SCIENCES
SORBONNE
UNIVERSITÉ

UE : LU3IN005

2019-2020

Projet 1 : Bataille Navale

Mohamed DHIFAOUI & Lucie CUBAUD

Table des matières

Introduction

La **bataille navale**, appelée aussi **touché-coulé**, est un jeu de société dans lequel deux joueurs doivent placer des « navires » sur une grille tenue secrète et tenter de « toucher » les navires adverses. Le gagnant est celui qui parvient à couler tous les navires de l'adversaire avant que tous les siens ne le soient. On dit qu'un navire est coulé si chacune de ses cases a été touchées par un coup de l'adversaire.

Dans ce petit projet, notre objectif est d'étudier le jeu de la bataille navale d'un point de vue probabiliste.

On va modéliser le jeu par une grille de 10 cases par 10 cases. L'adversaire place sur cette grille un certain nombre de bateaux qui sont caractérisés par leur longueur :

- un porte-avions (5 cases)
- un croiseur (4 cases)
- un contre-torpilleur (3 cases)
- un sous-marin (3 cases)
- un torpilleur (2 cases)

Il a le droit de les placer verticalement ou horizontalement. Le positionnement des bateaux reste secret, l'objectif du joueur est de couler tous les bateaux de l'adversaire en un nombre minimum de coup. A chaque tour de jeu, il choisit une case où il tire : s'il n'y a aucun bateau sur la case, la réponse est vide; si la case est occupée par une partie d'un bateau, la réponse est touché; si toutes les cases d'un bateau ont été touchées, alors la réponse est coulé et les cases correspondantes sont révélées. Lorsque tous les bateaux ont été coulés, le jeu s'arrête et le score du joueur est le nombre de coups qui ont été joués. Plus le score est petit, plus le joueur est performant.

1. Combinatoire du jeu

Q1 : Donner une borne supérieure du nombre de configurations possibles pour la liste complète de bateaux sur une grille de taille 10.

On dispose d'une matrice de taille 10 x 10, ainsi que de 5 bateaux. Pour obtenir une borne supérieure du nombre de configurations possibles pour la liste complète des bateaux, il suffit de calculer le nombre de façons possibles pour placer chaque bateau et multiplier les résultats trouvés. Le dessin qui se trouve en dessous nous donne une idée sur les façons possibles pour placer chaque bateau :

Couleur noir : les positions où on peut placer le bateau horizontalement ou verticalement

Couleur rouge : les positions où on peut placer le bateau que horizontalement

Couleur vert : les positions où on peut placer le bateau que verticalement

x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x				
x	x	x	x	x	x				
x	x	x	x	x	x				
x	x	x	x	x	x				

o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o	o	o	o
o	o	o	o	o	o	o			
o	o	o	o	o	o	o			
o	o	o	o	o	o	o			

s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s	s	s
s	s	s	s	s	s	s	s		
s	s	s	s	s	s	s	s		

t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	t
t	t	t	t	t	t	t	t	t	

x : porte-avions

o : croiseur

t : torpilleur

s : sous-marin ou contre-torpilleurs

longueur du bateau	nombre de cases occupés par le bateau horizontalement	nombre de cases occupés par le bateau verticalement	nombre de façons possibles pour placer le bateau
5	60	60	120
4	70	70	140
3	80	80	160
2	90	90	180

Donc d'après le tableau précédent on a 120 * 120 *160*160*180 = 66 355 200 000 configurations possibles pour la liste complète des bateaux sur une grille de taille 10.

Q2 : Donner d'abord une fonction qui permet de dénombrer le nombre de façons de placer un bateau donné sur une grille vide. Comparer au résultat théorique.

Un bateau de taille n sur une ligne (ou colonne) peut, théoriquement, être placé de (10 – n + 1) façons différentes. Sachant que notre matrice est de taille 10 x 10 et qu'un bateau peut être placé horizontalement ou verticalement, on aura donc 2 x 10 x (10 – n + 1) façons différentes de placer un bateau sur une grille vide.

