

# Rapport sur le projet Le monde des blocs

mis en page par  
DIARE Youssouf 22008756  
OLANGASSICKA Franck 22112035  
L3 info,  
Groupe 2A  
Chargé de Tp : **BONNET Gregory**  
Université de Caen Normandie  
2024–2025

19 novembre 2024



UNIVERSITÉ  
CAEN  
NORMANDIE

# Sommaire

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Modélisation du problème</b>	<b>2</b>
2.1	Variables et contraintes . . . . .	2
<b>3</b>	<b>Planification</b>	<b>2</b>
<b>4</b>	<b>Problèmes de satisfaction de contraintes</b>	<b>2</b>
<b>5</b>	<b>Extraction de connaissances</b>	<b>3</b>
<b>6</b>	<b>Difficultés rencontrées</b>	<b>3</b>
<b>7</b>	<b>Conclusion</b>	<b>3</b>

# 1 Introduction

Le projet **Le monde des blocs (BlocksWorld)** est un exemple classique utilisé en intelligence artificielle pour illustrer et tester des algorithmes de planification. L'objectif est de représenter et manipuler des configurations d'un monde constitué de blocs empilés en utilisant des variables et des contraintes en nous servant des algorithmes nécessaires pour chaque étapes, implémentés durant les Tp précédents.. Ce rapport décrit les étapes de modélisation, les algorithmes utilisés, ainsi que les résultats obtenus.

## 2 Modélisation du problème

### 2.1 Variables et contraintes

Pour représenter les configurations d'un monde de blocs, trois ensembles de variables ont été définis :

- **onb** : indique où se trouve un bloc  $b$  (sur un autre bloc  $b'$  ou sur la table dans une pile  $p$ ).
- **fixedb** : booléen indiquant si le bloc  $b$  est indéplaçable.
- **freep** : booléen indiquant si la pile  $p$  est libre.

Les contraintes binaires suivantes assurent la validité des configurations :

1. Deux blocs différents ne peuvent pas être sur le même support.
2. Si un bloc  $b$  est posé sur un bloc  $b'$ , alors  $b'$  est indéplaçable.
3. Si un bloc  $b$  est posé sur une pile  $p$ , alors  $p$  n'est pas libre.

Dans cette partie, nous avons créé deux classes **PileBlock** et **BlocksWord** qui nous instancient respectivement l'ensemble de variables et l'ensemble de contraintes ce qui nous permettra par la suite de tester les fonctions d'affichage de valeurs selon différentes configurations avant d'implémenter une configuration qui nous permet de vérifier notre contraintes de bases ou binaires, contraintes croissantes et régulières. Les deux dernières contraintes ont été implémenté dans les classes **ConfigReguliere** et **ConfigCroissante**.

## 3 Planification

En ce qui concerne cette partie , nous devons définir une liste d'actions applicables et nécessaires pour pouvoir déplacer nos blocs. Ensuite , il nous fallait tester nos actions et avec chacun des planificateurs implémentés durant les Tp précédents. Pour cette étape nous avons pu tester avec nos planificateurs mais aucun plan n'a pu être découvert. Tout cela a été fait dans une classe **BlocksWord** qui nous renvoie tout les actions possibles ensuite dans une main nommé **MainPlanification** qui était sensé nous rendre différents plans avec nos planificateurs. Aussi deux heuristiques ont été implémentés. Le premier **HeuristicBlocDeplaces** sert à trouver un bloc mal placé sur une pile quelconque et le deuxième **HeuristicMouvementsNecessaires** sert à définir les mouvements nécessaires pour partir d'une position initiale vers une position finale.

## 4 Problèmes de satisfaction de contraintes

Dans cette partie, nous avons défini tous les solveurs implémentés durant les premiers avec toutes leur variable afin de pouvoir calculer le temps de résolutions.

## 5 Extraction de connaissances

La classe DataMining destinée à l'analyse des configurations dans un monde des blocs. Cette classe permet de modéliser des relations entre les blocs, d'associer des états à des variables booléennes, et de générer une base de données . La classe MainDataming me permet d'initialiser tous les items d'un monde et de par la suite extraire les items fréquents et les règles d'associations.

## 6 Difficultés rencontrées

Durant le projet, la difficulté majeur a été de savoir comment utiliser les algorithmes déjà implémentés sur la résolution des problèmes. Bien qu'ayant compris le fonctionnement de ces algorithmes, pouvoir les utiliser sans pour autant les redéfinir étaient un challenge que nous pensons avoir réussi. Mais plusieurs points n'ont pas été résolu et le projet n'a pas été terminé. Nous n'avons pas utilisé la librairie donnée pour la représentation du monde de bloc car nous avons eu le temps de bien comprendre le fonctionnement de la librairie.

## 7 Conclusion

Ce projet a permis d'explorer la modélisation et la planification dans le monde des blocs. Des algorithmes efficaces ont été développés et testés. Les résultats obtenus ne montrent peut être pas le résultat attendu mais nous nous sommes tout de même efforcé de rendre un programme fonctionnel inachevé.