Mini-projet d'Intelligence artificielle
Indications de solution de la partie I

José MARTINEZ

2023-2024

► LE CUBE DE M. RUBIK ET AUTRES CASSE-TÊTE ●●●●○

1 ► Modélisez ce *type* de jeu avec l'approche (E, i, F, O, P, C).

Réponse
Il s'agit de jeux de *permutations*, dont le taquin et l'âne rouge. Un état de ce *type* de jeu se modélise comme une permutation de valeurs, traduisible sous forme quasi canonique par :

$$E = \{1, 2, \ldots, n\} \leftrightarrow V^n$$

ou encore

$$E \subset V^n,$$

sous la contrainte que chaque valeur $v \in V$ n'apparaît qu'une seule fois.

Réponse
Tout état est *a priori* un problème à résoudre :

$$i = (v_{\pi(1)}, v_{\pi(2)}, \ldots, v_{\pi(n)}), \forall \pi \in S_n$$

Réponse
Sans perte de généralité,

$$f = (v_1, v_2, \ldots, v_n)$$

et

$$F = \{f\}.$$

Réponse
Les opérations sont une *famille* de permutations :

$$\text{permutation}_k : E \leftrightarrow E.$$

Réponse
Toutes les permutations sont possibles de manière inconditionnelle :

$$P = \left\{\text{pré-condition}_{\text{permutation}_k} : E \to \mathcal{B} := 1 | \forall k \in K\right\}.$$

Réponse
Deux possibilités générales peuvent être envisagées pour les coûts :
— un coût unitaire, l'approche par défaut :

$$C = \left\{\text{coût}_{\text{permutation}_k} : E \to E := 1 | \forall k \in K\right\}.$$

— un coût égal au nombre de valeurs ayant changé de position,

$$C = \left\{\text{coût}_{\text{permutation}_k} : \begin{array}{ccc} E & \to & E \\ (v_1, \ldots, v_n) & \mapsto & |\{i : 1 \leq i \leq n, v_i \neq v_{\pi_k(i)}\}| \end{array} \middle| \forall k \in K\right\},$$

2 ► Dérivez-en une version particularisée pour le *Rubik's Cube*, dans sa version 2×2 pour simplifier le problème.

Réponse

$$E \subset (C \times \{1, 2, 3, 4\})^{6.n^2}$$

avec $C = \{\text{rouge}, \text{vert}, \ldots, \text{orange}\}$ et $n = 2$ pour le *Rubick's Cube* 2×2.

Réponse
Toutes les configurations du cube ne sont *pas* accessibles, sans démontage. En effet, un cube-problème s'obtient comme un mélange aléatoire du cube acheté. Nous le définissons après l'introduction des opérations.

Réponse
Comme dans le cas général,

$$F = \{f\}.$$

Arbitrairement, nous posons :

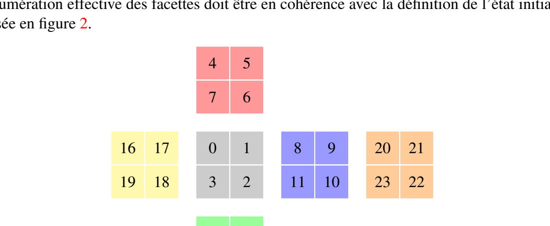
$$f = ((r, 1), (r, 2), (r, 3), (r, 4), (g, 1), (g, 2), \ldots, (o, 1), (o, 2), (o, 3), (o, 4)).$$

Réponse
Nous particularisons les permutations à des rotations :

$$O = \{\text{rotation}_{f,a} : E \leftrightarrow E\}$$

avec, comme paramètres :
— $f \in \{\text{avant}, \text{arrière}, \text{dessus}, \text{dessous}, \text{gauche}, \text{droite}\}$;
— $a \in \{90, -90, 180\}$.

Réponse
Pour la réalisation, il faut définir la rotation sur une face, illustrée en figure 1.



(a) « numérotation » avant la rotation (b) effet de la rotation

FIGURE 1 – Rotation à 90 degrés dans le sens des aiguilles d'une montre de la face « centrale »

Elle se traduit sous la forme :

$$\text{rotation\_angle}_{90}((\underbrace{a, b, c, d}_{\text{face}}, \underbrace{e, f, g, h, i, j, k, l}_{\text{contour}})) = (\underbrace{d, a, b, c}_{\text{face} \curvearrowright}, \underbrace{k, l, e, f, g, h, i, j}_{\text{contour} \curvearrowright})$$

Les deux autres rotations s'en déduisent par composition : $\text{rotation\_angle}_{180} = (\text{rotation\_angle}_{90})^2$ et $\text{rotation\_angle}_{-90} = (\text{rotation\_angle}_{90})^3$.

