Une question de tournois

Année 2020 – 2021

Élèves :
- Sophie PRADEL, élève de Seconde ;
- Manon DROB, élève de Terminale ;
- Tom CANTAREL, élève de Première.

Encadrés par les enseignants :
- Mme Sandrine VERNHET (professeur de mathématiques) ;
- M. Laurent THOMAS (professeur de sciences physiques).

Établissement : Lycée Raymond Savignac, Villefranche de Rouergue

Établissements jumelés :
- Collège Georges Pompidou, Cajarc ;
- Lycée français international Louis Massignon, Casablanca.

Chercheur : M. Bertrand JOUVE, CNRS Senior Researcher.

Sommaire :

1. Présentation du sujet

Le but de cet article est de décrire et caractériser les tournois qui vérifient les deux propriétés suivantes :

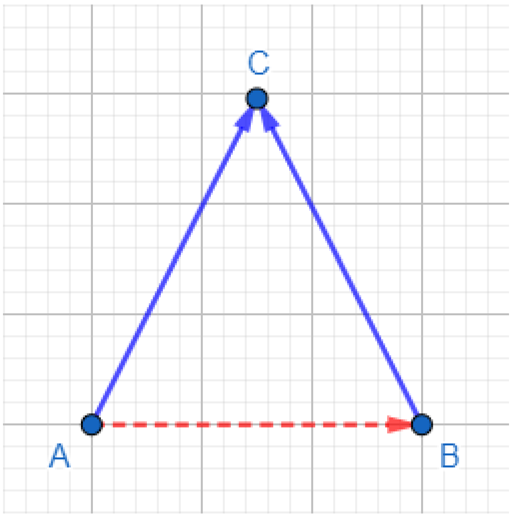
- **il n’existe pas deux équipes “jumelles” (propriété 1) ;**
- **si une équipe quelconque est retirée (par exemple si elle a été disqualifiée) alors il existe deux équipes au moins qui deviennent “jumelles” (propriété 2).**

Un tournoi est un ensemble de rencontres (de matchs) entre n équipes dont les résultats sont représentés par des flèches orientées de l'équipe gagnante vers la perdante. Par exemple, lorsque l’équipe A gagne l’équipe B, on écrit : **A ⟶ B**.

On suppose qu’il n’y a aucun match nul : il existe toujours un gagnant et un perdant.

De plus, les équipes ne se rencontrent qu’une fois.

Deux équipes A et B sont “jumelles” si elles ont obtenu exactement les mêmes résultats dans leurs matchs contre les autres équipes, sans compter celui qu’elles jouent entre elles (A contre B). Toutes les deux ont donc gagné face aux mêmes équipes.



Dans le cas de trois équipes A, B et C, si A gagne B et C et si B gagne C, alors les équipes A et B sont jumelles puisqu’on ne tient pas compte du match qu’elles jouent entre elles (ici A ⟶ B).

2. Annonce des conjectures et résultats obtenus

Voici quelques résultats importants de l'article :

1. La parité du nombre d'équipes n'influe pas sur la présence d'équipes jumelles.

2. Dans un tournoi aléatoire, si le nombre d'équipes tend vers l'infini, alors la probabilité d'avoir deux équipes jumelles tend vers 0.

3. Plus le nombre d'équipes sera important, plus la probabilité que la première propriété soit respectée sera grande.

4. Pour respecter la deuxième propriété, il est nécessaire que les équipes remplissent certaines conditions sur les victoires et les défaites.

Cet article exposera le cheminement qui a conduit à ces conclusions, notamment grâce à une réflexion sur plusieurs schémas, à un programme Python et à l'exploitation des résultats de ce programme au travers d'un graphe.

