Société Mathématique de France
Astérisque 24-25 (1975)

PROBLÈMES D'OPTIMISATION EN NOMBRES ENTIERS

par

Jean-Louis NICOLAS

-:-:-:-

§. I. - NOMBRES HAUTEMENT COMPOSÉS. - La définition suivante est due à Ramanujan ([6]) :

Un nombre n est hautement composé si tout nombre plus petit que n a strictement moins de diviseurs que n .

Soit $d(n)$ le nombre de diviseurs de n . Si la décomposition en facteurs premiers de n est : $n = \prod_i p_i^{\alpha_i}$, on sait que $d(n) = \prod_i (\alpha_i + 1)$. Le nombre n sera hautement composé, si l'on a :

$$m < n \Rightarrow d(m) < d(n) .$$

On trouvera les principaux résultats sur les nombres hautement composés dans [6], [1] et [4]. En particulier la quantité $Q(X)$ de nombres hautement composés inférieurs à X vérifie :

$$(\log X)^{C_1} \leq Q(X) \leq (\log X)^{C_2}$$

avec $C_i \to 1$ et la conjecture que :

$$\lim \frac{\log Q(X)}{\log \log X} = \frac{\log 30}{\log 16} = 1,227 \quad .$$

Soit a un entier ; cherchons $\max_{n \leq a} d(n)$, et soit n^* le plus petit entier où $d(n)$ atteigne ce maximum. Le nombre n^* est hautement composé et si l'on écrit :

$$(1) \qquad n = 2^{x_1} 3^{x_2} \ldots p_k^{x_k} \ldots \qquad \text{avec } x_i \geq 0$$

n^* est solution du problème de programmation mathématique en nombres entiers :

$$(2) \qquad \begin{cases} x_1 \log 2 + x_2 \log 3 + \ldots + x_k \log p_k + \ldots \leq A = \log a \\ \max(\log (x_1+1) + \log (x_2+1) \ldots + \log (x_k+1) + \ldots) \end{cases}$$

où p_k désigne le $k^{\text{ième}}$ nombre premier.

Rappelons qu'un problème de programmation mathématique consiste à maximiser une fonction de plusieurs variables, ici, $\sum_i \log(x_i+1)$, ces variables étant soumises à des contraintes, ici, une seule contrainte $\sum_i x_i \log p_i \leq A$. (cf. [2] et [3])

Il se trouve que les méthodes d'étude des nombres hautement composés, c'est-à-dire la résolution du problème de programmation (2) , peuvent s'appliquer à d'autres problèmes de programmation mathématique en nombres entiers.

§. II. - UN PROBLÈME DE T. L. SAATY. - Dans son livre ([7]) , "Optimization in integers and related extremal problems", T. L. Saaty pose le problème suivant :

PROBLÈME : Etant donné un entier C , trouver un entier n , et des entiers x_1 , x_2 , ... x_n satisfaisant la contrainte :

$$x_1 + x_2 + \ldots + x_n = C$$

et maximisant la quantité :

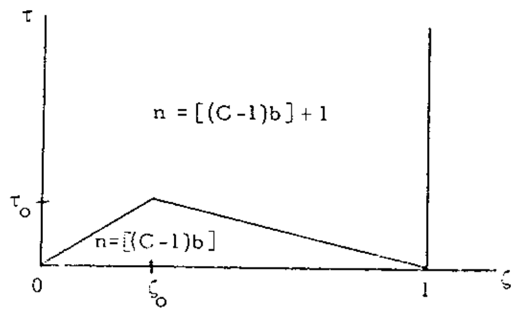
$$\prod_{i=1}^{n} x_i^i \ , \quad \text{ou ce qui est équivalent :} \ \sum_{i=1}^{n} i \log x_i \ .$$

<u>Solution</u> (cf. [5]). - Soit $a = \dfrac{\log 2}{\log 3} = 0,6309$, $b = \dfrac{3a\,(1-a)}{9a\,(1-a)-1} = 0,6375$ et $d = \dfrac{a}{9a\,(1-a)-1} = 0,5758$. On désigne par $[x]$ et $\{x\}$ les parties entières et fractionnaires de x . On pose $\tau = \{(C-1)b\}$.

1°) Si $\tau > \tau_o = \dfrac{9a\,(1-a)-2}{9a\,(1-a)-1} = 0,0874$, on a $n = [(C-1)b] + 1$.

