LU3IN005
Rida Taleb , Achraf Jday

Compte-rendu de projet

II . Combinatoire du jeu

1.
Nous avons 5 bateaux respectivement de taille 5 pour le porte-avions , 4 pour le croiseur , 3 pour le contre-torpilleur et le sous-marin et enfin 2 pour le torpilleur . Nous devons ainsi les placer sur une grille de taille 10x10 dans toutes les configurations possibles . Une façon naïve d'approcher la question serait de faire :

$$5+4+(2x3)+2 = 17$$

et de faire un arrangement pour sur 100 cases obtenir la somme de toutes les combinaisons possibles . Bien sur cette méthode fournit un résultat bien supérieur au nombre de configurations réel car elle ne prend pas en compte le faite que les cases composants un bateaux doivent être adjacentes .

2.
Un bateau de taille n dispose de 10-n+1 configurations sur une ligne de la grille de dimension 10x10 . Un bateau peut être positionner soit verticalement soit horizontalement et vu que la grille contient 10 lignes nous obtenons donc la formule 2x10x(10-n+1) qui représente le nombre de configuration possible sur toute la grille pour un bateau de taille n .

Nous obtenons donc les résultats théoriques suivants :

Taille n =	2	3	4	5
Nb de config	180	160	140	120

Les résultats obtenus avec la fonction «pos_config_bateau» valident ces résultats théoriques .

3.
Nous avons créer la fonction «pos_config_des_bateaux» qui prend en paramètres une grille et une liste de 3 numéros de bateaux de taille différent ou

égal et renvoie le nombre des configurations possibles . Nous obtenons les résultats suivants :

Nb bateaux Taille n =	1 n1= 3	2 n1= 3, n2= 2	3 n1=3, n2=2, n3=4
Nb de config	160	27 336	3 237 624

Cette méthode met énormément de temps pour une liste de 3 bateaux . Elle n’est pas viable pour une utilisation au-delà de cette quantité .

4.
Soit Ω un univers fini . La probabilité uniforme sur Ω est définie par la fonction de masse :

$$p(\omega) = 1/\mathrm{card}(\Omega)$$

Soit A1 l’événement : « tiré une grille donnée » . La probabilité d’un événement quelconque A de Ω est :

$$P(A) = \mathrm{card}(A)/\mathrm{card}(\Omega)$$

Soit g le nombres de grilles , on a donc :

$$\begin{aligned} P(A1) &= \mathrm{card}(A1)/\mathrm{card}(\Omega) \\ &= 1/g \end{aligned}$$

Nous avons la fonction «RandomEqualGrille» qui prend en paramètre une grille et une liste de bateaux . Elle générera des grilles aléatoirement tant qu’il n’en génère pas une égal à celle en paramètre . Voici les résultats obtenus en faisant la moyenne des réponses renvoyées par la fonction pour une liste :

Liste de bateau	La moyenne du nombre de grilles générées
[3]	113
[3,2]	32 663
[3,2,4]	1 257 345

Dès une liste 3 bateaux la fonction met énormément de temps retourner une réponse car le nombres de grille générée devient important .

Rida Taleb , Achraf Jday

5.
La fonction «approximationAlgorithm» prend en paramètre une liste de bateaux et applique la fonction «pos_config_bateau» à tous les éléments de cette liste et multiplie le résultat avec le précèdent pour finalement renvoyé le nombre total de grille . Voici les résultats obtenus comparé à la fonction «pos_config_des_bateaux» :

A: approximationAlgorithm
B: pos_config_des_bateaux

Nb Bateau Taille n	A	B	Différence entre A et B
1 n=3	160	160	0
2 n=3, n=2	28 800	27 336	1464
3 n=3, n=2, n=4	4 032 000	3 237 624	794 376
4 n=3, n=2, n=4, n=3	645 120 000	Trop de temps	/
5 n=3, n=2, n=4, n=3, n=5	77 414 400 000	Trop de temps	/

Nous pouvons remarquer que la différence entre A et B augmente avec le nombre de bateau dans la liste .