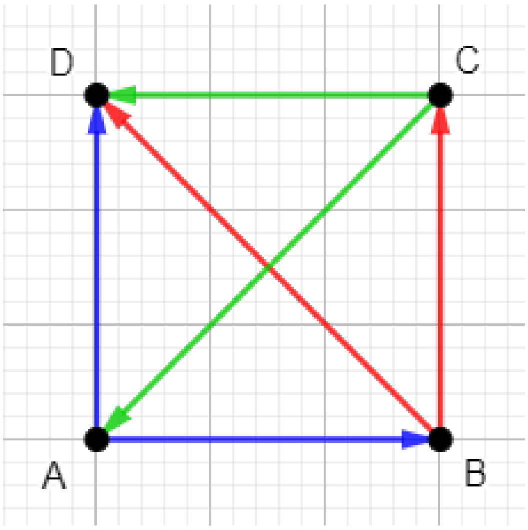
3. Texte de l’article

Sommaire de l’article :

I – Recherche empirique d'un tournoi vérifiant les deux propriétés

A – Raisonnement sur quatre équipes :

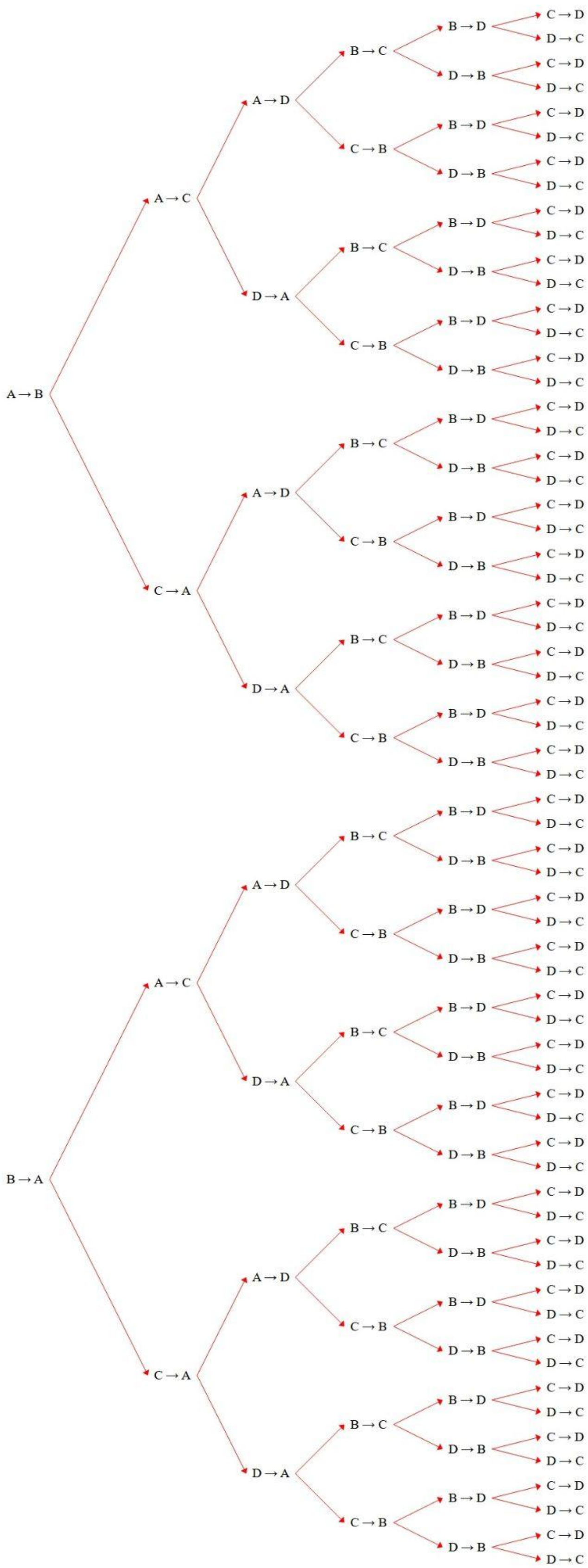
Il paraît intéressant et facile de raisonner avec un petit nombre d’équipes afin de comprendre les différentes situations de tournoi et de tester celles qui correspondent aux deux propriétés du sujet. Par exemple, le tournoi suivant à quatre équipes vérifie la première propriété (il n’existe pas deux équipes “jumelles”) :



A gagne B et D, B gagne C et D, C gagne A et D, D ne gagne personne.

