Jeu d'accessibilité lié à une marche de rotors sur un arbre non-orienté *

David Auger[1], Pierre Coucheney[1], Loric Duhazé[12]

[1] Université de Versailles Saint-Quentin, Laboratoire DAVID, Versailles, France
{david.auger,pierre.coucheney,loric.duhaze}@uvsq.fr

[2] Laboratoire de Recherche en informatique (LRI), Gif-sur-Yvette, France
loric.duhaze@lri.fr

Mots-clés : *Marche de Rotors, Jeu d'accessibilité.*

1 Introduction

Soit $G = (V, E)$ un graphe. Un *rotor de* $v \in V$ est un ordre cyclique sur les arcs sortants de v. Une configuration de rotors ρ est une application qui associe à chaque sommet de V un de ses arcs sortants. Si l'on place un jeton sur un sommet du graphe, on peut le déplacer selon la règle suivante : si le jeton est sur le sommet $v \in V$, il emprunte l'arc $\rho(v) = (v, w)$ pour arriver en w et ensuite, $\rho(v)$ est modifié selon l'ordre du rotor de v. Par analogie avec les marches aléatoires sur les graphes, la suite des configurations de rotors et des sommets visités par le jeton définit une *marche de rotors* sur G. Il a été montré que les propriétés statistiques des marches aléatoires sont en général bien approximées par leur équivalent déterministe qu'est la marche de rotor. Dans cet exposé, nous étudierons plutôt des problèmes de décisions associés à ces marches de rotors. Plus précisément, on considère des problèmes déterministes analogues à des problèmes d'accessibilité stochastiques (chaînes de Markov avec puits, processus de décision markoviens avec puits, jeux stochastiques simples) reposant sur des marches de rotors. La question est de déterminer, étant donnée une configuration de rotors et un sommet de départ pour le jeton, le puits dans lequel le jeton va finir. Ce problème peut se généraliser à un problème décisionnel si un ou des joueurs peuvent choisir la configuration initiale des rotors de certains sommets du graphe. Dans cet exposé nous étudierons le problème sur les arbres non orientés et nous montrerons qu'on peut le résoudre en temps linéaire dans ses variantes à 0, 1 ou 2 joueurs.

2 Problème ARRIVAL et problèmes de décision associés

Le problème de décision ARRIVAL [1] est le suivant :

> Etant donné une configuration de rotors et un sommet de départ pour le jeton, celui-ci va-t'il finir dans un puits donné ?

Ce problème appartient à la classe de complexité NP $\cap$ co-NP, sans pour autant que l'on ait exhibé d'algorithme polynomial pour le résoudre en général. Néanmoins, sur les graphes non-orientés, la marche de rotors atteint un puits en temps polynomial, ce qui constitue d'ailleurs le seul algorithme connu pour résoudre ce problème. Cet algorithme est quadratique sur les arbres non-orientés, alors que, dans le cas d'une chaîne avec deux puits p_1 et p_2 à ses extrémités, le problème se résout facilement grâce à la propriété énoncée dans le théorème suivant.

*Ces recherches ont été partiellement financées par le Labex DigiCosme (projet ANR11LABEX0045DIGICOSME) dirigé par l'ANR au sein du programme « Investissement d'Avenir» Idex ParisSaclay (ANR11IDEX000302)

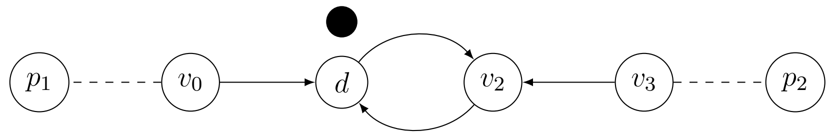

FIG. 1 – Les arcs représentent la configuration de rotors actuelle avec le jeton en d ; les arcs en pointillés sont ceux du graphe non-orienté sous-jacent

Théorème 1 *Soit $f_1(\rho)$ le nombre d'arcs orientés vers p_1 dans la configuration de rotor ρ, soit $\ell(p_1, d)$ le nombre d'arêtes entre p_1 et d, le jeton finit en p_1 si et seulement si $\ell(p_1, d) \leq f_1(\rho)$.*

A titre d'exemple, considèrons la chaîne non-orientée comme sur la Figure 1. La configuration des rotors est donnée par les arcs dessinés. Les seuls cycles induits par cette configuration sont d'ordre 2. Dans cet exemple, le jeton finit dans le puits p_1 après avoir visité les sommets $d, v_2, d, v_0, d, v_2, v_3, v_2, d, v_0, p_1$. Et dans ce cas $f_1(\rho)$ vaut 2 et $\ell(p_1, d)$ vaut 2. Le théorème 1 donne non seulement le puits de sortie de d mais également celui de tous les sommets de la chaîne : tous les v_i tels que $\ell(p_1, v_i) \leq f_1(\rho)$ terminent en p_1 alors que les autres vont en p_2.

Pour utiliser la propriété sur les chaînes dans le cas plus général des arbres non-orientés, nous définissons une opération de projection de la marche de rotors sur une sous-chaîne de l'arbre. Cette projection permet alors d'obtenir une caractérisation du puits de sortie pour un sommet v en fonction du nombre minimal d'arcs orientés vers v en venant d'un puits.

Le problème ARRIVAL peut être généralisé à un cadre décisionnel avec un ou deux joueurs qui peuvent choisir la configuration initiale des rotors d'un sous-ensemble des sommets du graphe. On fixe des sous-ensembles de puits disjoints pour chaque joueur et l'objectif de chacun est de faire en sorte que le jeton finisse dans un de ses puits. Dans ce cas, la complexité du calcul des stratégies optimales pour un graphe quelconque est NP-complet pour un joueur, et PSPACE-complet pour deux joueurs [2]. Pour les graphes non-orientés la question de la complexité reste ouverte. Pour ce qui concerne les arbres, nous montrons que notre algorithme pour le problème ARRIVAL s'étend naturellement à ce cadre décisionnel pour calculer les stratégies optimales en temps linéaire.

Références

[1] Dohrau, Jérôme and Gärtner, Bernd and Kohler, Manuel and Matoušek, Jiří and Welzl, Emo. *ARRIVAL : A zero-player graph game in NP $\cap$ coNP.* A journey through discrete mathematics 367–374. Springer, 2017.

[2] Fearnley, John and Gairing, Martin and Mnich, Matthias and Savani, Rahul. *Reachability switching games* arXiv preprint arXiv :1709.08991, 2017.