6.
L’approximation précédente repartait à chaque à fois d’une grille vide avant de placer un bateau et ne prend pas en compte la possibilité que 2 bateaux se chevauchent . Pour obtenir une meilleur représentation de la réalité nous allons donc créer la fonction «approximationAlgorithmV2» qui prend en paramètre une liste de bateaux et qui les placera aléatoirement au fur et à mesure sans repartir d’une grille vide à chaque fois.

A: approximationAlgorithm
B: pos_config_des_bateaux
C: approximationAlgorithmV2

Nb Bateau Taille n	A	B	C	Différence entre A et B	Différence entre C et B
1 n=3	160	160	160	0	0
2 n=3, n=2	28 800	27 336	27 200	1464	136
3 n=3, n=2, n=4	4 032 000	3 237 624	3 368 640	794 376	131 016
4 n=3, n=2, n=4, n=3	645 120 000	Trop de temps	394 400 000	/	/
5 n=3, n=2, n=4, n=3, n=5	77 414 400 000	Trop de temps	23 420 723 200	/	/

Nous pouvons remarquer que la fonction «approximationAlgorithmV2» produit belle est bien des approximations plus proches de la réalité .

III . Modélisation probabiliste du jeu

1.

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

Nous allons commencer par tirer aléatoirement un nombre k de cases .

Rida Taleb , Achraf Jday

Pour obtenir le nombre de combinaison correspondant à n cases de bateaux nous devons multiplier le nombre de possibilités de tirage de n cases de bateaux parmi q par le nombre de possibilités de tirage du reste cela représenterai k-n cases de l’océan parmi 100-q . On divisera le résultat obtenu par le nombre total de tirages .

Y étant le nombre de cases de bateaux tirées , la probabilité d’en avoir n s’exprime P(Y = n) et vaut :

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

De cette formule nous pouvons déduire la probabilité d’avoir au moins 17 cases de bateaux parmi k tirées :

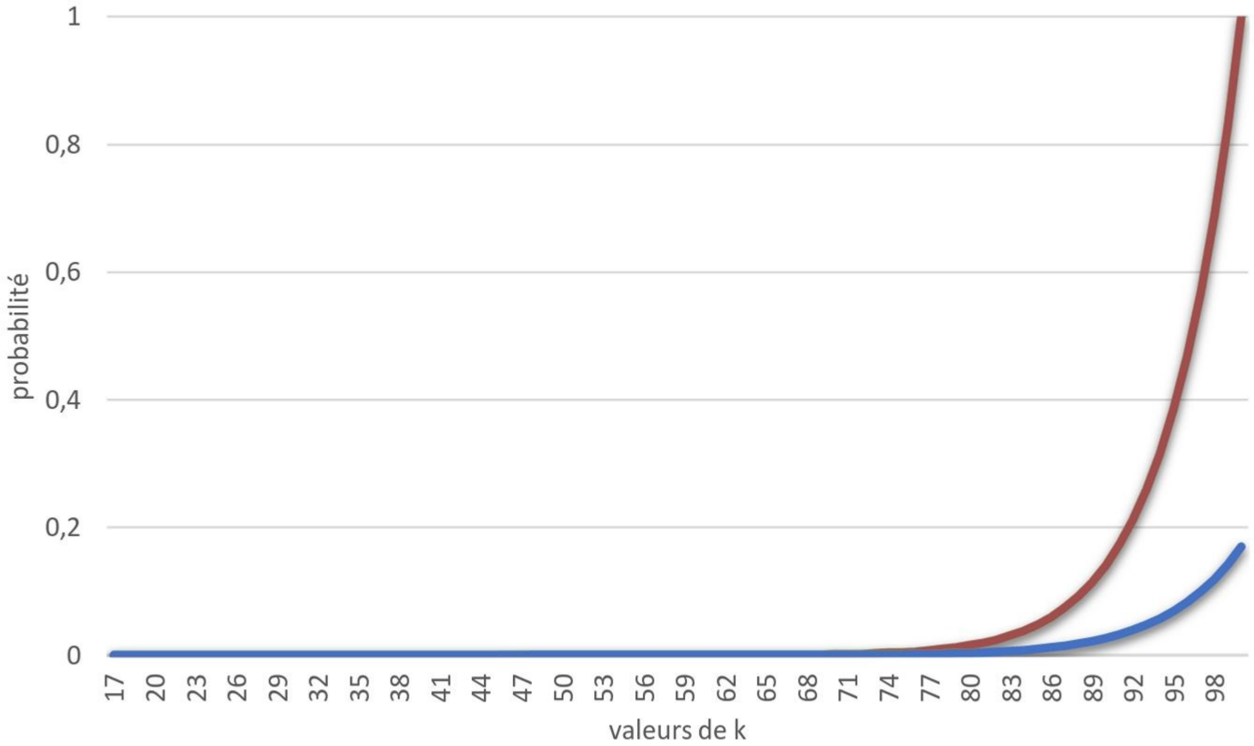
$$P(X \leq k) = \frac{\binom{17}{17}\binom{83}{k-17}}{\binom{100}{k}} = \frac{\binom{83}{k-17}}{\binom{100}{k}}$$

