LU3IN005 - Projet 1 : Bataille Navale

Rapport de Projet 1 par Emma CHAKMA, Clarisse LUONG et Bettina MUBILIGI

Partie 2 - Combinatoire du Jeu

Puisque l'on dispose d'une grille de taille $10 * 10$ cases, soit 100 cases, on doit regarder le nombre de manières de placer 1 bateau (avec un nombre de cases données) sur une ligne horizontale, puis verticale, sur 10 cases chacunes.

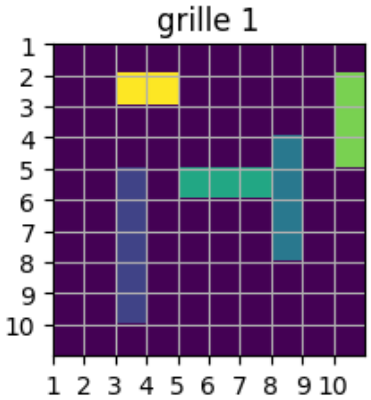
Pour placer le bateau 1, le porte-avions, qui prends 5 cases, on enlève le nombre de cases du bateau au nombre total de la ligne et on ajoute 1, soit $10 - 5 + 1 = 6$, donc il y a 6 manières de placer le bateau dans une ligne, fois 10 lignes, fois deux (lignes horizontales et verticales), soit $6 * 10 * 2 = 120$ manières de placer le porte-avions dans la grille.

En suivant le même raisonnement pour le reste des bateaux, on obtient pour le bateau 2 (croiseurs) à 4 cases : $7 * 10 * 2 = 140$, le bateau 3 (contre-torpilleurs) à 3 cases : $8 * 10 * 2 = 160$, de même pour le bateau 4 (sous-marin) à 3 cases, et enfin il y a $9 * 10 * 2 = 180$ manières de placer le bateau 5 (torpilleur) dans la grille.

La borne supérieure simple du nombre de configurations possibles pour la liste complète de bateaux sur une grille de taille 10 sera égale à $120 * 140 * 160 * 160 * 180$ ce qui est égal à environ 77 milliards de configurations.

La fonction qui permet de calculer le nombre de façons de placer un bateau donné sur une grille vide est nb_placements_possibles_bateau(id) avec id le numéro du bateau, et nous renvoie les même valeurs que celles trouvées lors du calcul théorique précédent, soit 120 pour le bateau 1, 140 pour le bateau 2, 160 pour les bateaux 3 et 4 et 180 pour le bateau 5.

En implémentant la fonction qui permet de calculer le nombre de façon de placer une liste de bateaux sur une grille vide, on observe que le nombre de placements possible varie selon les chevauchements : l'absence de chevauchements implique un nombre de placements inférieur à au cas où ils seraient permis. Ce nombre dépend également de la taille des bateaux de la liste.



Un exemple de grille

La probabilité de tirer une grille donnée $p = 1/n$, n le nombre de grilles, si on considère toutes les grilles comme équiprobables.

Remarque : notre fonction grilles_egales(grille) qui génère des grilles aléatoirement jusqu'à ce que la grille générée soit égale à la grille passée en paramètre a un temps d'exécution très élevé si on prend des grilles contenant plus de 3 bateaux. Ceci peut être expliqué par le fait que plus la grille contient de bateaux, plus il y a de combinaisons possibles à explorer, et plus on mettra de temps à trouver une grille identique à celle passée en paramètre de la fonction.

Le nombre de grilles total pour une liste de bateaux est n, $n = 1/p$, p la probabilité de tirer une grille donnée pour cette liste de bateaux, approximativement.

Pour trouver cette probabilité p, il faut calculer le nombre Nv de grilles valides générées avec la liste de bateaux sur le nombre d'essais total Nt de génération de grilles, soit $p = Nv/Nt$.

Donc le nombre de grilles total $n = 1/p = Nt/Nv$, soit le nombre d' essais total Nt sur le nombre Nv de grilles valides générées.

Ici, les grilles 'valides' sont les grilles sans débordements et chevauchements des bateaux, soit les grilles pour lesquelles la foncion peut_placer(grille, bateau, position, direction) était à 'True' pour tous les bateaux de la liste.
Nous avons implémenté une fonction nb_total_grilles(liste_bateau) qui calcule ce nombre total de grilles, à l'aide de nb_placements_possibles_liste_chev(liste_bateaux) / nb_placements_possibles_liste_sans_chev(liste_bateaux), qui calculent le nombre de façon de placer une liste de bateaux sur une grille vide, avec et sans chevauchements respectivement, cependant le résultat obtenu ne semble pas cohérent.

