée par une stratégie mixte dont le support est formé des deux stratégies pures τ' et τ''. Le lemme suivant permet de réaliser une telle élimination.

Lemme. *Pour $\tau_1 < \tau_2$, on a $(\tau_1, \tau_2) \prec [(\tau_1, \tau_1), (\tau_2, \tau_2)]$. Autrement dit, il suffit de considérer les stratégies pures de la forme (τ_1, τ_2) vérifiant $\tau_1 \geq \tau_2$. Cette restriction est naturelle : lors du premier choix, il reste encore deux lancers possibles, ce qui permet d'adopter une attitude plus risquée, autrement dit, de fixer un seuil plus élevé qu'au second lancer.*

Démonstration. On pose $R(x) \triangleq \sum_{s \geq x} p_s$, $A = 1 - R(\tau_1)$, $B = R(\tau_1) - R(\tau_2)$, $C = R(\tau_2)$ et $\alpha = \dfrac{A+B}{2A+B}$. Les expressions des probabilités de dépasser s sont alors les suivantes, avec $E(s)$ la probabilité que le score final soit supérieur à s, sachant qu'on a effectué deux relances (contre τ_1 et τ_2). Dans le cas présent, on a $E(s) = R(s)$, mais cette égalité ne sera plus vérifiée dans la question suivante.

$$\mathbb{P}(X_{(\tau_1,\tau_2)} \geq s) = R(\max(s,\tau_1)) \ + \ A\Big(R(\max(s,\tau_2)) + (A+B)\,E(s)\Big),$$

$$\mathbb{P}(X_{(\tau_1,\tau_1)} \geq s) = R(\max(s,\tau_1)) \ + \ A\Big(R(\max(s,\tau_1)) + A\,E(s)\Big),$$

$$\mathbb{P}(X_{(\tau_2,\tau_2)} \geq s) = R(\max(s,\tau_2)) \ + \ (A+B)\Big(R(\max(s,\tau_2)) + (A+B)\,E(s)\Big).$$

On pose pour tout s,

$$T(s) \ \triangleq \ \mathbb{P}(X_{(\tau_1,\tau_2)} \geq s) - \alpha\,\mathbb{P}(X_{(\tau_1,\tau_1)} \geq s) - (1-\alpha)\,\mathbb{P}(X_{(\tau_2,\tau_2)} \geq s).$$

On observe que les termes en $E(s)$ se compensent exactement dans l'expression de $T(s)$. Afin de montrer que $T(s) \leq 0$, on distingue trois cas :
Cas 1 : $s < \tau_1$**.** Ici $\max(s,\tau_1) = \tau_1$ et $\max(s,\tau_2) = \tau_2$. On a donc

$$\begin{aligned} T(s) &= B + C + AC - \alpha(B + C + AB + AC) - (1-\alpha)C(1 + A + B) \\ &= \frac{AB(1 - A - B - C)}{B + 2A} = 0. \end{aligned}$$

Cas 2 : $\tau_1 \leq s < \tau_2$**.** Ici $\max(s,\tau_1) = s$ et $\max(s,\tau_2) = \tau_2$. On a

$$\begin{aligned} T(s) &= R(s) + AC - \alpha\big(1 + A\big)R(s) - (1-\alpha)\big(C + (A+B)C\big) \\ &= -\frac{A(1-A)C}{B+2A} + \frac{A(1-A-B)}{B+2A}R(s). \end{aligned}$$

Pour $R(s) = C$, on a $T(s) = -\frac{ABC}{B+2A} \leq 0$, et pour $R(s) = B+C$, on a $T(s) = \frac{AB(1-A-B-C)}{B+2A} = 0$. Comme $R(s) \in [C, B+C]$, on a $T(s) \leq 0$.