Nous trouvons donc ainsi que la probabilité d’avoir besoin exactement de k coups pour obtenir les 17 cases de bateaux est :

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

Nous pouvons remarquer qu’il est impossible d’avoir 17 cases de bateaux en jouant moins de 17 coups (P(X ≤ 16) = 0) et on ne peut jouer plus de 100 coups (P(X > 100) = 0) .

LU3IN005
Rida Taleb , Achraf Jday



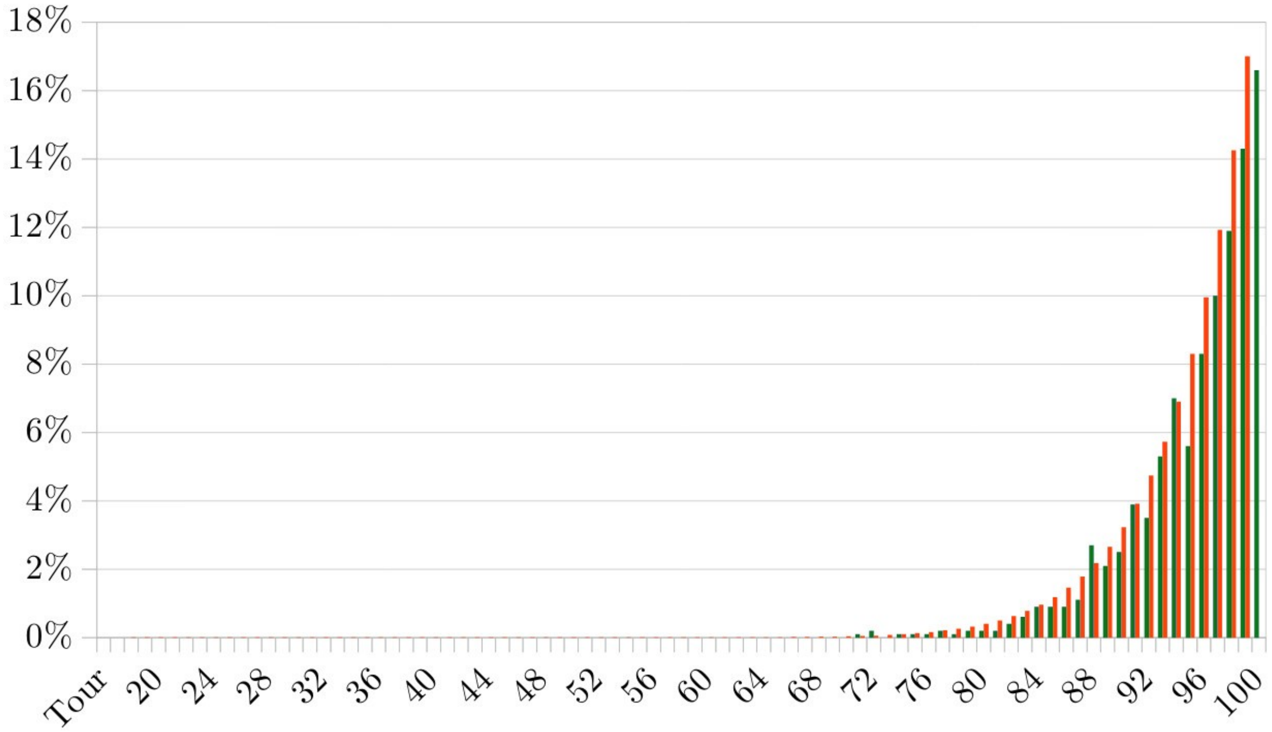
Rouge : Probabilité de gagner en au plus k coups
Bleu : Probabilité de gagner en exactement k coups