Partie 3 - Modélisation probabiliste du jeu

Version aléatoire

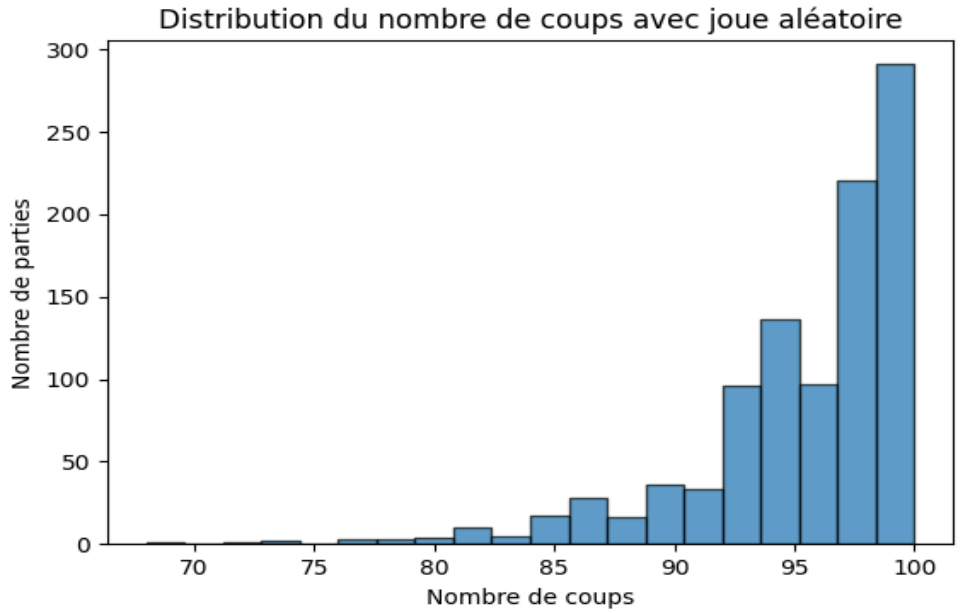
Nos hypothèses sont les suivantes pour le calcul l'espérance du nombre de coups joués avant de couler tous les bateaux si chaque coup est tiré aléatoirement :
- Il y a un seul bateau de chaque taille dans la grille
- La grille est de taille 10*10 soit 100 cases
- Il n'y a pas de chevauchements possibles entre les cases des bateaux

On pose une variable aléatoire X comptant le nombre d'épreuves de Bernoulli indépendantes entre elles (le nombre de coups joués) de probabilité de succès $p \in]0, 1[$ chacunes, X permet donc de compter le nombre de coups nécessaires pour couler tous les bateaux de la grille, soit le nombre de coups pour toucher chaque case de la grille contenant un bateau. X est donc une somme de Xi, avec i le nombre de coups joués.

Ces Xi, suivent une loi géométrique de paramètre Pi, Pi étant la probabilité de toucher une case contenant un bateau, avec donc une espérance E(Xi) $= 1/Pi$.

Il y a au total $5 + 4 + 3 + 3 + 2 = 17$ cases occupées sur 100 qui contiennent un bateau. Au premier coup, on aura 17 cases réellement occupées par un bateau sur 100 cases possibles, soit la probabilité $P1$ de toucher une case contenant un bateau pour $X1$ est $P(X1) = 17/100$, puis au second coup, il nous restera 16 cases réellement occupées sur 99 cases possibles, soit $P(X2) = 16/99$, puis $P(X3) = 15/98$, pour le troisième coup, etc, jusqu'à ce qu'il nous reste 1 case contenant un bateau sur 84 possibles, soit $P(X17) = 84/1$.

Ce qui nous donne une espérance E($X1$) $= 1/P(X1) = 100/17 \approx 5.88$ pour le premier coup, E($X2$) $= 1/P(X2) = 99/16 \approx 6.19$ pour le deuxième, etc.
Donc l'espérance de X est $\mathrm{E}(X) = \Sigma\,(E\,(Xi))$ pour i allant de 1 à 17, soit $E(X) = 100/17 + 99/16 + 98/15 + 97/14 + 96/13 + 95/12 + 94/11 + 93/10 + 92/9 + 91/8 + 90/7 + 89/6 + 88/5 + 87/4 + 86/3 + 85/2 + 84/1 \simeq 302$.