Cas 3 : $s \geq \tau_2$**.** Ici $\max(s,\tau_1) = \max(s,\tau_2) = s$. On a

$$\begin{aligned} T(s) &= R(s)\Big[1 + A - \alpha\big(1 + A\big) - (1-\alpha)\big(1 + A + B\big)\Big] \\ &= -\frac{AB}{B+2A}R(s) \leq 0. \end{aligned}$$

Ainsi, dans chacun des trois cas on a montré $T(s) \leq 0$. □

Par les arguments précédents (symétrisation de la coalition et élimination des stratégies de type $\tau_1 < \tau_2$ par combinaison convexe), on peut se restreindre, sans perte de généralité, à l'étude du jeu réduit suivant :

— la coalition est supposée jouer des profils *symétriques* : tous ses $K-1$ membres utilisent la même paire de seuils ;

— pour les deux camps, on ne considère que des stratégies pures (τ_1, τ_2) telles que $\tau_1 \geq \tau_2$.

Rappelons que les seuils possibles prennent les valeurs entières $\{4, 5, \dots, 18\}$, soit 15 valeurs. Le nombre de couples (τ_1, τ_2) satisfaisant $\tau_1 \geq \tau_2$ est donc

$$\#\{(\tau_1, \tau_2) \ : \ \tau_1, \tau_2 \in \{4, \dots, 18\},\ \tau_1 \geq \tau_2\} = \frac{15 \cdot 16}{2} = 120.$$

3. Par la question précédente, on est restreint à un jeu à 2 joueurs. Pour $t, \tau_1 \dots, \tau_k \in \{4, 18\}^\ell$ on généralise la notion de dominance à $t \prec [\tau_1, \dots, \tau_k]$. On voit clairement que $t \prec [t, \tau_1, \dots, \tau_k] \Rightarrow t \prec [\tau_1, \dots, \tau_k]$. Par la question précédente, on a que si $t_i < t_{i+1}$ pour un $i \in \{1, \dots, \ell - 1\}$, alors $(t_1, \dots, t_\ell) \prec [(t_1, \dots, t_{i-1}, t_i, t_i, t_{i+2}, \dots, t_\ell), (t_1, \dots, t_{i-1}, t_{i+1}, t_{i+1}, t_{i+2}, \dots, t_\ell)]$. Plus généralement, on a le lemme suivant.

Lemme. *Soit $a, b \in \{4, 18\}$ avec $a < b$. Soit $r \in \{2, \dots, \ell\}$ et $i \in \{0, \dots, \ell - r\}$. Pour $x \in \{0, \dots, r\}$, on pose*

$$H(x) \triangleq (t_1, \dots, t_i, \underbrace{a, \dots, a}_{x \text{ fois}}, \underbrace{b, \dots, b}_{r-x \text{ fois}}, t_{i+r+1}, \dots, t_\ell).$$

Alors, $H(x) \prec [H(0), H(r)]$ pour $0 < x < r$.

Démonstration. On veut montrer par récurrence sur $k \in \{2, \dots, r\}$ que

$$H(x) \prec [H(x'), H(x'')], \quad 0 \leq x' < x < x'' \leq r, \quad x'' - x' = k. \qquad (P_k)$$

Comme déjà mentionné, par la question précédente, on a $H(x) \prec [H(x-1), H(x+1)]$ pour $0 < x < r$, i.e., on a (P_2). Si on a $(P_2), \dots, (P_k)$, alors soit $0 \leq x' < x < x'' \leq r$, $x'' - x' = k+1$.

— Si $x' + 1 < x$, alors, par (P_k), on a $H(x) \prec [H(x'+1), H(x'')]$. Par $(P_{x-x'})$, on a $H(x'+1) \prec [H(x'), H(x)]$. On a donc $H(x) \prec [H(x'), H(x), H(x'')]$, et donc $H(x) \prec [H(x'), H(x'')]$.

— Si $x' + 1 = x$, alors, par (P_2), on a $H(x) \prec [H(x'), H(x+1)]$. Si $x + 1 = x''$, alors il n'y a rien à montrer. Si $x + 1 < x''$, alors par (P_k), on a $H(x+1) \prec [H(x), H(x'')]$. On a donc $H(x) \prec [H(x'), H(x), H(x'')]$, et donc $H(x) \prec [H(x'), H(x'')]$.