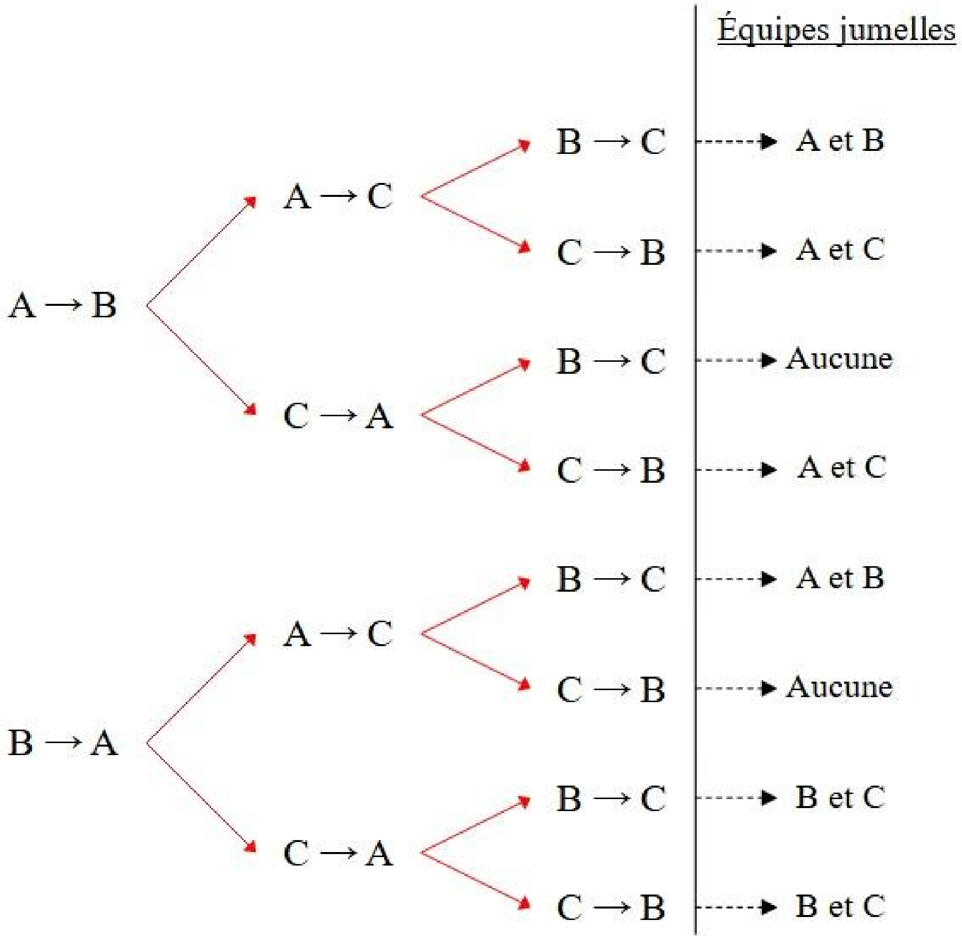
Cependant, il ne vérifie pas la deuxième condition.
Si on enlève l’équipe D, il n’y a pas d’équipes “jumelles”. Donc cela ne fonctionne pas.

Comme le montre l'arbre suivant, un tournoi avec seulement quatre équipes conduit déjà à 64 issues différentes pour les résultats :

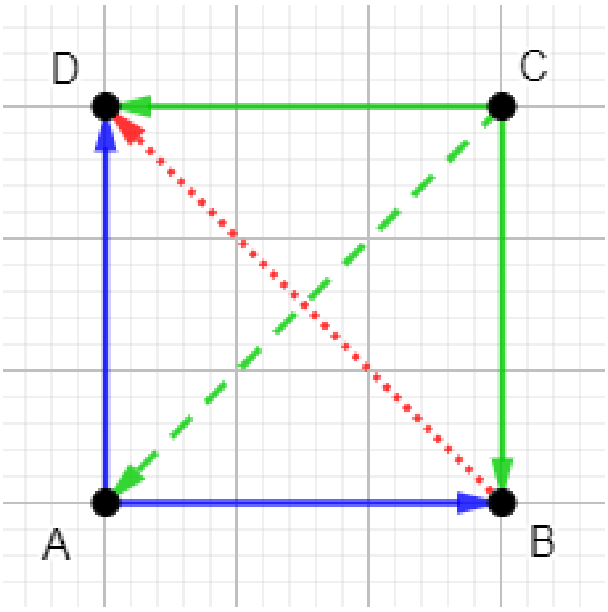


B – Chercher tous les tournois possibles serait impossible

Ainsi, quand le nombre d'équipes augmente, la quantité de résultats à traiter devient rapidement trop importante. Il faut donc trouver une ou des condition(s) générale(s) sur les tournois pour laquelle (lesquelles) les deux propriétés sont validées quel que soit le nombre d'équipes.
En simulant les issues pour un nombre impair d'équipes (par exemple 3), les résultats obtenus sont les suivants :