Réponse
La numération effective des facettes doit être en cohérence avec la définition de l'état initial, comme celle utilisée en figure 2.

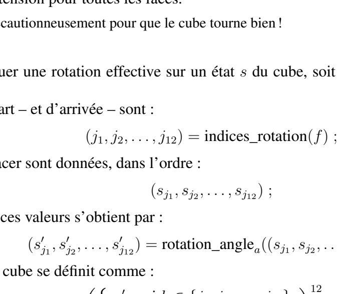


FIGURE 2 – Une numérotation globale des facettes

En plaquant cette numérotation sur la rotation à 90 degrés horaires de la figure 1, nous obtenons les indices de deux cycles d'une permutation :

$$\text{indices\_rotation}(\text{face}) = \left(\underbrace{0, 1, 2, 3}_{\text{centre}}, \overbrace{\underbrace{7, 6}_{\text{haut}}, \underbrace{8, 11}_{\text{droite}}, \underbrace{13, 12}_{\text{bas}}, \underbrace{18, 17}_{\text{gauche}}}^{\text{cycle extérieur}}\right).$$

Il faut la définir en extension pour toutes les faces.

Remarque. À faire précautionneusement pour que le cube tourne bien !

Réponse
À partir de là, appliquer une rotation effective sur un état s du cube, soit $\text{rotation}_{f,a}(s)$ se définit par les étapes suivantes :
— les indices de départ – et d'arrivée – sont :

$$(j_1, j_2, \ldots, j_{12}) = \text{indices\_rotation}(f) ;$$

— les valeurs à déplacer sont données, dans l'ordre :

$$(s_{j_1}, s_{j_2}, \ldots, s_{j_{12}}) ;$$

— la permutation de ces valeurs s'obtient par :

$$(s'_{j_1}, s'_{j_2}, \ldots, s'_{j_{12}}) = \text{rotation\_angle}_a((s_{j_1}, s_{j_2}, \ldots, s_{j_{12}})) ;$$

— l'état d'arrivée du cube se définit comme :

$$\left(\begin{cases} s'_k & \text{si } k \in \{j_1, j_2, \ldots, j_{12}\} \\ s_k & \text{sinon} \end{cases}\right)_{k=1}^{12}.$$

Remarque. Il s'agit là des étapes détaillées de l'application d'une permutation où seulement une partie des cycles est fournie, le reste devant être complété par l'identité. Nous avons donc $\text{rotation}_{f,a}(s) = \pi_{f,a}(s)$ où :

$$\pi_{f,a} = \left\{ J_i \mapsto K_i \middle| \begin{array}{l} J = \text{indices\_rotation}(f) \\ K = \text{rotation\_angle}_a(J) \\ i \in \{1, \ldots, 12\} \end{array} \right\} \bigcup \{j \mapsto j | j \notin \text{indices\_rotation}(f)\},$$

avec une notation impropre pour l'appartenance à un vecteur.

Réponse
Comme dans le cas général, il n'y a pas de contrainte sur les rotations.

Réponse
Nous leur associons un coût unitaire.

Réponse
Finalement, la génération de tous les successeurs d'un état, telle que nécessaire pour utiliser les solveurs, est traduite par :

$$\Gamma^+(s) = \left\{(\text{rotation}_{f,a}(s), 1) \middle| \begin{array}{l} f \in \{\text{avant}, \text{arrière}, \text{dessus}, \text{dessous}, \text{gauche}, \text{droite}\} \\ a \in \{90, -90, 180\} \end{array}\right\}.$$

3 ► Étudiez la structure du graphe de recherche :

(i) Quelle est sa taille ?

Réponse
Il y a un maximum de $8!$ permutations des sommets entre eux. Pour chacune des permutations, chaque sommet peut être orienté de trois façons, ce qui donne 3^8 possibilités. Seulement un tiers des permutations ont toutes les facettes bien positionnées les unes par rapport aux autres. Nous arrivons ainsi à un total de

$$3^8.8!/3 = 88\,179\,840.$$

Le nombre d'arcs est étudié plus bas.

