HAL
archives-ouvertes.fr

Algorithmes gloutons avec la classe

Karim Zayana, Pierre Michalak, Clément Beauseigneur, Hélène Tanoh

▶ **To cite this version:**

Karim Zayana, Pierre Michalak, Clément Beauseigneur, Hélène Tanoh. Algorithmes gloutons avec la classe. CultureMath, ENS, 2019. hal-02931346

HAL Id: hal-02931346
https://hal.telecom-paris.fr/hal-02931346

Submitted on 6 Sep 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Algorithmes gloutons avec la classe

Karim ZAYANA[1,2], Pierre MICHALAK[3], Clément BEAUSEIGNEUR[4] et Hélène TANOH[3]

[1] LTCI, Télécom Paris, Institut Polytechnique de Paris.
[2] IGEN, Ministère de l'Éducation nationale.
[3] IAIPR, Ministère de l'Éducation nationale.
[4] Google France, Paris.

Résumé

Mentionnés dans les nouveaux programmes de Numérique et Science Informatique du cycle terminal des lycées [1], les algorithmes gloutons offrent une solution pratique, mais pas toujours optimale, à de nombreux problèmes arithmétiques. Nous en donnons ici deux exemples qu'il est loisible de mener en classe, dès celle de première, sous forme d'activité ou d'approfondissement et qui peuvent ensuite fournir de la matière au grand oral du baccalauréat. Sans être élémentaire, le bagage mathématique que nous mobiliserons demeurera toutefois raisonnable [4, 7].

Mots-clés

Algorithmes gloutons, fractions égyptiennes et algorithme de Fibonacci, coefficients de Bézout, algorithme du monnayeur, suite de Fibonacci, one-point theorem.

1 Introduction

Pour un algorithme glouton, tout ce qui est pris n'est plus à prendre : une stratégie à courte vue car en se ruant sur les entrées, on n'a parfois plus de place au dessert. Ainsi résumée (et simplifiée), la logique du glouton - *greedy algorithm* en anglais - ne consiste cependant pas « à goûter à tout » [9, 2]. En effet, bien que son caractère vorace puisse le laisser supposer, le glouton n'est pas un algorithme de recherche par force brute. Il ne teste pas exhaustivement tous les possibles. Mais il maximise, à la manière d'un gradient discret et sans vision d'ensemble, donc sans revenir dessus, chaque prise de décision. Nous le mettons ici en œuvre dans deux contextes simples, la décomposition en fractions égyptiennes et le problème du rendu de monnaie, et nous en discutons les performances.

2 Fractions égyptiennes

2.1 Une réécriture élégante des fractions rationnelles

Définition 1 (fraction unitaire, dite aussi égyptienne)**.** *Nous appellerons ici fraction égyptienne tout rationnel positif de la forme* $\frac{1}{n}$ *où* $n \in \mathbb{N}^*$*, comme* $1, \frac{1}{2}, \frac{1}{3} \ldots$.

Le rôle de ces fractions est remarquable dans le

Théorème 1 (décomposition)**.** *Un rationnel positif* x *(non nul) peut toujours s'écrire comme somme de fractions égyptiennes deux à deux distinctes.*

Avant d'en démontrer le résultat (quand $x < 1$ pour commencer), vérifions-le sur quelques exemples. Ainsi, $\frac{3}{2} = 1 + \frac{1}{2}$. Et $\frac{2}{3} = \frac{1}{2} + \frac{1}{6}$, ou, plus long, $\frac{2}{3} = \frac{1}{3} + \frac{1}{6} + \frac{1}{9} + \frac{1}{18}$. On peinerait davantage avec le nombre $\frac{5}{2} = 1 + 1 + \frac{1}{2}$, qu'il faut débarrasser de ses doublons. La formule des « matching pairs »,

$$\frac{1}{k} = \frac{1}{k+1} + \frac{1}{k(k+1)} \tag{1}$$

peut nous y aider. L'une des fractions $1 = \frac{1}{1}$ se ramifie en $\frac{1}{2} + \frac{1}{2}$, l'une des fractions $\frac{1}{2}$ en $\frac{1}{3} + \frac{1}{6}$, etc. De fil en aiguille et après remise en ordre, $\frac{5}{2} = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{6} + \frac{1}{7} + \frac{1}{12} + \frac{1}{42}$ ainsi que l'illustre l'arborescence de la figure 1. Mais en éliminant une redondance on en crée parfois d'autres. À cette entreprise artisanale et hasardeuse, on préfère la ligne plus stricte qu'apporte la preuve suivante, valable tant que $x = \frac{p}{q} \in]0, 1[$.

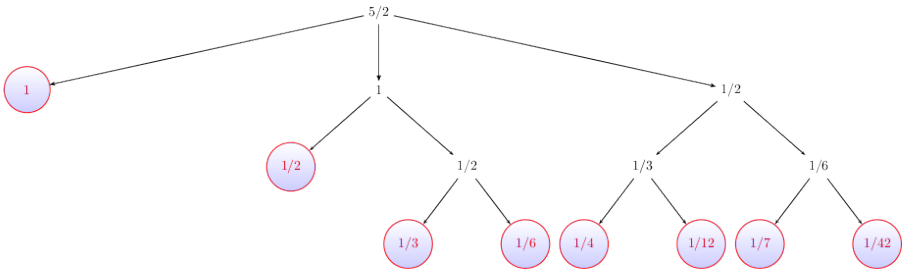

FIGURE 1 – Décomposition « artisanale » de $\frac{5}{2}$.

Démonstration. Raisonnons par récurrence forte sur le numérateur p de la fraction à décomposer. Quand $p = 1$, la propriété est acquise. Soit maintenant $x = \frac{p}{q}$ où $p < q$. Il serait vain de scinder x en $\frac{1}{q} + \frac{p-1}{q}$ car la deuxième fraction porte potentiellement $\frac{1}{q}$ elle aussi dans son écriture au rang $p-1$. Commençons plutôt par serrer x au plus près sur sa gauche grâce à une première fraction égyptienne, $\frac{1}{k}$. Ainsi défini, l'entier k remplit les conditions $\frac{1}{k} \leq x < \frac{1}{k-1}$, avec $k \geq 2$ puisque $x < 1$. Remplacer x par $\frac{1}{k}$ cause une erreur $e = x - \frac{1}{k}$. Ainsi définie, e est une fraction et e est positive. De plus,

$$e = \frac{p}{q} - \frac{1}{k} \tag{2}$$

$$= \frac{pk - q}{kq} \tag{3}$$

Mais $\frac{p}{q} < \frac{1}{k-1}$ donc le numérateur $P = pk - q = p + \overbrace{(p(k-1) - q)}^{<0}$, qu'on sait déjà positif, est aussi strictement inférieur à p. Nous venons d'établir que $0 \leq P < p$ et d'écrire

