DANS NOS CLASSES

Une année MATh.en.JEANS à Épinal (suite)

Cet article fait suite à celui consacré aux ateliers MATh.en.JEANS 2013-2014 d'Epinal, paru dans le Petit Vert n°119. Un groupe d'élèves nous rend compte de son travail de recherche autour du jeu du Colonel Blotto.

Le jeu du Colonel Blotto

Clara Pierre, Joseph Laurent, Quentin Michel, Clément Roy (Lycée Mendes France)
Cassandra Clément, Marie Paillard, Mélissa Durand Flon (Lycée Lapicque)

Le jeu du Colonel Blotto est un jeu qui se joue à deux. Il se déroule sur 3 terrains. Chaque joueur dispose d'une armée de 12 soldats qu'il faut répartir sur trois terrains en suivant ces règles : Le nombre de soldats placés sur le terrain A doit être supérieur ou égal au nombre de soldats placés sur le terrain B et le nombre de soldats sur le terrain B doit être supérieur ou égal à celui du terrain C. Sur chacun des terrains, les soldats de chaque joueur s'affrontent et gagnent à chaque fois qu'ils sont en supériorité numérique.
Par exemple :

	Terrain A	Terrain B	Terrain C
Joueur 1	9	3	0
Joueur 2	7	4	1

Sur le terrain A, c'est le joueur 1 qui gagne car il a plus de soldats que le joueur 2 (9>7)
Sur le terrain B, c'est le joueur 2 qui gagne car il a plus de soldats que le joueur 1 (4>3)
Sur le terrain C, c'est le joueur 2 qui gagne car il a plus de soldats que le joueur 1 (1>0)
En résumé, le joueur 2 remporte plus de batailles que le joueur 1 (2 batailles sur 3) donc c'est J2 qui gagne la partie.
Notre chercheur, Julien, après nous avoir exposé ce jeu, nous a posé la problématique suivante :

« Comment jouer pour perdre le moins souvent possible ? »

Pour répondre à cette question nous avons décidé de lister toutes les combinaisons possibles.

Nombre a de soldats à placer sur le terrain A
12 est le plus grand nombre de soldats que l'on puisse placer sur ce terrain car c'est le nombre de soldats dont on dispose. La combinaison 12/0/0 (12 soldats sur le terrain A, 0 sur B et 0 sur C) est possible car elle respecte les règles imposées : $12 \geqslant 0 \geqslant 0$
a peut prendre la valeur 4, par exemple en choisissant la combinaison 4/4/4. Mais 4 est la plus petite valeur possible pour a : car si on mettait moins de 4 soldats sur le terrain A, par exemple si on enlevait 1 soldat du terrain A, il faudrait le reporter sur le terrain B ou sur le terrain C, ce qui n'est pas possible car il y aurait alors plus de soldats sur ces terrains que sur le terrain A. Le nombre minimum à placer dans le terrain A est donc 4. Donc le nombre de soldats possibles à mettre dans le terrain A est compris entre 4 et 12.

Nombre b de soldats à placer sur le terrain B
Grâce à la combinaison 12/0/0, on voit que l'on peut mettre 0 au minimum. Et d'ailleurs, il n'est pas possible que le nombre de soldats soit négatif.
Si le nombre de soldats sur le terrain B pouvait être supérieur à 6 par exemple 7, il faudrait mettre alors au moins 7 soldats sur le terrain A. Or 7+7=14, ce qui n'est pas possible car on ne dispose que de 12 soldats. On peut effectivement mettre 6 soldats dans le terrain B en choisissant la combinaison 6/6/0. Le nombre de soldats à placer dans le terrain B est donc compris entre 0 et 6.

Nombre c de soldats à placer sur le terrain C

c peut prendre au minimum la valeur 0, c'est le cas dans les combinaisons 6/6/0 et 12/0/0 et d'ailleurs, le nombre de soldats ne peut être négatif.

Le maximum pour c est 4. Parce que si l'on mettait 5 sur ce terrain, il faudrait également placer au moins 5 sur les terrains A et B. Mais 5+5+5=15 et on ne dispose que de 12 soldats, ce n'est donc pas possible.

