

Problème de Correspondance de Post

Chejjari Adnane Cordeiro Fonseca Loïc Leloup Loïc

Noubissi Kamgang Allan Perraudin Benjamin

Taheri Amirsaeed

Février 2022



1 Introduction

Dans le cadre du cours "INFO-F308 - Projet transdisciplinaire" et de ce Printemps des Sciences, nous nous sommes penchés sur le Problème de Correspondance de Post, ou PCP, un problème introduit par Emile Louis Post en 1946.

Soit un alphabet Σ de symboles fini et un ensemble de blocs $(w_i, x_i) : w_i, x_i \in \Sigma$ $i = 1, \dots, n$, existe-t-il une séquence d'entiers j_1, \dots, j_l telle que $w_{j_1} w_{j_2} \dots w_{j_l} = x_{j_1} x_{j_2} \dots x_{j_l}$?

Évidemment il arrive que pour une certaine instance (une liste de blocs), il n'existe aucune solution. Par exemple l'instance suivante : (10, 0), (100,011), (1,01), (11, 10) ne possède aucune solution (on peut le voir facilement car aucun bloc ne peut être placé en premier). Nous utilisons la notation en matrice pour représenter les instances.

$$\begin{pmatrix} 10 & 100 & 1 & 11 \\ 0 & 011 & 01 & 10 \end{pmatrix} \quad (1)$$

Ce qui rend PCP particulièrement intéressant est que ce problème est indécidable, c'est à dire qu'il n'existe pas d'algorithme qui peut déterminer avec certitude si cette solution existe.

Dans le cadre du Printemps des Sciences nous avons tâché de présenter cette problématique relativement complexe et abstraite, à une audience de différents âges et sans connaissances à priori sur le sujet. Pour rendre cela didactique et intuitif, nous avons développé un mini jeu en ligne jouable sur navigateur à l'adresse suivante : <https://pcp.digitalstudios.lu/>, permettant à une personne d'affronter différentes instances de PCP et de tenter de les résoudre en déplaçant des blocs.

2 Critères pour un bon jeu

La première question est de savoir quel type d'instance nous voulons que nos joueurs affrontent dans ce jeu qui fera pâlir les plus grands titres du monde du jeu vidéo.

Le but est de placer le joueur face à des instances de plus en plus difficiles à résoudre. Certains critères sont de mise pour considérer une instance comme "intéressante" pour le joueur, de plus ces critères peuvent permettre de favoriser nos chances de trouver une solution, et ainsi de générer plus d'instances.

3 niveaux de difficultés sont présents dans le jeu, facile intermédiaire et difficile. Ces trois niveaux diffèrent par la solution minimale des instances qui les composent. De 3 à 4 pour le niveau facile, intermédiaire de 4 à 7 et 8 à 16 pour le dernier. Exemple d'instance "facile":

$$\begin{pmatrix} 11 & 0 & 001 \\ 1 & 101 & 0 \end{pmatrix} \quad (2)$$

→ solution de longueur 3 : (1)(2)(1)

Pour que le joueur ne retombe pas sur une même instance, nous devons générer une base de donnée suffisamment remplie, et ainsi permettre au jeu une certaine durée de vie.

3 Méthodologie

Comme mentionné plus haut il n'existe pas d'algorithme nous permettant de résoudre le PCP, or afin de rendre le jeu correct, il nous fallait des instances dont on sait qu'elles sont résolubles. Pour cela il suffit de trouver au moins une solution pour une instance donnée.

L'approche retenue est celle d'un **backtracking itératif** à partir d'instances générées aléatoirement. L'idée est simplement de résoudre une instance pour une certaine profondeur (16 maximum) et de parcourir l'arbre des possibilités en plaçant les blocs à la suite des autres, jusqu'à trouver (ou non) une solution. Le backtracking a l'avantage d'être assez simple à implémenter (voir algorithme 1).

Algorithm 1 solve(depth, instance)

```
1: if isSolution() then
2:   return true
3: end if
4: for bloc in instance do
5:   if bloc can be push then
6:     push(bloc)
7:     if solve(depth - 1, instance) == true then
8:       return true
9:     end if
10:    pop(bloc)
11:   end if
12: end for
```

4 Résultats

Notre base de donnée finale se compose actuellement de 38687 instances pour le niveau 1 (solution max 4), 13533 pour le niveau 2 (max 7) et 3835 pour le niveau 3 (max 16). Ce chiffre varie puisque notre de base de donnée augmente en temps réel au fur et à mesure que des instances sont trouvées.