$$x = \frac{p}{q} \tag{4}$$

$$= \frac{1}{k} + \frac{P}{Q} \tag{5}$$

où $Q = kq$. Si $P = 0$, l'affaire est pliée. Sinon, l'hypothèse $\mathscr{H}$ au rang P fournit une décomposition égyptienne de $\frac{P}{Q}$, dont il faut juste s'assurer qu'elle ne contienne pas $\frac{1}{k}$. Or,

$$\frac{P}{Q} = e < \frac{1}{k-1} - \frac{1}{k} = \frac{1}{k(k-1)} \tag{6}$$

et

$$\frac{1}{k(k-1)} \leq \frac{1}{k} \tag{7}$$

d'où $\frac{P}{Q} < \frac{1}{k}$. Les fractions unitaires constituant $\frac{P}{Q}$ et relevant de $\mathscr{H}(P)$ se cumulent ; aucune d'elle n'égale donc $\frac{1}{k}$ et la conclusion suit. On notera que réduire les fractions au fur et à mesure dans cette démonstration ne modifierait en rien le cours des choses et conduirait au même développement.

2.2 Mise en œuvre algorithmique quand $x < 1$

De notre approche découle un algorithme glouton, déjà connu de Fibonacci au XIII^{e} siècle, où chaque itération donne lieu à la plus grande fraction égyptienne qui soit. On le rédigerait de la sorte pour fournir, en sortie, la chaîne de caractères à afficher présentant la décomposition obtenue.

```
DECOMPOSITION_V1(r)  # r rationnel dans ]0,1[
1   x, L, k ← r, [], 1
2   while (x ≠ 0)
3     while (x < 1/k)
4       k ← k + 1
5     x ← x − 1/k
6     L.append("+ 1/" + str(k))
7   return(L)
```

Ordonnées, les fractions égyptiennes vont en décroissant strictement. Il n'est donc pas utile, et serait même dispendieux, de réinitialiser la variable k à 1 avant la seconde boucle.

Implémenté en machine, le programme ne tourne pourtant pas comme prévu. Voyons pourquoi sur l'exemple où $x = \frac{3}{5}$, en l'exécutant pas à pas. En théorie, le procédé génère d'abord la fraction égyptienne $\frac{1}{2}$. En ligne 5, il remplace x par $x - \frac{1}{2}$, soit $\frac{1}{10}$; puis par $x - \frac{1}{10}$, qui est nul. Il s'arrête donc et retourne $\frac{1}{2} + \frac{1}{10}$. La machine se comporte autrement. Elle débute bien en calculant $\frac{1}{2}$. Sur ce elle remplace x par $x - \frac{1}{2}$, soit $\frac{1}{10}$, mais elle entâche le décimal $\frac{1}{10}$, qui n'est pas un nombre dyadique, d'une petite erreur [10]. Elle lui substitue $\frac{1}{16} + \frac{1}{32} + \frac{1}{256} + \frac{1}{512} + \ldots = 0,0999\ldots$ qui, malgré toute la puissance informatique, lui est légèrement inférieur. Dès lors, la deuxième fraction égyptienne renvoyée n'est plus $\frac{1}{10}$, mais $\frac{1}{11}$, ce qui entraîne l'ordinateur dans une boucle sans fin, ligne 2, et donne : $\frac{3}{5} = \frac{1}{2} + \frac{1}{11} + \frac{1}{111} + \frac{1}{12211} + \frac{1}{149096311} + \ldots$.

On évite cet écueil en bannissant tout calcul flottant. On s'oblige alors à ne gérer que des entiers, correspondant aux paires de numérateurs et dénominateurs des rapports manipulés. D'où la version ajustée :

```
DECOMPOSITION_V2(a, b)  # r = a/b
1   p, q, L, k ← a, b, [], 1
2   while (p ≠ 0)
3     while (p * k − q < 0)
4       k ← k + 1
5     p, q ← p * k − q, q * k
6     L.append("+ 1/" + str(k))
7   return(L)
```

Plus fiable, le script produit cependant des décompositions laborieuses. En témoignent $\frac{14}{19} = \frac{1}{2} + \frac{1}{5} + \frac{1}{28} + \frac{1}{887} + \frac{1}{2359420}$ ou l'astronomique $\frac{5}{31} = \frac{1}{7} + \frac{1}{55} + \frac{1}{3979} + \frac{1}{23744683} + \frac{1}{1127619917796295}$. Et pour cause : si le processus a déjà engrangé la somme $s_n = \frac{1}{k_1} + \frac{1}{k_2} + \ldots + \frac{1}{k_n}$ approchant par défaut la fraction $x = \frac{p}{q}$ passée en argument, et s'il n'a pas encore terminé, il s'apprête à rechercher k_{n+1}. La somme s_n engage $\frac{1}{k_n}$ et non $\frac{1}{k_n - 1}$ pour ne pas excéder x. Nécessairement,

$$\frac{1}{k_{n+1}} < \frac{1}{k_n - 1} - \frac{1}{k_n} = \frac{1}{k_n(k_n - 1)} \tag{8}$$

donc

$$k_{n+1} > k_n(k_n - 1) \tag{9}$$

On comprend ainsi mieux l'inflation observée. Quant au nombre d'étapes requises, on ne sait que le majorer par p, limite d'ailleurs atteinte avec $\frac{5}{31}$.

La taille des dénominateurs n'est pas rédhibitoire en soi – le langage Python composerait avec ce gigantisme sans provoquer de débordement de la mémoire – mais elle verrouille la boucle en ligne 3. Le compteur k, qui ne s'y incrémente que d'une unité par itération, doit dépasser des seuils toujours plus vertigineux. Les temps de calcul en deviennent prohibitifs.

En pratique, sorti des fractions les plus simples, il faut choisir une autre stratégie – non gloutonne cette fois. Habilement, les dénominateurs successifs émergeront des relations de Bézout. L'effet ne sera plus d'augmenter k à chaque tour, mais à l'inverse, pour un meilleur contrôle, de le diminuer. Partons de la fraction *préalablement réduite* $\frac{p}{q} = \frac{p_0}{q_0} \in]0,1[$. Bien entendu, $0 < p_0 < q_0$. Tant que $p_0 \neq 1$ il existe une paire (p_1, q_1) unique telle que :

$$p_0 q_1 - q_0 p_1 = 1 \tag{10}$$

avec $0 < q_1 < q_0$ et $0 < p_1 < p_0$. Les entiers p_1 et q_1 sont à leur tour premiers entre eux. Ajoutons que $\frac{p_0}{q_0} = \frac{1}{q_0 q_1} + \frac{p_1}{q_1}$. Nécessairement, $p_1 < q_1$. Tant que $p_1 \neq 1$ on répète l'opération qui donne ici naissance au couple (p_2, q_2), avec $p_1 q_2 - q_1 p_2 = 1$, $0 < q_2 < q_1 < q_0$, $0 < p_2 < p_1 < p_0$. À ce stade, $\frac{p_0}{q_0} = \frac{1}{q_0 q_1} + \frac{1}{q_1 q_2} + \frac{p_2}{q_2}$, un scenario qui se répète