Le nombre de soldats dans le terrain C est donc compris entre 0 et 4.

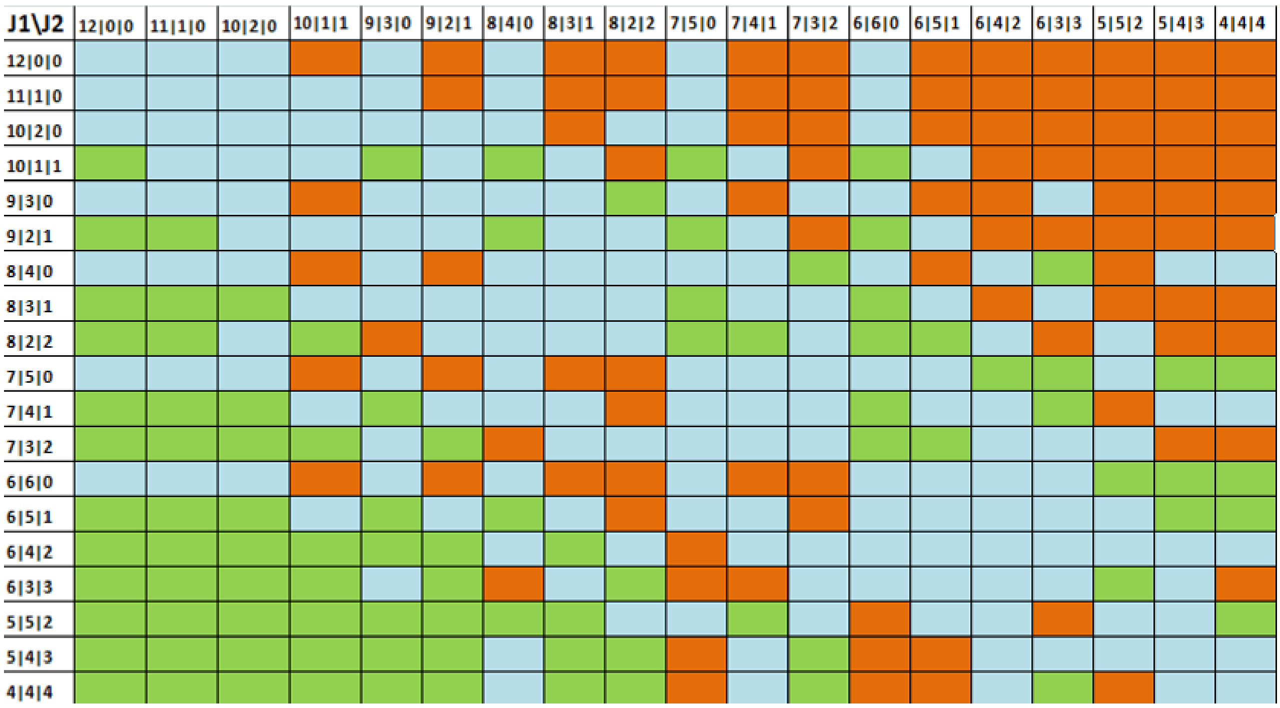
Finalement on obtient le tableau suivant :

	Nombre de soldats a dans le terrain A	Nombre de soldats b dans le terrain B	Nombre de soldats c dans le terrain C
Min	4	0	0
Max	12	6	4

Nous avons recherché toutes les possibilités et nous avons trouvé 19 combinaisons.