(ii) Est-il symétrique ?

Réponse
C'est un graphe symétrique car la rotation inverse immédiate,

$$\text{rotation}_{f,\pm 90} \circ \text{rotation}_{f,\mp 90},$$

ou encore la composition avec la rotation complémentaire

$$\text{rotation}_{f,180} \circ \text{rotation}_{f,180},$$

sur une même face f s'annulent.

Si oui, quels problèmes cela pose-t-il ?

Réponse
Il y a autant de circuits de longueur 2 que d'états, puis une explosion combinatoire de circuits de longueurs 3, 4, etc. !

Si oui, quels avantages cela apporte-t-il ?

Réponse
Les recherches peuvent se faire :
— aussi bien de manière « naturelle », c'est-à-dire de l'état initial vers l'état final,
— que dans le sens inverse ;
— voire de manière *bidirectionnelle*.

Remarque. Bien réalisée, une recherche bidirectionnelle est bien plus efficiente qu'une recherche unidirectionnelle.

Éventuellement, comment peut-on remédier aux problèmes ?

Réponse
Éviter d'utiliser deux fois de suite la même face.

Réponse
Éventuellement, éviter de faire tourner successivement une face et son *opposée* sur l'axe.

(iii) Quel est son facteur de branchement ?

Réponse
Le facteur de branchement est égal à $3 \times 6 = 18$, soit trois rotations possible par face.
En évitant d'utiliser deux fois de suite la même face, il est réduit à $3 \times 5 = 15$, sauf pour l'état initial de résolution où il reste à 18.
En évitant d'utiliser une face et sa face opposée immédiatement après, il est même réduit à 12, sauf pour l'état initial.

Remarque. *Attention* ! Cette double opération pourrait être faite par un joueur et rendrait la recherche de la solution *impossible* ! Il faut donc y prendre garde lorsque l'on utilise cette version.

(iv) Est-il dense ?

Réponse
Le nombre de voisins de chaque état est 18. Nous avons donc le ratio transitions sur états :

$$\frac{18.n}{n^2 - n} = \frac{18}{n - 1}.$$

La densité tend vers zéro quand le graphe *exploré* augmente. À la limite, nous avons

$$\frac{18}{88\,179\,840 - 1} \simeq 204 \times 10^{-9}.$$

(v) Avec le facteur de branchement, quelle est l'implication sur un *arbre* de recherche ?

Réponse
Le facteur de branchement est tel qu'une très grande partie du graphe est générée en quelques niveaux. La densité est si faible que le nombre de doublons devrait être très faible dès lors que l'on évite les rotations inverses ou se compensant pour partie. En effet, tant que tous les états n'ont pas été générés, les 15 ou 12 arcs sortants ont, en toute probabilité, plus de chances d'aboutir au très grand nombre de nouveaux états qu'au très petit nombre d'anciens états.

	$b = 15$		$b = 12$	
profondeur (p)	cumul sommets	exhaustivité (en %)	sommets	exh.
0	1	0,00	1	0,00
1	19	0,00	19	0,00
2	289	0,00	235	0,00
3	4 339	0,00	2 827	0,00
4	65 089	0,07	33 931	0,04
5	976 339	1,11	407 179	0,46
6	14 645 089	16,61	4 886 155	5,54
7	**219 676 339**	249,12	58 633 867	66,49
8	**3 075 468 750**	3 736,85	**703 606 411**	797,92

TABLE 1 – Nombres d'états cumulés pour les deux facteurs de branchement sans retour inverse immédiat

Les nombres d'états sont reportés sur le tableau 1 en fonction des facteurs de branchements et de la profondeur de recherche. Il est divisé par la taille de l'espace de recherche ce qui *approxime* grossièrement mais utilement la densité à ce niveau :
— Sur les premiers niveaux, la détection des doublons est *absolument inutile*.
— Au sixième niveau, nous sommes certains de la présence des doublons mais dans des proportions *a priori* supportables, surtout pour un facteur de branchement de 12.
— À partir de la profondeur 7, la présence des doublons devient problématique mais la taille de l'espace exploré l'est encore plus : *tout* pour $b = 15$ ou plus de la moitié pour $b = 12$.
En conclusion, la détection des doublons n'est pas envisageable sur la totalité du graphe. Toutefois, elle demeurerait utilisable sur chaque branche de l'arbre. Mais, ce ne sera vraiment utile qu'à partir de la profondeur 6 et n'évitera qu'une partie de l'explosion sur-exponentielle qu'à partir de la profondeur 7, où l'on a déjà tout visité ou presque...

