Rapport MADI

Mathieu Berthellemy - Élodie Cayez

6 janvier 2016

Résumé

On s'intéresse aux déplacements d'un cavalier dans une grille donnée dont certaines cases sont interdites. On considère que le cavalier est initialement dans le coin supérieur gauche de la grille et l'on veut atteindre une case cible donnée (typiquement le coin inférieur droit) en un minimum de mouvements. En chaque case, le cavalier choisit une action parmi l'un des huit coups légaux autorisés aux échecs

Introduction

Ce rapport présentera differentes approches du problème. En première partie on s'interessera à une approche prudente,le cavalier devra se rendre d'un bout à l'autre de la grille en minimisant les pénalités induites par certaine cases colorées. Pour ce faire on utilisera l'algorithme d'itération de la valeur. En seconde partie le cavalier devra se rendre d'un bout à l'autre de la grille en équilibrant le nombre de fois qu'il passe sur les cases bleues et rouges. Pour ce faire on utilisera la programmation mathématique pour résoudre un PDM bi objectifs.

1 Recherche d'une trajectoire prudente

1.1 Modélisation

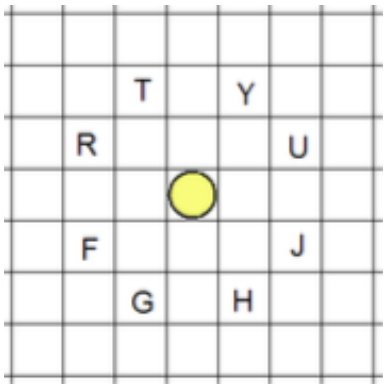
PDM = <S,A,T,R>

S Pour une matrice de tailler $n \times m$ contenant b cases noires, il y a $n \times m - b$ états du monde. On notera S_i^j l'état du monde où le cavalier est sur la case (i, j).

A Actions possibles : R, T, Y, U, J, H, G, F chaque élément correspond à un mouvement décrit sur cette figure.

Ainsi les voisins de l'état S_i^j sont les états :

- S_{i-2}^{j+1}si mouvement R
- S_{i-1}^{j+2}si mouvement T
- S_{i+1}^{j+2}si mouvement Y
- S_{i+2}^{j+1}si mouvement U
- S_{i+2}^{j-1}si mouvement J
- S_{i+1}^{j-2}si mouvement H
- S_{i-1}^{j-2}si mouvement G
- S_{i-2}^{j-1}si mouvement F



T Soit s l'état actuel du monde, a l'action effectuée et s' l'état attendu de l'action a. Soit A(x) l'ensemble des adjacents de x , N(x) l'ensemble des voisins de x au sens précédement définie. On a alors :

$$T(s, a, s') = 1 - \frac{|N(s)|}{16} \tag{1a}$$

$$T(s, a, s'') = \frac{|N(s)|}{16}, \forall s'' \in N \tag{1b}$$

R Soit :

- V l'ensemble des cases marquées verte
- B l'ensemble des cases marquées bleue
- R l'ensemble des cases marquées rouge
- N l'ensemble des cases marquées noire

On a alors :

$$R(x) = c(blanc) - 1 = -1, \forall x \in Bl \tag{2a}$$

$$R(x) = c(vert) - 1 = -11, \forall x \in V \tag{2b}$$

$$R(x) = c(bleu) - 1 = -21, \forall x \in B \tag{2c}$$

$$R(x) = c(rouge) - 1 = -31, \forall x \in R \tag{2d}$$

$$R(x) = c(noir) - 1 = -41, \forall x \in N \tag{2e}$$

$$R(S_n^m) = 999 \tag{2f}$$

1.2 Équations de Bellman associées

Les équations de Bellman associées à un tel procéssus décisionel Markovien sont l'ensemble des équations de la forme :

$$V^*(s) = \max_a \left[-1 + \gamma \left(\frac{V^*(s'(a)) \cdot (1 - |N|)}{16} + \frac{|N|}{16} \cdot \sum_{s'' \in N(s'(a))} V^*(s'') \right) \right], \forall s \in Bl \tag{3a}$$

$$V^*(s) = \max_a \left[-11 + \gamma \left(\frac{V^*(s'(a)) \cdot (1 - |N|)}{16} + \frac{|N|}{16} \cdot \sum_{s'' \in N(s'(a))} V^*(s'') \right) \right], \forall s \in V \tag{3b}$$

$$V^*(s) = \max_a \left[-21 + \gamma \left(\frac{V^*(s'(a)) \cdot (1 - |N|)}{16} + \frac{|N|}{16} \cdot \sum_{s'' \in N(s'(a))} V^*(s'') \right) \right], \forall s \in B \tag{3c}$$

$$V^*(s) = \max_a \left[-31 + \gamma \left(\frac{V^*(s'(a)) \cdot (1 - |N|)}{16} + \frac{|N|}{16} \cdot \sum_{s'' \in N(s'(a))} V^*(s'') \right) \right], \forall s \in R \tag{3d}$$

$$V^*(s) = \max_a \left[-41 + \gamma \left(\frac{V^*(s'(a)) \cdot (1 - |N|)}{16} + \frac{|N|}{16} \cdot \sum_{s'' \in N(s'(a))} V^*(s'') \right) \right], \forall s \in N \tag{3e}$$