Exemple :

Prenons par exemple un **porte-avions** (taille 5)

X : les positions où on peut placer le porte-avions horizontalement ou verticalement (il y en a 6*6=36)

X : les positions où on peut placer le porte-avions que horizontalement (il y en a 4*6=24)

X : les positions où on peut placer le porte-avions que verticalement (il y en a 4*6=24)

x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x				
x	x	x	x	x	x				
x	x	x	x	x	x				
x	x	x	x	x	x				

Donc au total on a 36*2+24+24=120 ce qui est aussi le cas avec notre formule (2*10*(10-5+1)=120).

Pour calculer le nombre de façons de placer un bateau donné sur une grille vide on va utiliser la fonction **placer_un_bateau** qui prend en paramètre un bateau et retourne le nombre possibilité de le place sur une grille vide. Elle parcourt toutes les cases de la grille et appelle la fonction **peut_placer** qui retourne vrai s'il est possible de placer le bateau sur la grille et qui comme dans la bataille navale fait l'hypothèse qu'on ne colle pas les bateaux et qu'on ne connaît pas le type de bateau que l'on touche. On a mis les résultats trouvés dans le tableau ci-dessous.

TABLE : Comparaison du nombre de façons de placer un bateau sur une grille vide

Liste de bateaux	Résultats théoriques	Résultats expérimentaux
[5]	180	180
[3]	160	160
[2]	140	140
[1]	120	120

Q3 : Donner une fonction qui permet de dénombrer le nombre de façons de placer une liste de bateaux sur une grille vide. Calculer le nombre de grilles différentes pour 1, 2 et 3 bateaux. Est-il possible de calculer de cette manière le nombre de grilles pour la liste complète de bateaux ?

Pour dénombrer le nombre de façons de placer une liste de bateaux sur une grille vide on va utiliser la fonction **placer_des_bateaux** qui va prendre en paramètre la liste de bateaux (qui contient au moins un bateau) et retourner le nombre de possibilité de placer la liste des bateaux sur la grille. Cette fonction fait le traitement suivant : On commence par placer le premier bateau à la première position possible en utilisant la fonction **peut_ placer** puis le deuxième bateau puis le troisième ... Une fois tous les bateau sont placés, on place le dernier bateau à la deuxième position possible puis la troisième jusqu'à ce qu'ils épuisent toutes les positions possibles. Maintenant, l'avant dernier bateau est placé à la deuxième position possible et on recommence à placer le dernier bateau exactement comme on a fait précédemment à toutes les positions possibles.

TABLE : le nombre de grilles différentes pour 1,2 et 3 bateaux donné par la fonction **placer_des_bateaux**

Nombre de bateaux	Liste de bateaux	Placer_des_bateaux
Un bateau	[5]	180
Deux bateaux	[5 , 4]	24 744
Trois bateaux	[5 , 4 , 3]	2 754 616

On remarque que le nombre de façons de placer une liste de bateaux sur une grille vide se multiplie par ≈ 120 à chaque fois qu'on ajoute un bateau à la liste. Du coup, si avec 3 bateaux on a 2 754 616 façons différentes de placer les bateaux, on aura donc ≈ 330 553 920 façons pour 4 bateaux et ça sera ≈39 666 470 400 façons. On remarque dès qu'on commence à avoir 3 bateaux la fonction **placer_des_bateaux** consomme trop de ressources et prend beaucoup de temps pour trouver le nombre de façons de placer la liste de bateaux donc ce n'est pas une bonne idée de calculer de cette manière le nombre de grilles pour la liste complète de bateaux.

Q4 : En considérant toutes les grilles équiprobables, quel est le lien entre le nombre de grilles et la probabilité de tirer une grille donnée ? Donner une fonction qui prend en paramètre une grille, génère des grilles aléatoirement jusqu'à ce que la grille générée soit égale à la grille passée en paramètre et qui renvoie le nombre de grilles générées.

