Rapport Projet 1

Bataille navale

2 – Combinatoire du jeu

Question 1

Une manière simple de déterminer la borne supérieure du nombre de configuration possibles pour la liste complète de bateaux est de faire le produit du nombre de configurations de chaque bateau. On a respectivement 120, 140, 160 et 180 configurations possible pour les bateaux de taille 5, 4, 3 et 2. Ce qui fait un total de 120*140*160*160*190 = 77 414 400 000 configurations pour la liste complète de bateaux.

Question 2

Un bateau de taille n a 10-n+1 configurations sur une ligne de 10 cases. Donc pour une grille de 10x10, il y a 10*(10-n+1) configurations. Et comme il y a les positions verticales et horizontales, on a donc 2*10*(10-n+1) configurations possibles au total.

Taille d'un bateau	2	3	4	5
Nb de config en théorie	180	160	140	120
Valeurs obtenues avec la fonction	180	160	140	120

On peut donc voir que les résultats obtenues avec la fonction *calcul_nombre_facon* correspond à notre théorie.

Question 3

On utilise la fonction *calcul_nombre_facon_liste2* pour une liste de 2 bateaux :

Liste de bateaux (taille)	[5, 4]	[5, 3]	[5, 2]	[4, 3]	[4, 2]	[3, 3]	[3, 2]
Nombre de façon de placer	20760	25488	30724	31968	38520	39160	47148

On utilise la fonction *calcul_nombre_facon_liste3* pour une liste de 3 bateaux :

Liste de bateaux (taille)	[5, 4, 3]	[5, 4, 2]	[5, 3, 3]	[5, 3, 2]	[4, 3, 3]	[4, 3, 2]	[3, 3, 2]
Nombre de façon de placer	2 558 256	3 133 568	3 183 216	3 894 596	4 042 616	4 943 064	6 127 740

Il est possible de calculer de cette manière le nombre de grilles pour la liste complète de bateaux, mais ça serait trop long.

Question 4

P, la probabilité de tirer une grille donnée. Et N, le nombre total de grilles.

En considérant toutes les grilles équiprobables, on a : P = $\frac{1}{N}$

On utilise la fonction *calcul_egalite_grille*.
Elle a été exécutée 10 fois par liste de bateaux afin d'obtenir une moyenne.

Liste de bateaux (taille)	[5]	[5, 4]	[5, 4, 3]
Moyenne du nombre d'essais	123	13051	2374301
Temps moyen (s)	0,003	0,48	128

Au delà d'une liste de 3 bateaux, cela commence à prendre beaucoup de temps pour retrouver la grille générée.

Question 5

Une manière d'approximer le nombre total de grilles pour une liste de bateaux est de faire le produit du nombre total de chaque bateau en prenant en compte les cas précédents ainsi que les orientations horizontales et verticales.

On a un grille de taille l*L (ici, l=10 et L=10). On se place d'abord sur la ligne d'indice le plus petit. Si on peut placer le bateau, alors on retourne l*L. Sinon on retourne (l - 1)*L. Puis on incrémente l'indice de la ligne. On fait de même avec les colonnes. Cependant cette méthode ne prend pas en compte les cases vides des lignes/colonnes précédentes.

On utilise la fonction *approximation_nombre_total_grille_ameliorer()*. Ce qui nous donne un résultat de 43 882 020 000 configurations.

La fonction d'approximation a été déterminée avec la réflexion suivante :

On part du porte-avions (5 cases). Par ligne, il y a 6 possibilités. Donc pour 10 lignes, ça en fait 60. De même pour les colonnes. Il y a donc 120 possibilités pour de placer le porte-avion.

Ensuite, on place le croiseur (4 cases). Sur la même ligne ou colonne que le porte-avion, il reste 5 cases. Il y a donc 2 possibilités de placer le croiseur sur cette ligne. Et sur une ligne de 10 cases, il y a 7 possibilités. Soit 7*9=63 possibilités sur les 9 lignes de 10 cases qui restent. De même pour les colonnes, donc (63+2)*2=130 possibilités.

Puis, on place le contre-torpilleurs (3 cases). Il n'y a plus de place pour le placer sur une ligne qui contient déjà un porte-avions et un croiseur, donc il reste 9 lignes de disponible. Sur 10 cases, il y a 8 possibilités, soit 8*9=72 possibilités sur les 9 lignes. Cependant, si on se place en colonne, il y a 1 colonne de 10 cases et 9 de 9 cases. Pour une colonne de 9 cases, il y a 7 possibilités, soit 7*9=63 pour les 9 lignes. Et 8 pour une colonne de 10 cases. Donc 63+8=71 possibilités, soit 72+71=143 possibilités en tout.