jusqu'au premier entier p_i égal à 1. La décomposition finale compte moins de p membres, tous distincts puisque la suite $(q_n q_{n+1})_n$ décroît strictement. Contenus par $q_0 q_1$, lui-même inférieur à q^2, les dénominateurs n'enflent plus démesurément. C'est avec cette méthode que nous ramenons $\frac{14}{19}$ et $\frac{5}{31}$ respectivement à $\frac{1}{285}+\frac{1}{165}+\frac{1}{21}+\frac{1}{6}+\frac{1}{2}$ et à $\frac{1}{775}+\frac{1}{475}+\frac{1}{247}+\frac{1}{91}+\frac{1}{7}$. Retranscrite ci-après, elle charge une routine **bezout**( ) associant à tout couple (a,b) non trivial les traditionnels coefficients u et v respectant l'identité $au+bv=pgcd(a,b)$, et, moyennant une retouche mineure à l'algorithme d'Euclide étendu, les conditions $u\in[1,b-1]$ et $-v\in[1,a-1]$. Sa complexité, de l'ordre de $\log(ab)$, reste mesurée.

```
DECOMPOSITION_V3(a, b)  # x = a/b et a ∧ b = 1
1  p, q, L ← a, b, []
2  while (p ≠ 1)
3     Q, P = bezout(p, q)
4     L.append("+ 1/" + str(q × Q))
5     p, q ← −P, Q
6  L.append("+ 1/" + str(q))
7  return(L)
```

2.3 Le programme a encore faim!

Notre traitement échoue quand $x>1$: la solution gloutonne accumule des fractions égyptiennes toutes égales (en l'occurence, à 1), l'autre mène à une fraction du type $\frac{p}{1}$ (en définitive, entière) qui l'enlise. Une première phase consiste dès lors à approcher x par défaut à l'unité près avant d'enclencher l'algorithme qui terminera le travail. On se sert pour cela de la série harmonique, qui présente le double intérêt d'être formée de fractions égyptiennes distinctes et de diverger. Un rationnel $x>1$ étant donné, il existe un rang j pour lequel

$$H_j = 1+\frac{1}{2}+\ldots+\frac{1}{j} < x \leq 1+\frac{1}{2}+\ldots+\frac{1}{j}+\frac{1}{j+1} = H_{j+1} \tag{11}$$

Reste à décomposer $x-H_j$, élément de $\mathbb{Q}_+^*$, sans craindre de collision avec l'expression de H_j puisque $x-H_j<H_{j+1}-H_j=\frac{1}{j+1}$. Sur ce modèle, $\frac{5}{2}=\left(1+\frac{1}{2}+\frac{1}{3}+\frac{1}{4}+\frac{1}{5}+\frac{1}{6}\right)+\frac{1}{20}$. En partant non pas d'une somme partielle mais d'une tranche de Cauchy, $H_{i,j}=\frac{1}{i}+\frac{1}{i+1}+\ldots+\frac{1}{j}$, on mettrait en évidence une autre décomposition égyptienne. En choisissant i assez grand, on adapte même cette idée au cas de $x\in]0,1]$. Ainsi,

$$\frac{2}{3} \quad = \quad \left(\frac{1}{5}+\frac{1}{6}+\frac{1}{7}+\frac{1}{8}\right) \quad + \quad \left(\frac{1}{69720}+\frac{1}{54282}+\frac{1}{40766}+\frac{1}{29172}+\frac{1}{19500}+\frac{1}{11750}+\frac{1}{5922}+\frac{1}{2016}+\frac{1}{32}\right) \tag{12}$$

En conclusion, tout rationnel strictement positif admet une infinité de développements égyptiens.

3 Le problème du monnayeur

3.1 Rendre efficacement la monnaie

Pour régler 48 euros en liquide, on peut sortir de son portefeuille **5** billets ou pièces : **2** billets de 20, **1** billet de 5, **1** pièce de 2 et **1** pièce de 1. D'autres combinaisons existent, telles que $\mathbf{2}\times 10+\mathbf{2}\times 5+\mathbf{6}\times 2+\mathbf{6}\times 1$, mais elles nécessitent davantage d'espèces, ici **16** billets ou pièces (nous dirons coupures, pour simplifier). L'approche la plus naturelle, celle du « moindre » effort, celle-là même que nous avons appliquée et que pendant longtemps les distributeurs automatiques des banques ou les caisses électroniques de supermarché ont suivie, est gloutonne. On avance vers le montant désiré en piochant d'abord autant que de possible dans sa liasse de 500, puis dans celle de 200, de 100, etc. jusqu'à utiliser les pièces. L'écart à la valeur nominale se réduit à grandes enjambées, à plus courtes, et, à tout petits pas, finit par s'annuler. Nous allons étudier si cette manière de faire l'appoint fonctionne toujours, est optimale au regard du nombre de coupures à réunir, et, le cas échéant, conserverait de bonnes performances dans des systèmes monétaires aux devises différemment échelonnées.

Par commodité, nous n'envisagerons que des prix ronds – les sommes seront donc entières (sans centimes), et nous nous rêverons riches à millions – nos réserves de "cash" étant illimitées.

3.2 Formalisation

Le contexte une fois posé, il est utile de le modéliser.

Définition 2 (système de valeurs, complétude)**.** *Un système S de n valeurs est une suite strictement ordonnée d'entiers naturels non nuls* $(v_1, v_2, \ldots, v_n)$ *où* $0 < v_1 < v_2 < \ldots < v_n$*. Le système est dit complet quand la première graduation,* v_1*, vaut l'unité :* $v_1 = 1$.

Le système $S = (4, 8, 12)$ n'est pas complet tandis que le système numéraire en vigueur sur la zone euro, $S_€ = (1, 2, 5, 10, 20, 50, 100, 200, 500$[1]$)$, l'est.

Définition 3 (décomposition)**.** *Une décomposition d'un entier naturel x sur un système* $S = (v_1, v_2, \ldots, v_n)$ *est une famille* $(\omega_1, \omega_2, \ldots, \omega_n)$ *d'entiers naturels, appelés coefficients énumérateurs ou poids, tels que*

$$x = \omega_1 v_1 + \omega_2 v_2 + \ldots + \omega_n v_n \tag{13}$$

Abusivement, nous qualifierons également de décomposition de x l'écriture (13)*. Nous désignerons par*

$$\Omega_S(x, \omega_1, \omega_2, \ldots, \omega_n) = \omega_1 + \omega_2 + \ldots + \omega_n \tag{14}$$

le poids de la décomposition.

