Pavage de Penrose et Problème des tours d'Hanoï

Alexandre Debant & Louis Cohen

5/10/2014

Résumé

Comment paver le plan est un problème mathématique assez répandu et nous allons étudier la méthode proposée par Roger Penrose. Pour cela nous présenterons deux pavages puis nous en étudierons quelques propriétés de périodicité. Dans un second temps nous présenterons une recherche sur les tours d'Hanoï par l'étude de deux algorithmes, en particulier leur complexité.

Mots-clés. Penrose, pavage, quasi-périodique, Hanoï, récursif, itératif, complexité, optimalité.

1 Pavage de Penrose : généralités et exemples

1.1 Principe de base

Le pavage de Penrose est un type de pavage découvert dans les années 1970 par Roger Penrose recherchant une distraction intellectuelle. Ce dernier chercha à paver le plan avec seulement deux types de figures : le triangle d'or obtus et le triangle d'or aigu. Ce sont des triangles isocèles ayant pour longueurs de cotés 1 et $\varphi = \frac{1+\sqrt{5}}{2}$. Intuitivement nous serions tentés de vouloir paver le plan de manière itérative c'est-à-dire de chercher à dessiner les triangles les uns à coté des autres en s'astreignant à ne pas "créer des blancs". Mais si nous essayons de faire cela, nous nous rendons vite compte que cette méthode devient rapidement très compliquée lorsque le nombre de triangles déjà posés devient important.

Solution proposée par Penrose : effectuer un découpage puis agrandir pour retrouver l'échelle d'origine : chaque triangle de base va être découpé de telle sorte qu'après le découpage, les nouvelles figures obtenues soient des figures de base (à réduction près). Par exemple, dans FIGURE 1 nous pouvons voir une possibilité pour découper les deux triangles d'or ; l'obtus est scindé en 1 obtus et 1 aigu et l'aigu en 1 obtus et 2 aigus.

Comme nous pouvons le voir dans FIGURE 2, le rendu esthétique du pavage présenté ci-dessus est assez limité. Aucune organisation ne semble apparaitre.

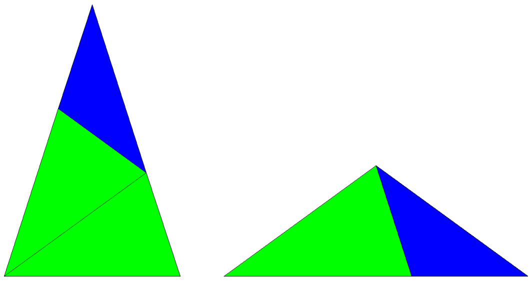

FIGURE 1 – Découpe des triangles de base

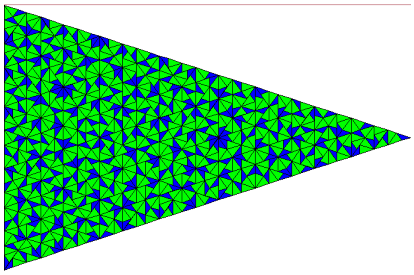

FIGURE 2 – Exemple d'une découpe (7 itérations)

1.2 Un pavage plus esthétique

Pour obtenir un pavage "plus joli" une des solutions est de modifier les figures de base. Par exemple, nous pouvons choisir d'utiliser ce que l'on va appeler un cerf-volant et une fléchette. Leur découpe, comme le montre la FIGURE 3, respecte bien les règles de découpe énoncées ci-avant.

Comme nous pouvons le voir sur FIGURE 3, la découpe des cerfs-volants et des fléchettes génère des demi-fléchettes et donc il faut s'assurer que les demi-fléchettes se dessinent de façon a bien se compléter.

Notre découpage est correct :

FIGURE 3 représente la première division du cerf-volant. Pour justifier que le pavage est correct, il faut montrer que les demi-fléchettes et les demi-cerfs-volants se dessinent de telle façon que les figures n'ayant pas de côté sur le bord du pavage soient entières.
Sur FIGURE 4 nous pouvons voir que la seconde itération fait apparaitre des demi-cerfs-volant sur le bords mais que les demi-fléchettes à l'intérieur se recollent correctement de façon à former des fléchettes entières. Au bout de la troisième itération, nous pouvons voir que toutes les figures intérieures sont entières et que les demi-fléchettes sont présentent sur le bord dans la configuration présente lors de la première découpe. Par conséquent, par itération, nous avons que les demi-figures seront exclusivement sur le bord du pavage à la fin de n'importe quelle découpe.