Soit un univers fini Ω.

Un évènement E1 : « tirer une grille donnée »

On définit la probabilité uniforme sur Ω est par la fonction de masse $\mathrm{p}(\omega) = \frac{1}{\mathrm{card}(\Omega)}$.

La probabilité d'un événement E quelconque de Ω est : $\mathrm{P}(\mathrm{E}) = \frac{\mathrm{card}(\mathrm{E})}{\mathrm{card}(\Omega)}$.

On aura donc $\mathrm{P}(\mathrm{E1}) = \frac{\mathrm{card}(\mathrm{E1})}{\mathrm{card}(\Omega)} = \frac{1}{\text{le nombre de grilles}}$.

On a la fonction **grille_egale_alea** qui prend en paramètre une grille et une liste de bateaux et qui va générer des grilles aléatoirement jusqu'à ce que la grille générée soit égale à la grille passée en paramètre et qui renvoie le nombre de grilles générées. Pour tester cette fonction, nous avons implémenté une fonction **test_grille_egale_alea** qui prend en paramètre une durée et une liste de bateaux, et qui appelle jusqu'à la fin de la durée la fonction **grille_egale_alea** et retourne la moyenne des résultats obtenus. On a mis les résultats trouvés dans le tableau ci-dessous :

TABLE : la moyenne de nombre de grilles générées pendant une durée donnée par la fonction **test_grille_egale_alea**

Liste de bateaux	Nombre d'itérations en 60 secondes	La moyenne de nombre de grilles générées
[5]	4747	178
[5 , 4]	30	25 235
[5 , 4 , 3]	2	1 353 375

On remarque que dès qu'on passe trois bateaux la moyenne de nombre de grilles générées devient énormément grand ce qui prend beaucoup de temps.

Q5 : Donner un algorithme qui permet d'approximer le nombre total de grilles pour une liste de bateaux. Comparer les résultats avec le dénombrement des questions précédentes. Est-ce une bonne manière de procéder pour la liste complète de bateaux ?

On trouve en dessous l'algorithme de la fonction **approx_nb_grilles1** qui prend en paramètre la liste de bateaux et qui utilise la fonction **place_un_bateau** pour placer chaque bateau de la liste. Elle retourne le nombre de possibilité de placer un bateau donné sur une grille donnée :

```
approx_nb_grilles1(liste_bateau) :
    res := 1
    Pour chaque bateau i de liste_bateau
        Res := res × placer_un_bateau(i)
    renvoyer res
```

On trouve dans le tableau ci-dessous la comparaison des résultats de la fonction placer_des_bateaux et approx_nb_grilles1 :

Liste de bateaux	placer_des_bateaux	approx_nb_grilles1
[5]	180	180
[5 , 4]	24 744	28800
[5 , 4 , 3]	2 754 616	4 608 000
[5 , 4 , 3 , 2]	Prends beaucoup de temps	645 120 000
[5 , 4 , 3 , 2 , 1]	Prends beaucoup de temps	77 414 400 000

Q6 (bonus) : Proposer des solutions pour approximer plus justement le nombre de configurations.

On trouve en dessous l'algorithme de la fonction **approx_nb_grilles2** qui commence par calculer le nombre de grilles possibles en plaçant le premier bateau à l'aide de la fonction **place_un_bateau2**, puis elle utilise la fonction **place_alea** pour placer ce bateau dans une nouvelle grille et utilisée cette dernière pour placer le deuxième bateau jusqu'à ce que tous les bateau soient placés :

```
approx_nb_grilles2(liste_bateau) :
    res := 1
    grille := [ ]
    Pour chaque bateau i de liste_bateau
        Res := res × placer_un_bateau2(grille ,i)
        Place_alea(grille,i)
    renvoyer res
```

