Léane Texier

TP2 – Programmation dynamique

Question 1 : Méthode

Si au moins un des successeurs est négatif ou nul alors la valeur de la configuration est:

-1*max(val_neg_config)+1 avec val_neg_config tous les successeurs négatifs ou nuls

Sinon (i.e. toutes les valeurs sont strictement positives), la valeur de la configuration est:

-1*max(val_pos_config)-1 avec val_pos_config tous les successeurs (positifs)

Question 2 : Version naïve

Fonction copier_dispo:

Cette fonction permet de copier une disposition donnée. Elle parcourt alors chaque ligne afin de mémoriser les valeurs de chaque dans la nouvelle variable que l'on retourne. Cette fonction nous permet de copier une disposition sans changer l'actuelle. En effet, en python, on ne peut pas simplement copier une variable en faisant new = old. Effectivement, si l'on fait cela et que l'on modifie new, old sera alors modifiée et inversement. Or, on ne veut pas que cela soit le cas, il est donc utile pour créer une copie de le faire soit avec la fonction que j'ai créée soit en utilisant le module python 'copy' et en faisant une copie en profondeur.

Fonction tour_blanc:

Cette fonction permet de connaître toutes les dispositions possibles suivantes quand c'est au joueur blanc de jouer.

Pour faire cela, nous initialisons tout d'abord une variable qui regroupera toutes les dispositions suivantes possibles. Puis, nous parcourons chaque case de l'hexapawn. Pour chacune, nous regardons alors s'il y a un pion blanc dessus (i.e. la case contient la valeur 'P'). Si c'est le cas, nous regardons:

- si le pion peut avancer (i.e. il n'y a personne sur la case devant).
- si le pion peut aller en diagonale droite (i.e. il y a un pion adverse sur la diagonale droite, c'est-à-dire qu'il y a un pion noir ('p') sur la cellule en diagonale droite du pion).
- si le pion peut aller en diagonale gauche (i.e. il y a un pion adverse sur la diagonale gauche, c'est-à-dire qu'il y a un pion noir ('p') sur la cellule en diagonale gauche du pion).

Dans chacun des trois cas précédents, si c'est possible, nous ajoutons la disposition que cela ferait en effectuant le mouvement à la variable regroupant toutes les dispositions suivantes possibles.
Il n'est pas utile de regarder si le pion ne va pas sortir du plateau de jeu en avancant car si cela est son tour, cela signifie qu'il n'a aucun pion sur la dernière ligne.

Fonction tour_noir:

Cette fonction est similaire à 'tour_blanc'. Elle permet de savoir les dispositions possibles suivantes quand c'est au joueur noir de jouer.

Fonction eval_hexapawn:

Cette fonction permet d'évaluer une configuration en fonction de ces successeurs.

Quand c'est le tour du joueur blanc et qu'un pion noir a atteint la dernière rangée, alors on retourne 0 car cela signifie que le joueur blanc a perdu. Si c'est au tour du joueur blanc mais qu'il n'y a pas de pion noir sur la dernière rangée, on regarde alors tous les coups possibles pour le joueur blanc. Quand c'est le tour du noir, nous effectuons les mêmes actions mais en regardant si un pion blanc a

atteint la dernière rangée et en évaluant les coups possibles pour le joueur noir. Une fois cette liste de possibilités récupérées (que ce soit au joueur blanc ou noir de jouer), on regarde si elle est vide. Si c'est le cas, cela signifie que le joueur qui doit jouer ne peut faire aucun mouvement. Il est alors bloqué et a donc perdu. C'est pourquoi on retourne la valeur 0 dans ce cas. Dans le cas où le joueur peut jouer, on va alors pour chaque disposition possible suivante, évaluer la nouvelle configuration grâce à cette même fonction en mettant en paramètre la nouvelle disposition qu'on souhaite évaluer et en indiquant que c'est à l'autre joueur de jouer. Pour chaque disposition, nous aurons une valeur. Nous allons alors regarder si cette valeur est négative ou nulle. Si c'est le cas, on l'enregistrera dans la liste regroupant les valeurs négatives et nulles. Dans le cas contraire, nous enregistrons cette valeur dans celle contenant les valeurs positives. Une fois toutes les dispositions étudiées, nous regardons si nous avons au moins une valeur négative ou nulle, c'est-à-dire que la liste contenant les valeurs négatives et nulles n'est pas vide. Si c'est le cas, nous appliquons la formule adaptée (cf Q.1) et on retourne la valeur. Dans le cas contraire (c'est-à-dire que toutes les valeurs des dispositions suivantes possibles étudiées sont strictement positives), alors on applique l'autre formule (cf Q.1) et on retourne la valeur.