Après, on place le sous-marin (3 cases). La ligne avec le porte-avion et le croiseur n'est toujours pas disponible pour placer un sous-marin. Et pour la ligne avec le contre-torpilleur, il reste 7 cases de disponible. De plus, il reste 8 lignes de 10 cases. Sur 7 cases, il y a 5 possibilités de placer le sous-marin et sur 10 cases, il y a 8 possibilités, soit 8*8+5=69 possibilités sur la formation en ligne. Pour les colonnes, on a 9 colonnes de 9 cases et 1 colonne de 7 cases (occupé par le contre-torpilleur). Sur une colonne de 9 cases, il y a 7 possibilités, et toujours 5 possibilités pour une colonne de 7 cases. Soit 7*9+5=68 possibilités sur la formation en colonne. Donc 69+68=137 possibilités en tout. *[OU Le sous-marin a autant de nombres de cases que le contre-torpilleurs. Il a donc aussi 143 possibilités. Cependant le contre-torpilleurs bloque 6 possibilités. Donc cela fait 143-6=137 possibilités en tout.]*

Et enfin, on place le torpilleur (2 cases). Sur une formation en ligne, la ligne avec le porte-avion et le croiseur n'est toujours pas disponible. Sur la ligne avec le contre-torpilleur et le sous-marin, il reste 4 cases. De plus, il reste 8 lignes de 10 cases. Pour la ligne de 4 cases, il y a 3 possibilités de placer le torpilleur. Sur une ligne de 10 cases, il y a 9 possibilités, soit 9*8=72 possibilités pour les 8 lignes. Donc 72+3=75 possibilités pour une formation en ligne. Et pour une formation en colonne, il y a 9 colonnes de 9 cases et 1 de 4 cases. Sur une colonne de 4 cases, il y a 3 possibilités. Sur une colonne de 9 cases, il y a 8 possibilités, soit 8*9=72 possibilités pour les 9 colonnes de 9 cases. Donc 72+3=75 possibilités pour une formation en colonne. Donc 75+75=150 possibilités en tout.

Donc le nombre de configurations possibles pour la liste complète de bateaux sur une grille de taille 10 est de 120*130*143*137*150=45 842 940 000 configurations.

Question 6

Pour approximer plus justement le nombre de configurations, on peut faire :

On se place sur la ligne d'indice le plus petit. Si o peut placer le bateau, alors on retourne l*L. Sinon on retourne (l - 1)*L + (L – l0) avec l0 le nombre de cases vides. De ce fait, contrairement à l'approximation faite à la question précédente, on prend en compte les cases vides.

3 - Modélisation probabiliste du jeu

Chaque version a été testée sur la même grille qui a été reset (avec la fonction *reset*) entre chaque exécution. Cela permet de pouvoir comparer l'efficacité des différentes version.

Version aléatoire

Soit Y, le contenu d'une case. Soit p, la probabilité qu'une case contient un bateau. Y est un succès si la case contient un bateau. On a alors P(Y = 1) = p. Y est donc une variable aléatoire qui suit une loi de Bernoulli de paramètre p (et q = 1 – p).

On a un bateau de taille 5, un de taille 4, deux taille 3 et un de taille 2. Ce qui nous fait au total 17 cases occupées par un bateau. Et la grille fait 10x10, soit 100 cases au total. La probabilité p est donc de $\frac{17}{100}$.

Soit N, le nombres de cases total de la grille avec A, le nombre de cases qui contient un bateau, et B, le nombre de cases qui ne contient pas de bateau. On a N = A + B.

Soit X, le nombre de cases qui contient un bateau sur n tirages indépendants, non ordonnés et sans remise dans une grille de N cases. X est alors une variable aléatoire discrète qui suit une loi hypergéométrique de paramètres (n, p, N) dont l'univers X(Ω) est l'ensemble des entiers de 0 à n. On peut alors définir la probabilité de X pour un k appartenant à X(Ω) :

$$P(X = k) = \frac{\binom{A}{k}\binom{B}{n-k}}{\binom{N}{n}} = \frac{\binom{A}{k}\binom{N-A}{n-k}}{\binom{N}{n}}$$

On a alors E(X) = np

p vaut $\frac{17}{100}$ seulement pour le 1er tirage, soit pour n=1.