Avec $S = (4, 8, 12)$, $2 \times 4 + 1 \times 8 + 0 \times 12$ et $1 \times 4 + 0 \times 8 + 1 \times 12$ sont des décompositions de $x = 16$, tandis que $x = 10$ et $x = 13$ n'en ont aucune. Lorsque le système S est complet – nous le supposerons dorénavant – tout $x \in \mathbb{N}$ s'y décompose ne serait-ce que par autoréférencement : $x = x \times v_1 + 0 \times v_2 + \ldots + 0 \times v_n$. L'algorithme glouton ci-après termine toujours ; au pire solde-t-il la fin de l'addition en pièces de 1. Il délivre « sa » décomposition, formatée dans une chaîne de caractères renvoyée en sortie.

```
INITIALISATIONS  # A adapter, ici système €, S[1] = 1, S[2] = 2, S[3] = 5, ...
1   n = 9
2   S = (1;2;5;10;20;50;100;200;500)

MONNAYEUR(x)  # x entier naturel
1   L ← []
2   while (x ≠ 0)
3      k = 0
4      while (x >= S[n])
5         x ← x − S[n]
6         k ← k + 1
7      L.append("+" str(k) + "*" + str(S[n]) )
8      n ← n − 1
9   return(L)
```

L'ensemble des décompositions possibles d'un entier x donné est fini, les coefficients étant tous bornés par x. On le dénombrerait même en multipliant des séries génératrices comme suit :

$$\frac{1}{1-z} \times \frac{1}{1-z^{v_2}} \times \frac{1}{1-z^{v_3}} \times \ldots \times \frac{1}{1-z^{v_n}} \tag{15}$$

dont on sélectionnerait le monôme z^x du produit après distribution. Surtout, il est légitime d'introduire la

Définition 4 (poids minimal)**.** *Soient S un système complet et* $x \in \mathbb{N}$*. On appelle poids de x relativement au système S, et on note* $\mu_S(x)$*, le plus petit poids parmi ceux des décompositions admissibles de x sur S. Soit :*

$$\mu_S(x) = \min_{\substack{(\omega_1, \omega_2, \ldots, \omega_n) \\ x = \omega_1 v_1 + \omega_2 v_2 + \ldots + \omega_n v_n}} \Omega_S(x, \omega_1, \omega_2, \ldots, \omega_n) \tag{16}$$

Par propriété d'un minimum et car $x = x \times 1$ compte parmi les décompositions admissibles de x, on a la

Proposition 1. $0 \leq \mu_S(x) \leq x$.

Reprenons $S = (4, 8, 12)$, avec lequel $\mu_S(16) = 2$. Ce poids minimal s'obtient de deux façons : $16 = 1 \times 4 + 0 \times 8 + 1 \times 12$, déjà rencontrée, mais aussi $16 = 0 \times 4 + 2 \times 8 + 0 \times 12$

Les poids des décompositions produites par l'algorithme glouton retiendront notre attention :

1. Le billet de 500 n'est plus émis par les banques centrales mais il conserve sa valeur et circule toujours.

Définition 5 (poids glouton)**.** *Soient S un système complet et $x \in \mathbb{N}$. On posera $g_S(x)$ le poids de la décomposition que donne de x l'algorithme glouton.*

Bien sûr, nous avons la

Proposition 2. $g_S(x) \geq \mu_S(x)$.

Définition 6 (système canonique)**.** *Soit S un système complet. On dit que S est un système canonique lorsque l'algorithme glouton produit toujours des décompositions optimales, à savoir quand,*

$$\forall x \in \mathbb{N}, \quad g_S(x) = \mu_S(x) \tag{17}$$

Avant la réforme de 1971, le système monétaire du Royaume Uni reposait sur la suite $S_£ = (1,3,6,12,24,30)$. Il n'était pas canonique. En cause par exemple $x = 48$, dont la décomposition gloutonne, $0\times1+0\times3+1\times6+1\times12+0\times24+1\times30$ (à lire de droite à gauche) pèse $g_S(x) = 3$ et coûte par conséquent davantage que l'écriture $x = 2\times24$.

Il n'est pas si aisé de caractériser les systèmes canoniques. Une chose est sûre, en dépit de légendes aux arguments plus ou moins intuitifs, ils ne correspondent pas aux suites supercroissantes, par conséquent sans objet dans cet article. Nous en rappelons néanmoins la

Définition 7 (suite supercroissante)**.** *Une suite croissante et positive $(v_1, v_2, \ldots, v_n)$ est supercroissante quand*

$$\begin{cases} v_1 < v_2 \\ v_1 + v_2 < v_3 \\ \ldots \\ v_1 + v_2 + \ldots + v_{n-1} < v_n \end{cases} \tag{18}$$

La supercroissance ne garantit pas le caractère canonique d'un système complet. En atteste $S = (1,6,11,19)$. Le choix de $x = 22$ donne lieu à $\mu_S(22) = 2$ quand $g_S(22) = 4$. À travers des cas d'école, nous constaterons dans le paragraphe suivant que la condition de supercroissance n'est pas non plus nécessaire, puis nous déboucherons vers le cas le plus général, section 3.4.

3.3 Quelques exemples canoniques

Avant d'illustrer le propos, caractérisons les décompositions gloutonnes :

Théorème 2 (caractérisation gloutonne)**.** *Soit $S = (v_1 = 1, v_2, \ldots, v_n)$ un système complet. Une écriture du type $\omega_1 v_1 + \omega_2 v_2 + \ldots + \omega_n v_n$ dont les coefficients sont entiers naturels est une décomposition gloutonne (du nombre x à laquelle elle est égale) si, et seulement si,*

$$\begin{cases} \omega_1 v_1 < v_2 \\ \omega_1 v_1 + \omega_2 v_2 < v_3 \\ \ldots \\ \omega_1 v_1 + \omega_2 v_2 + \ldots + \omega_{n-1} v_{n-1} < v_n \end{cases} \tag{19}$$

Démonstration. Qu'une de ces inégalités soit renversée, par exemple $\omega_1 v_1 + \omega_2 v_2 + \ldots + \omega_{k-1} v_{k-1} \geq v_k$ et l'algorithme glouton n'a pas fait son office au moment de déterminer ω_k. Qu'elles soient toutes bien orientées et les coefficients successifs, $\omega_n, \omega_{n-1}, \ldots$, en remontant jusqu'à ω_1 sont conformes aux choix de l'algorithme.

Une grande variété de situations types s'en déduit.

