

La pipopipette

I. Préambule

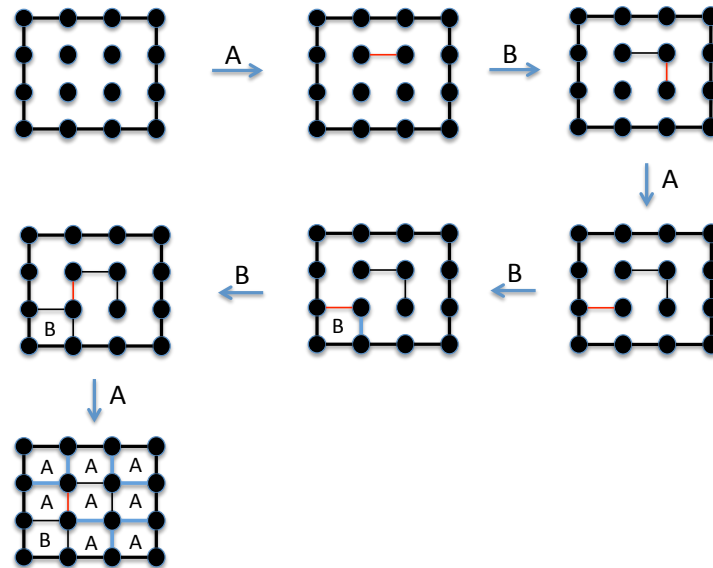


Figure 1: Une partie de pipopipette

La Pipopipette ou « jeu des petits carrés » est un jeu de société se pratiquant à deux joueurs en tour par tour. Le jeu se joue généralement avec papier et crayon sur du papier quadrillé. À chaque tour, chaque joueur trace un petit trait suivant le quadrillage de la feuille. Le but du jeu est de former des carrés. Le gagnant est celui qui a fermé le plus de carrés. Le fait de fermer un carré impose de rejouer, ce qui peut aboutir à fermer de nombreux carrés à la suite lorsque se créent des couloirs. (source Wikipédia)

De façon relativement abstraite, on peut voir l'espace des configurations du jeu de pipopipette comme un graphe où chaque sommet représente une configuration et où les successeurs d'un sommet sont les configurations obtenues grâce à un coup quelconque. Une partie consiste en un chemin à travers ce graphe entre un sommet de départ et un sommet d'arrivée.

II. Le sujet

Réaliser un programme en java calculant des stratégies pour jouer à la pipopipette. Votre programme permettra:

- de jouer en mode interactif sur un terminal,
- de concevoir des stratégies par une analyse du graphe des configurations,
- de concevoir des stratégies des méthodes d'apprentissage par renforcement,
- de comparer des stratégies par des méthodes exactes et par simulation,

Pour candidater à une note supérieur 16/20, votre système devra mettre en œuvre toutes les méthodes proposées au jeu de « qui perd-gagne ».

III. Stratégies pour la pipipopette

A. Stratégies élémentaires

Nous définissons 4 stratégies élémentaires :

- 1) Joueur Simplet : Le joueur choisit au hasard le coup suivant
- 2) Joueur Prévoyant : Le joueur choisit au hasard un coup parmi ceux donnant le moins de points immédiatement
- 3) Joueur Idiot : Le joueur choisit au hasard un coup parmi ceux donnant le plus de points immédiatement
- 4) Joueur Pondéré : Pour chaque configuration, le joueur a défini un poids. Le joueur choisit au hasard un coup parmi ceux minimisant la formule : $point + poids$ où $point$ désigne le nombre de points perdus immédiatement et $poids$ étant le poids de la configuration atteinte après le coup.

Le tableau suivant donne la valeur de l'option `strategie` utilisée dans la définition des commandes demandées

| | |
|--|---|
| Joueur Simplet | <code>.-simplet type nblignes nbcolonnes</code> |
| Joueur Prévoyant | <code>-prevoyant</code> |
| Joueur Idiot | <code>-idiot</code> |
| Joueur Pondéré : les pondérations des configurations sont données sous la forme d'un fichier pip | <code>-pondere fichier.pip</code> |

Pour les trois premières options, les paramètres `type` `nblignes` `nbcolonnes` définissent la configuration initiale : `nblignes` `nbcolonnes` donnent la taille du tableau et `type` indique si les traits du contour sont initialement dessinés : `type = C` avec contour, `type=S` sans contour. Pour le joueur Pondéré, l'information est donnée dans le fichier.