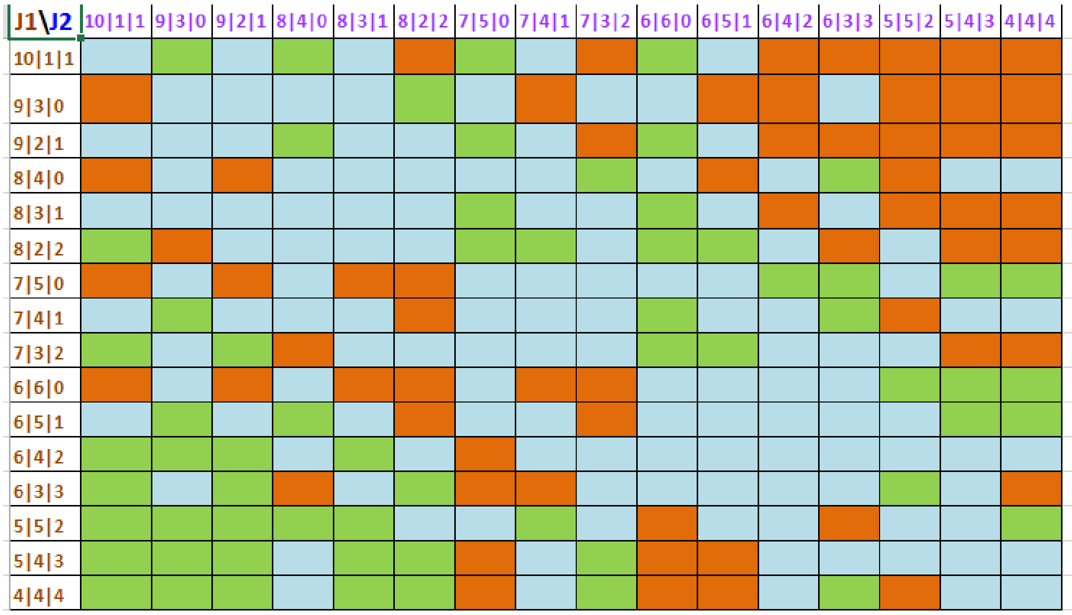
Ensuite, nous avons fait combattre chaque combinaison contre toutes les autres et nous avons noté les résultats de chaque combat.

Ce qui nous donne le tableau suivant, qui recense tous les combats :



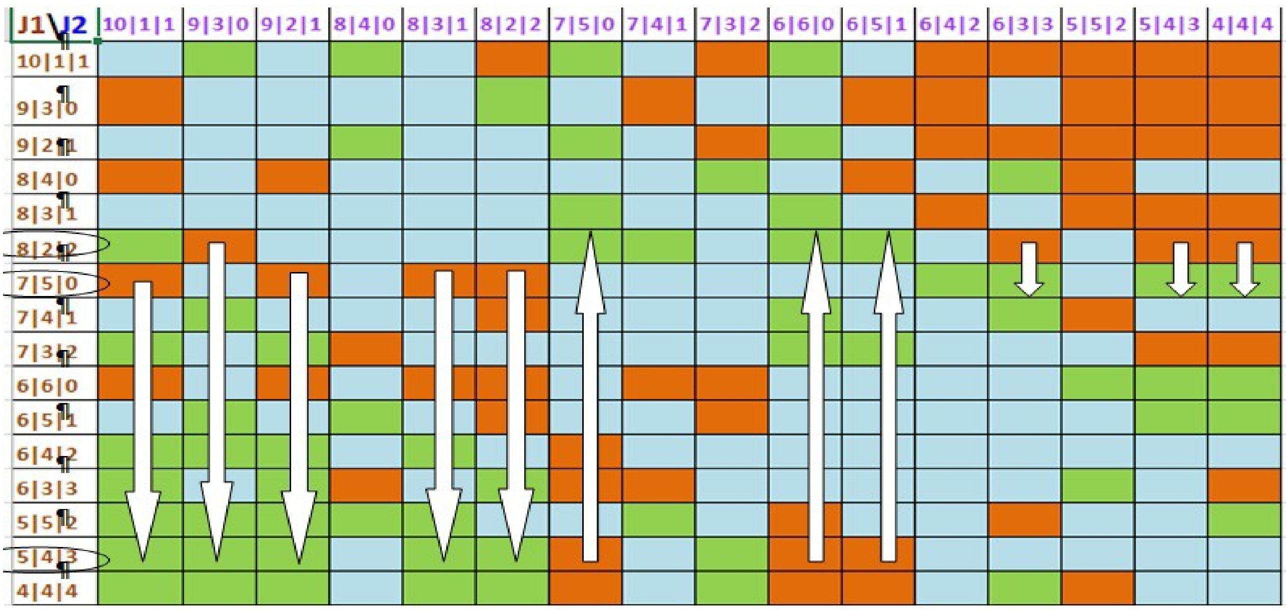
Grâce à ce tableau nous avons pu remarquer qu'aucune combinaison ne perdait jamais, mais également que 3 combinaisons ne gagnaient jamais (12/0/0 ; 11/1/0 ; 10/2/0). Donc nous avons refait notre tableau en enlevant ces 3 combinaisons, ce qui nous donne le tableau suivant.