Il apparaît qu'un grand nombre d'équipes jumelles est obtenu. Il est alors tentant d'envisager l'hypothèse selon laquelle des équipes jumelles sont possibles si le nombre d'équipes est impair, mais pas s'il est pair, ce qui confirmerait l'exemple du tournoi à quatre équipes considéré précédemment (dans la partie A).



Pour tester cette hypothèse, il est possible d'étudier un tournoi avec un nombre pair d'équipes, par exemple 4. Dans ce cas, une des issues montre que A et C ont toutes les deux gagné les mêmes équipes (B et D), car le match A/C ne compte pas. Elles sont donc jumelles. Or le nombre d'équipes est pair.

Cette hypothèse était donc fausse.

Ainsi, **la parité du nombre d'équipes n'est pas une condition de la présence ou non d'équipes jumelles**.

Elle ne peut donc pas permettre de décrire ou de caractériser les tournois répondant aux deux propriétés énoncées.

II – La première propriété - Le programme Python

Un programme Python[1] calculant le nombre d'équipes jumelles en fonction du nombre d'équipes total paraît être l'outil adapté pour réfléchir sur la première propriété.
Après plusieurs simulations, il fait naître la conjecture que lorsque le nombre d'équipes tend vers l'infini, la probabilité d'avoir deux équipes jumelles tend vers zéro.

```python
from random import*


n = int(input("Nombre d'equipes : "))                   #n est le nombre d'equipes
cptj = 0                                                #cptj est le compteur de paires d'equipes jumelles
listeEquipesGagnees = [set() for i in range(n)]         #liste dans laquelle on stockera les equipes gagnees par chaque equipe


for i in range(n) :

    for k in range (i+1 , n) :

        GagnePerdu = randint(1 , 2)                 #le programme choisit de façon aleatoire 1 ou 2 sachant que 1 est "gagne" et 2 est "perdu" dans le match de l'équipe i contre l'équipe k

        if GagnePerdu == 1 :
            listeEquipesGagnees[i].add(k)           #on rajoute l'equipe k gagnee par i dans l'ensemble de l'equipe i

        elif GagnePerdu == 2 :
            listeEquipesGagnees[k].add(i)           #on rajoute l'equipe i gagnee par k dans l'ensemble de l'equipe k


print("listeEquipesGagnees : ",listeEquipesGagnees)


#recherche des équipe jumelles
for a in range(n) :

    for b in range(a+1 , n) :

        #on crée 2 nouvelles listes (pour éviter de changer les listes initiales qui sont mutables) sans le match b contre a

                l1 = [i for i in listeEquipesGagnees[a] if i != b]      #prend dans la liste les valeurs de l'ensemble de liste[a] s'ils ne sont pas égaux a b
                l2 = [i for i in listeEquipesGagnees[b] if i != a]      #prend dans la liste les valeurs de l'ensemble de liste[b] s'ils ne sont pas égaux a a

        #on compare les 2 listes créées pour voir si elles sont jumelles
                if l1 == l2 :
                    cptj = cptj + 1                 #on compte le nombre de paires d'equipes jumelles sur tous les matchs joues
                    print("Les equipes ",a," et ",b," sont jumelles")


if cptj == 1 :
    print("Il y a 1 paire d'equipes jumelles.")         #on obtient cptj au singulier
else :
    print("Il y a " , cptj , " paires d'equipes jumelles.") #on obtient cptj au pluriel
```

Il serait intéressant de pouvoir tester sur un grand nombre de simulations chaque configuration de tournoi (c'est-à-dire avec un nombre d'équipes fixé).

En modifiant ce programme, il permet de choisir le nombre de simulations de tournois à réaliser.