On trouve dans le tableau ci-dessous la comparaison des résultats de la fonction placer_des_bateaux , approx_nb_grilles1 et approx_nb_grilles2 :

Liste de bateaux	Placer_des_bateaux	approx_nb_grilles1	approx_nb_grilles2
[5]	180	180	180
[5 , 4]	24 744	28 800	24 480
[5 , 4 , 3]	2 754 616	4 608 000	2 815 200
[5 , 4 , 3 , 2]	Prends beaucoup de temps	645 120 000	200 154 240
[5 , 4 , 3 , 2 , 1]	Prends beaucoup de temps	77 414 400 000	5 554 080 000

On remarque que mettre à jour le grille à chaque fois qu'on place un bateau de la liste de bateaux avec la fonction **approx_nb_grilles2** nous donne une approximation plus proche de la réalité, ce qui est différent avec la fonction **approx_nb_grilles1** qui à chaque fois repart de la grille vide pour placer un bateau.

2. Modélisation probabiliste du jeu

Partie 1 : Version aléatoire

Q1.1 : En faisant des hypothèses que vous préciserez, quelle est l'espérance du nombre de coups joués avant de couler tous les bateaux si on tire aléatoirement chaque coup?

Soit
- N = 100 le nombre de cases de la grille
- q le nombre de cases occupés par les bateaux
- N - q le nombre de cases occupés par l'océan
- Y : le nombre de cases de bateaux tirées

On commence par tirer aléatoirement k cases.

Pour calculer le nombre de combinaisons correspondant à n cases bateaux on doit multiplier le nombre de possibilités de tirage de n cases de bateaux parmi q par le nombre de possibilités de tirage du reste ce qui représente $k - n$ cases de l'océan parmi $100 - q$. Après on divise le résultat trouvé par le nombre total de tirages.

La probabilité d'avoir n cases de bateaux tirés est donc :

$$p(Y = n) = \frac{\binom{q}{n}\binom{N-q}{k-n}}{\binom{N}{k}} = \frac{\binom{17}{n}\binom{100-17}{k-n}}{\binom{100}{k}} = = \frac{\binom{17}{n}\binom{83}{k-n}}{\binom{100}{k}}$$

On peut donc déduire la probabilité d'avoir au moins 17 cases de bateaux parmi k tirées :

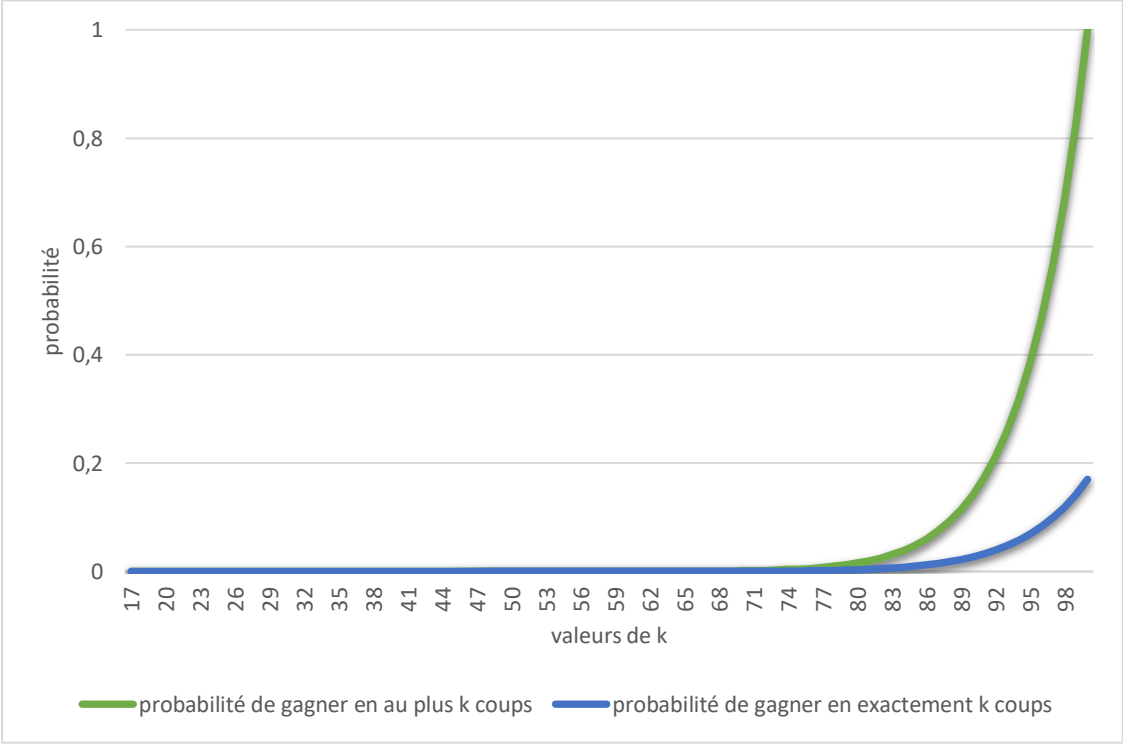
$$p(X \leq k) = \frac{\binom{17}{17}\binom{83}{k-17}}{\binom{100}{k}} = \frac{\binom{83}{k-17}}{\binom{100}{k}}$$