sommaire

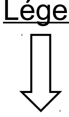


Si on choisit de jouer avec une combinaison, notre adversaire peut toujours trouver une combinaison qui la bat. On a donc décidé de chercher 2 combinaisons qui se complètent afin qu'aucune autre ne les battent les deux à la fois. Mais cette fois encore, nous n'avons trouvé aucun couple de combinaison qui reste invaincu. Exemple : Nous avons donc essayé d'associer : 8/4/0 et 7/5/0. Cependant, la combinaison 9/2/1 bat les deux à la fois, les deux combinaisons précédentes ne sont donc pas optimales.

Ensuite, nous avons cherché un groupe de 3 combinaisons qui se complètent et nous avons trouvé le groupe suivant : 8/2/2 ; 7/5/0 ; 5/4/3. Ce groupe n'est pas imbattable certes, mais si nous jouons un grand nombre de fois aléatoirement ces trois combinaisons de façon équiprobable, aucune combinaison choisie par notre adversaire ne pourra les battre toutes à la fois, et en moyenne, nous pouvons espérer gagner davantage que notre adversaire. En effet, quand une de nos combinaisons (ex : 8/2/2) perd contre une autre combinaison (ex. : 6/3/3), une autre combinaison de notre groupe (ex. : 7/5/0) bat cette dernière.



Légende


La flèche part de l'endroit où une des combinaisons du groupe perd et se termine à l'endroit où une autre gagne et donc compense la perte.

sommaire

Une fois tout ce travail réalisé, nous avons créé un algorithme permettant de simuler un combat entre l'ordinateur qui appliquera notre stratégie (il jouera aléatoirement une de nos 3 combinaisons) et une combinaison quelconque que notre adversaire pourrait choisir parmi les 19 (cette combinaison sera entrée dans la liste P). L'algorithme permet de simuler un grand nombre de combats très rapidement, l'ordinateur va jouer x fois notre stratégie face à la même combinaison. Grâce à cet algorithme nous avons pu vérifier expérimentalement qu'en moyenne notre stratégie est efficace face à toute combinaison choisie par l'adversaire.

Voici ci-contre, les résultats de 500 combats entre notre groupe de combinaisons et la combinaison 4/4/4.

```
***Algorithme lancé***
Veuillez entrer les trois nombres de votre combinaison
Entrer le terme de rang i de la liste P : 4
Entrer le terme de rang i de la liste P : 4
Entrer le terme de rang i de la liste P : 4
Combien de fois voulez-vous jouer cette combinaison contre l'ordinateur:
Entrer x : 500
Vous avez gagné 159 fois.
Vous avez fait 166 égalité(s);
Vous avez perdu 175 fois.
Si vous voulez continuer à jouer, entrez 1 en valeur de u
Sinon entrez 0 en valeur de u
Entrer u :
```

Avec notre stratégie, nous pouvons remarquer que, contre la combinaison 4/4/4, nous pouvons espérer perdre moins souvent que gagner ou faire égalité.