[1] Le Python est un langage de programmation.

```python
from random import*


n = int(input("Nombre d'equipes : "))                      #n est le nombre d'equipes
cptj = 0                                                   #cptj est le compteur de paires d'equipes jumelles
cptt = 0                                                   #cptt est le compteur de nombre de fois sur z où il y a des equipes jumelles
percent = 0                                                #pourcentage de simulations avec des equipes jumelles
z1 = 0                                                     #variable pour compter le nombre de simulations
z2 = int(input("Nombre de simulations : "))                #nombre de simulations pour le nombre n d'equipes fixe
print("")
print("")

while z1 < z2:

    z1 = z1 + 1

    listeEquipesGagnees = [set() for i in range(n)]     #listeEquipesGagnees est la liste dans laquelle on stockera les equipes gagnees par chaque equipe

    for i in range(n) :

            for k in range (i+1 , n) :

                GagnePerdu = randint(1 , 2)             #le programme choisit de façon aleatoire 1 ou 2 : 1 si l'equipe i gagne la k et 2 si l'equipe i perd contre la k

                if GagnePerdu == 1 :
                    listeEquipesGagnees[i].add(k)       #on rajoute l'equipe k gagnee par i dans l'ensemble de l'equipe i

                elif GagnePerdu == 2 :
                    listeEquipesGagnees[k].add(i)       #on rajoute l'equipe i gagnee par k dans l'ensemble de l'equipe k


    #print("listeEquipesGagnees : ",listeEquipesGagnees)


        #on recherche maintenant les equipes jumelles
    for a in range(n) :

            for b in range(a+1 , n) :

                #on crée 2 nouvelles listes sans le match a contre b (pour eviter de changer les listes initiales qui sont mutables)

                l1=[i for i in listeEquipesGagnees[a] if i!=b]
                l2=[i for i in listeEquipesGagnees[b] if i!=a]

                #on compare les 2 listes fabriquees pour voir si elles sont jumelles
                if l1 == l2 :
                    cptj = cptj + 1                         #on compte le nombre de paires d'equipes jumelles sur tous les matchs joues
                    print("- Les equipes ",a," et ",b," sont jumelles")


    if cptj == 1 :
        print("Il y a 1 paire d'equipes jumelles.")         #on obtient cptj au singulier
        print("")

    else :
        print("Il y a" , cptj , "paires d'equipes jumelles.") #on obtient cptj au pluriel
        print("")

    if cptj > 0 :
      cptt = cptt + 1


    listeEquipesGagnees = []
    l1 = []
    l2 = []
    cptj = 0
    i = 0
    k = 0


percent = (cptt*100)/z2
print ("")
print ("Sur", z2, "simulations, avec", n, "equipes, il y a eu", cptt, "fois des equipes jumelles, soit", percent, "%.")
```

Il devient ainsi possible de chercher un seuil (un nombre d'équipes) à partir duquel la probabilité d'obtenir deux équipes jumelles dans un tournoi tiré aléatoirement devient très faible.

Pour un nombre donné d'équipes, dès que le pourcentage d'équipes jumelles est nul, le nombre de simulations est augmenté jusqu'à ce que le pourcentage diffère de zéro. À partir de cet instant, les simulations peuvent se faire sur le nombre d'équipes suivant (c'est-à-dire en ajoutant une équipe) et ainsi de suite jusqu'à ce que soit trouvé le seuil recherché.

Le tableau suivant récapitule l'occurrence[2] d'équipes jumelles (en %) en fonction du nombre d'équipes.