Par conséquent, la probabilité d'avoir besoin d'exactement de k coups pour obtenir les 17 cases de bateaux est :

$$p(X = k) = p(X \leq k) - p(X \leq k-1)$$

Remarque : $p(X \leq 16) = 0$ (C'est impossible d'avoir 17 cases de bateaux en jouant moins de 17 coups)

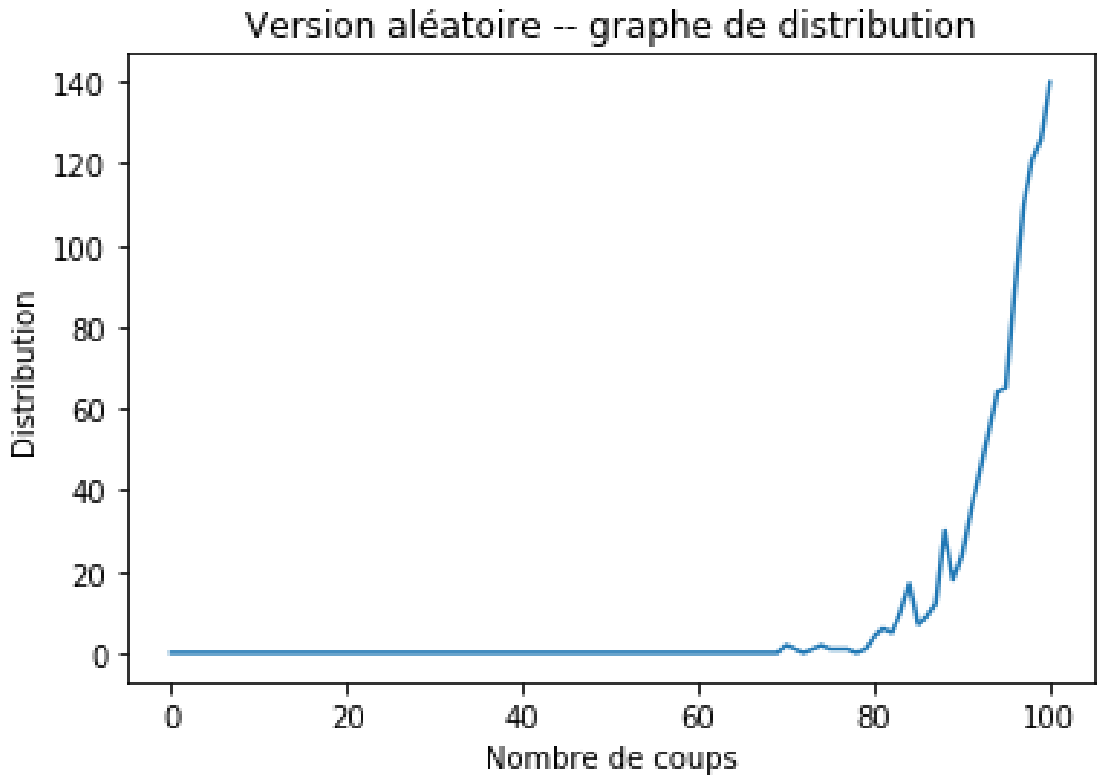
$p(X > 100) = 0$ (On ne peut pas jouer plus de 100 coups).