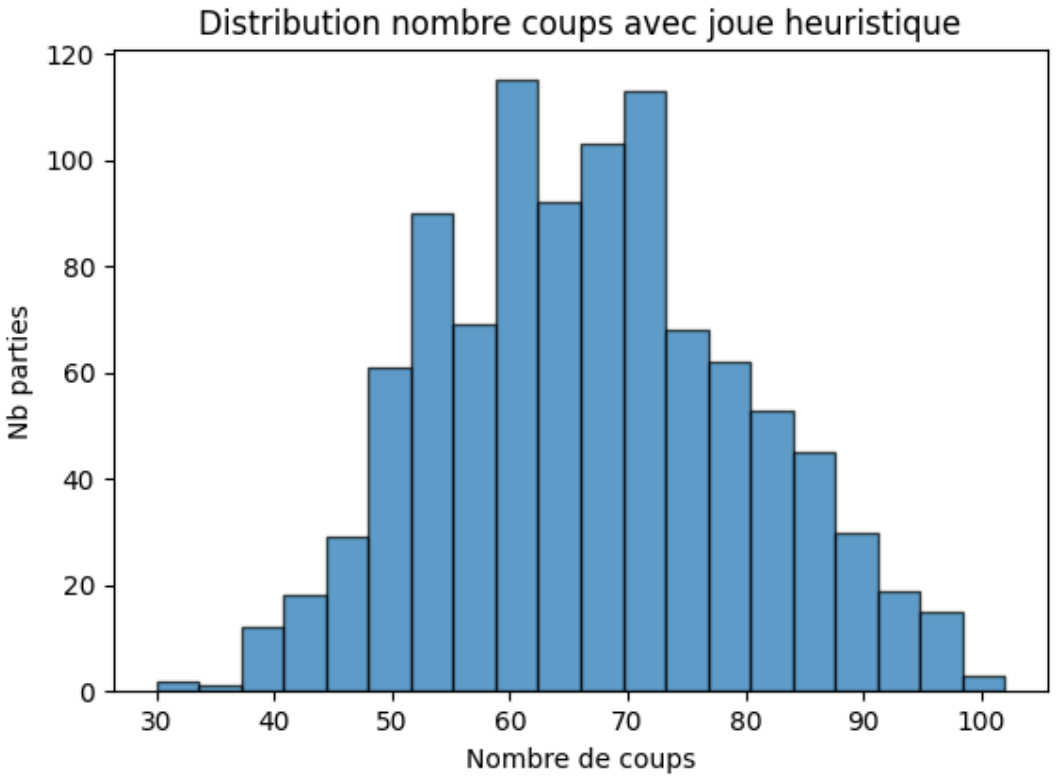


.

En observant le graphique de distribution du nombre de coups pour la version aléatoire, on voit que la distribution de la variable aléatoire X (nombre de coups) tourne en moyenne autour de 100 coups, donc $E(X) \simeq 100$, ce qui ne correspond pas du tout à la valeur de $E(X)$ théorique calculée précédemment, soit $E(X) \simeq 302$. Cela peut s'expliquer que la version théorique ne prends pas en compte le cas où une case jouée l'est définitivement (on ne pourrait donc pas toucher deux fois la même case), mais la version implémentée dans la fonction joue_aléatoire prends ce cas en compte, et stocke les cases déjà explorées dans un ensemble. L'espérance triple donc avec la version théorique, car on explore plusieurs fois la même case.

Version heuristique

La version heuristique mettait en œuvre deux comportements : un comportement aléatoire tant que rien n'est touché et un comportement qui va explorer les cases connexes lorsqu'un coup touche un bateau. Ci-dessous se trouve le graphique de la distribution de la variable aléatoire correspondant au nombre de coups joués par partie pour couler tous les bateaux, en fonction du nombre de parties :



On observe ici que le nombre de coups moyen par partie pour couler tous les bateaux se situe en moyenne entre 60 et 75 coups, ce qui est inférieur à la valeur de $E(X)$ trouvée pour la version aléatoire, qui était environ 100. Cela s'explique par le caractère semi-aléatoire de la version heuristique : même si la case tirée est tirée aléatoirement, si la case contient un bateau, le comportement aléatoire devient un parcours de case ordonné, car on sait maintenant qu'une des cases adjacentes à cette case contient elle aussi un bateau, donc la probabilité qu'une de ces cases contienne un bateau augmente. On cherche donc le bateau dans les cases adjacentes en priorité, avant de revenir à une recherche aléatoire, ce qui diminue le nombre de coups joués.