Nous pouvons ainsi calculer une espérance égal à 95,39 . Si nous jouons au hasard nous pouvons espérer gagner une partie en 95 tours en moyenne .

Pour calculer l'espérance de la variable aléatoire représentant le nombre coups pour terminer une partie nous allons lancer 1000 parties de batailles navales après cela nous divisons la somme des nombres de coups joués par 1000 . Nous remarquons que nous obtenons un résultat de 95,30 ce qui est très proche du résultat théorique qui est égal à 95,39 .

Dans le cas de la distribution nous allons lancer 1000 parties et mettre le nombres de victoires en fonction du nombres de coups dans un tableau .

Nous obtenons le diagramme suivant :



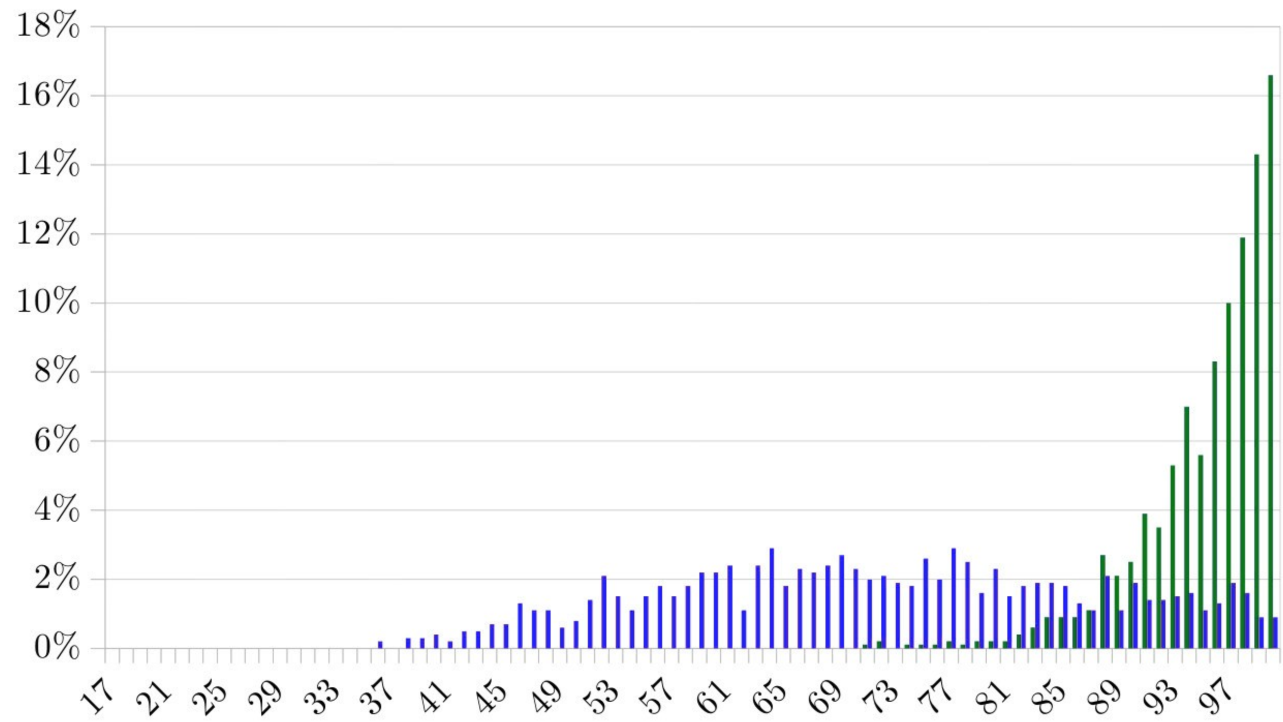
Orange : Résultats théoriques
Vert : Résultats expérimentaux