On a donc E(X) = 0,17.

Soit m, le nombre moyen de tires. On peut voir qu'expérimentalement m converge vers 95,4 (cf. *Figure1*).

Pour trouver l'espérance expérimentale E'(X), on fait E'(x) = $\frac{A}{m}$ = $\frac{17}{95,4}$ = 0,178

On a alors $E'(X) \simeq E(X)$.

On peut alors dire que les hypothèses formulées (variable aléatoire discrète et tirages indépendants non ordonnés et sans remise) sont probablement correctes.

L'hypothèse « variable aléatoire discrète » se justifie par le fait que l'univers est fini. Celle des « tirages indépendants » se justifie par le fait que la case suivante est choisie sans prendre en compte la case touchée précédente si elle contient un bateau ou non. Celle des « tirages non ordonnés » se justifie par le fait qu'il n'est pas nécessaire de les ordonner. Et celles des « tirage sans remise » se justifie par le fait qu'une case touchée reste touchée jusqu'à la fin du jeu.

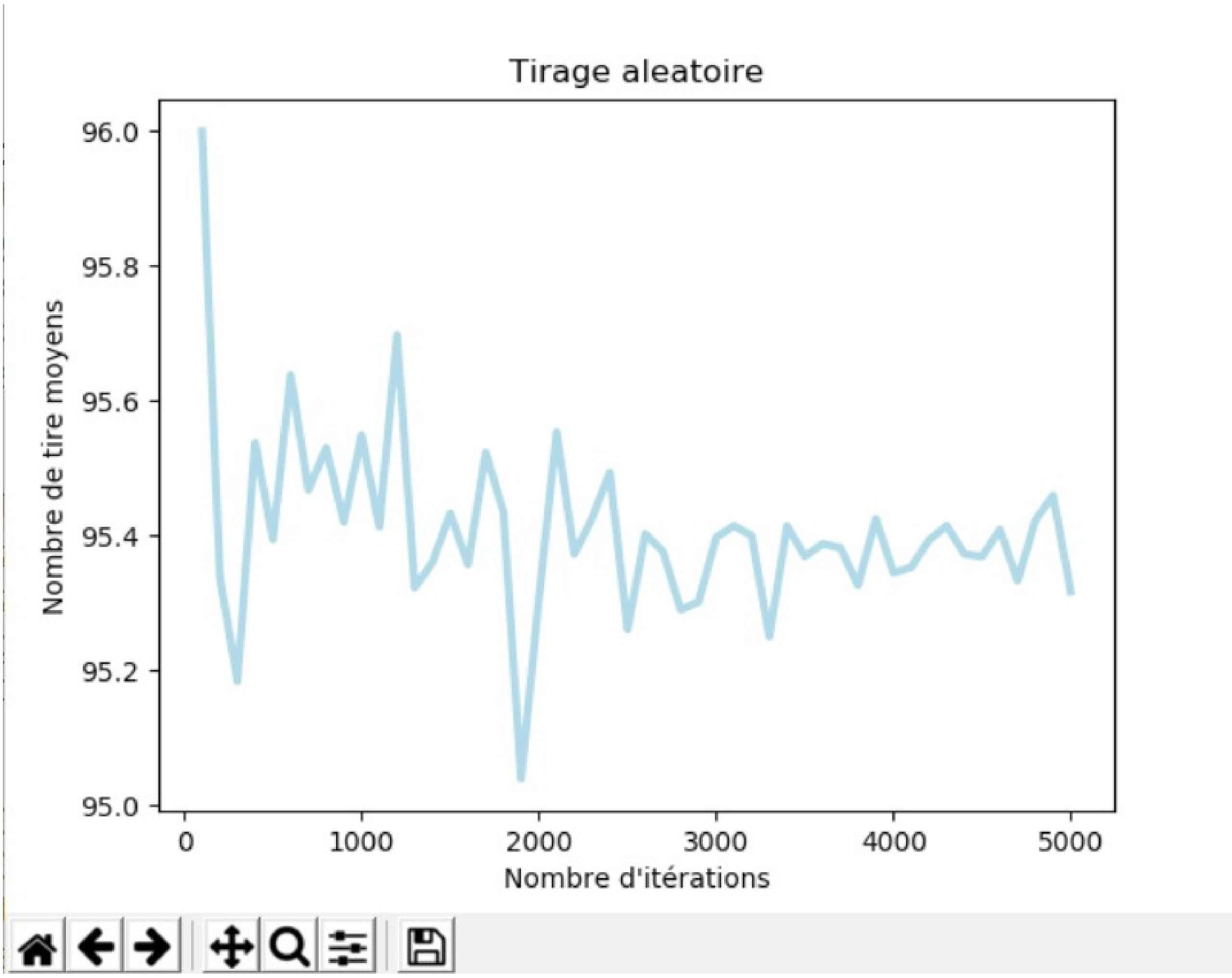


Figure1 : Graphique représentant le nombre moyen de tires effectués pour terminer le jeu en fonction du nombre d'itérations (soit le nombre de fois où on a répété l'expérience) pour la version aléatoire avec une grille de 10x10 et 5 bateaux de taille 5, 4, 3, 3 et 2.

Version heuristique

Précédemment, on a vu qu'il fallait en moyenne 95,4 coups pour finir le jeu avec la version aléatoire. On peut voir qu'avec la version heuristique, il faut en moyenne 61,5 coups pour finir (cf. *Figure2*). On peut alors en déduire que le version heuristique est plus efficace que la version aléatoire.