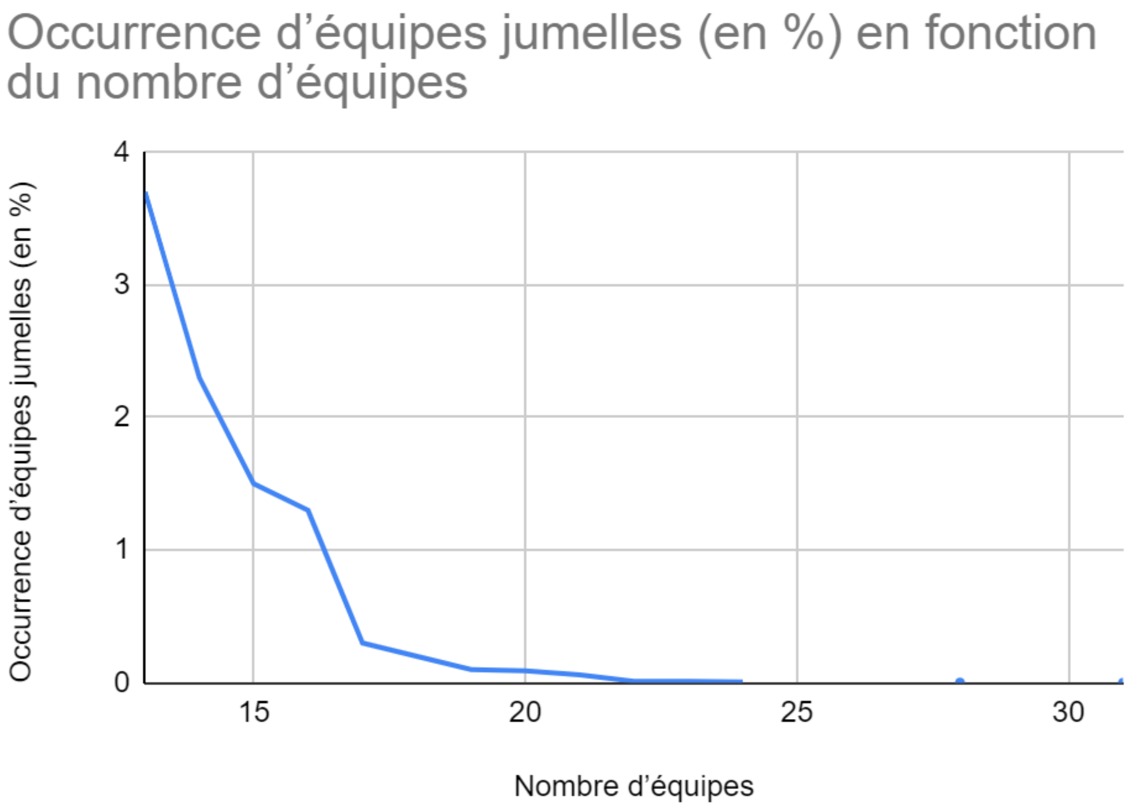
	A	B	C
1	Nombre d'équipes	Nombres de simulations	Occurrence d'équipes jumelles (en %)
2	13	1000	3,7
3	14	1000	2,3
4	15	1000	1,5
5	16	1000	1,3
6	17	1000	0,3
7	18	1000	0,2
8	19	1000	0,1
9	20	10000	0,09
10	21	10000	0,06
11	22	10000	0,01
12	23	10000	0,01
13	24	100000	0,005
14	25	100000	0.001
15	26	100000	0.001
16	27	100000	0.002
17	28	200000	0,0015
18	29	200000	0.0005
19	30	400000	0.00025
20	31	1000000	0

Ce programme met en évidence qu'**à partir de 31 équipes par tournoi, la probabilité de tirer un tournoi contenant deux équipes jumelles est très faible, inférieure à 10^{-5}**.

[2] L'occurrence d'un événement est son apparition dans le temps ou dans l'espace.

III – Le graphe

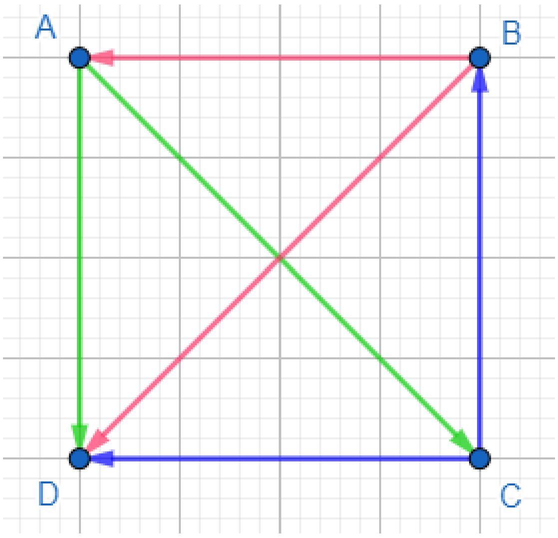
Un graphe extrapolé représentant l'occurrence d'équipes jumelles (en %) en fonction du nombre d'équipes peut être réalisé.