La distribution trouvé avec les résultats expérimentaux est très proches de celle obtenu théoriquement .

2.
Nous allons implémenter une version Heuristique qui consistera à jouer aléatoirement tant qu'une case de bateau n'est pas touchée mais dans le cas où une est touchée elle explorera les cases adjacentes .

Nous allons suivre le même chemin que pour la version précédente et comparer les résultats avec ceux qui seront obtenus avec la version heuristique .

Nous obtenons le diagramme suivant :



Bleu : Résultats expérimentaux Ver. Aléatoire
Vert : Résultats expérimentaux Ver. Heuristique

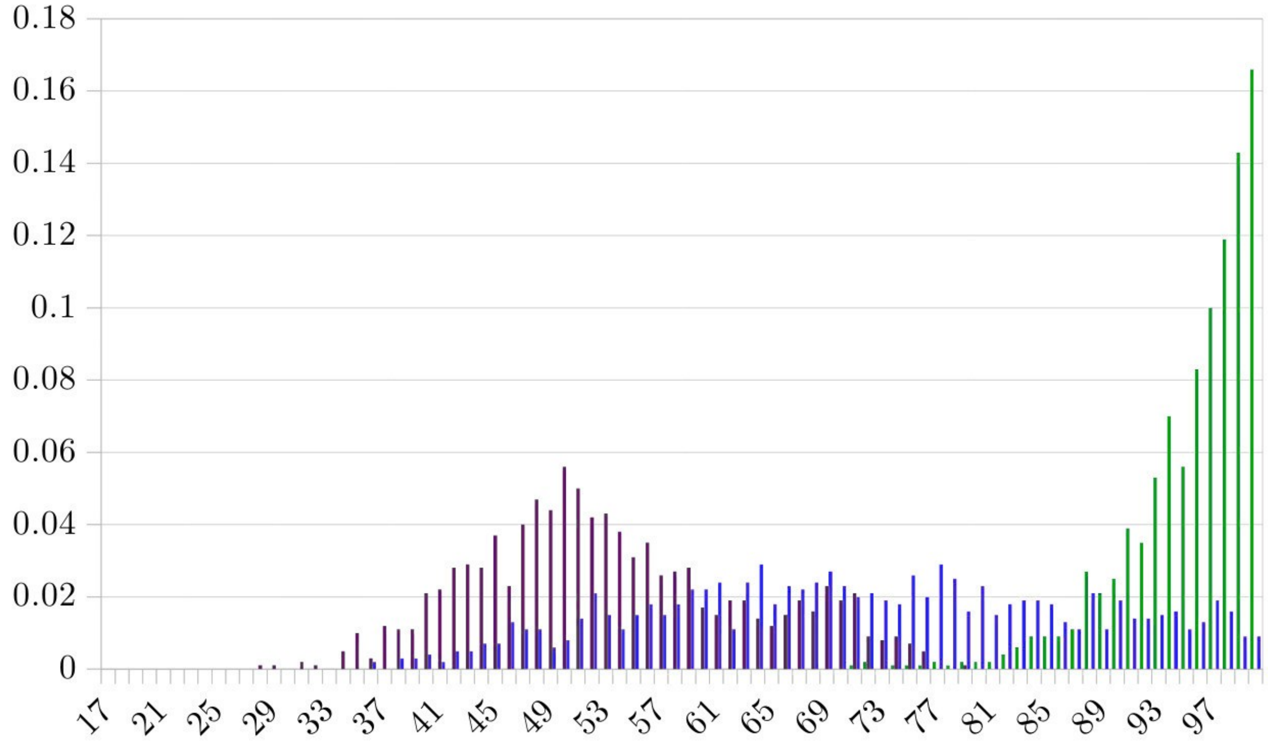
Nous obtenons une espérance de 71,60 pour la Ver. Heuristique contre 95,30 pour la Ver. Aléatoire ce qui montre que la nouvelle stratégie déployé dans la version heuristique est plus efficace .