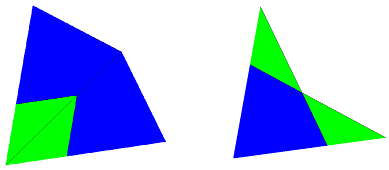

FIGURE 3 – *
Cerf volant

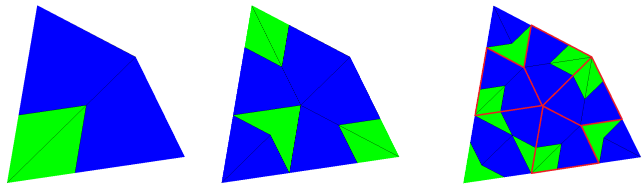

FIGURE 4 – Fléchette

1.3 Quelques propriétés de périodicité

1.3.1 Le pavage n'est pas périodique

Maintenant que nous connaissons un peu plus précisément les pavages de Penrose, nous pouvons nous intéresser à certaines de ses propriétés. Par exemple, lorsque nous considérons des pavages, un élément souvent étudié est la périodicité. En effet, cela permet de caractériser le pavage.

Un pavage non-périodique

Dans le cas des pavages de Penrose, nous pouvons montrer qu'ils ne sont pas périodiques. Pour ce faire, il suffit de considérer les suites définies ci-après et de remarquer que ces dernières permettent de montrer (voir ci-après) que le rapport $\frac{\text{nombre de cerfs volants}}{\text{nombre de fléchettes}}$ après chaque division tend vers un irrationnel . En effet, un pavage périodique est engendré par un unique motif $\mathcal{A}_0$ ce qui a pour conséquence que le rapport $\frac{\text{nombre de cerfs volants}}{\text{nombre de fléchettes}}$ est constant et a pour valeur $\frac{\text{nombre de cerfs volants dans } \mathcal{A}}{\text{nombre de fléchettes dans } \mathcal{A}} \in \mathbb{Q}$.

<u>**Définition des suites :**</u>

— $(c_n)_{n\in\mathbb{N}} =$ nombre de cerfs volants à la n^{ieme} itération
— $(f_n)_{n\in\mathbb{N}} =$ nombre de fléchettes à la n^{ieme} itération
— $(x_n)_{n\in\mathbb{N}} = \begin{cases} f_p & \text{si } n = 2p \\ c_p & \text{si } n = 2p+1 \end{cases}$
— On peut montrer que : $\forall n \in \mathbb{N}, x_{n+2} = x_{n+1} + x_n$
— Donc $\lim\limits_{n\to+\infty} \frac{x_{n+1}}{x_n} = \lim\limits_{n\to+\infty} \frac{c_n}{f_n} = \varphi$

1.3.2 Mais il est quasi-périodique

Les pavages de Penrose sont donc non-périodiques. Cependant, lorsque nous regardons certains types de ces pavages, alors nous pouvons remarquer "une sorte de régularité" ; c'est grâce à cela que certains peuvent donner des figures ayant un aspect esthétique considéré comme beau. Par conséquent nous pouvons nous demander : d'où cette "régularité" vient-elle ? La réponse est que les pavages de Penrose sont quasi-périodiques.

<u>**Définition :**</u> Un pavage est dit quasi-périodique si lorsque nous avons un motif $\mathcal{A}$ qui apparait dans le pavage alors il y apparait une infinité de fois.

