

Rapport de TIPE

Le problème de Berlekamp : résolution théorique, résolution pratique

1 Préambule

L'optimisation est une branche des Mathématiques qui trouve de très nombreuses applications, tant en Physique qu'en Chimie ou encore dans l'Industrie. Comme expliqué dans le MCOT fourni en Mars, résoudre un tel problème nécessite de modéliser judicieusement les objets étudiés (mathématiquement puis informatiquement), d'exploiter les propriétés que ces modélisations mettent à notre disposition et d'optimiser les algorithmes de résolution, afin de garantir une complexité temporelle suffisamment faible pour permettre une résolution effective du problème considéré. En s'attachant à résoudre le *problème de Berlekamp*, ce TIPE s'inscrit précisément dans cette démarche.

2 Introduction : principe du *Berlekamp's Switching Game*

Considérons un tableau d'ampoules de taille $n \times n$ dans lequel ampoule est soit éteinte, soit allumée, autour duquel s'affrontent 2 joueurs. Le 1^{er} doit imposer une configuration initiale (nombre d'ampoules allumées et positionnement sur le tableau), et le 2^{ème} n'a accès qu'à des interrupteurs situés en bout de lignes et de colonnes, dont le basculement a pour effet d'intervenir les états des ampoules sur la ligne ou la colonne correspondante (les ampoules allumées s'éteignent, et réciproquement). Il peut manipuler ces interrupteurs à l'infini, dans l'ordre qu'il souhaite.

L'objectif du 2nd joueur est d'amener le tableau dans une situation où le nombre d'ampoules allumées est minimal. Le 1^{er} joueur cherche donc à imposer une configuration initiale qui maximise le nombre minimal d'ampoules allumées que peut obtenir son adversaire.

La taille n du tableau étant fixée, l'objectif de cette étude est de déterminer les configurations initiales optimales pour le 1^{er} joueur. J'effectuerai pour cela une étude mathématique fournissant une résolution théorique exacte, puis une étude informatique permettant la résolution effective de ce problème (complexité temporelle acceptable).

3 Déroulé opérationnel de l'étude

Pour tenter de résoudre analytiquement un tel problème d'optimisation, il a tout d'abord fallu déterminer une modélisation mathématique adéquate. De manière intuitive, j'ai choisi de modéliser les tableaux d'ampoules du *problème de Berlekamp* par des matrices de $\mathcal{M}_n(\mathbb{F}_2)$ (où $\mathbb{F}_2 = \{0, 1\}$), dans lesquelles un coefficient 0 en position (i, j) correspond à une ampoule éteinte, et un coefficient 1 à une ampoule allumée. De plus, en définissant le *poids* d'une matrice de $\mathcal{M}_n(\mathbb{F}_2)$ comme le nombre de coefficients 1 qu'elle comporte, on obtient avec cette modélisation que le poids d'une matrice est exactement égal au nombre d'ampoules allumées dans le tableau correspondant. L'intérêt d'une telle définition est de travailler avec une donnée numérique (le poids d'une matrice de $\mathcal{M}_n(\mathbb{F}_2)$), qu'on

cherche à maximiser ou minimiser en jouant sur un certain nombre de paramètres.

J'ai alors pu mettre au jour certaines relations entre les objets (classes d'équivalence pour une relation bien choisie...) et certaines structures algébriques (espaces vectoriels et sous-espaces vectoriels, groupe quotient...), qui m'ont permis de réduire cette étude à un problème de théorie des codes. J'ai alors pu adapter une méthode classique de « décodage » (« décodage au maximum de vraisemblance par tableau standard ») afin de résoudre le *problème de Berlekamp* d'un point de vue théorique.

Toutefois, si l'algorithme de résolution obtenu met en évidence d'intéressantes propriétés algébriques, il n'en présente pas moins une très mauvaise complexité temporelle et des besoins en mémoire excessifs. Cela rend son utilisation impossible pour la résolution effective du problème, même pour des tableaux de taille réduite. J'ai par conséquent mené une seconde étude, afin de déterminer un compromis algorithmique permettant une résolution dans un laps de temps raisonnable et avec une fiabilité convenable.

Un travail préalable à cette seconde étude a été de choisir une nouvelle modélisation des objets du problème, cette fois-ci informatique. Afin de minimiser l'occupation de la mémoire vive de l'ordinateur, j'ai choisi une modélisation faisant intervenir des entiers naturels, dont la décomposition en base 2 permet, par unicité, de retrouver la matrice correspondante. Ce choix présente un double intérêt : la faible place qu'occupent les entiers en mémoire, et la facilité d'implémentation d'opérations sur les décompositions en base 2, dans le langage Python.

J'ai alors adapté à la résolution du *problème de Berlekamp* l'algorithme dit « du recuit simulé » : une méthode algorithmique récursive qui permet, au moyen d'un procédé inspiré de la physique statistique, de détecter les extrema globaux d'une fonction numérique sur un certain ensemble. J'ai donc implémenté puis optimisé une version de cet algorithme en Python, afin de modéliser l'action du 2nd joueur dans le *Berlekamp's Switching Game* : étant donnée une configuration initiale imposée par le 1^{er} joueur, l'algorithme recherche les configurations accessibles de poids minimal. J'ai ensuite écrit un second algorithme, qui exploite le premier afin de déterminer les poids minimaux obtenus pour un grand nombre de configurations initiales choisies aléatoirement dans l'ensemble des configurations possibles. On retiendra les configurations initiales fournissant le poids minimal le plus élevé.

C'est grâce à ce second algorithme que j'ai effectivement pu résoudre le *problème de Berlekamp* pour différentes tailles de tableaux.

4 Conclusion générale

Mon choix de réaliser deux études du *Berlekamp's Switching Game*, l'une mathématique et l'autre informatique, m'a donc permis une approche particulièrement complète. J'ai ainsi pu mettre en évidence d'intéressantes propriétés algébriques, puis implémenter un algorithme de résolution et l'optimiser, afin de pouvoir résoudre effectivement le *problème de Berlekamp* dans un grand nombre de cas.

5 Bibliographie additionnelle et contacts

Aucune information à rajouter par rapport à celles fournies dans le MCOT.