3.
Dans cette nouvelle version nous allons améliore la stratégie employé dans la version heuristique car elle ne prend pas en compte le faite que selon la taille du bateaux (obtenus en prenant en compte les bateaux restants) et la localisation du premier contact certaines positions ne sont pas possible et il ne sert à rien d'essayer de les toucher .

Nous remploierons la même méthode d'obtention de résultats que pour les versions précédentes .

Rida Taleb , Achraf Jday

Le diagramme suivant compare les résultats obtenus avec les 3 Ver. mis en place :



Bleu : Résultats expérimentaux Ver. Aléatoire
Vert : Résultats expérimentaux Ver. Heuristique
Violet : Résultats expérimentaux Ver. Probabiliste

L'espérance des résultats obtenus avec la Ver. Probabiliste est de 53,30 ce qui est bien mieux que l'espérance obtenu avec la Ver. Aléatoire ou Heuristique qui sont respectivement de 95,30 et 71,60 .

IV . Senseur imparfait: à la recherche de l'USS Scorpion

1. La loi de la variable aléatoire Yi est celle de Bernoulli de paramètre π et pour Zi|Yi nous avons une loi Bernoulli de paramètre Ps .

2. P(Zk = 0 et Yk = 1) = P(Yk = 1) x P(Zk = 0 | Yk = 1)

3. P(Zk = 0 et Yk = 1) = $\pi^1 \times (1-\pi)^0 \times Ps^0 \times (1 - Ps)^1$

$= \pi \times (1 - Ps)$

D’après la formule de Bayes :

$P(Z_k = 0 \text{ et } Y_k = 1) = P(Y_k = 1) \times P(Z_k = 0 \mid Y_k = 1)$
$= P(Z_k = 0) \times P(Y_k = 1 \mid Z_k = 0)$

Alors :

$P(Y_k = 1 \mid Z_k = 0) = P(Z_k = 0 \text{ et } Y_k = 1) / P(Z_k = 0)$

Or :

$P(Z_k = 0) = 1 - P(Z_k = 1) = 1 - \pi \times P_s$

Donc :

$P(Y_k = 1 \mid Z_k = 0) = (\pi (1 - P_s)) / (1 - \pi \times P_s)$

Conclusion :

Nous avons pu remarqué tout au long de ce projet que le jeu de la bataille navale possède une dimension combinatoire importante . Par le biais des nombreuses fonctions nous avons mis en avant le fait que jouer au hasard est une méthode assez médiocre face à une personne qui jouerai de manière un peu plus probabiliste . De l’aléatoire à Monte-Carlo nous avons assister à l’évolution d’un raisonnement de plus en plus poussée mettant en lumière l’efficacité de certaines méthodes de jeu pour la bataille navale .