4 ► Au vu de l'étude de la question 3, choisissez un solveur adapté parmi ceux vus en cours, depuis la version récursive naïve par retours-arrière jusqu'à A^*. Expliquez votre choix en détaillant les avantages et inconvénients de chaque approche ou famille d'approches. Discutez les difficultés restantes. Éventuellement, proposez une « nouvelle » variante.

Réponse
Le solveur naïf n'a presqu'aucune chance de fonctionner correctement. Il va entrer dans une recherche infinie sur le premier circuit trouvé, ce qui devrait se produire à la profondeur 6.

Réponse
Nous savons que le nombre d'opérations maximal est de 20 pour le *Rubik's Cube* 3×3. Nous pouvons donc limiter la profondeur de recherche. Mais, il reste à déterminer cette limite...
Le *minorant* est 7. Au vu de la dernière ligne du tableau 2, on peut estimer qu'il est de 8 ou 9...

Réponse
La détection des circuits n'est nécessaire que pour les niveaux profonds, c'est-à-dire à partir de la profondeur 6 ou 7. Mais, le nombre d'états à envisager est déjà tellement important, que le bénéfice semble minime *a priori*.
De plus, ce contrôle ne protège pas des développements répétés associés aux *cycles*.

Réponse
Dans la famille des recherches en arbre, la recherche par « séparation et évaluation » peut être retenue. Encore faut-il trouver un bonne *heuristique d'évaluation*...

Réponse
Les recherches itérées présentent les mêmes difficultés que les recherches précédentes. Notons toutefois que le facteur de branchement est tellement élevé qu'elles ne coûtent pour ainsi dire rien de plus.
Le tableau 2 montre que redévelopper tous les niveaux depuis la racine se stabilise autour de la constante multiplicative 1,09 dans le cas d'un arbre dodécanaire.

			$b = 15$			$b = 12$
p	développés (d)	re-développés (r)	ratio r/d (en %)	d	r	r/d
0	1	1	100,00	1	1	100,00
1	19	20	105,26	19	20	105,26
2	289	309	106,92	235	255	108,51
3	4 339	4 648	107,12	2 827	3 082	109,02
4	65 089	69 737	107,14	33 931	37 013	109,08
5	976 339	1 046 076	107,14	407 179	444 192	109,09
6	14 645 089	15 691 165	107,14	4 886 155	5 330 347	109,09
7	219 676 339	235 367 504	107,14	58 633 867	63 964 214	109,09

TABLE 2 – Nombres d'états re-développés par une recherche itérée suivant la profondeur

Réponse
Les recherches en graphe se heurtent *toutes* à la taille de l'espace de recherche dès lors que la solution nécessite six ou sept étapes et plus, avec déjà des millions d'états à stocker. Or, la détection des doublons n'est pas nécessaire en dessous.

Réponse
Finalement, la solution la plus simple consiste à effectuer une recherche en largeur d'abord mais de manière itérée. Nous emploierons donc IDS, sans besoin de détection des circuits. Si une bonne heuristique d'évaluation est trouvée, nous pourrons passer à IDA.
À des profondeurs trop importantes, aucune des deux approches candidates ne reste viable ! Mais...

Indication
Le problème étant symétrique, nous pouvons envisager une recherche bidirectionnelle :
(a) depuis le cube-problème non pas vers la solution mais une pré-solution ;
(b) depuis l'état final vers des états proches, c'est-à-dire des pré-solutions.
Comme la solution est unique, la recherche inversée à partir de l'état final peut n'être exécutée qu'une seule fois et stockée. On peut stocker tous les états jusqu'à la profondeur 3 ou 4, en fonction des temps de calcul ou du fait qu'il aient été stockés dans un fichier à relire. Combinée à une recherche IDS d'une profondeur maximale 4 ou 5, voire 6, nous avons donc au total une recherche sur une profondeur de 7 à 10 niveaux, et cela sans avoir à détecter les doublons.

Remarque. Mais ce n'est pas ce qui est demandé dans la suite...