Ci-dessous, notre algorithme écrit sur Algobox.

```
VARIABLES
  L EST_DU_TYPE LISTE
  M EST_DU_TYPE LISTE
  N EST_DU_TYPE LISTE
  P EST_DU_TYPE LISTE
  i EST_DU_TYPE NOMBRE
  a EST_DU_TYPE NOMBRE
  C EST_DU_TYPE LISTE
  combat EST_DU_TYPE LISTE
  x EST_DU_TYPE NOMBRE
  compteurgain EST_DU_TYPE NOMBRE
  compteurégalité EST_DU_TYPE NOMBRE
  compteurdéfaite EST_DU_TYPE NOMBRE
  j EST_DU_TYPE NOMBRE
  u EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
  u PREND_LA_VALEUR 1
  L[1] PREND_LA_VALEUR 8
  L[2] PREND_LA_VALEUR 2
  L[3] PREND_LA_VALEUR 2
  M[1] PREND_LA_VALEUR 7
  M[2] PREND_LA_VALEUR 5
  M[3] PREND_LA_VALEUR 0
  N[1] PREND_LA_VALEUR 5
  N[2] PREND_LA_VALEUR 4
  N[3] PREND_LA_VALEUR 3
  TANT_QUE (u==1) FAIRE
    DEBUT_TANT_QUE
    compteurgain PREND_LA_VALEUR 0
    compteurégalité PREND_LA_VALEUR 0
    compteurdéfaite PREND_LA_VALEUR 0
    AFFICHER "Veuillez entrer les trois nombres de votre combinaison"
    POUR i ALLANT_DE 1 A 3
      DEBUT_POUR
      LIRE P[i]
      FIN_POUR
    AFFICHER "Combien de fois voulez-vous jouer cette combinaison contre l'ordinateur: "
    LIRE x
    POUR j ALLANT_DE 1 A x
      DEBUT_POUR
      a PREND_LA_VALEUR ALGOBOX_ALEA_ENT(1,3)
      SI (a==1) ALORS
        DEBUT_SI
        POUR i ALLANT_DE 1 A 3
          DEBUT_POUR
          C[i] PREND_LA_VALEUR L[i]
          FIN_POUR
        FIN_SI
      SI (a==2) ALORS
        DEBUT_SI
        POUR i ALLANT_DE 1 A 3
          DEBUT_POUR
          C[i] PREND_LA_VALEUR M[i]
          FIN_POUR
          FIN_SI
        SI (a==3) ALORS
          DEBUT_SI
          POUR i ALLANT_DE 1 A 3
            DEBUT_POUR
            C[i] PREND_LA_VALEUR N[i]
            FIN_POUR
        FIN_SI
      POUR i ALLANT_DE 1 A 3
        DEBUT_POUR
        SI (C[i]>P[i]) ALORS
          DEBUT_SI
          combat[i] PREND_LA_VALEUR 1
          FIN_SI
          SINON
            DEBUT_SINON
            SI (C[i]==P[i]) ALORS
              DEBUT_SI
              combat[i] PREND_LA_VALEUR 0
              FIN_SI
              SINON
                DEBUT_SINON
                combat[i] PREND_LA_VALEUR -1
                FIN_SINON
            FIN_SINON
        FIN_POUR
      SI (combat[1]+combat[2]+combat[3]>0)
```

sommaire

```
ALORS
        DEBUT_SI
        compteurdéfaite PREND_LA_VALEUR
      compteurdéfaite+1
        FIN_SI
        SINON
          DEBUT_SINON
          SI (combat[1]+combat[2]+combat[3]==0)
ALORS
            DEBUT_SI
            compteurégalité PREND_LA_VALEUR
      compteurégalité+1
            FIN_SI
            SINON
              DEBUT_SINON
              compteurgain PREND_LA_VALEUR
compteurgain+1
              FIN_SINON
            FIN_SINON
          FIN_POUR
        AFFICHER "Vous avez gagné "
        AFFICHER compteurgain
        AFFICHER " fois."
        AFFICHER "Vous avez fait "
        AFFICHER compteurégalité
        AFFICHER " égalité(s);"
        AFFICHER "Vous avez perdu "
        AFFICHER compteurdéfaite
        AFFICHER " fois."
        AFFICHER "Si vous voulez continuer à jouer,
entrez                                   en valeur de
u "
        AFFICHER "Sinon entrez 0 en valeur de u"
        LIRE u
        FIN_TANT_QUE

  FIN_ALGORITHME
```

En examinant le tableau pour chercher des combinaisons qui « se complètent », nous n'avons pas trouvé de meilleur choix que notre proposition mais nous n'avons pas prouvé que notre choix de 3 combinaisons était le meilleur. Pour étudier tous les triplets possibles il y aurait (16 x 15 x14)/(3 x 2x1) = 560 triplets de combinaisons à comparer.