Situation 1. *Un système complet $S = (d_1 = 1, d_2, \ldots, d_n)$ dont les barreaux se divisent successivement – $d_1 \mid d_2$, $d_2 \mid d_3$, $\ldots$, $d_{n-1} \mid d_n$ – est canonique.*

Démonstration. Soit $\omega_1^* d_1 + \omega_2^* d_2 + \ldots + \omega_n^* d_n$ une décomposition optimale d'un certain x. Obligatoirement $\omega_1^* d_1 \leq d_2 - d_1$. Sans quoi $\omega_1^* d_1$, qui saute par crans d'amplitude d_1, égale ou dépasse d_2 de $(\omega_1^* - \frac{d_2}{d_1})d_1$. Or il est sans conteste plus rentable de remplacer $\omega_1^* d_1$ par $(\omega_1^* - \frac{d_2}{d_1})d_1 + 1d_2$ dont la contribution à la décomposition de x descend à $(\omega_1^* - \frac{d_2}{d_1}) + 1 < \omega_1^*$. De même, $\omega_2^* d_2 \leq d_3 - d_2$, $\omega_3^* d_3 \leq d_4 - d_3$, etc. Si bien que :

$$\forall k < n, \ \omega_1^* d_1 + \omega_2^* d_2 + \ldots + \omega_k^* d_k \leq (d_2 - d_1) + (d_3 - d_2) + \ldots (d_{k+1} - d_k) \tag{20}$$

et, après télescopage :

$$\omega_1^* d_1 + \omega_2^* d_2 + \ldots + \omega_k^* d_k \leq d_{k+1} - d_1 < d_{k+1} \tag{21}$$

La conclusion tombe via le théorème 2 : non seulement l'écriture gloutonne est optimale mais c'est ici la seule à l'être et par conséquent *la* meilleure.

Cette propriété vaut notamment pour les suites en progression géométrique $(1, q, q^2, \ldots, q^{n-1})$ de raisons entières $q \geq 2$. Quand $q = 2$ et $q = 10$, on y retrouve les jalons des bases 2 et 10. Signalons en passant la supercroissance de tous ces systèmes : $v_2 \leq 2v_1$, $v_3 \leq 2v_2$, … donc,

$$\begin{aligned} v_1 + v_2 + \ldots + v_k &\leq (2^k - 1)v_1 && (22)\\ &< 2^k v_1 \leq v_{k+1} && (23)\end{aligned}$$

Situation 2. *Le système euro* $S_€ = (1, 2, 5, 10, 20, 50, 100, 200, 500)$ *est canonique.*

Démonstration. Nous l'avons souligné, la supercroissance du système (ici vérifiée) n'est pas un critère déterminant. Utilisons plutôt le théorème 2 et autorisons-nous à indexer les coefficients par les valeurs faciales qu'ils pondèrent. Soit donc $\omega_1^* \times 1 + \omega_2^* \times 2 + \omega_5^* \times 5 + \omega_{10}^* \times 10 + \omega_{20}^* \times 20 + \omega_{50}^* \times 50 + \omega_{100}^* \times 100 + \omega_{200}^* \times 200 + \omega_{500}^* \times 500$ une décomposition optimale d'un certain x.

— Une pièce de 2 se substitue à deux de 1, abaissant strictement le poids de la décomposition si $\omega_1^* \geq 2$. Donc $\omega_1^* \leq 1$. De même, $\omega_{10}^* \leq 1$ et $\omega_{100}^* \leq 1$.

— Un billet de 5 et une pièce de 1 font gagner trois pièces de 2. Donc $\omega_2^* \leq 2$, majoration qui se transpose aux dizaines et aux centaines : $\omega_{20}^* \leq 2$ et $\omega_{200}^* \leq 2$.

— Un billet de 10 vaut mieux (en contexte) que deux de 5. Donc $\omega_5^* \leq 1$ et, à l'avenant, $\omega_{50}^* \leq 1$.

Un dernier point mérite précision :

— Quand $\omega_1^* = 1$, alors $\omega_2^* \leq 1$: on tirerait sinon meilleur profit de ω_5^*. De même, quand $\omega_{10}^* = 1$, $\omega_{20}^* \leq 1$ et quand $\omega_{100}^* = 1$, $\omega_{200}^* \leq 1$.

Ces inégalités aboutissent à

$$\begin{cases} \omega_1^* \times 1 < 2 \\ \omega_1^* \times 1 + \omega_2^* \times 2 < 5 \\ \omega_1^* \times 1 + \omega_2^* \times 2 + \omega_5^* \times 5 < 10 \\ \omega_1^* \times 1 + \omega_2^* \times 2 + \omega_5^* \times 5 + \omega_{10}^* \times 10 < 20 \\ \ldots \\ \omega_1^* \times 1 + \omega_2^* \times 2 + \ldots + \omega_{50}^* \times 50 + \omega_{100}^* \times 100 + \omega_{200}^* \times 200 < 500 \end{cases} \tag{24}$$

C'est en dosant les coefficients ω_1 et ω_2 qu'on balaye les cinq premiers entiers : $0 = 0 \times 1 + 0 \times 2$, $1 = 1 \times 1 + 0 \times 2$, $2 = 0 \times 1 + 1 \times 2$, $3 = 1 \times 1 + 1 \times 2$, $4 = 0 \times 1 + 2 \times 2$. Réglé à 1, le poids ω_5 translate le curseur dans la fourchette $[\![4, 9]\!]$. On joue sur les dizaines en actionnant ω_{10}, ω_{20}, ω_{50}. Et sur les centaines au moyen de ω_{100}, ω_{200}, ω_{500}. De sorte qu'avec un exemplaire de 1, deux de 2, un de 5, …, deux de 200, un de 500, on couvre toutes les valeurs entières allant de 0 à 999 (et même un peu au-delà). Bien que cela ne corresponde pas à la logique gloutonne, on peut aussi faire avec deux exemplaires de 1, un de 2, un de 5, deux de 10, …, un de 500. Les coffrets de masses des balances Roberval sont garnis selon cette répartition, et sont ainsi (un peu) moins lourds à transporter.

Situation 3. *Un système complet en progression arithmétique* $S_A = (a_1 = 1, a_2 = a_1 + r, a_3 = a_2 + r, \ldots, a_n = a_{n-1} + r)$ *gouverné par une raison entière* $r \geq 1$ *est canonique. Il n'est en revanche pas supercroissant.*

Démonstration. Commençons par la fin. Dans le cas général,

$$a_{k-1} + a_k = a_{k-2} + a_{k+1} \tag{25}$$

donc $a_{k+1} = a_k + (a_{k-1} - a_{k-2}) < a_k + a_{k-1} < a_k + a_{k-1} + \ldots + a_1$ et le système n'est pas supercroissant.