5 ► Y a-t-il des heuristiques applicables à ce (type de) problème, aussi bien en théorie qu'en pratique au vu de leur complexité de mise en œuvre, pour mémoire :

(i) ordonnancement ;

Réponse
A priori, aucune rotation n'est préférable à une autre...

(ii) contraintes ;

Réponse
Deux contraintes ont déjà été utilisées afin de limiter le facteur de branchement. En poursuivant l'analyse, il faut attendre la profondeur 6 pour faire émerger le motif générique

$$(\text{rotation}_{f,180} \circ \text{rotation}_{g,180})^3$$

où f est une face quelconque et g, une face perpendiculaire à f, c'est-à-dire toutes les autres sauf son opposée.
Or, à la profondeur 6, on génère au minimum $12^6 = 2\,985\,984$ séquences de rotations distinctes. Chercher à détecter les $6 \times 4 = 24$ séquences qui sont idempotentes est bien ridicule (0,000 8 %)...

Remarque. Nous savons maintenant que 6 est la longueur minimale d'un circuit dans un graphe d'états avec un facteur de branchement de 12. Détecter les circuits n'est donc nécessaire qu'à partir de cette profondeur.

En fait, il y a trois variations supplémentaire sur ce même thème, qui sont donc au total :

$$\begin{cases} r_{f,180} \circ r_{g,180} \circ r_{f,180} \circ r_{g,180} \circ r_{f,180} \circ r_{g,180} \\ r_{f,180} \circ r_{g,180} \circ r_{f,180} \circ r_{g',180} \circ r_{f',180} \circ r_{g',180} \\ r_{f,180} \circ r_{g,180} \circ r_{f',180} \circ r_{g,180} \circ r_{f,180} \circ r_{g',180} \\ r_{f,180} \circ r_{g,180} \circ r_{f',180} \circ r_{g',180} \circ r_{f',180} \circ r_{g180} \end{cases}$$

où f' et g' sont les faces opposées de f et g respectivement. Nous avons donc réellement $4 \times 24 = 96$ séquences, mais pas de quoi changer notre conclusion puisque nous passons à 0,003 %...

Réponse
Exploiter d'autres contraintes devrait pouvoir amener à des situations où une rotation est « imposée », bien quelle ne soit jamais unique, voire « interdite ». On voit mal quelle propriété exploiter afin de fournir des indications aussi fortes...

(iii) équivalences ;

Réponse
Un cube peut être *réorienté dans l'espace*, en faisant pivoter ce dernier suivant l'un des axes et même deux d'entre eux, ce qui nous donne sa classe d'équivalence. Chacune contient les 24 façons de le faire.
Le nombre d'états développés peut au mieux être réduit d'un ordre de magnitude.
Nous pouvons choisir le représentant de chaque classe comme l'élément minimal au sens de l'ordre lexicographique des séquences :

$$\text{représentant}(s) = \min\,[s].$$

Ce calcul est très coûteux. Il faut calculer 23 réorientations du cube, chacune nécessitant de créer de nouveaux cubes de 24 couples de valeurs scalaires. (Il faut créer de nouvelles opérations ou réutiliser les rotations mais ce sera plus coûteux encore.) Enfin, il y a 24 comparaisons d'inégalités afin de déterminer le minimum, chacune variant entre une et 24 comparaisons. Un *minorant* de la constante multiplicative associée à cette opération en temps borné constant est déjà $23 \times 24 \times 24 = 13\,248$.
Nous avons donc :
— d'une part, un ordre de magnitude gagné sur le nombre d'états, devenant applicable et sensible seulement pour les niveaux profonds, c'est-à-dire à partir de la profondeur 7 ;
— d'autre part, un surcoût de plusieurs ordres de magnitude s'appliquant à chaque état développé.
Le choix est vite fait ! Pour une recherche arborescente, on ne doit en attendre aucun gain.

Remarque. Pour une recherche en graphe, la réduction du nombre d'états-représentants pourrait compenser le surcoût de calcul, mais elles ont été exclues...

(iv) évaluation.

Réponse
S'agissant d'un jeu de permutations, comme le taquin, l'heuristique standard est celle de Manhattan, c'est-à-dire la somme des distances minimales entre les positions courantes des facettes et celles de destination.
Mais, elle n'est pas suffisamment discriminante... Dans sa version simple, chaque facette se trouve à une distance moyenne de 2 de sa position finale.