Fonction hexapawn_naif:

Cette fonction permet de connaître la valeur de la configuration donnée par l'utilisateur. Une vérification est faite afin de s'assurer que le joueur blanc n'est pas dans une position gagnante car si c'est le cas, cela signifie qu'il ne doit pas jouer étant donné qu'il a déjà gagné. Si ce n'est pas le cas, nous évaluons alors la disposition grâce à la fonction ci-dessus.

Question 3 : Version “dynamique”

Les fonctions 'copier_dispo', 'tour_blanc' et 'tour_noir' sont les mêmes que dans la version naïve.

La fonction 'hexapawn_dynamique' est similaire à la fonction 'hexapawn_naif' à la seule différence que la première fait appel à la fonction d'évaluation dynamique, que je vais expliquer ci-dessous, contrairement à la seconde fonction qui fait appel à la fonction d’évaluation naïve.

Fonction “eval_hexapawn_d”:

Cette fonction ressemble fortement à la fonction de la question 2 'eval_hexapawn'. Cependant, celle-ci est enrichie.

Tout d'abord nous utilisons deux variables globales. Une d'entre elles regroupe toutes les dispositions déjà évaluées quand c'est au joueur blanc de jouer avec la valeur de l’évaluation de la configuration associée. L'autre variable globale fait pareil mais pour quand c'est au joueur noir de jouer. Quand c'est au joueur blanc de jouer, on regarde tout d'abord si on a déjà évalué cette configuration pour ce joueur. Si c'est le cas, on retourne alors la valeur enregistrée dans la variable globale. Dans le cas contraire, nous regardons si un pion noir (= pion adverse) a atteint la dernière rangée. Si c'est le cas, on enregistre cette nouvelle disposition dans notre variable globale associée et on retourne 0 étant donné que cela signifie que le joueur blanc a perdu. Si la disposition n'a pas encore été évaluée et qu'aucun pion adverse a atteint la dernière rangée, alors on évalue les différents coups possibles à jouer. Quand, c'est au tour du joueur noir, nous faisons pareil mais adapté au joueur noir. La suite est globalement pareille que la fonction d'évaluation précédente à la seule différence que nous enregistrons dans la variable globale associée la configuration évaluée ainsi que sa valeur.

Version “dynamique” avec symétries

A la suite de cette fonction, je me suis demandée s'il était utile de regarder non pas seulement si on avait déjà eu à faire à cette disposition mais aussi si on avait déjà évalué une configuration symétrique à celle-ci. J'ai donc implémenté cela dans le fichier 'version_dynamique_miroir'. Cependant, j'ai pu remarquer que sur les jeux de données présents sur le fil et sur la plateforme, même si cela permet parfois de réduire le nombre d'appels à la fonction d’évaluation, le temps d'exécution est plus important. Cela est dû au fait que, dans les exemples, il est plus coûteux en temps de modifier la disposition afin d’avoir une de ses symétries puis de regarder si on a déjà évalué cette configuration plutôt que d'évaluer cette configuration symétrique.

Les fonctions sont pour la plupart identiques ou fortement similaires à celles expliquées précédemment. La différence est que dans la fonction d'évaluation nous ne regardons pas seulement si on a déjà évalué cette configuration pour ce joueur mais aussi:

- si on a déjà évalué la configuration miroir pour ce joueur.
- si on a déjà évalué la configuration dans le cas inverse, c'est-à-dire que les pions noirs étaient les pions blancs et inversement et que le plateau était donc à l'inverse.
- si on a déjà évalué la configuration miroir inverse.

Afin de mettre en place cela, j'ai implémenté deux nouvelles petites fonctions qui permettent suivant une configuration donnée de:

- retourner la configuration miroir.
- retourner la configuration inverse.

Grâce à ces deux fonctions, je peux également avoir la configuration miroir inverse en les “combinant”.