

# AAIA TP 3IF - Monde des Blocs

Florian Rascoussier

Christine Solnon

2 janvier 2024



## **Introduction**

# Table des Matières

<b>1 Partie 1 : Modélisation du problème</b>	<b>3</b>
1.1 Définition et déclaration des variables . . . . .	3
1.2 Question 1 . . . . .	3
1.3 Question 2 . . . . .	3
1.4 Question 3 . . . . .	3
1.5 Question 4 . . . . .	3
<b>2 Partie 2 : Définition du graphe d'états</b>	<b>4</b>
2.1 Question 5 . . . . .	4
2.2 Question 6 . . . . .	4
2.3 Question 7 . . . . .	4
2.4 Question 8 . . . . .	5
2.5 Question 9 . . . . .	5
<b>3 Partie 3 : Heuristiques pour le monde des blocs</b>	<b>5</b>
3.1 Question 10 . . . . .	5
3.2 Question 11 . . . . .	6
3.3 Question 12 . . . . .	6
3.4 Question 13 . . . . .	6
3.5 Question 14 . . . . .	6
3.6 Question 15 . . . . .	7
<b>Acronymes</b>	<b>8</b>
<b>Bibliographie additionnelle</b>	<b>8</b>

## Table des figures

1 Exemples de deux états $E_1$ et $E_2$ , pour $k = 3$ et $n = 5$ . . . . .	7
---	---

## **Codes et programmes**

1	Résultats de l'exécution du programme de recherche de plus court chemin . . . . .	5
---	---	---

# 1 Partie 1 : Modélisation du problème

Dans cette partie, nous nous intéressons à la modélisation du problème du *Monde des Blocs* en tant que problème de planification.

## 1.1 Définition et déclaration des variables

Voici une liste des variables utilisées, avec leur domaine et leur signification :

- $n$  : le nombre de blocs
- $k$  : le nombre de piles
- $E$  : un état donné du Monde des Blocs
- $actions(E)$  : l'ensemble de toutes les actions possibles pour l'état  $E$
- $t(E, i \rightarrow j)$  : le nouvel état obtenu en appliquant l'action  $i \rightarrow j$  à l'état  $E$
- $G = (S, A)$  : le graphe d'états, où  $S$  est l'ensemble des états possibles et  $A$  est l'ensemble des transitions entre les états

## 1.2 Question 1

*Combien d'actions différentes sont-elles possibles pour l'état  $E_1$  ?*

Supposons que l'état  $E_1$  dispose de  $k = 3$  piles non-vides. Chaque pile peut envoyer un bloc vers  $k - 1$  autres piles. Donc, pour chaque pile, il y a  $k - 1$  actions possibles. Par conséquent, le nombre total d'actions possibles pour l'état  $E_1$  est  $(k - 1) \times k = 6$  actions.

## 1.3 Question 2

*Étant donné un état  $E$  de  $n$  blocs sur  $k$  piles, quelle est la taille maximale de  $actions(E)$  ?*

Pour maximiser le nombre d'actions possibles, chaque pile doit contenir au moins un bloc, permettant ainsi des déplacements de blocs entre toutes les piles. Ainsi, chaque pile peut envoyer un bloc à  $k - 1$  autres piles. Le nombre total d'actions possibles est donc  $k(k - 1)$ .

## 1.4 Question 3

*Étant donné un état  $E$  de  $n$  blocs sur  $k$  piles ayant  $v$  piles vides, quelle est la taille de  $actions(E)$  ?*

Avec  $v$  piles vides, il reste  $k - v$  piles contenant des blocs. Chaque pile active peut envoyer un bloc à  $k - 1$  autres piles, donc le nombre total d'actions est  $(k - v)(k - 1)$ .

## 1.5 Question 4

*Quel est l'ordre de grandeur du nombre total d'états différents possibles ?*

Le nombre total d'états est exponentiel par rapport au nombre de blocs et linéaire par rapport au nombre de piles.

Pour s'en convaincre, on peut considérer que pour atteindre l'état final, chaque bloc doit être déplacé au moins une fois. Considérons cette situation qui est donc une borne minimale du chemin optimal parmi le graphe d'états possibles. Dans ce cas, à chaque niveau de profondeur, on a vu avec la question 2 qu'il y a  $k(k-1)$  actions possibles au maximum. Ainsi, chaque état à chaque niveau possède  $k(k-1)$  états fils. On peut donc en déduire que le nombre total d'états est de l'ordre de  $O(k(k-1)^n) = O((k^2 - k)^n) = O(k^2n)$ .

On se retrouve alors avec une borne minimale du nombre maximum d'états possible en  $O(k^2n)$ , ce qui est exponentiel par rapport au nombre de blocs et linéaire par rapport au nombre de piles.

## 2 Partie 2 : Définition du graphe d'états

### 2.1 Question 5

*Le graphe d'états  $G$  est-il orienté ?*

Oui, le graphe d'états  $G$  est orienté. Chaque action menant d'un état  $E$  à un état  $E'$  représente un arc orienté dans le graphe.