1.3 Itération de la valeur - transitions déterministes

Data: S, A, T, R, la description du PDM. ϵ, la précision voulue. γ, le facteur d'actualisation
Result: D*, l'ensemble des politiques optimales.
$t \leftarrow 0$;
$V_0(s) \leftarrow 0, \forall s \in S$;
repeat
  $t \leftarrow t + 1$;
  foreach $s \in S$ **do**
    foreach $a \in A$ **do**
      $Q_t^a(s) \leftarrow R(s,a) + \gamma \sum_{s' \in S} T(s,a,s') \cdot V_{t-1}(s')$;
    end
    $V_t(s) \leftarrow max_{a \in A}\{Q_t^a(s)\}$;
  end
until $max_{s \in S}\{|V_t(s) - V_{t-1}(s)|\} < \epsilon$;
foreach $s \in S$ **do**
  $d(s) \leftarrow argmax_{a \in A}\{Q_t^a(s)\}$;
end
return *d(s),* $\forall s \in S$*;*

Algorithm 1: algorithme d'itération de la valeur

1.4 Itération de la valeur - transitions aléatoires

1.5 Calibration de gamma

Le choix d'un γ adéquat est d'une importance cruciale pour maximiser l'efficassité de la politique qui en dépend. Si trop petit, l'agent ne se préocupe pas assez du futur, il y a peut de chance qu'il arrive jusqu'au but. Au contraire si trop grand l'agent regarde trop dans le futur et tous ses voisins semble attirants. Nous avons réfléchis a un moyen de calculer le γ le plus approprié à une grille donnée. Notons W, H respectivement la longueur et la hauteur d'une telle grille. la distance de Manhattan pour aller d'un bout à l'autre est D = W + H (la diagonale). Dans ce problème le cavalier peut se déplacer de 4 cases d'un coup, on peut donc approximer le nombre de nécessaire de coups qu'il lui faudrait pour se rendre dans le coin inférieur gauche depuis le coin suppérieur droit par $D_c = D/4$ (la diagonale pour le cavalier). On peut donc calculer la proportion de la récompense final qui arrivera une case x donnée avec la suite suivante :

$$U(x) = U(x-1)^\gamma$$

ou plus précisement la proportion de la récompense final qui arrive à la case départ :

$$U(D_c) = U(D_c - 1)^\gamma$$

Par la suite nous posons B = [start, end] la fenêtre dans laquel nous souhaitons que la récompense perçut depuis la case départ se trouve. En résolvant par recherche dichotomique :

$$U(D) \in B$$

nous obtenons une valeur de γ de façon computationelle.

Data: g, la grille
Result: γ, le facteur d'actualisation le plus approprié pour g
$\gamma \leftarrow 0.5$;
$B_\gamma \leftarrow \{0, 1\}$;
$B \leftarrow \{min, max\}$;
$D_c \leftarrow (g.width + g.height)/4$;
repeat
  $tmp \leftarrow 1000$;
  foreach *i in range(D_c)* **do**
    $tmp \leftarrow pow(tmp, \gamma)$;
  end
  if $tmp \leq min$ **then**
    $B_\gamma \leftarrow \{B_\gamma[0], \gamma\}$;
  end
  if $tmp \geq max$ **then**
    $B \leftarrow \{\gamma, B_\gamma[1]\}$;
  end
until $tmp \geq min$ *and* $tmp \leq max$;
return γ

Algorithm 2: algorithme de recherche dichotomique de gamma

2 Recherche d'une politique équilibrée

2.1 Modélisation

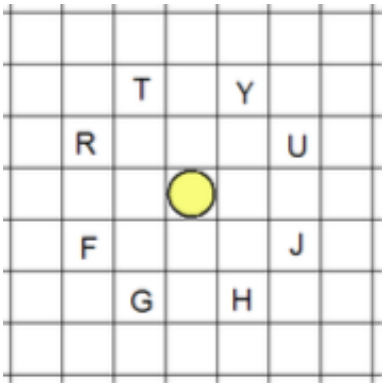
PDM = <S,A,T,R>

S Pour une matrice de tailler $n \times m$ contenant b cases noires, il y a $n \times m - b$ états du monde. On notera S_i^j l'état du monde où le cavalier est sur la case (i, j).

A Actions possibles : R, T, Y, U, J, H, G, F chaque élément correspond à un mouvement décrit sur cette figure.