B. Graphe réduit des configurations (voir le cours)

C. Calcul d'une stratégie optimum (voir le cours)

D. Méthode d'apprentissage par renforcement (voir le cours)

E. Comparaison de deux stratégies (voir le cours)

F. Jeu du « qui perd-gagne »

Dans ce jeu, le gagnant est celui qui complète le moins de carré. Un joueur qui a l'obligation de compléter un carré si cela est possible.

IV. Que doit faire votre programme ?

Le programme java sera composé d'une unique archive jar **pipopipette.jar**. Ce programme devra s'exécuter dans un terminal avec les options suivantes :

- o `java -jar pipopipette.jar -name` affiche vos noms et prénoms
- o `java -jar pipopipette.jar -h` rappelle la liste des options du programme
- o `java -jar pipopipette.jar -graphe type nblignes`

`nbcolonnes` renvoie le graphe des configurations réduit au format dot.

- `java -jar pipopipette.jar -joue strategie` offre la possibilité de jouer sur la console de manière interactive.
- `java -jar pipopipette.jar -cal strategie` calcule une stratégie optimum face un joueur suivant la stratégie passée en paramètre. La commande renvoie le résultat au format pip (voir l'exemple simple C3x3.pip)
- `java -jar pipopipette.jar -cal -graphe strategie` calcule une stratégie optimum face un joueur suivant la stratégie passée en paramètre. La commande renvoie le résultat le graphe réduit au format dot et la pondération des configurations définissant la stratégie calculée.
- `java -jar pipopipette.jar -apprend parametres strategie` calcule une stratégie par une méthode d'apprentissage (détail à venir).
- `java -jar taquin.jar -eval strategie strategie` évalue les deux stratégies en paramètre de manière exacte et une table des probabilités de tous les scores possibles. Le résultat est un script gnuplot donnant le résultat sous la forme d'un diagramme en batons.
- `java -jar taquin.jar -simul N strategie strategie` évalue les deux stratégies en paramètre par simulation en lançant N parties et une table des probabilités de tous les scores possibles. Le résultat est un script gnuplot donnant le résultat sous la forme d'un diagramme en batons..

Pour obtenir une note supérieure à 16, vous reprenez les commandes précédentes le jeu « qui perd-gagne » en ajoutant l'option `-perd`.

V. Quoi et quand rendre ?

A rendre le 15 mai 2016.

Quoi :

- format papier : rapport à remettre au secrétariat
- par mail :
 - contenant les personnes du groupe,
 - rapport (format pdf) ,
 - le programme (fichier jar exécutable, contenant aussi les sources),
 - dossier compressé contenant les données d'expérimentation.

VI. Quelles références disponibles sur la toile et à la bibliothèque des sciences

1. http://fr.wikipedia.org/wiki/La_Pipopipette pour connaître les règles du jeu
2. http://fr.wikipedia.org/wiki/Apprentissage_par_renforcement donne quelques éléments sur les méthodes d'apprentissage par renforcement.
3. <http://www.termsys.demon.co.uk/vtansi.htm> contient des commandes de contrôle du terminal
4. <http://docs.oracle.com/javase/8/> API java
5. <http://java.sun.com/docs/codeconv/index.html> donne les conventions d'écriture d'un programme java
6. La programmation en pratique, Brian W. Kernigan et Rob Pike. Editeur Vuibert informatique. Lire le premier chapitre sur la manière de commenter un programme et choisir des noms de variables
7. Au coeur de Java 2 (Volume 1), Notions Fondamentales, Cay S. Horstmann et Gary

Cornell. Editeur CampusPress. Livre recommandé.

8. Algorithmique - 3ème édition - Cours avec 957 exercices et 158 problèmes, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein. Dunod.