Concours Général Maths 2024
Autour de l'exercice 2

Les bonbons cachés

Eléments de correction de ce problème, à l'exception de la partie 3. Je ne suis pas en mesure de proposer une piste de résolution consistante de cette partie.

Partie 1 : Sophie et Germain testent les trois boîtes.

1. Notons B l'évènement « Sophie choisit la boîte contenant les bonbons » et $\overline{B}$ son évènement contraire : « Sophie choisit une des deux boîtes n'en contenant pas ».

Vu qu'il y a équiprobabilité du choix de chacune des trois boîtes : $P(B) = \frac{1}{3}$

2.a. Si Sophie ne modifie pas son choix, le jeu précédent n'est pas modifié. $P(B) = \frac{1}{3}$

2.b. Supposons que Sophie modifie son choix et change sa boîte. Notons B_1 l'évènement « Sophie choisit au premier tour la boîte contenant les bonbons », évènement dont la probabilité est égale à $\frac{1}{3}$ d'après la question précédente, et B l'évènement « Sophie choisit la boîte contenant les bonbons à l'issue du jeu »

Nous avons : $P_{B_1}(B) = 0$ car si B_1 est réalisé, au second tour Sophie échange la boîte gagnante contre une vide et $P_{\overline{B_1}}(B) = 1$ car si B_1 n'est pas réalisé, au second tour Sophie échange sa boîte vide contre la boîte gagnante, l'autre boîte vide étant éliminée.

Dans ce protocole : $P(B) = P(B_1) \times P_{B_1}(B) + P(\overline{B_1}) \times P_{\overline{B_1}}(B) = \frac{1}{3} \times 0 + \frac{2}{3} \times 1 = \frac{2}{3}$

2.c. Le second protocole est plus avantageux pour Sophie puisqu'elle double ses changes de gagner.

Partie 2 : Une stratégie pour Sophie

On reprend la notation B_1 de la partie 1.

3.a. Vu qu'il y a équiprobabilité du choix de chacune des *n* boîtes : $P(B_1) = \frac{1}{n}$.

3.b. Si Sophie ne modifie pas son choix : $P(B) = P(B_1) = \frac{1}{n}$ car le second tour n'a aucune influence.

Si Sophie modifie son choix, il ne reste que deux boîtes au second tour. Nous avons toujours $P_{B_1}(B) = 0$ et $P_{\overline{B_1}}(B) = 1$. Dans ce protocole : $P(B) = P(B_1) \times P_{B_1}(B) + P(\overline{B_1}) \times P_{\overline{B_1}}(B) = \frac{1}{n} \times 0 + \frac{n-1}{n} \times 1 = \frac{n-1}{n}$

Pour tout entier $n \geq 3$, l'inégalité $\frac{n-1}{n} > \frac{1}{n}$ est vérifiée.

Sophie a toujours intérêt à modifier son choix.

4.a. Si Sophie ne modifie pas son choix : $P(B) = P(B_1) = \frac{1}{n}$ car les tours suivants n'ont aucune influence.

4.b. Si Sophie conserve son choix à chaque tour sauf au dernier où elle change son choix, ce protocole revient à éliminer $(n-2)$ boîtes vides successivement. Il est équivalent au protocole de la question **3.b**. La probabilité de gagner est la même qu'en **3.b.**

En gardant *toujours la même boîte* jusqu'au dernier tour (où elle l'échange) Sophie s'assure de gagner avec une probabilité exactement égale à $\frac{n-1}{n}$.

Partie 3 : Une stratégie pour Germain

?

5

Partie 4 : Une stratégie commune

Je propose une méthode basée sur des congruences et cousine du « théorème chinois », méthode permettant de déterminer le numéro de la boîte contenant les bonbons sans pour autant le « communiquer ».

On choisit deux nombres premiers *p* et *q* distincts et supérieurs ou égaux à 3 et on pose $n = p \times q$

Les entiers de 1 à $n = p \times q$ sont caractérisés par leurs restes dans les divisions euclidiennes par *p* et par *q*.

Sophie choisit la boîte numéro *n* et la garde jusqu'à connaître le numéro de la boîte contenant les bonbons.

En éliminant seulement deux boîtes, Germain peut arriver à coder le numéro *b* de la boîte où se trouvent les bonbons :

- La première boîte porte un numéro congru à *b* modulo *p*
- La deuxième boîte porte un numéro congru à *b* modulo *q*

Ci-dessous un algorithme Python et son application avec $p = 3\,;q = 5$. On constate que cet algorithme fonctionne non seulement pour $n = 3 \times 5 = 15$ mais aussi pour les valeurs de *n* telles que $11 \leq n \leq 15$.