Ainsi les voisins de l'état S_i^j sont les états :

- S_{i-2}^{j+1}si mouvement R
- S_{i-1}^{j+2}si mouvement T
- S_{i+1}^{j+2}si mouvement Y
- S_{i+2}^{j+1}si mouvement U
- S_{i+2}^{j-1}si mouvement J
- S_{i+1}^{j-2}si mouvement H
- S_{i-1}^{j-2}si mouvement G
- S_{i-2}^{j-1}si mouvement F



T Soit s l'état actuel du monde, a l'action effectuée et s' l'état attendu de l'action a. Soit A(x) l'ensemble des adjacents de x , N(x) l'ensemble des voisins de x au sens précédement définie. On a alors :

$$T(s, a, s') = 1 - \frac{|N(s)|}{16} \tag{4a}$$

$$T(s, a, s'') = \frac{|N(s)|}{16}, \forall s'' \in N \tag{4b}$$

R Soit :

- Bl l'ensemble des cases non marquées
- B l'ensemble des cases marquées bleue
- R l'ensemble des cases marquées rouge

On a alors :

$$R(x) = -2, \forall x \in Bl \tag{5a}$$

$$R(x) = c(bleu) - 2 = -1, \forall x \in B \tag{5b}$$

$$R(x) = c(rouge) - 2 = -1, \forall x \in R \tag{5c}$$

$$R(S_n^m) = 1000 - 2 = 998 \tag{5d}$$

2.2 Résolution par programmation mathématiques

$$max \sum_{s \in S} V_s - z \tag{6a}$$

$$sc : z \leq \sum_{s \in Red} V_s \tag{6b}$$

$$z \leq \sum_{s \in Blue} V(s) \tag{6c}$$

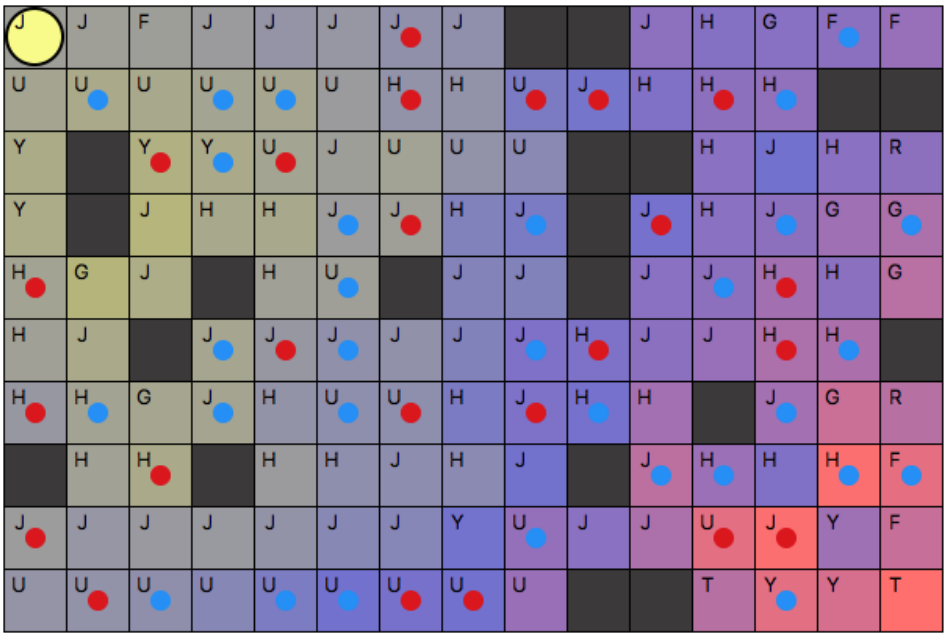
$$V_s \geq R(s') + \gamma \left[\left(1 - \frac{|N(s)|}{16} \right) \cdot V_{s'} + \sum_{s'' \in A(s')} \frac{1}{16} \cdot V_{s''} \right] \tag{6d}$$

$$V(s) = 1000, \forall s \in End \tag{6e}$$

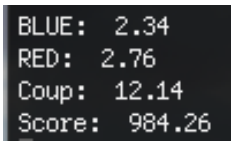
$$V(s) \in \Re, \forall s \in S \tag{6f}$$

avec :
— S, ensemble des états
— Blue, Red, ensemble des états de S tel quel la case est marquée respectivement bleu ou rouge
— End, l'ensemble des états finaux, ici End = (width, height)
— V(s), ensemble des états voisins de s
— A(s), ensemble des états adjacents de s
— (6b), (6c) on minimise la plus grande espérance entre $C_R et C_B$
— (6d) $\forall s \in S, \forall s' \in N(s)$
— (6e) Récompense induite pour arriver à l'état final
— (6f), V_s, la valeur de la recompense espérée pour l'état S

2.3 Résultats



Ci-dessus une capture d'écran du la grille une fois la politique calculée. Chaque case est coloriée selon son espérance d'utilité, de plus on trouve dans le coin superieur gauche de chaque case le coup à jouer pour suivre la politique.



résultat moyen sur 1000 répétitions

2.4 Résolution par itération de la valeur