Lorsque le nombre d'équipes augmente, l'occurrence d'équipes jumelles tend vers 0. Cela confirme la conjecture précédente.

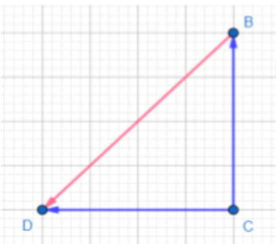
IV – La deuxième propriété

Après la première propriété, il convient de considérer la deuxième. Il peut être considéré à nouveau une simulation à quatre équipes dans laquelle :



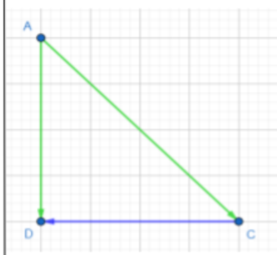
A a gagné C et D, B a gagné A et D, C a gagné B et D, D n’a gagné aucun match. Il n'existe pas d’équipes “jumelles” dans ce tournoi.

- Si on supprime l’équipe A :



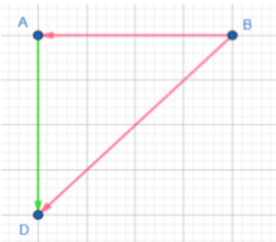
B et C deviennent jumelles

- Si on supprime l’équipe B :



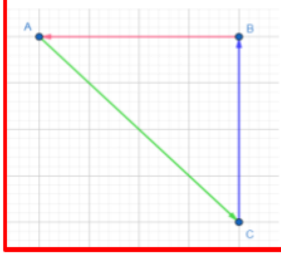
A et C deviennent jumelles

- Si on supprime l’équipe C :



A et B deviennent jumelles

- Si on supprime l’équipe D :



Aucune équipe jumelle n’apparaît

Ainsi, la suppression successive des équipes A, B et C entraîne l’apparition systématique d'équipes jumelles. Par contre, en supprimant l’équipe D, aucune équipe jumelle n’apparaît.

À la fin du tournoi dans lequel il n’existe pas deux équipes “jumelles”, si une équipe n’ayant gagné aucun match est supprimée, cela revient à enlever un gain de match à toutes les autres. Ces dernières

se retrouvent alors avec le même nombre de gains moins un et avec les mêmes équipes gagnées moins une : celle qui a été supprimée.
Donc soustraire une équipe n’ayant rien gagné à un ensemble d’équipes sans équipes jumelles n’en fera pas apparaître.

Suivant le même raisonnement, supprimer une équipe ayant gagné tous ses matchs ne fera pas non plus apparaître d’équipes jumelles.

Ainsi, des tournois vérifiant la deuxième propriété peuvent être caractérisés. Dans un tournoi sans équipes jumelles, pour que deux équipes deviennent jumelles après avoir supprimé n'importe quelle équipe, **il ne faut pas qu’il y ait d’équipe ayant gagné ou perdu tous ses matchs** dans ce tournoi. Ceci est une condition nécessaire.

4. Conclusion

Ainsi, quel que soit le nombre d'équipes dans le tournoi, il est impossible d’avoir “à coup sûr” un tournoi respectant les deux propriétés de l’énoncé. En effet, quel que soit le nombre d’équipes dans le tournoi, il serait toujours possible de construire 2 équipes jumelles puis d’ajouter aléatoirement tous les matchs restants.
Cependant, pour améliorer la probabilité d’obtenir un tournoi qui vérifie les deux propriétés parmi l’ensemble des tournois possibles, on peut **prendre un nombre d’équipes élevé** et **empêcher d’avoir une équipe qui a gagné ou perdu tous ses matchs**.

5. Remerciements

Nous tenons à remercier :
- M. Bertrand Jouve qui a proposé ce sujet et suggéré des pistes de réflexion ;
- M^{me} Sandrine Vernhet et M. Laurent Thomas pour leur soutien et leurs conseils ;
- L'association MATh.en.JEANS pour la richesse de l'expérience qu'elle offre aux élèves.