Montrons que le pavage cerf-volant/fléchette est quasi-périodique

Considérons $\mathcal{A}$ un motif apparaissant dans le pavage.
Nous pouvons noter $n \geq 0$ le nombre de découpages qu'il faut (à partir de la figure initiale : cerf-volant) pour faire apparaitre $\mathcal{A}$ pour la première fois.
Notons $x_k \quad k \geq 1$ le nombre d'apparitions du motif $\mathcal{A}$ après $(n+2) \times k$ découpages.
Étant donné qu'après deux découpages le motif A contient au moins deux cerfs volants, après $n+2$ découpages le pavage contient au moins deux fois le motif $\mathcal{A}$. Finalement, après $k \times n + 2$ découpages le pavage contient au moins k fois le motif $\mathcal{A}$.
Comme le pavage du plan s'obtient en découpant à l'infini le motif de base, le nombre d'apparitions de $\mathcal{A}$ est alors $\lim\limits_{k \to +\infty} x_k = +\infty$
<u>Conclusion :</u> le pavage est quasi-périodique.

Remarque. *Pourquoi devons-nous considérer $n+2$ découpages et pas n ? La raison vient du fait que si le motif $\mathcal{A}$ ne contient que des fléchettes alors il faut au moins effectuer une découpe pour faire apparaitre des cerfs volants. Il y en a alors autant qu'il y avait de fléchettes. Par conséquent, si le motif $\mathcal{A}$ est une unique fléchette alors une unique découpe ne fait apparaitre qu'un seul cerf volant ; il faut donc en effectuer deux.*

1.4 Bilan

Si nous voulons faire un bilan de cette étude des pavages de Penrose, nous pouvons dire que bien que ce soit des pavages relativement simples à mettre en œuvre (une simple récursion suffit) ils permettent de créer des pavages esthétiquement jolis et possédant des propriétés mathématiques particulières et non triviales (non périodiques mais quasi-périodiques). De plus, grâce à cette quasi-périodicité, ils permettent de modéliser de façon correcte la structure de certains minéraux appelés quasi-cristaux par exemple.

2 Tours d'Hanoï

2.1 Principe Général

2.1.1 Règles et notations

Le jeu des tours d'Hanoï est un jeu de réflexion de principe relativement simple. Le jeu se présente sous la forme d'un plateau à trois piquets. Sur le premier piquet est posé au départ n disques de tailles décroissantes (voir figure 1 avec n=4).
Le but du jeu est de déplacer ces n disques du piquet de départ vers un piquet d'arrivée. Pour se faire le joueur doit respecter deux règles :

- Ne déplacer qu'un seul disque à la fois
- Ne jamais poser un disque plus gros sur un disque plus petit. (Le déplacement de la figure 2 est autorisé tandis que celui de la figure 3 ne l'est pas)

Terminologie :

- On notera n le nombre de disque à déplacer.
- On appellera les disques en fonction de leur taille ; ainsi le disque de plus petite taille sera le disque 1 tandis que le plus gros (si il y a n disques) sera le disque n.

2.1.2 Intérêt

L'intérêt de ce jeu est de tester certaines capacités du joueur (particulièrement en médecine : trouble de concentration, manque d'inhibition, mémoire immédiate ...). Mais si le jeu semble relativement difficile à résoudre pour un esprit humain, nous verrons par la suite que le principe de récursivité le rend très simple à résoudre pour un ordinateur.

2.2 Les deux implémentations

2.2.1 récursivité

Le principe de la méthode récursive est de se ramener au problème le plus simple possible : déplacer un disque déja au sommet d'un piquet vers un autre. Ainsi la résolution fonctionne par étape :

— Si je n'ai qu'un disque à déplacer de A vers B je le déplace
— Si j'ai n+1 disques à déplacer de A vers C, je déplace d'abord les n disques du haut (par rappel de la fonction) de A vers B (figure 4) puis je déplace le disque n+1 de A vers C (figure 5) puis je redéplace les n disques de B vers C (toujours par récursion) (figure 6)

Remarque. *Le problème de cette méthode est que malgré son efficacité de résolution pour un ordinateur elle ne permet aucunement à un humain de résoudre le problème.*

2.2.2 Itération

Au contraire la méthode itérative peut être mise en pratique par l'humain. Le principe est de choisir chacun de ses coups uniquement en se rappelant d'informations sur les deux coups précédents.
De plus après avoir joué un certain nombre de coups on se souviendra de :

— La taille du disque déplacé précédemment (plus exactement : si le disque déplacé était le disque 1 ou un autre : le disque moyen)
— La provenance du disque 1 lors son dernier déplacement (pour éviter de revenir dessus au déplacement suivant).