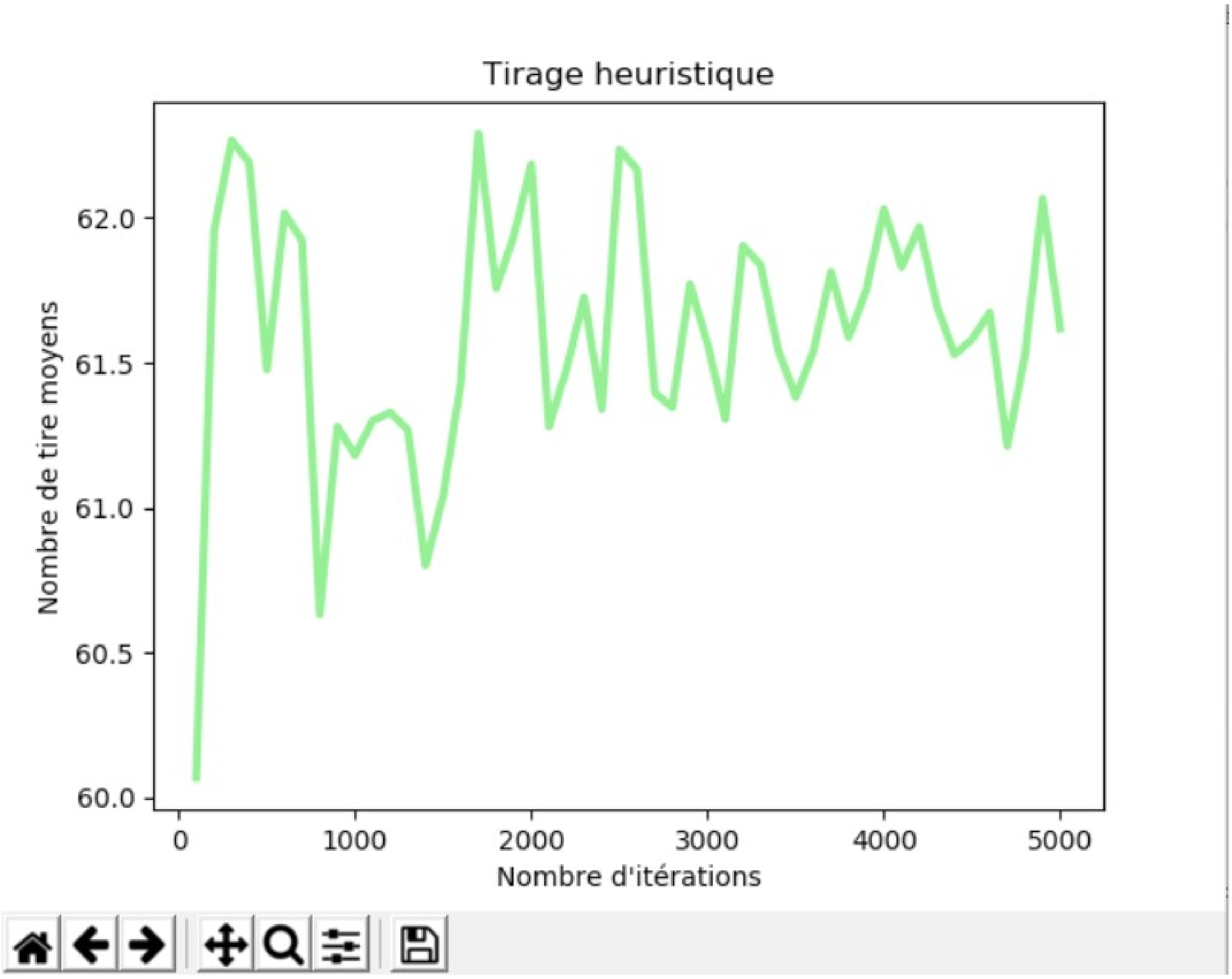


Figure2 : Graphique représentant le nombre moyen de tires effectués pour terminer le jeu en fonction du nombre d'itérations (soit le nombre de fois où on a répété l'expérience) pour la version heuristique avec une grille de 10x10 et 5 bateaux de taille 5, 4, 3, 3 et 2.

Version probabiliste simplifiée

Il est hors de question de calculer à chaque tout la nouvelle distribution jointe sur tous les bateaux, car ça serait trop long et fastidieux.

L'hypothèse d'indépendance est fausse ici, car dorénavant le choix de la case suivante se fait on fonction de la case précédente.

On peut voir qu'expérimentalement il faut en moyenne 46,75 coups pour finir le jeu (cf. *Figure3*). Ce résultat est plus faible que la version aléatoire et heuristique. On peut alors conclure que la version probabiliste simplifiée est plus efficace que les versions heuristique et aléatoire.