On a donc (P_{k+1}). Par récurrence, on a donc (P_r). □

Lemme. *Il suffit de considérer les stratégies pures de la forme $t = (t_1, \dots, t_\ell)$ avec $t_1 \geq \dots \geq t_\ell$. On peut donc se restreindre à un sous-ensemble de stratégies pures à $\binom{14+\ell}{\ell}$ éléments.*

Démonstration. On procède par récurrence sur le nombre d'indices i tels que $t_i \neq t_{i+1}$. Si ce nombre vaut 0, il n'y a rien à montrer. Supposons maintenant que ce nombre soit égal à $k+1$. Si les $k+1$ indices i vérifient $t_i > t_{i+1}$, il n'y a rien à montrer. Sinon, un des indice i vérifie $t_i < t_{i+1}$ et le lemme précédent permet alors de supprimer cet indice, ce qui conclut la récurrence. □

4. On utilise le code Python suivant. Sur un ordinateur portable classique, le calcul nécessite environ 150 secondes. Les équilibres obtenus sont résumés dans le tableau ci-dessous.

K	t_1	Structure de $t_2(L)$
2	13	$13\{L = 0\},\ 12\{L = 1\}$
3–5	12	$12\{0 \leq L \leq 1\},\ 11\{2 \leq L \leq K-1\}$
6	12	$12\{0 \leq L \leq 1\},\ 11\{2 \leq L \leq 4\},\ 10\{L = 5\}$
7–12	11	$11\{0 \leq L \leq 4\},\ 10\{5 \leq L \leq K-1\}$
13–17	11	$11\{0 \leq L \leq 4\},\ 10\{5 \leq L \leq 11\},\ 9\{12 \leq L \leq K-1\}$
18–30	10	$10\{0 \leq L \leq 11\},\ 9\{12 \leq L \leq K-1\}$
31–48	10	$10\{0 \leq L \leq 11\},\ 9\{12 \leq L \leq 29\},\ 8\{30 \leq L \leq K-1\}$
49–50	10	$10\{0 \leq L \leq 11\},\ 9\{12 \leq L \leq 29\},\ 8\{30 \leq L \leq 49\}$
51–87	9	$9\{0 \leq L \leq 29\},\ 8\{30 \leq L \leq K-1\}$
88–100	9	$9\{0 \leq L \leq 29\},\ 8\{30 \leq L \leq 87\},\ 7\{88 \leq L \leq K-1\}$