Remarque. *Il est tout-à-fait possible pour une personne réelle de se souvenir de ces informations.*

Avant de détailler l'algorithme commençons par remarquer que le nombre de coups possible est très restreint, 3 coups seulement :

— Déplacer le disque 1 (qui est toujours en haut d'un piquet vu qu'il ne peut pas être recouvert). On peut donc le déplacer vers n'importe lequel des deux autres piquets : deux déplacements possibles.
— A part au départ où à la fin (quand tous les disques sont sur un seul disque) il y a toujours un disque en haut d'un (ou des deux autres) piquet.

Le pseudo code en lui-même :
Tant que tous les disques ne sont pas sur un autre piquet que celui de départ faire :

Si le petit disque n'a pas été déplacé au coup précédemment :
On déplace le disque 1 vers le piquet d'où il ne venait pas précédemment et on note le nouveau piquet d'où provient le disque 1 (par exemple si le dernier coup du disque 1 a été $A \rightarrow B$ alors on fait le déplacement $A \rightarrow C$)
Sinon on déplace le disque moyen avec le seul coup autorisé.
Fin si
Fin tant que

Remarque. *Nous ne démontrerons pas la finitude de cet algorithme, cependant la démonstration serait analogue à celle de la complexité de la partie suivante.*

Le problème qui s'est naturellement posé lors de la suite de mes recherches était la comparaison de ces deux méthodes. Mais l'expérience pratique a montré que les deux méthodes renvoyaient exactement la même série de coups. La troisième partie de ce rapport est donc dédiée à une démonstration, via la complexité, de cette équivalence de méthode.

2.3 Complexité

2.3.1 Définitions

Définitions :
- Algorithme optimal : algorithme ayant la plus basse complexité parmi ceux résolvant le problème.
- Complexité d'Hanoï à n disques (notée C_n) : Nombre de déplacements de disque.

2.3.2 Minoration de la complexité minimale et complexité de la méthode récursive

Pour déplacer (n+1) disques de $A \rightarrow C$ il faut obligatoirement déplacer les n premiers de $A \rightarrow B$ puis déplacer le disque n+1 de $A \rightarrow C$ puis redéplacer les n premiers disques de $B \rightarrow C$. Cela nous donne une inégalité récursive sur la complexité de la résolution optimale :
$C_{n+1} \geq 2 \times C_n + 1$
Par récurrence vu que $C_1 = 1$ on obtient l'inégalité suivante :
$C_n \geq \sum_{k=1,n-1} C_k \geq 2^n - 1$
Cette inégalité est en réalité une égalité vu que l'algorithme récursif a une complexité de C_n égale a $2^n - 1$ (d'après sa construction même les inégalités deviennent des égalités).

Ainsi l'algorithme récursif est un algorithme optimal pour le problème. Montrons maintenant que l'algorithme itératif est lui aussi optimal.

2.3.3 complexité de la méthode itérative et unicité de la solution optimale

La démonstration de l'optimalité de l'algorithme itératif ne reposera pas sur le même principe que la démonstration précédente.

Nous allons montrer que la liste de coups explicitement décrit par cet algorithme est obligatoire pour avoir un algorithme optimal.

Nous allons étudier les trois coups possibles (décris dans la partie précédente) dans les deux configurations possibles de l'algorithme et montrer qu'un seul coup n'est possible dans un algorithme optimal.

