

Bataille Navale

AIT OUAZZOU Zakari
BENATTAR Ethan

March 25, 2020

1 Modélisation et fonctions simples

Voir code.

2 Combinatoire du jeu

2.1 Borne Supérieure

Borne simple La borne la plus simple que l'on puisse calculer est la somme des dispositions possibles pour chaque bateau de la liste s'il était seul, car dans la réalité deux bateaux ne peuvent pas se superposer donc on devrait avoir plus de contraintes, mais pour avoir une estimation grossière on commence avec cette méthode.

Pour calculer le nombre de dispositions d'un bateau pour une grille vide, on remarque au début que le nombre de dispositions horizontales est égal à celui des dispositions verticales, donc il ne faut que calculer une puis la multiplier par 2.

Pour calculer le nombre de dispositions horizontale, on calcule le nombre de dispositions par ligne, puis on multiplie par le nombre de lignes (10 dans ce cas). On pourrait imaginer que le bateau glisse sur la ligne pour calculer le nombre de dispositions, on le dépose le plus à gauche possible puis on le fait glisser vers la droite jusqu'à ce qu'il atteigne le bord, et le nombre de positions et alors le nombre de cases que la case de base du bateau peut occuper, qui est donc égal au nombre de cases dans une ligne, moins le nombre des cases secondaires du bateau.

On obtient donc la relation pour un bateau B:

$$Nb_{dispositions}(B) = (Nb_{colonnes} - (Taille(B) - 1)) * Nb_{lignes} * 2$$

Bateau	Dispositions
Porte-avions	$(10-(5-1))*10*2 = 120$
Croiseur	$(10-(4-1))*10*2 = 140$
Contre-torpilleurs	$(10-(3-1))*10*2 = 160$
Sous-marin	$(10-(3-1))*10*2 = 160$
Torpilleur	$(10-(2-1))*10*2 = 180$
Total (Borne simple)	77,414,400,000

Borne avancée On va déposer les bateaux un par un, en rajoutant des restrictions au fur et à mesure pour calculer une borne supérieure pour le nombre des configurations possibles. On commence par les plus gros bateau jusqu'au plus petit. Pour le premier bateau, on n'a pas de restrictions donc c'est égal au nombre de dispositions pour une grille vide. Pour le reste des bateaux, on doit s'assurer qu'il ne se superposent pas avec des bateaux déjà déposés, donc pour chaque bateau, le nombre de dispositions autorisées est égal au nombre total de dispositions moins les dispositions "interdites" qui dépend de la dispositions des autres bateaux.

Pour l'instant, on ne prends en compte que le nombre de cases déjà occupées par les autres bateau pour estimer une meilleure borne (en réalité, même les cases non occupée peuvent causer problème selon le positionnement des bateaux, par exemple si l'on positionne le porte avion horizontalement au milieu d'une ligne, on ne peut plus y déposer de Croiseur, alors que s'il était collé au bord ce ne serait pas le cas).

On enlève donc 10 dispositions pour le 2ème bateau (5 horizontales, 5 verticales), 18 pour le 3ème (5+4=9 horizontales, 9 verticales), 24 pour le 4ème et 30 pour le dernier.

On a donc comme borne

$$120 * (140-10) * (160-18) * (160-24) * (180-30) = 45,190,080,000 \text{ dispositions}$$

Borne plus avancée

2.2 Façons de placer un bateau

Les résultats de la fonction (ligne:130 dans le code) sont identiques aux résultats théoriques, la logique est donc correcte.

2.3 Façons de placer une liste de bateaux

Nombre de grilles différentes pour:

1 Porte-Avion: 120 dispositions

1 Porte-Avion + 1 Croiseur: 15600 dispositions

1 Porte-Avion + 1 Croiseur + Contre-Torpilleur: 2254176 dispositions

On ne peut pas procéder de cette manière pour la liste complète car pour 3 bateau (1000000 itérations) le temps d'exécution est 6.339157819747925 secondes, donc si on rajoute 2 autres bateau (10000 fois plus d'itérations), on peut estimer que l'on va attendre 17 heures.

2.4 Probabilité de tirer une grille donnée

En considérant toutes les grilles équiprobables, la probabilité de tirer une grille donnée est:

$$\frac{1}{Nb_{grilles}}$$

avec $NB_{grilles}$: Nombre total de grilles possibles.

2.5 Approximation du nombre total de grilles

Algorithme Basique On parcourt la liste de bateau, et on initialise le nombre de grilles avec le nombre de dispositions possibles pour le premier bateau dans une grille vide, puis on parcourt le reste de la liste en multipliant à chaque fois ce nombre par le nombre de dispositions possible du bateau courant moins le nombre de cases déjà occupées.

L'approximation est correcte pour une liste de 3 bateaux, mais très fausse (différence de 50% pour la liste complète) pour des listes plus grandes.

Algorithme DLX (Bonus) Je n'ai hélas pas réussi à améliorer l'approximation sans revenir à une implementation brute-force en testant tout les cas possibles (donc très lent pour liste complète), mais en recherchant des méthodes d'approxiamtion pour puzzle, je suis tombé sur l'algorithme DLX de Knuth qui a déjà été implémenté:

<https://possiblywrong.wordpress.com/2016/03/12/analysis-of-hexagonal-battleship/>

3 Modélisation probabiliste du jeu

3.1 Version aléatoire

3.2 Version probabiliste simplifiée

3.3 Version Monte-Carlo

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

4.1

4.2

4.3

4.4