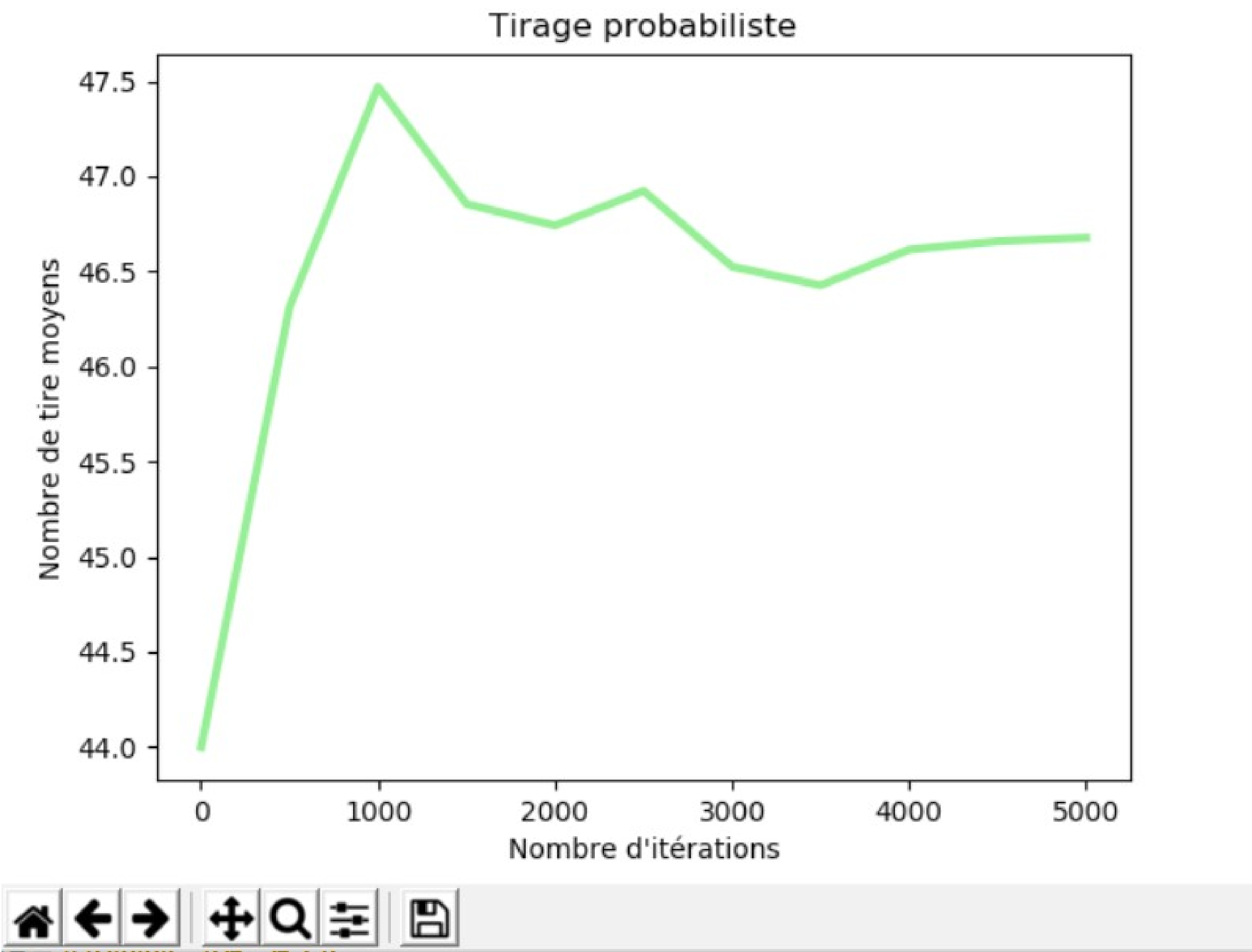


Figure3 : Graphique représentant le nombre moyen de tires effectués pour terminer le jeu en fonction du nombre d'itérations (soit le nombre de fois où on a répété l'expérience) pour la version probabiliste simplifiée avec une grille de 10x10 et 5 bateaux de taille 5, 4, 3, 3 et 2.

Version Monte-Carlo

Pour la version Monte-Carlo, il n'y a pas autant d'itérations que les autres versions parce que cela prend trop de temps. Cependant, on peut tout de même comparé avec les autres versions.

Avec les observations expérimentales (cf. *Figure4*), on peut qu'il faut en moyenne 43 coups pour finir le jeu. Ce qui est beaucoup moins que les versions aléatoire et heuristique, mais seulement un peut moins que la version probabiliste simplifiée. On peut donc conclure que la version Monte-Carlo est bien plus efficace que les version aléatoire et heuristique, mais pas beaucoup plus que la version probabiliste simplifiée.

Figure4 : Graphique représentant le nombre moyen de tires effectués pour terminer le jeu en fonction du nombre d'itérations (soit le nombre de fois où on a répété l'expérience) pour la version Monte-Carlo avec une grille de 10x10 et 5 bateaux de taille 5, 4, 3, 3 et 2.