- Premier cas : le disque 1 a été déplacé, son dernier déplacement était $A \rightarrow B$
  - Déplacer le disque 1 de $B \rightarrow A$, la série de coups n'est pas optimale : on peut ne pas jouer $A \rightarrow B \rightarrow A$ et arriver au même résultat en économisant 2 coups.
  - Déplacer le disque 1 de $B \rightarrow C$, la série de coup n'est pas optimale : on peut remplacer $A \rightarrow B \rightarrow C$ par $A \rightarrow C$ et arriver au même résultat en économisant 1 coup.
  - Il faut donc jouer le déplacement du disque moyen.
- Deuxième cas : le disque 1 n'a pas été déplacé au coup précédent, son dernier déplacement était $A \rightarrow B$
  - Déplacer le disque moyen, vu que le disque moyen ne change pas apres un déplacement de celui ci (un piquet est occupé par le disque 1, le piquet qu'on libère est occupé par un disque plus grand que le moyen ou reste vide, et le moyen est sur le dernier piquet), cela induirait de le redéplacer une nouvelle fois ce qui, par le même raisonnement que dans le cas précédent n'est pas optimal.
  - On sait donc maintenant que l'alternance du déplacement du petit et du moyen est obligatoire. La démonstration du fait que déplacer le disque 1 de $B \rightarrow A$ n'est pas optimale est un peu plus complexe. On montrera plutôt que l'alternance des piquets A B et C lors des déplacements du disque 1 est obligatoire. Pour cela on raisonnera par récurrence sur la propriété P(N) = si n (nombre de disques à déplacer) impair il y a alternance $A \rightarrow C \rightarrow B \rightarrow A \rightarrow C \rightarrow B$ lors des déplacements du disque 1 si pair alternance $A \rightarrow B \rightarrow C \rightarrow A \rightarrow B \rightarrow C$ (pour arriver sur le piquet C) :
    - Si n=1 déplacement $A \rightarrow C$ résout le problème. Propriété vérifiée.
    - Si n+1>1 (on pose ici n pair, la démonstration pour n impair repose sur exactement le même principe) alors en supposant P(n) vraie. Avec toujours le même principe : pour déplacer (n+1) disques de $A \rightarrow C$ il faut obligatoirement déplacer les n premiers de $A \rightarrow B$, l'enchaînement de déplacements du disque 1 sera (vu que n est pair et que l'on va de $A \rightarrow B$ et d'après P(n)) $A \rightarrow C \rightarrow B \rightarrow \ldots A \rightarrow C \rightarrow B$, puis on déplace le disque n+1 de $A \rightarrow C$, puis enfin redéplacer les n premiers disques de $B \rightarrow C$, l'enchaînement de déplacements du disque 1 sera (vu que n est pair et que l'on va de $A \rightarrow B$ et d'après P(n)) $B \rightarrow A \rightarrow C \rightarrow \ldots B \rightarrow A \rightarrow C \rightarrow$.

      Finalement les déplacements du disque 1 ont été $A \rightarrow C \rightarrow B \rightarrow \ldots B \rightarrow A \rightarrow C \rightarrow \ldots A \rightarrow C$ d'où P(n+1). La propriété P(n) est donc vérifiée pour tout $n \in \mathbb{N}$

      Ainsi on a prouvé que les déplacements du disque 1 doivent alterner de piquets.

On voit grâce à cette démonstration que pour le premier coup, si on veut déplacer la pile de n disques de $A \rightarrow C$, il faut jouer le coup $A \rightarrow C$ si n impair et $A \rightarrow B$ si n pair.

Ainsi on a montré que non seulement la méthode itérative est une méthode optimale mais en plus que toute méthode optimale renvoie la même suite de coups que la méthode itérative, ce résultat étant vrai en particulier pour la méthode récursive.

3 Conclusion

Ce projet m'a permis, je pense, d'un peu mieux appréhender l'idée de la recherche. En effet, la direction qu'a pris ce projet n'était pas du tout celle que je pensais lui donner au départ, ceci dû aux résultats obtenus au fil du projet.

A Annexe

A.1 auto-évaluation

Points forts

— Respect des délais et organisation du temps : les objectifs de temps que je m'étais fixés ont été respectés. Je n'ai pas eu de travail de dernière minute à produire en urgence.
— Dernière partie du projet relativement personnelle et tentative d'approche personnelle du sujet.
— Bonne relation de groupe pour le rendu du projet en commun (même si ce n'était pas l'aspect le plus développé du projet).
— Approche différente du problème des pavages (plus mathématique) en comparaison avec ce qu'ont pu faire les autres binômes
— Implémentation d'un second pavages avec de nouvelles figures de base

Points faibles

— Énorme perte de temps lors de l'implémentation due à un manque de pratique
— Le premier point a entrainé un manque de profondeur du projet effectué à mon gout, par manque de temps pour traiter d'autres aspects.
— Manque de clarté dans certains points du rapport.
— Mauvaise anticipation des difficultés que le projet allait créer.
— L'implémentation graphique reste limitée
— La présentation reste sur des éléments relativement basiques