On trouve alors une espérance de 95,389 ce qui veut dire qu'avec la version aléatoire on peut espérer gagner la partie en moyenne de 95 tours.

<u>Q1.2 :</u> Comment calculer la distribution de la variable aléatoire correspondant au nombre de coups pour terminer une partie ? Tracer le graphique de la distribution. Comparer à l'espérance théorique précédemment calculée. Que remarquez-vous par rapport aux hypothèses formulées ? Y'a-t-il une justification ?

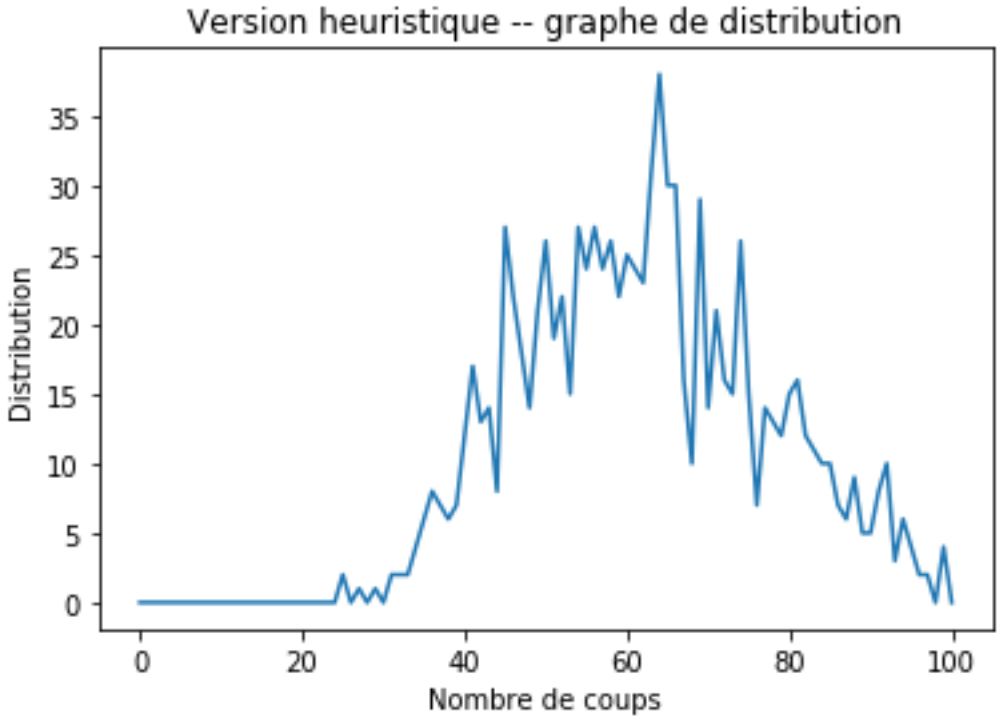
Pour calculer la distribution, on lance 1000 parties et on met le nombre de victoires en fonction du nombre de coups dans un tableau :



Pour calculer l'espérance de la variable aléatoire, on lance 100 parties et on divise la somme de nombres de coups joués par 100. On remarque qu'on obtient un résultat de 95,305 ce qui es presque égale au résultat théorique qui est de 95,389.