4 – Senseur imparfait : à la recherche de l'USS Scorpion

Théorie

Soit la variable aléatoire Y_i suit une loi de Bernoulli de paramètre π_i. On a :

$P(Y_i = y) = (\pi_i)^y (1 - \pi_i)^{1-y}$ (avec i = 1, …, n)

Et $Z_i \mid Y_i$ une autre variable aléatoire qui suit une loi Bernoulli de paramètre p_s. On a :

$P(Z_i \mid Y_i = z) = (p_s)^z (1 - p_s)^{1-z}$ (avec i = 1, …, n)

Supposons que $Y_k = 1$ et $Z_k = 0$, on cherche $P(Z_k = 0, Y_k = 1)$.

D'après la formule de Bayes, on a :

$P(Z_k = 0, Y_k = 1) = P(Z_k = 0 \mid Y_k = 1)\, P(Y_k = 1)$

Or, d'après l'énoncé, la probabilité de l'objet se trouve sur la case et que le senseur le détecte est de p_s, soit $P(Z = 1 \mid Y = 1) = p_s$. Donc la probabilité que l'objet se trouve sur la case et que le senseur ne le détecte pas est de $1 - p_s$, soit $P(Z = 0 \mid Y = 1) = 1 - p_s$. Et on a $P(Y_k = 1) = \pi_i$, d'après la loi de Bernoulli.

On a donc : $P(Z_k = 0, Y_k = 1) = \pi_i (1 - p_s)$

Ensuite, on cherche $P(Y_k = 1 \mid Z_k = 0)$. D'après la formule de Bayes, on a :

$P(Y_k = 1 \mid Z_k = 0) = \dfrac{P(Z_k=0, Y_k=1)}{P(Z_k=0)}$

Il nous manque $P(Z_k = 0)$.