Améliorations possibles

— Le manque d'aisance en programmation pure, devrait se résorber au cours de l'année (j'y travaille).
— Meilleure appréhension des problèmes qu'un projet pose.
— Meilleure organisation du temps entre recherches et développement de mon propre projet.
— Développer l'implémentation de pavages "plus esthétiques"
— Essayer de comprendre théoriquement pourquoi certaines découpes sembles plus régulières que d'autres

Difficultés à anticiper

— Clarté des présentations
— Réel travail avec une autre personne sur un même code
— Retard en aisance de programmation (qui ne sera pas totalement réglé lors du prochain projet)
— Apprendre à utiliser, se servir de façon optimale, la bibliothèque graphique limitée de OCaml
— Oser étudier des propriétés moins classiques quitte à ne pas aboutir entièrement à ce que nous voulions

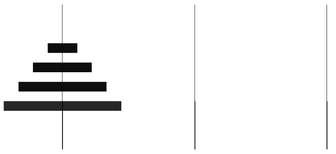

FIGURE 5 – Départ du jeu d'Hanoï

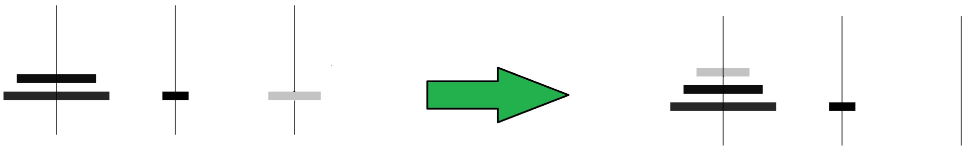

FIGURE 6 – exemple de déplacement autorisé

A.2 Les dessins ! !

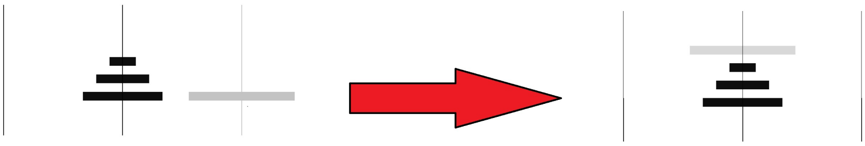

FIGURE 7 – exemple de déplacement non-autorisé

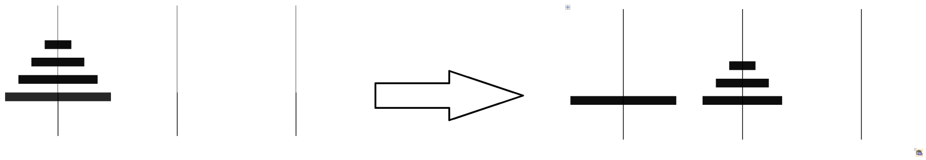

FIGURE 8 – méthode récursive : premier déplacement

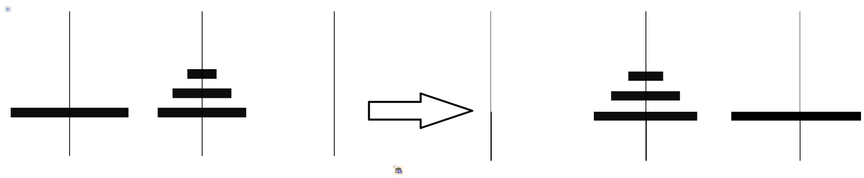

FIGURE 9 – méthode récursive : second déplacement

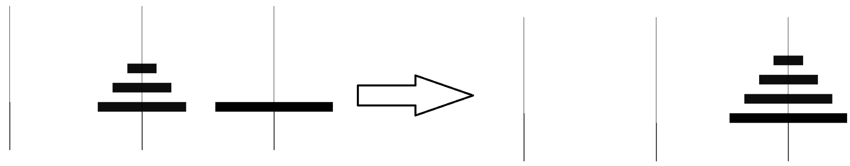

FIGURE 10 – méthode récursive : troisième déplacement