Étudions la canonicité. Soit un entier x. Parmi toutes les décompositions optimales de x, une seule, $\omega_1^{**} a_1 + \omega_2^{**} a_2 + \ldots + \omega_n^{**} a_n$, est maximale au regard de l'ordre lexicographique « arabe », où $(\omega_1^*, \omega_2^*, \ldots, \omega_n^*) \preceq (\widehat{\omega}_1^*, \widehat{\omega}_2^*, \ldots, \widehat{\omega}_n^*)$ quand $\omega_n^* < \widehat{\omega}_n^*$ ou, s'ils sont égaux, $\omega_{n-1}^* < \widehat{\omega}_{n-1}^*$ ou, s'ils sont eux aussi égaux, etc [2]. Cette écriture « suroptimale » qui privilégie les grosses coupures a des chances d'être gloutonne. Elle l'est effectivement et ne reposera que sur de rares coefficients :

— S'il se trouve un k, $1 < k < n$, pour lequel $\omega_k^{**} \geq 2$, on aurait remplacé $2a_k$ par $a_{k-1} + a_{k+1}$, engendrant une écriture de même poids mais classée au-dessus selon $\preceq$. C'est impossible.

— S'il se trouve k et ℓ, $1 < k < \ell < n$, pour lesquels ω_k^{**} et ω_ℓ^{**} sont non nuls (donc valent 1), on aurait remplacé $a_k + a_\ell$ par $a_{k-1} + a_{\ell+1}$, engendrant une écriture de même poids mais classée au-dessus selon $\preceq$. C'est impossible.

2. Par exemple, $(1, 4, 3) < (10, 5, 3) < (1, 0, 8)$.

— Le développement « suroptimal » de x possède donc au plus trois coefficients significatifs : ω_1^{**}, un certain $\omega_k^{**} \leq 1$, et ω_n^{**}. De plus $\omega_1^{**} \leq r$ sans quoi nous aurions subtilisé avantageusement a_2 à $(1+r)a_1$. Enfin, ω_1^{**} et ω_k^{**} ne sont pas conjointement saturés sans quoi $\omega_1^{**} a_1 + \omega_k^{**} a_k = r + a_k = a_{k+1}$. Le théorème 2 s'enclenche.

Situation 4. *Bien qu'il ne soit pas supercroissant, le système $S_F = (F_1 = 1, F_2 = 2, F_3 = 3, \ldots, F_n)$ régi par la récurrence de Fibonacci est canonique.*

Démonstration. L'inégalité $F_1 + F_2 + \ldots + F_k \geq F_{k-1} + F_k = F_{k+1}$ valable dès que $2 \leq k < n$ contredit la supercroissance. On déduirait même d'une récurrence que

$$F_1 + F_2 + \ldots + F_k = F_{k+2} - 2 \tag{26}$$

Pour l'heure, envisageons une décomposition optimale $\omega_1^* F_1 + \omega_2^* F_2 + \ldots + \omega_n^* F_n$ d'un certain x.

Jusqu'au rang $n-1$, au moins un coefficient sur deux est nul. Si deux poids consécutifs ω_{k-1}^* et ω_k^*, $k < n$, étaient non nuls on grouperait en effet $F_{k-1} + F_k$ en F_{k+1}, de valence strictement moindre.

On peut alors toujours se ramener à une écriture de x à la fois minimale et dont les coefficients sont binaires jusqu'à l'ordre $n-1$. Expliquons comment on a pu l'aplanir ainsi. Quand le poids ω_{n-1}^* dépasse 2, on allège d'autant sa charge en la redistribuant entre ses voisins selon la règle $2F_{n-1} = F_n + F_{n-3}$. Cela laisse inchangé le poids total. On répète l'opération sur ω_{n-1}^* autant que de besoin. On aborde ensuite ω_{n-2}^*. S'il excède 2, on le lisse à son tour sachant que $2F_{n-2} = F_{n-1} + F_{n-4}$. On ne compromet pas le travail réalisé sur F_{n-1} car le coefficient (actualisé) ω_{n-1}^* devait être nul. À ce stade, $\omega_{n-1}^* = 1$ et (donc) $\omega_{n-2}^* = 0$. Cahin caha, on en arrive à ω_2^*, que l'équation $2F_2 = F_3 + F_1$ dissout à son tour. Enfin ω_1^* se traite en reversant $2F_1$ dans F_2. Une alternative à cette démarche constructive consistait à raisonner comme sur les systèmes arithmétiques, en considérant la décomposition « suroptimale » de x.

Le reformatage une fois accompli, nous récueillons une écriture mimale de x dont les poids sont binaires et non mitoyens de ω_1^* à ω_{n-1}^*. Soit $k < n$. Comme $F_k = F_{k-1} + F_{k-2}$ et $F_{k-1} = F_{k-2} + F_{k-3}$, et puisque F_k et F_{k-1} ne sont pas concomitants dans la nouvelle décomposition, $\omega_{k-1}^* F_{k-1} + \omega_k^* F_k \leq F_{k-3} + F_{k-2} + F_{k-1}$. En étendant ce raisonnement,

$$\begin{aligned} \omega_1^* F_1 + \omega_2^* F_2 + \ldots + \omega_k^* F_k &\leq 1 + F_1 + F_2 + \ldots + F_{k-1} && (27)\\ &= F_{k+1} - 1 \text{ d'après (26)} && (28)\\ &< F_{k+1} && (29) \end{aligned}$$

On conclut grâce au théorème 2.

3.4 Cas général

L'étude qui suit, inspirée de [6, 8, 5, 3], vise à qualifier directement les systèmes canoniques en nous affranchissant du théorème 2. Nous noterons $\lceil x \rceil$ et $\lfloor x \rfloor$ les parties entières respectivement supérieure et inférieure d'un réel x. Par exemple, $\lceil 10.4 \rceil = 10$ et $\lfloor 10.4 \rfloor = 11$. Commençons par ce

Théorème 3. *Si le système complet $S = (v_1 = 1, v_2, \ldots, v_n)$ est canonique alors, pour tout k, $2 \leq k \leq n$,*

$$g_S(v_k^+) \leq \frac{v_k^+}{v_{k-1}} \tag{30}$$

où v_k^+ est le premier multiple entier de v_{k-1} supérieur ou égal à v_k, soit :

$$v_k^+ = \lceil \frac{v_k}{v_{k-1}} \rceil v_{k-1} \tag{31}$$

Démonstration. Le terme $\frac{v_k^+}{v_{k-1}} v_{k-1}$ est une représentation, certes ramassée, de v_k^+ sur le système S. Donc $\mu_S(v_k^+) \leq \frac{v_k^+}{v_{k-1}}$. Or, $g_S(v_k^+) = \mu_S(v_k^+)$. La conclusion s'impose. La condition énoncée, qui porte sur les performances du glouton en certains nœuds bien choisis, s'avère suffisante quand les échelons de S sont assez espacés. C'est ce qu'affirme le

Théorème 4 (One-Point Theorem)**.** *Soit $S = (v_1 = 1, v_2, \ldots, v_n)$ un système complet tel que pour tout k, $2 \leq k \leq n-1$,*

$$v_k^+ < v_{k+1} \tag{32}$$

Si, pour tout k, $2 \leq k \leq n-1$,

$$g_S(v_k^+) \leq \frac{v_k^+}{v_{k-1}} \tag{33}$$

alors le système S est canonique.