On pourrait vérifier qu'avec $p = 5\,;q = 7$ l'algorithme fonctionne pour $15 \leq n \leq 35$ et qu'avec $p = 7\,;q = 13$ l'algorithme fonctionne pour $27 \leq n \leq 91$

```
>>> def bonbons(p,q):
        for b in range(1,p*q+1):
                i=1
                while i%p!=b%p:
                        i=i+1
                if i==b:
                        i=i+p
                j=1
                while j%q!=b%q:
                        j=j+1
                if j==i or j==b:
                        j=j+q
                print("si bonbons dans",b,"restes",b%p,"et",b%q,
"Germain enlève la boîte",i,"puis la boîte",j)
```

```
>>> bonbons(3,5)
si bonbons dans 1 restes 1 et 1 Germain enlève la boîte 4 puis la boîte 6
si bonbons dans 2 restes 2 et 2 Germain enlève la boîte 5 puis la boîte 7
si bonbons dans 3 restes 0 et 3 Germain enlève la boîte 6 puis la boîte 8
si bonbons dans 4 restes 1 et 4 Germain enlève la boîte 1 puis la boîte 9
si bonbons dans 5 restes 2 et 0 Germain enlève la boîte 2 puis la boîte 10
si bonbons dans 6 restes 0 et 1 Germain enlève la boîte 3 puis la boîte 1
si bonbons dans 7 restes 1 et 2 Germain enlève la boîte 1 puis la boîte 2
si bonbons dans 8 restes 2 et 3 Germain enlève la boîte 2 puis la boîte 3
si bonbons dans 9 restes 0 et 4 Germain enlève la boîte 3 puis la boîte 4
si bonbons dans 10 restes 1 et 0 Germain enlève la boîte 1 puis la boîte 5
si bonbons dans 11 restes 2 et 1 Germain enlève la boîte 2 puis la boîte 1
si bonbons dans 12 restes 0 et 2 Germain enlève la boîte 3 puis la boîte 2
si bonbons dans 13 restes 1 et 3 Germain enlève la boîte 1 puis la boîte 3
si bonbons dans 14 restes 2 et 4 Germain enlève la boîte 2 puis la boîte 4
si bonbons dans 15 restes 0 et 0 Germain enlève la boîte 3 puis la boîte 5
```

Avec $p = 3$; $q = 7$, l'algorithme fonctionne pour $15 \leq n \leq 21$:

```
>>> bonbons(3,7)
si bonbons dans 1 restes 1 et 1 Germain enlève la boîte 4 puis la boîte 8
si bonbons dans 2 restes 2 et 2 Germain enlève la boîte 5 puis la boîte 9
si bonbons dans 3 restes 0 et 3 Germain enlève la boîte 6 puis la boîte 10
si bonbons dans 4 restes 1 et 4 Germain enlève la boîte 1 puis la boîte 11
si bonbons dans 5 restes 2 et 5 Germain enlève la boîte 2 puis la boîte 12
si bonbons dans 6 restes 0 et 6 Germain enlève la boîte 3 puis la boîte 13
si bonbons dans 7 restes 1 et 0 Germain enlève la boîte 1 puis la boîte 14
si bonbons dans 8 restes 2 et 1 Germain enlève la boîte 2 puis la boîte 1
si bonbons dans 9 restes 0 et 2 Germain enlève la boîte 3 puis la boîte 2
si bonbons dans 10 restes 1 et 3 Germain enlève la boîte 1 puis la boîte 3
si bonbons dans 11 restes 2 et 4 Germain enlève la boîte 2 puis la boîte 4
si bonbons dans 12 restes 0 et 5 Germain enlève la boîte 3 puis la boîte 5
si bonbons dans 13 restes 1 et 6 Germain enlève la boîte 1 puis la boîte 6
si bonbons dans 14 restes 2 et 0 Germain enlève la boîte 2 puis la boîte 7
si bonbons dans 15 restes 0 et 1 Germain enlève la boîte 3 puis la boîte 1
si bonbons dans 16 restes 1 et 2 Germain enlève la boîte 1 puis la boîte 2
si bonbons dans 17 restes 2 et 3 Germain enlève la boîte 2 puis la boîte 3
si bonbons dans 18 restes 0 et 4 Germain enlève la boîte 3 puis la boîte 4
si bonbons dans 19 restes 1 et 5 Germain enlève la boîte 1 puis la boîte 5
si bonbons dans 20 restes 2 et 6 Germain enlève la boîte 2 puis la boîte 6
si bonbons dans 21 restes 0 et 0 Germain enlève la boîte 3 puis la boîte 7
```

On peut faire mieux :

Solution proposée par N.I.E. (élève de Terminale et participant au concours 2024) :

Pour tout $n \geq 4$:

- Sophie choisit la boîte 4. Si c'est la bonne, Germain ôte la boîte 1, sinon il ôte une des boîtes 2 ou 3 (il en faut deux au cas où une des deux serait la bonne).
- Si Sophie a la boîte correcte elle la garde jusqu'à la fin, et sinon elle garde sa "mauvaise boîte" et l'échange au dernier tour avec la bonne.