python Télécharger le code

```python
from scipy.special import comb; import time, numpy as np

def expand(a, b):
    coeffs = np.array([1.0])
    for i in range(len(a)):
        coeffs = np.convolve(coeffs, [a[i], b[i]])
    return coeffs

def conditional_losing_probability(others_list):
    K, n, p = len(others_list) + 1, others_list[0].n, []
    for s in range(n):
        a = [o.pdf[s] for o in others_list]
        b = [o.tail[s] for o in others_list]
        p.append(np.dot(expand(a, b), [1/(k+1) for k in range(K)[::-1]]))
    return np.array(p)

class Strategy:
    p = np.bincount([i + j + k for i in range(1,7) for j in range(1,7) for k
         in range(1,7)]) / 216
    def __init__(self, oponents=None, K=None, t1=None, t2=None):
        self.n, self.t1, self.t2, self.pdf = len(self.p), t1, t2, None
        if self.t1 is None and self.t2 is None:
            self.t1, self.t2 = self.BR(oponents, K)
        if self.t1 is None and self.t2 is not None:
            self.pdf = self.pdf1(self.t2)
        if self.t1 is not None and self.t2 is None:
            self.pdf, self.p_redraw = self.pdf1(self.t1, redraw=True)
        if self.pdf is not None:
            self.tail = 1 - np.cumsum(self.pdf)
    def pdf1(self, t, p=None, redraw=False):
        if p is None:
            p = self.p
        p_redraw, out = 0, np.zeros(self.n)
        for i, pi in enumerate(self.p):
            if i >= t:
                out[i] += pi
            else:
                p_redraw += pi
        if redraw:
            return out/(1-p_redraw), p_redraw
        return out + p_redraw * p
    def BR(self, oponents, K):
        o = Strategy(t1=oponents.t1)
        loss_continue, loss_stop, t2, p2, q = [], [], [], [], []
        for L in range(K):
            loss_continue.append(conditional_losing_probability([Strategy(t2
                =oponents.t2[L]) for _ in range(L)] + [o for _ in range(K-1-
                L)]))
            loss_stop.append(conditional_losing_probability([Strategy(t2=
                oponents.t2[L-1]) for _ in range(L)] + [o for _ in range(K
                -1-L)]))
            t2.append(np.where(np.sum(loss_continue[L] * self.p) >=
                loss_continue[L])[0][0])
            p2.append(self.pdf1(t2[L]))
            q.append(comb(K-1, L) * o.p_redraw**L * (1-o.p_redraw)**(K-1-L))
        stop_1 = sum(loss_stop[L] * q[L] for L in range(K))
        continue_1 = sum(loss_continue[L] * p2[L] * q[L] for L in range(K))
        t1 = np.where(np.sum(continue_1) >= stop_1)[0][0]
        return (t1, t2)
```

```python
def compress_t2(t2):
    segments, start = [], 0
    for i in range(1, len(t2)):
        if t2[i] != t2[start]:
            end = i - 1
            val = t2[start]
            if start == 0:
                if end == 0:
                    segments.append(f"{val}{{L=0}}")
                else:
                    segments.append(f"{val}{{0<=L<={end}}}")
            else:
                if start == end:
                    segments.append(f"{val}{{L={start}}}")
                else:
                    segments.append(f"{val}{{{start}<=L<={end}}}")
            start = i
    val = t2[start]
    end = len(t2) - 1
    if start == 0:
        if end == 0:
            segments.append(f"{val}{{L=0}}")
        else:
            segments.append(f"{val}{{0<=L<={end}}}")
    else:
        if start == end:
            segments.append(f"{val}{{L={start}}}")
        else:
            segments.append(f"{val}{{{start}<=L<={end}}}")
    return ", ".join(segments)

start_time = time.time()
previous_thresholds = (0,0)
for K in range(2, 101):
    S = Strategy(t1=4, t2=[4 for L in range(K)])
    rS = Strategy(oponents=S, K=K)
    while (S.t1, S.t2) != (rS.t1, rS.t2):
        S = rS
        rS = Strategy(oponents=S, K=K)
    if (S.t1, S.t2) != previous_thresholds:
        t1, t2 = S.t1, compress_t2(S.t2)
        print(f"thresholds=({t1}, [{t2}]) for K={K}")
    previous_thresholds = (S.t1, S.t2)
print(f"({time.time() - start_time} secondes)")
```

Notes et références

La situation étudiée est inspirée du deuxième wagon de la vidéo *Stop the Train* (2025) de Squeezie, vidéaste français. D'un point de vue mathématique, cette énigme s'inscrit dans le cadre de la *théorie des jeux*, branche des mathématiques qui modélise les interactions stratégiques entre agents rationnels [1, 2]. Plus précisément, la question 1 consiste à déterminer des *équilibres de Nash*, notion introduite par John Nash en 1950 [3]. Un équilibre de Nash correspond à un profil de stratégies tel qu'aucun joueur ne puisse améliorer son gain en modifiant seul sa stratégie.
Pour analyser la situation considérée à la question 2, on met en œuvre la *méthode d'élimination itérative des stratégies dominées*, procédé classique en théorie des jeux [4]. Cette méthode consiste à retirer successivement les stratégies qui sont toujours moins performantes que d'autres, quels que soient les choix de l'adversaire. Dans le contexte de cette énigme, la notion de dominance entre stratégies est étroitement liée à celle de *dominance stochastique* [5], qui compare les lois de probabilité associées aux gains des joueurs. Cette relation permet de réduire significativement l'espace stratégique tout en conservant l'ensemble des équilibres pertinents du jeu.

Sources

[1] John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.

[2] Martin J. Osborne. *An Introduction to Game Theory*. Oxford University Press, 2003.

[3] John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences of the USA*, 36(1):48–49, 1950.

[4] Martin J Osborne and Ariel Rubinstein. *A course in game theory*. MIT press, 1994.

[5] James P. Quirk and Ronald M. Saposnik. Admissibility and measurable utility functions. *Review of Economic Studies*, 29(2):140–146, 1962.