Démonstration. On procède par l'absurde. L'arme est à « double détente » … D'abord en déclarant l'existence d'un premier rang t, $2 \leq t \leq n$, tel que $S_t = (v_1, v_2, \ldots, v_t)$ ne soit pas canonique. Puis, conditionnellement à ce rang t, en invoquant un premier contre-exemple $x \geq 2$ pour lequel $g_{S_t}(x) > \mu_{S_t}(x)$, ou, abréviativement, $g_t(x) > \mu_t(x)$.

Forcément $x \geq v_t$. Dans le cas contraire, aucune représentation de x ne peut mobiliser l'échelon v_t et donc $\mu_t(x) = \mu_{t-1}(x)$. Mais alors $\mu_{t-1}(x) = g_{t-1}(x)$ par minimalité de t. On remonte ensuite à $g_t(x)$, du fait de $x < v_t$. Bref, l'enchaînement $\mu_t(x) = \mu_{t-1}(x) = g_{t-1}(x) = g_t(x)$ nous dément. Évidemment $x \neq v_t$, où le glouton serait imbattable avec $g_t(v_t) = 1$. Ainsi a-t-on même $x > v_t$.

Puisque $x \geq v_t$, $g_t(x)$ impliquera v_t. Pas $\mu_t(x)$. Justifions tout cela :

— Naturellement, x étant supérieur au barreau v_t, le glouton relatif à S_t lui ôte un premier v_t. Puis il se lance aux trousses de $x - v_t$. Si bien que $g_t(x) = g_t(x - v_t) + 1$.

— De la somme $x = (x - v_t) + v_t$ provient une première inégalité, que l'on pourrait baptiser de triangulaire : $\mu_t(x) \leq \mu_t(x - v_t) + \mu_t(v_t) = \mu_t(x - v_t) + 1$. À supposer que v_t participe de $\mu_t(x)$, nous aurions $x = \left(\omega_1^* v_1 + \omega_2^* v_2 + \ldots + (\omega_t^* - 1) v_t\right) + 1 v_t$. Il est virtuellement possible de condenser encore le premier membre, c'est pourquoi $\mu_t(x) \geq \mu_t(x - v_t) + 1$. L'un dans l'autre, $\mu_t(x) = \mu_t(x - v_t) + 1$.

— Tout mis bout à bout, $\mu_t(x - v_t) < g_t(x - v_t)$, ce qu'interdit pourtant la minimalité de x.

Dual de v_t^+, premier multiple de v_{t-1} supérieur ou égal à v_t, définissons x^-, dernier multiple de v_{t-1} inférieur ou égal x : $x^- = \lfloor \frac{x}{v_{t-1}} \rfloor v_{t-1}$. La figure 2, qui n'est pas à l'échelle, résume l'alignement des principaux jalons concernés, les positions relatives de x_- et v_t^+ étant permutables.

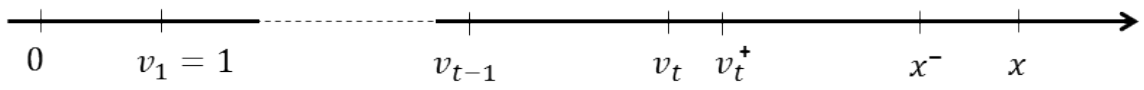

FIGURE 2 – Alignement des principaux curseurs.

Dès lors que $\mu_t(x)$ n'accorde aucun poids à v_t, $\mu_t(x) = \mu_{t-1}(x)$, lui-même égal à $g_{t-1}(x)$, conséquence de la minimalité de t. À l'assaut de x, le glouton adossé à S_{t-1} s'élance résolument sur x^-. Il lui en coûte $\lfloor \frac{x}{v_{t-1}} \rfloor$. Puis $g_{t-1}(x - x^-)$ pour couvrir la distance restante. Le bilan s'écrit :

$$\mu_t(x) = \lfloor \frac{x}{v_{t-1}} \rfloor + g_{t-1}(x - x^-) \tag{34}$$

Soupesons en outre $g_t(x)$. L'algorithme s'acquitte dès le départ de $1 v_t$. Puis de $g_t(x - v_t) = \mu_t(x - v_t)$ (minimalité de x), pour finir. L'emboîtement de S_{t-1} dans S_t prolonge cette chaîne en la majorant :

$$g_t(x) = 1 + g_t(x - v_t) \underset{\text{min. de } x}{=} 1 + \mu_t(x - v_t) \tag{35}$$

$$\leq 1 + \mu_{t-1}(x - v_t) \underset{\text{min. de } t}{=} 1 + g_{t-1}(x - v_t) \tag{36}$$

Scindons par ailleurs $x - v_t$ en

$$x - v_t = (x - x^-) + (x^- - v_t^+) + (v_t^+ - v_t) \tag{37}$$

$$= \left(\lfloor \frac{x}{v_{t-1}} \rfloor - \lceil \frac{v_t}{v_{t-1}} \rceil \right) v_{t-1} + (x - x^-) + (v_t^+ - v_t) \tag{38}$$

Les entiers naturels $x - v_t$ d'un côté et $(x - x^-) + (v_t^+ - v_t)$ de l'autre sont égaux modulo $\left(\lfloor \frac{x}{v_{t-1}} \rfloor - \lceil \frac{v_t}{v_{t-1}} \rceil \right) v_{t-1}$. Leurs poids glouton s'en trouvent liés :

$$g_{t-1}(x - v_t) = \left(\lfloor \frac{x}{v_{t-1}} \rfloor - \lceil \frac{v_t}{v_{t-1}} \rceil \right) + g_{t-1}\left((x - x^-) + (v_t^+ - v_t) \right) \tag{39}$$

Puis $g_{t-1}\left((x - x^-) + (v_t^+ - v_t) \right) = \mu_{t-1}\left((x - x^-) + (v_t^+ - v_t) \right)$ (minimalité de t) qui, toujours à la manière d'une inégalité triangulaire, est moindre que $\mu_{t-1}(x - x^-) + \mu_{t-1}(v_t^+ - v_t)$, quantité elle même égale à $g_{t-1}(x - x^-) + g_{t-1}(v_t^+ - v_t)$ (minimalité de t). Récapitulons :

$$g_{t-1}\left((x - x^-) + (v_t^+ - v_t) \right) \leq g_{t-1}(x - x^-) + g_{t-1}(v_t^+ - v_t) \tag{40}$$