Version probabiliste simplifiée

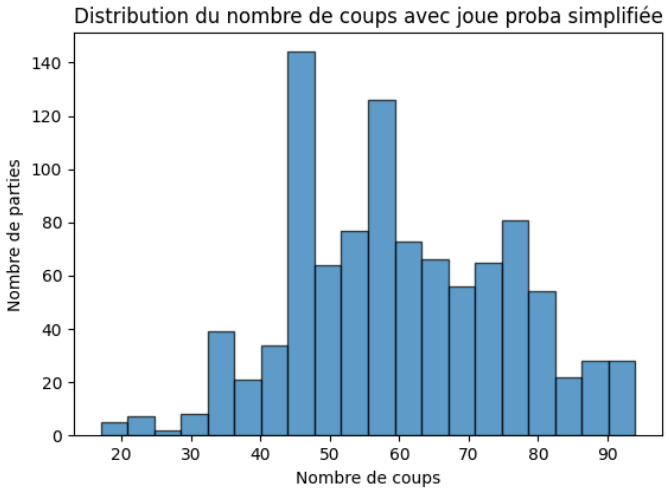
Les étapes de l'algorithme de recherche probabiliste :

1/
Création de la matrice mat_total avec la somme de toutes les probabilités pour chaque bateau et chaque case.

2/
On remplit un ensemble avec les coordonnées à explorer dans l'ordre (coordonnées avec la plus grande probabilité de contenir un bateau), et on modifie systématiquement cet ensemble avec les nouvelles probabilités de contenir un bateau dans les cases à chaque tour (on met à jour mat_total et l'ensemble des coordonnées à explorer).

3/
Parcours de la grille similaire à la version heuristique, en commençant par les cases avec la plus grande probabilité de contenir un bateau tout en modifiant les probabilités dans les cases en fonction du résultat du coup : si la case est vide, on enlève 1 à la probabilité de présence de tous les bateaux dans la case (-5), si on touche un bateau, on ne touche pas aux probabilités dans cette case, mais on parcourt les cases adjacentes pour couler le bateau, et en le coulant, on enlève 1 à la probabilité de présence de tous les autres bateaux (-4), à chaque case contenant le bateau.

L'hypothèse d'indépendance est fausse, car la probabilité de présence des bateaux dans une case varie selon le résultat d'un coup dans une autre case : par exemple, si une case contient le bateau 1, alors la probabilité de présence des autres bateaux est nulle. Les bateaux ne pouvant pas se chevaucher, toutes les probabilités sont liées entre elles.



En observant le graphique de distribution de la variable aléatoire pour la version probabiliste, on voit que l'espérance du nombre de coups par partie pour couler tous les bateaux de la grille, $E(X)$, se situe entre 40 et 60 coups, avec un pic à environ 45 coups. Ce résultat est nettement plus bas que les deux résultats pour les versions aléatoires (≈ 100) coups et heuristique (≈ 70) coups. En effet, la version probabiliste implique un parcours des cases selon la probabilité de présence d'un bateau, et non aléatoirement ou semi aléatoirement comme les versions précédentes. Plus cette probabilité sera élevée, plus la case correspondante sera explorée en priorité. En conséquence, on trouvera plus vite les cases occupées, soit en moins de coups par partie.

Partie 4 - Senseur Imparfait: à la recherche de l'USS Scorpion

La loi de probabilité pour Y=1 est P(Y) = 1/N = πi car il y a une seule case sur les N du quadrillage qui contient l'objet. La loi de probabilité pour Y=0 est 1-(1/N) (ou 1 - πi) car le reste des N cases du quadrillage est vide.

Selon l'énoncé, la loi de probabilité de Zi|Yi est la suivante :
- p(Z=0|Y=0) = 1, le senseur ne détecte rien et l'objet ne s'y trouve pas
- p(Z=1|Y=0) = 0, le senseur détecte un objet et l'objet ne s'y trouve pas
- p(Z=0|Y=1) = 1-ps, le senseur ne détecte pas d' objet et l'objet s'y trouve
- p(Z=1|Y=1) = ps, le senseur détecte un objet et l'objet s'y trouve

Dans le cas ou la détection n'a pas fonctionné (Zk=0) alors que l'objet s'y trouvait (Yk=1), on peut exprimer la probabilité de la manière suivante :
p(Zk=0|Yk=1) =(p(Yk= 1|Zk= 0) * p(Zk =0))/ p(Yk=1)

Si la détection en cas de probabilité de contenir l'objet recherché πk, alors πk = 0 au prochain tour et on augmente la probabilité sur toutes les autres cases car il y a une case en moins parmi les N dans laquelle l'objet aurait pu se trouver. Ce qui donne πi = 1/(N-nb_cases_zéro).