### 2.2 Question 6

*Quels sont les algorithmes qui peuvent être utilisés pour rechercher ce plus court chemin ?*

Plusieurs algorithmes peuvent être utilisés pour rechercher le plus court chemin, tels que Bellman-Ford, BFS, DFS et Dijkstra, en fonction des spécificités du graphe (poids, orientation, etc.).

### 2.3 Question 7

*Quel est l'algorithme le plus efficace pour rechercher ce plus court chemin ?*

Dans le contexte d'un graphe non pondéré comme celui du Monde des Blocs, BFS (Breadth-First Search) est généralement le plus efficace, car il trouve le plus court chemin en explorant uniformément autour du nœud source.

DFS est un autre algorithme de parcours de graphe qui se concentre sur l'exploration en profondeur d'une branche du graphe avant de revenir en arrière pour explorer d'autres branches. Il peut également être utilisé pour trouver le plus court chemin.

Chacun de ces algorithmes a ses avantages et inconvénients. Mais surtout, ils peuvent être améliorés en utilisant des heuristiques pour guider la recherche vers des solutions plus optimales.

## 2.4 Question 8

Quelle est la complexité en temps de cet algorithme par rapport à  $|S|$  et  $|A|$  ?

La complexité temporelle de BFS est linéaire, soit  $O(|S| + |A|)$ , où  $|S|$  est le nombre de sommets (états) et  $|A|$  est le nombre d'arêtes (actions) dans le graphe.

## 2.5 Question 9

Quelle est la complexité en temps de cet algorithme par rapport au nombre de blocs ( $n$ ) et de piles ( $k$ ) ?

La complexité temporelle de l'algorithme par rapport au nombre de blocs ( $n$ ) et de piles ( $k$ ) est exponentielle, car le nombre total d'états possibles croît exponentiellement avec l'augmentation de  $n$  et  $k$  (voir question 4, section 1.5).

# 3 Partie 3 : Heuristiques pour le monde des blocs

## 3.1 Question 10

On commence par exécuter le programme de recherche de plus court chemin pour 4 piles et un nombre de blocs variant de 6 à 8. On obtient les résultats suivants :

```
1      $ make run
2      Enter the number of stacks: 4
3      Enter the number of blocs: 6
4      Optimal solution of length 7 found in 12982 iterations and 0.060902 seconds
5      [...]
6      $ make run
7      Enter the number of stacks: 4
8      Enter the number of blocs: 7
9      Optimal solution of length 9 found in 188569 iterations and 1.04769 seconds
10     [...]
11     $ make run
12     Enter the number of stacks: 4
13     Enter the number of blocs: 8
14     Optimal solution of length 11 found in 2224481 iterations and 15.6446 seconds
15     [...]
```

Code 1 – Résultats de l'exécution du programme de recherche de plus court chemin

Les résultats montrent une augmentation exponentielle du nombre d'itérations et du temps d'exécution avec l'augmentation du nombre de blocs. Pour améliorer les performances de l'algorithme, on peut adopter des heuristiques adéquates. Trois heuristiques sont introduites :

- $h_1$  : Nombre de blocs ne se trouvant pas sur la dernière pile.
- $h_2$  : Nombre de blocs ne se trouvant pas sur la dernière pile, plus deux fois le nombre de blocs  $b$  tels que  $b$  se trouve sur la dernière pile mais devra nécessairement être enlevé de cette pile pour ajouter et/ou supprimer d'autres blocs sous lui.

- $h_3$  : Nombre de blocs de  $E_i$  ne se trouvant pas sur la dernière pile, plus le nombre de blocs se trouvant au-dessus de chaque bloc ne se trouvant pas sur la dernière pile.

Pour qu'une heuristique soit admissible, elle doit toujours sous-estimer le coût réel pour atteindre l'objectif. Formellement, une heuristique  $h$  est admissible si pour tout état  $E$ ,  $h(E) \leq \delta(E, E')$  où  $\delta(E, E')$  est la distance réelle entre  $E$  et l'état cible  $E'$ .

Une heuristique  $h$  est dite plus informée qu'une heuristique  $h'$  si, pour tout état  $E$ ,  $h(E) \geq h'(E)$ , tandis que les deux heuristiques sont incomparables s'il existe deux états  $E$  et  $E'$  tels que  $h(E) < h'(E)$  et  $h(E') > h'(E')$ .

### 3.2 Question 11

*L'heuristique  $h_1$  est-elle admissible ?*

L'heuristique  $h_1$  est admissible, car le nombre de blocs ne se trouvant pas sur la dernière pile représente une limite inférieure du nombre de mouvements nécessaires pour atteindre l'état final. En effet, chaque bloc doit au moins une fois être déplacé vers la dernière pile, sous-estimant ainsi le coût réel pour atteindre l'objectif.