Les hypothèses du théorème interviennent ici. Nous savons que :

$$g_S(v_t^+) \leq \frac{v_t^+}{v_{t-1}} = \lceil \frac{v_t}{v_{t-1}} \rceil \tag{41}$$

et que

$$v_t^+ < v_{t+1} \tag{42}$$

Donc $g_S(v_t^+) = g_t(v_t^+)$ via ((42)). Ainsi (41) devient-elle $g_t(v_t^+) \leq \lceil \frac{v_t}{v_{t-1}} \rceil$. Mais $g_t(v_t^+) = 1 + g_t(v_t^+ - v_t)$. Comme, $v_t^+ - v_t < v_t$ (car $v_t^+ - v_t < v_{t-1}$), $g_t(v_t^+) = 1 + g_{t-1}(v_t^+ - v_t)$, bref :

$$g_t(v_t^+) = 1 + g_{t-1}(v_t^+ - v_t) \tag{43}$$

puis :

$$1 + g_{t-1}(v_t^+ - v_t) \leq \lceil \frac{v_t}{v_{t-1}} \rceil \tag{44}$$

On infère, en articulant (34), (36), (39), (40), (44) :

$$g_t(x) \leq \mu_t(x) \tag{45}$$

La contradiction éclate.

3.5 Retour aux cas d'école et conclusion

Apprécions l'efficacité du théorème 4 en contraste au paragraphe 3.3 dont nous reprendrons le plan et les notations.

Reprise du résultat 1 où $d_1 = 1 \mid d_2 \mid \ldots \mid d_n$. Pour $k < n$,

$$\lceil \frac{d_k}{d_{k-1}} \rceil = \frac{d_k}{d_{k-1}} \tag{46}$$

et donc

$$d_k^+ = d_k < d_{k+1} \tag{47}$$

De plus, $g_n(d_k) = 1$. En particulier,

$$g_n(d_k^+) \leq \frac{d_k^+}{d_{k-1}} \tag{48}$$

Nous homologuons le système.

Reprise du résultat 2 où $v_1 = 1, v_2 = 2, v_5 = 5, \ldots, v_{500} = 500$. Les seuls doutes éventuels porteraient sur $v_5^+ = 6$ et $v_{50}^+ = 60$. Ils sont bien strictement inférieurs respectivement à $v_{10} = 10$ et $v_{100} = 100$ (où les notation indicielle ont été adaptées pour l'occasion). De surcroît, 6 a pour développement glouton $6 = 1 \times 5 + 1 \times 1$, de poids 2 inférieur ou égal à $\frac{v_5^+}{v_2} = 3$. On traiterait de même l'entier 60. Nous homologuons le système.

Reprise du résultat 3 où $v_1 = a_1 = 1, v_2 = a_2, \ldots, v_n = a_n$ est arithmétique de premier terme 1 et de raison entière r. De manière générale,

$$\frac{a_k}{a_{k-1}} = \frac{a_{k-1} + r}{a_{k-1}} = 1 + \frac{r}{a_{k-1}} \tag{49}$$

et donc $a_k^+ = 2a_{k-1} \ldots$ qui surclasse a_{k+1}. Mais l'entrefilet qu'encadre les formules (42) et (43) prouve que seule l'inégalité $g_k(a_k^+) \leq \frac{a_k^+}{a_{k-1}}$ compte. C'est une chance : $a_k^+ = 2a_{k-1} = a_k + a_{k-2}$, et donc $g_k(a_k^+) = 1 + 1 = 2$. Nous homologuons le système.

Reprise du résultat 4 où $v_1 = F_1 = 1, v_2 = F_2, \ldots, v_n = F_n$. De manière générale,

$$\frac{F_k}{F_{k-1}} = \frac{F_{k-1} + F_{k-2}}{F_{k-1}} = 1 + \frac{F_{k-2}}{F_{k-1}} \tag{50}$$

donc $F_k^+ = 2F_{k-1}$. Or $F_k = F_{k-1} + F_{k-2} < 2F_{k-1}$. La condition $F_k^+ < F_{k+1}$ du théorème 4 n'est certes pas réalisée, mais $2F_{k-1} = F_k + F_{k-3}$, dont la décomposition gloutonne sur S_k a pour valence $1 + 1 = 2$ (voire 1 seulement quand $k = 2$). Nous homologuons le système.

L'analyse des deux dernières situations dicte une version plus opérationnelle du théorème 4 :

Théorème 5 (One-Point Theorem, version 2)**.** *Un système complet $S = (v_1 = 1, v_2, \ldots, v_n)$ est canonique si pour tout k, $2 \leq k \leq n-1$,*

$$g_k(v_k^+) \leq \frac{v_k^+}{v_{k-1}} \tag{51}$$

Remerciements

Les auteurs remercient l'équipe de relecteurs de l'ENS Paris, Maxime BERGER, Florian REVERCHON, et Frédéric JAËCK, pour leurs corrections et suggestions.

Références

[1] Bulletin Officiel de l'Éducation nationale, janvier 2019. https://cache.media.education.gouv.fr/file/SP1-MEN-22-1-2019/26/8/spe633_annexe_1063268.pdf.

[2] Laurent GODEFROY, « Algorithmes gloutons », https://www.supinfo.com/cours/2ADS/chapitres/06-algorithmes-gloutons, consulté en 2018–19.

[3] M.J. MAGAZINE, G.L. NEMHAUSER, L.E. TROTTER, Jr., "When the Greedy Solution Solves a Class of Knapsack Problems", Operations Research, 1975. https://www.jstor.org/stable/169525?seq=1#metadata_info_tab_contents.

[4] MATh.en.JEANS, Actes du congrès d'Angoulème, 2006. http://docplayer.fr/21135232-Decompositions-de-fractions-en-fractions-egyptiennes.html.

[5] Andy MIRZAIAN, "Greedy Coin Change Making", tutorial given at York University. https://slideplayer.com/slide/4071796/.

[6] Anna NIEWIAROWSKA and Michal ADAMASZEK, "Combinatorics of the Change-Making Problem", Preprint, 2007. https://arxiv.org/pdf/0801.0120.pdf. '

[7] Olympiades nationales de mathématiques, exercice national n°2, 2016. https://cache.media.eduscol.education.fr/file/Domaines_artistiques/31/8/sujet_2016_metropoleeuropeafriqueorient_817318.pdf.

[8] B.N. TIEN and T.C. HU, "Error Bounds and the Applicability of the Greedy Solution to the Coin-Changing Problem", Operations Research, 1977. https://www.jstor.org/stable/169929?seq=1#metadata_info_tab_contents.

[9] Wikipedia, "Greedy algorithm", https://en.wikipedia.org/wiki/Greedy_algorithm, consulté en 2018–19.

[10] Karim ZAYANA et Edwige CROIX, « Informat*h*ique », *Au fil des Maths*, 2018. https://perso.telecom-paristech.fr/kzayana/articles/Informathique_KZayana_ECroix.pdf.