2°) Si $\tau < \tau_o$, on pose $\zeta = \{(C-1)d\}$ et on a : $n = [(C-1)b]$ ou $[(C-1)b]+1$ suivant le schéma ci-dessous :

$$\zeta_o = \frac{(6a-1)(2-3a)}{9a\,(1-a)-1} = 0,2726$$



3°) Lorsque n est fixé, on a :

$$\begin{cases} x_1 = \ldots = x_r = 1 & \\ x_{r+1} = \ldots = x_{r+s} = 2 & \text{avec : } r = [(3n-C+1)(1-a)] \\ x_{r+s+1} = \ldots = x_n = 3 & \text{et : } r+s = [(3n-C+1)\,a] \end{cases}$$

Démonstration : 1ère étape. - Il est facile de voir que la solution vérifie :

$$1 \leq x_1 \leq x_2 \leq \ldots \leq x_n \leq 3 .$$

2ème étape. - On fixe n et on regarde la meilleure solution pour n fixé et $1 \leq x_1 \leq \ldots \leq x_n \leq 3$, ce qui permet de calculer, lorsque C est donné :

$$h(n) = \max_{\substack{x_1 + \ldots + x_n = C \\ n \text{ fixé, } x_i \in \{1,2,3\}}} \prod_{i=1}^{n} x_i^i .$$

3ème étape. - On cherche l'entier n qui maximise $h(n)$ et qui fournit ainsi la solution. Cette étape présente des difficultés de calcul, mais pas de difficultés théoriques.

§. III. - LES MULTIPLICATEURS DE LAGRANGE. - Revenons sur la deuxième étape de la démonstration précédente. On doit résoudre, pour n fixé, le problème de programmation mathématique :

$$(3) \qquad \begin{array}{l} x_1 + x_2 + \ldots + x_n = C \\ \max(\log x_1 + 2 \log x_2 + \ldots + n \log x_n) \end{array} \quad \text{avec } x_i \in \{1, 2, 3\} .$$

Si l'on devait résoudre en nombres réels ce problème qui est de la forme :

$$(4) \qquad \begin{cases} g(x) = g(x_1, \ldots, x_n) = C \\ \max f(x) = f(x_1, \ldots, x_n) \end{cases}$$

on utiliserait la méthode des multiplicateurs de Lagrange. On écrirait :

$$\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial g}{\partial x_1}} = \frac{\frac{\partial f}{\partial x_2}}{\frac{\partial g}{\partial x_2}} = \ldots = \frac{\frac{\partial g}{\partial x_n}}{\frac{\partial g}{\partial x_n}} = \rho$$

la valeur commune ρ de ces rapports s'appelant le multiplicateur de Lagrange. Autrement dit, on écrirait que pour une certaine valeur de ρ, la différentielle de la fonction $f(x) - \rho\, g(x)$ vérifie :

$$(5) \qquad d(f(x) - \rho\, g(x)) = 0$$

au point $x = (x_1, \dots, x_n)$ solution du problème 2 .

La justification de cette technique (cf. [2], chapitre 3, §. 4 et 5, et bien d'autres ouvrages) est basée sur le théorème des fonctions implicites et ne peut pas s'appliquer lorsque les variables $x_1, \dots, x_n$ sont entières. Mais on peut l'adapter en remplaçant la condition (5) par une condition plus restrictive :

(6) $\qquad f - \rho\, g$ a un maximum en x .

Cela nous donne le théorème :

THÉORÈME. (des multiplicateurs de Lagrange en nombres entiers) - Soit $E \subset \mathbb{R}^n$; f et g deux fonctions de E dans $\mathbb{R}$. On suppose que pour $\rho \in R$, $f - \rho\, g$ a un maximum absolu sur E qu'elle atteint en $x^* \in E$. On pose $C = g(x^*)$. Alors x^* est solution du problème de programmation mathématique :

$$\begin{cases} g(x) = C \\ \max\limits_{x \in E} f(x) \end{cases} .$$

Démonstration : Soit $x \in E$ tel que $g(x) = C$. On a :

$$f(x) - \rho\, g(x) \leq f(x^*) - \rho\, g(x^*) .$$

Comme $g(x) = g(x^*) = C$, cela entraine $f(x) \leq f(x^*)$.

Remarque : Cette démonstration peut servir, lorsque $E = \mathbb{R}^n$ pour justifier la technique des multiplicateurs de Lagrange en nombres réels. Mais elle est moins générale, car il peut se faire que la condition (5) soit réalisée, sans que la condition (6) le soit et qu'il y ait effectivement un extremum. Exemple :

$$\begin{cases} x_1 - x_2 = C \\ \max x_1^2 - 2x_2^2 \end{cases} .$$

Remarquons toutefois que si la fonction f est concave et g linéaire, $f - \rho g$ est concave et les conditions (5) et (6) sont équivalentes.

