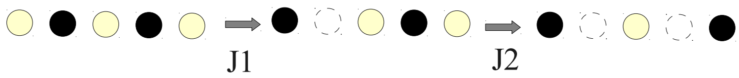
Un jeu combinatoire : Le Clobber

Définition et règle du jeu :

<u>Règle du Clobber</u> : Deux joueurs jouent sur une ligne où sont disposés des pions de couleur noire et blanche. Un joueur joue au choix sur un pion blanc ou noir et doit "manger" (c'est à dire déplacer le pion choisi sur l'emplacement du pion mangé et enlever ce dernier) un pion immédiatement voisin et de couleur différente. Les pions ne peuvent pas se déplacer sur des trous. Le premier joueur ne pouvant plus jouer perd la partie.

<u>Exemple</u> : 

: Le second joueur gagne

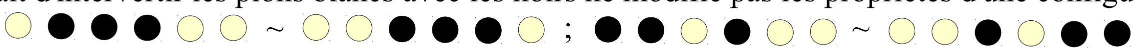
On appellera "configuration" toute association de pions et de trous, "option" un coup permettant de passer d'une configuration à une autre et "configuration fille de C", une configuration obtenue en jouant immédiatement sur C. On notera O (resp. X et _) les pions blancs (resp. pions noirs et trous)

Problématique et intérêt du sujet :

La problématique du sujet et d'essayer de caractériser les configurations perdantes et gagnantes. On dit qu'une configuration est gagnante si le premier joueur qui joue peut gagner contre toutes défenses du second joueur. Dans le cas contraire on dit qu'elle est perdante.

<u>Intérêt</u> : Le Clobber est un jeu introduit récemment dont on connait assez peu de résultats. L'intérêt d'étudier un jeu combinatoire est de trouver des résultats généraux sur des familles de jeux qui peuvent avoir des applications mathématiques. Par exemple l'étude d'un certain type de jeux combinatoire a permis à J. Comway d'introduire les nombres "surréels". Les chercheurs du Labri utilisent les jeux pour tester la sécurité de systèmes informatiques. Pour finir le Clobber est un jeu qui a été utilisé lors de compétitions universitaires.

Premiers résultats triviaux :

Une configuration a les mêmes propriétés si elle est lue de gauche à droite ou de droite à gauche.
Le fait d'intervertir les pions blancs avec les noirs ne modifie pas les propriétés d'une configuration.
Ex : 

Méthode mise en œuvre et programme de résolution :

Pour étudier ce jeu j'ai utilisé la fonction Grundy définie par :

$$G : \mathcal{Cf} \to \mathbb{N}, \quad C \mapsto G(C)$$

Avec $\mathcal{Cf}$ l'ensemble des configurations

Avec G(C) = 0 si C n'admet aucune option (si on ne peut pas jouer sur la configuration C) et
pour toute configuration C qui admet au moins une option et de configurations induites $(C_1, C_2, ...C_n)$
$G(C) = \text{mex}\ \{G(C_1), G(C_2), ..., G(C_n)\}$, avec mex la fonction qui a toute partie de $\mathbb{N}$ associe le plus petit entier qui n'y figure pas, autrement dit à toute partie A de $\mathbb{N}$, $mex(A) = min(\mathbb{N} \setminus A)$

<u>Théorèmes fondamentaux</u> :
- Une configuration C est perdante si et seulement si G(C)=0
- Pour toutes configurations C et C', on a $G(C\_C') = G(C) \oplus G(C')$

Avec $\oplus$ désignant la fonction xor définie de $\mathbb{N}^2$ dans $\mathbb{N}$ (Ex $7 \oplus 2 = [111]_2 \oplus [10]_2 = [101]_2 = 5$)

Exemple :

C1 G(C1) = 2

C • C2 G(C2) = 12 On en déduit que G(C) = 0 donc C est perdante

C3 G(C3) = 3

Grâce à cet outil mathématique j'ai pu réaliser un programme récursif qui permet de calculer le grundy de n'importe quelle configuration (mais au delà de 18 pions le calcul d'une configuration devient assez important). J'ai également fait un programme qui a enregistré le grundy de 250 millions de configurations différentes (jusqu'à 27 pions).