### 3.3 Question 12

*L'heuristique  $h_2$  est-elle admissible ?*

L'heuristique  $h_2$  est également admissible. Elle prend en compte le coût de déplacement des blocs qui sont sur la dernière pile, mais doivent être déplacés pour permettre à d'autres blocs de se placer en dessous. En multipliant ce nombre par deux, on reste dans une estimation inférieure du coût réel. En effet, pour chaque bloc sur la dernière pile, mais n'étant pas à la bonne hauteur, il faudra au minimum 2 déplacements de ce bloc considéré pour l'enlever puis le remettre sur la dernière pile. Il s'agit d'une estimation minimum, car chaque bloc ainsi déplacé peut nécessiter plus de 2 mouvements pour atteindre sa position finale.

### 3.4 Question 13

*L'heuristique  $h_2$  est-elle plus informée, moins informée, ou incomparable par rapport à l'heuristique  $h_1$  ?*

L'heuristique  $h_2$  est plus informée que l'heuristique  $h_1$ . En plus de compter les blocs ne se trouvant pas sur la dernière pile, elle prend en compte le coût supplémentaire pour déplacer les blocs déjà sur la dernière pile, mais qui doivent être temporairement déplacés. Ainsi,  $h_2$  donnera une valeur de coût égale ou supérieur à celle fournie par  $h_1$ , et donne donc une meilleure estimation du coût réel. Elle est donc plus informée.

### 3.5 Question 14

*L'heuristique  $h_3$  est-elle admissible ?*

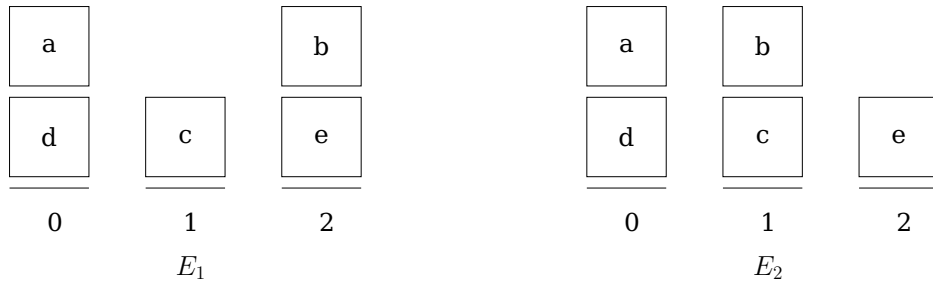


Figure 1 – Exemples de deux états  $E_1$  et  $E_2$ , pour  $k = 3$  et  $n = 5$ .

L'heuristique  $h_3$  est admissible. Elle compte non seulement les blocs qui ne sont pas sur la dernière pile, mais aussi le nombre de blocs qui doivent être déplacés pour permettre le placement d'autres blocs. Cette mesure fournit une sous-estimation du coût réel, car elle ne prend pas en compte les déplacements multiples ni l'ordre optimal de déplacement des blocs.

### 3.6 Question 15

*L'heuristique  $h_3$  est-elle plus informée, moins informée, ou incomparable par rapport à l'heuristique  $h_2$  ?*

Tandis que  $h_2$  compte de manière spécifique les déplacements nécessaires pour les blocs sur la dernière pile,  $h_3$  se concentre sur une vue plus globale de tous les blocs qui ne sont pas à leur place. En fonction de la situation, par exemple si la dernière pile est déjà bien remplie,  $h_2$  peut donner une meilleure estimation du coût réel que  $h_3$  puisqu'il reste peu de blocs sur les autres piles. C'est bien sûr l'inverse quand la dernière pile est vide. Ainsi, les deux heuristiques sont incomparables.

On peut ainsi trouver des contre-exemples :

$$\exists E_1, E_2 \in E \mid h_2(E_1) < h_2(E_2) \wedge h_3(E_1) > h_3(E_2).$$

Dans l'illustration suivante 1, on peut voir :

$$h(E_1) = \begin{cases} h_1(E_1) = 3 \\ h_2(E_1) = 5 \\ h_3(E_1) = 4 \end{cases} \quad h(E_2) = \begin{cases} h_1(E_2) = 4 \\ h_2(E_2) = 4 \\ h_3(E_2) = 6 \end{cases}$$

On a donc  $h_2(E_1) > h_3(E_2) \wedge h_2(E_1) < h_3(E_2)$ , ce qui montre que les deux heuristiques sont incomparables.



## Acronymes

**AAIA** Algorithmique Avancée pour l'Intelligence Artificielle et les graphes. 2

## Bibliographie additionnelle

- [1] Christine Solnon. *Première partie : Algorithmique avancée pour les graphes*. AAIA. CITI Lab, INSA Lyon dépt. Informatique. 2016. url : <http://perso.citi-lab.fr/csolnon/supportAlgoGraphes.pdf> (visité le 28/12/2023).