D'après le théorème des probabilités totales, on a :

$P(Z_k = 0) = P(Z_k = 0 \mid Y_k = 1) * P(Y_k = 1) + P(Z_k = 0 \mid Y_k = 0) * P(Y_k = 0)$

$P(Z_k = 0) = (1 - p_s) * \pi_k + 1 * (1 - \pi_k)$

$P(Z_k = 0) = 1 - \pi_k p_s$

On a donc : $P(Y_k = 1 \mid Z_k = 0) = \dfrac{\pi_k (1 - p_s)}{1 - \pi_k p_s}$

On cherche à mettre à jour π_i .

On considère la probabilité p_s comme constante parce qu'on suppose que les appareils de détection restent les mêmes tout au long de notre recherche. Et la probabilité π_i va changer au fur et à mesure de nos recherches.

D’après les questions précédentes, on a : $\pi_{k,nouveau} = \frac{\pi_{k,ancien}(1-p_s)}{1-p_s\pi_{k,ancien}}$

Si l’objet n’est pas détecté à la cellule k, alors on pourrait croire qu’il se trouve dans une autre cellule. La probabilité que l’objet se trouve dans une cellule i (avec i ≠ k) devrait donc augmenter. On veut donc trouver la probabilité que la cellule i contienne l’objet, mais que le senseur ne l’ait pas détecté, soit $P(Y_i = 1 \mid Z_k = 0)$.

D’après la formule de Bayes, on a : $P(Y_i = 1 \mid Z_k = 0) = \frac{P(Z_k=0 \mid Y_i=1)P(Y_i=1)}{P(Z_k=0)}$

Or, $P(Z_k = 0 \mid Y_i = 1)$ représente la probabilité de ne pas détecter l’objet à la cellule k étant donné que l’objet se trouve à la cellule i. Elle est donc forcément de 1.

On a donc : $P(Y_i = 1 \mid Z_k = 0) = \frac{\pi_i}{1-\pi_k p_s}$

On obtient donc : $\pi_{i,nouveau} = \frac{\pi_{i,ancien}}{1-p_s\pi_{k,ancien}}$

Expérimentation

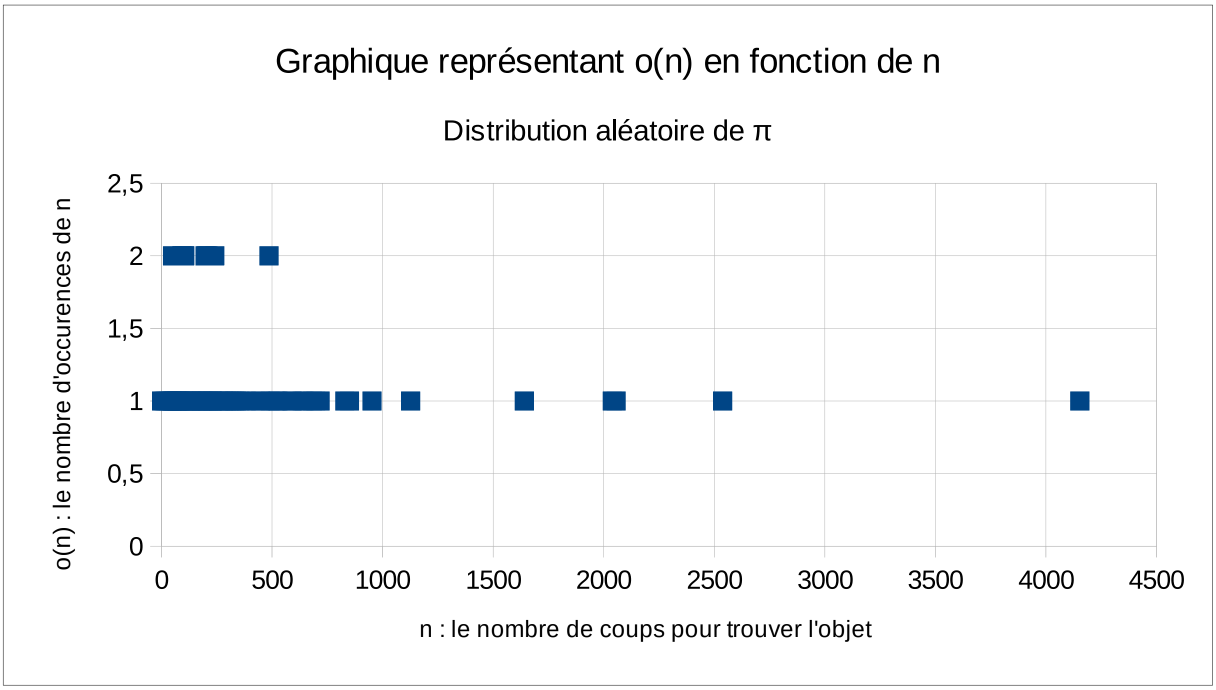
On étudie plusieurs distribution de π :