Quelques chiffres :
Sur les 250 millions premières configurations (sans trou) voici les résultats trouvés :

Grundy	0	1	2	3	4	5	6	7	8	9	10/11/12
%	19.184	19.139	13.841	13.819	9.449	9.054	5.519	4.619	3.556	1.503	< 0,3

Répartition des configurations (sans trou) perdantes en fonction du nombre de pion :

pions	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
%	50	25	37,5	37,5	37,5	21,5	23,4	23,4	27,1	21,4	22,2	20,8	22,3	19,9	20.0	18,5	18,9

Les $(ox)^n$ et $(ooxx)^n$:
J'ai fait un programme qui calcul le grundy des configurations du type $(ox)^n$ (c'est à dire de la forme oxoxox..). Le programme est basé sur le fait qu'il existe un ensemble fini de configurations contenant $(ox)^n$ et dépendant de n (de la forme $E_n=\{C_{1,n};C_{2,n};...;C_{p,n}\}$) tel que pour déterminer le grundy des configurations de E_{n+1} il suffise de connaitre le grundy des configurations de $E_0 \cup E_1 \cup ... \cup E_n$. Pour $(ox)^n$ on a :
$E_n = \{ (ox)^n ; (ox)^n x ; (ox)^n o ; o(ox)^n o ; o(ox)^n x ; o(ox)^n oo \}$
De même pour $(ooxx)^n$ on peut appliquer cette méthode mais cette fois avec $card(E_n) = 19$

<u>Résultats</u> : Pour les $(ox)^n$ je suis allé jusqu'à n=1500. Je n'ai pas trouvé de périodicité dans le grundy malgré une petite ressemblance qui peut me faire penser qu'il existe (peut-être) une période de 23 à partir d'un rang N (plus grand que 1500). J'ai aussi remarqué que le nombre de configuration perdante est faible (environ 1,67%) et qu'au delà n=390 il n'y a plus aucune configuration perdante. Je remarque aussi que globalement le grundy des configurations croit avec n. Pour conclure sur les $(ox)^n$, je n'ai pas pu caractériser les configurations perdante et j'ai pu uniquement faire des conjectures.

Pour les $(ooxx)^n$ j'ai remarqué que le grundy reste constant égal à 3 (à partir de n=2). Cela s'explique par le fait que pour tout n le grundy des configurations $C_{k,n}$ de E_n reste constant lorsque n varie. J'ai donc pu démontrer par récurrence que la configuration $(ooxx)^n$ est gagnante pour $n \geq 2$. De plus, pour $n \geq 3$, le premier coup gagnant est le suivant :
Tout d'abord il existe p et q tels que $(ooxx)^n = (ooxx)^p ooxxooxx(ooxx)^q$ avec $p \geq 1$ et p+q+2=n
Pour pouvoir gagner il faut jouer ainsi $(ooxx)^n \rightarrow (ooxx)^p oox\_xoxx(ooxx)^q$

Jouer contre l'ordinateur :
J'ai réfléchi sur une intelligence artificielle capable de jouer contre un humain . Cette intelligence artificielle est assez simple. A partir d'une configuration quelconque, il détermine toutes les options possibles et deux cas se présentent :
- S'il existe une option qui mène à une configuration perdante alors l'ordinateur joue dessus et gagne
- Si toutes les options mènent à des configurations gagnantes alors on choisit la configuration qui a le moins d'option menant à une configuration perdante. Au coup d'après, le joueur humain sera sur une configuration gagnante mais il aura le moins de coups gagnants (les coups qui peuvent le faire gagner sont réduits et il a donc plus de chance de perdre)

<u>Contact</u> : Eric SOPENA, chercheur au LaBRI (Laboratoire Bordelais de Recherche en Informatique)