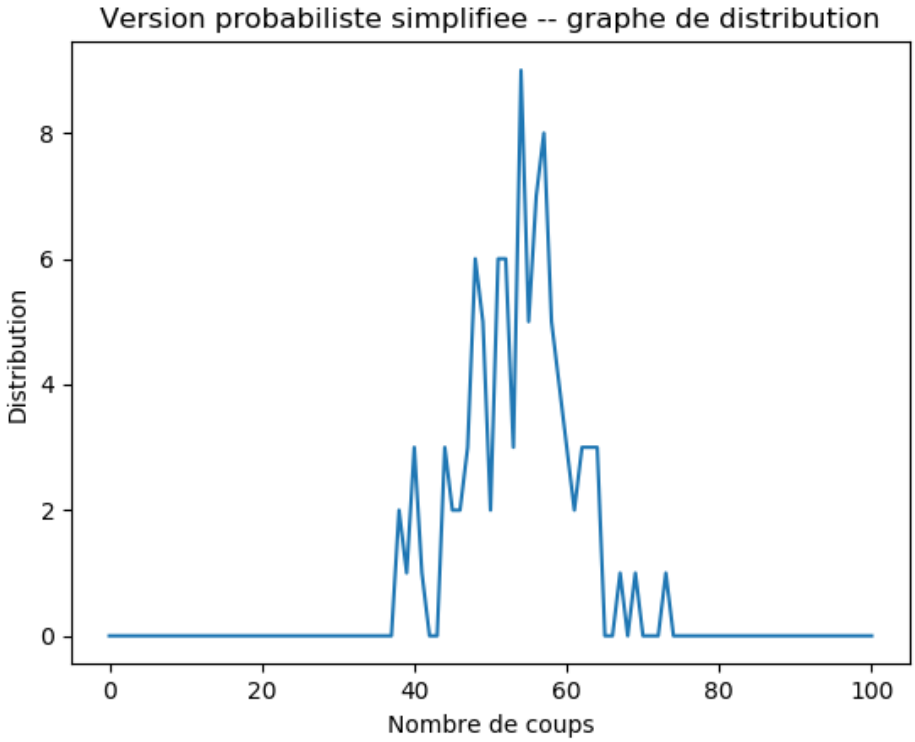
Partie 2 : Version heuristique

Q1.1 : La version aléatoire n'exploite pas les coups victorieux précédents, elle continue à explorer aléatoirement tout l'échiquier. Proposer une version heuristique (à base de règles) composée de deux comportements : un comportement aléatoire tant que rien n'est touché, un comportement qui va explorer les cases connexes lorsqu'un coup touche un bateau. Comparer les résultats et tracer le graphique de la distribution de la variable aléatoire.



Pour calculer l'espérance de la variable aléatoire, on refait exactement ce qu'on a fait avec la version aléatoire : on lance 100 parties et on divise la somme de nombres de coups joués par 100. On remarque qu'on obtient un résultat de 62,688 ce qui est mieux que ce qu'on a trouvé avec la version aléatoire 95,305.

Partie 3 : Version probabiliste simplifiée



On remarque qu'avec la version probabiliste simplifiée on peut gagner une partie en 53,700 coups en moyenne, ce qui est vraiment mieux par rapport à ce qu'on a avec la version aléatoire 95,305 coups et la version heuristique 62,688 coups.

Conclusion

A travers ce projet nous avons vu qu'il existe un très grand nombre de configurations différentes de grille à la bataille navale. A première vue, la bataille navale est un jeu de chance, où celui qui en aura le plus trouvera les bateaux de son adversaire plus rapidement. Cependant, nous avons vu que grâce au calcul de la probabilité de trouver un bateau sur chaque case, un joueur peut considérablement diminuer son nombre de coup. Le calcul de probabilité seul ne garantit pas forcément la victoire contre un autre joueur, mais la combinaison de stratégie et de calcul de probabilité peut néanmoins augmenter ses chances.