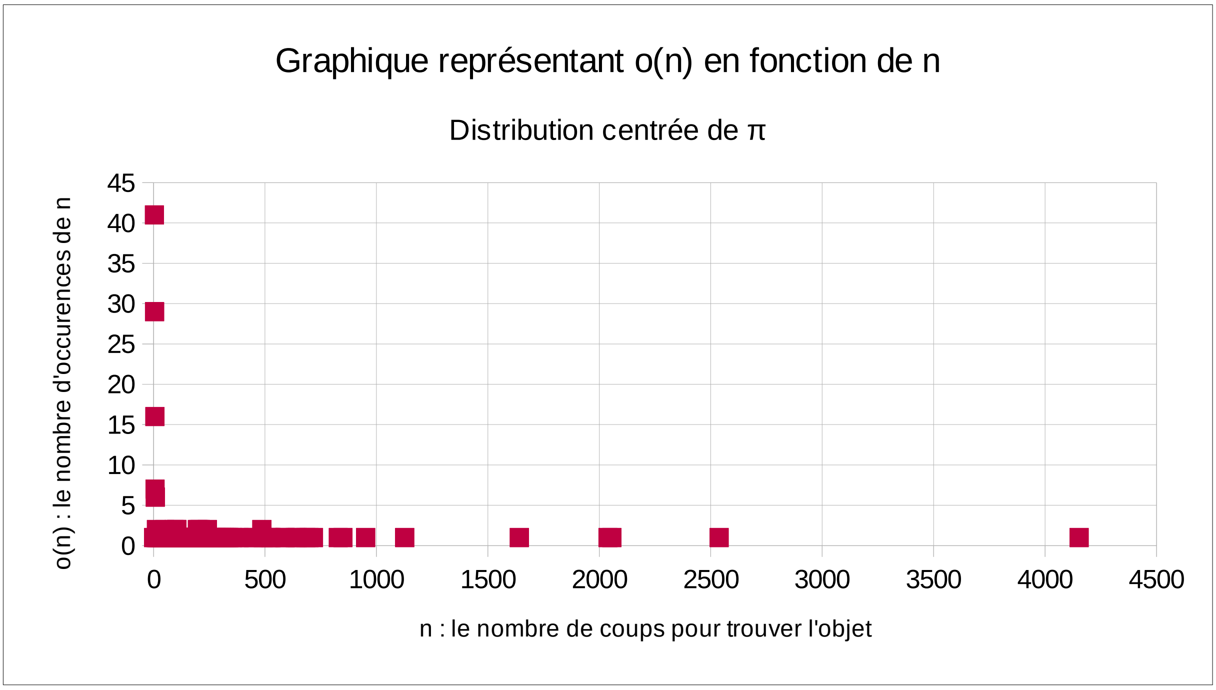
- distribution aléatoirement uniforme
- distribution centrée

Soit n, le nombre de coups effectués pour trouver l’objet. Et soit o(n), le nombre d’occurrence de n.

On regarde combien de fois on a obtenu n sur 100 exécution de la fonction *senseur_imparfait.*

On peut voir sur les 2 différents graphiques suivants qu’on trouve l’objet en moins de coups pour une distribution centrée qu’une distribution aléatoire. On peut donc conclure que c’est plus rapide de trouver un objet s’il est centrée qu’aléatoire.


Graphique représentant o(n) en fonction de n
Distribution aléatoire de π
o(n) : le nombre d'occurences de n
n : le nombre de coups pour trouver l'objet
0
0,5
1
1,5
2
2,5
0
500
1000
1500
2000
2500
3000
3500
4000
4500


Graphique représentant o(n) en fonction de n
Distribution centrée de π
o(n) : le nombre d'occurences de n
n : le nombre de coups pour trouver l'objet
0
5
10
15
20
25
30
35
40
45
0
500
1000
1500
2000
2500
3000
3500
4000
4500