			9	8	7	7	6	6	5	5	5	4	4	3
			0	1	2	1	3	2	4	3	2	4	3	3
			0	0	0	1	0	1	0	1	2	1	2	3
9	0	0	3	3	3	4	3	4	3	4	4	4	4	4
8	1	0	3	3	3	3	3	4	3	4	4	4	4	4
7	2	0	3	3	3	3	3	3	3	4	3	4	4	4
7	1	1	2	3	3	3	2	3	2	3	4	3	4	4
6	3	0	3	3	3	4	3	3	3	3	2	4	3	3
6	2	1	2	2	3	3	3	3	2	3	3	3	4	4
5	4	0	3	3	3	4	3	4	3	3	3	3	2	2
5	3	1	2	2	2	3	3	3	3	3	3	3	3	3
5	2	2	2	2	3	2	4	3	3	3	3	2	3	4
4	4	1	2	2	2	3	2	3	3	3	4	3	3	2
4	3	2	2	2	2	2	3	2	4	3	3	3	3	3
3	3	3	2	2	2	2	3	2	4	3	2	4	3	3

Le joueur 1 gagne.

Le joueur 1 perd.

Les joueurs 1 et 2 à égalité.

Les élèves du Lycée Lapicque d'Épinal, jumelé avec notre lycée sur ce projet, ont dirigé leurs recherches sur le cas où on avait 9 soldats à répartir, puis 10 soldats. Dans chacun de ces cas, ils ont également construit un tableau similaire au nôtre montrant les différentes combinaisons et les résultats des affrontements entre ces combinaisons.

			10	9	8	8	7	7	6	6	6	5	5	5	4	4
			0	1	2	1	3	2	4	3	2	5	4	3	4	3
			0	0	0	1	0	1	0	1	2	0	1	2	2	3
10	0	0	3	3	3	4	3	4	3	4	4	3	4	4	4	4
9	1	0	3	3	3	3	3	4	3	4	4	3	4	4	4	4
8	2	0	3	3	3	3	3	3	3	4	3	3	4	4	4	4
8	1	1	2	3	3	3	2	3	2	3	4	2	3	4	4	4
7	3	0	3	3	3	4	3	3	3	3	2	3	4	3	4	3
7	2	1	2	2	3	3	3	3	2	3	3	2	3	4	4	4
6	4	0	3	3	3	4	3	4	3	3	3	3	3	2	3	2
6	3	1	2	2	2	3	3	3	3	3	3	2	3	3	4	3
6	2	2	2	2	3	2	4	3	3	3	3	2	2	3	3	4
5	5	0	3	3	3	4	3	4	3	4	4	3	3	3	2	2
5	4	1	2	2	2	3	2	3	3	3	4	3	3	3	3	2
5	3	2	2	2	2	2	3	2	4	3	3	3	3	3	3	3
4	4	2	2	2	2	2	2	2	3	2	3	4	3	3	3	3
4	3	3	2	2	2	2	3	2	4	3	2	4	4	3	3	3

Dans le cas de 9 soldats, contrairement à notre étude sur 12 soldats, ils ont trouvé qu'une combinaison (5/3/1) permettait de ne jamais perdre contre l'adversaire.

Avec 10 soldats, on ne peut de nouveau plus trouver une telle combinaison. Ils ont également proposé un triplet de combinaisons (6/4/0, 6/2/2 et 4/4/2) qui permet sur un grand nombre de parties, de perdre moins souvent que de gagner ou de faire égalité.