Application aux nombres hautement composés

On prend pour E l'ensemble des suites $(x_1, \dots, x_i, \dots)$ avec x_i entier ≥ 0, et un nombre fini de x_i non nuls. E est en bijection avec $\mathbb{N}$ par la formule (1). On pose :

$$g(x) = \sum_i x_i \log p_i$$

et
$$f(x) = \sum_i \log (x_i + 1) .$$

Pour tout $\varepsilon > 0$, on sait que $\lim_{n\to\infty} \frac{d(n)}{n^\varepsilon} = 0$. Par conséquent $\frac{d(n)}{n^\varepsilon}$ a un maximum absolu sur $\mathbb{N}$ et donc $f(x) - \varepsilon\, g(x)$ a un maximum absolu sur E. Le (ou les) point(s) où ce maximum est atteint est d'après le théorème précédent hautement composé. L'ensemble de ces points, pour les différentes valeurs de ε, constitue l'ensemble des nombres que Ramanujan a appelés hautement composés supérieurs (cf. [6], §. 32).

Application au problème (3)

On applique le théorème précédent avec $E = \{1, 2, 3\}^n \subset \mathbb{R}^n$. E est un ensemble fini et pour tout ρ, la fonction $\sum_{i=1}^{n} (i \log x_i - \rho\, x_i)$ admet un maximum sur E qu'elle atteint en $x^* = (x_1^*, \dots, x_n^*)$. Les coordonnées de x_i^* sont aisément calculables : Pour chaque i, x_i^* réalise le maximum de $(i \log x_i - \rho\, x_i)$ et pour $\rho > 0$, on a :

$$\begin{cases} x_1^* = \dots = x_r^* = 1 & \text{avec} \quad r = \min(n, [\frac{\rho}{\log 2}]) \\ x_{r+1}^* = \dots = x_{r+s}^* = 2 & \text{avec} \quad r+s = \min(n, [\frac{\rho}{\log 3/2}]) \\ x_{r+s+1}^* = \dots = x_n^* = 3 & \end{cases}$$

On calcule ensuite $g(x^*) = r+2s+3(n-r-s)$ en fonction de ρ. Et il se trouve que l'on peut choisir ρ pour que $g(x^*)$ prenne toutes les valeurs C possibles entre n et $3n$, ce qui permet de résoudre complètement le problème 1.

§. IV. - <u>LA MÉTHODE DES BÉNÉFICES.</u> - Pour résoudre des problèmes de programmation mathématique, la méthode précédente s'applique sans difficultés si la fonction f est concave, si g est linéaire et si f et g sont séparables (i. e. somme de fonctions portant chacune sur une coordonnée) et même dans des conditions un peu plus générales. Ce qui est exceptionnel dans le problème précédent, c'est que l'on puisse choisir ρ pour que $g(x^*)$ prenne toutes les valeurs C possibles. Cela vient de ce que les coefficients de la forme linéaire g sont égaux à 1.

S'il n'en est pas ainsi, au lieu de se contenter de chercher le maximum de la fonction $f - \rho g$, on range les éléments x de l'ensemble discret E suivant les valeurs décroissantes de la fonction $f - \rho g$. On obtient ainsi une suite $x^{(1)} = x^*, x^{(2)}, \dots, x^{(k)}, \dots$ avec :

$$(f-\rho g)(x^{(1)}) \geq (f-\rho g)(x^{(2)}) \geq \dots \geq (f-\rho g)(x^{(k)}) \geq \dots$$

Si l'on a $g(x^{(2)}) = C_2 \neq g(x^*) = C_1$, alors $x^{(2)}$ sera solution du problème de programmation :

$$\begin{cases} g(x) = C_2 \\ \max\limits_{x \in E} f(x) \end{cases}$$

Si l'on a $C_2 > C_1$ et si l'on veut que $x^{(2)}$ soit une solution du problème analogue, avec la contrainte $g(x) \leq C_2$, il faudra s'assurer en plus que l'on a $f(x^{(2)}) \geq f(x^*)$.

On peut continuer en examinant $x^{(3)}$: Si l'on a $g(x^{(3)}) = C_3$ avec $C_3 \neq C_2$ et $C_3 \neq C_1$, alors $x^{(3)}$ sera solution du problème de programmation :

$$\begin{cases} g(x) = C_3 \\ \max f(x) \end{cases} .$$

Pour chaque k, on appelle bénéfice de $x^{(k)}$ la quantité :
bén $x^{(k)} = (f-\rho g)(x^*) - (f-\rho g)(x^{(k)})$.

D'après la définition de la suite $x^{(k)}$, bén $x^{(k)}$ croît avec k. Grâce à un théorème de majoration des bénéfices (cf. [3] et [4] proposition 2), il suffit d'examiner $x^{(k)}$ pour les petites valeurs de k : lorsque bén $x^{(k)}$ devient trop grand, $x^{(k)}$ ne sera sûrement pas solution du problème de programmation :

$$\begin{cases} f(x) \leq C_k = g(x^{(k)}) \\ \max f(x) \end{cases} .$$

<u>Problème ouvert</u> : Etant donné un entier C, trouver des entiers n, $x_1, x_2, \dots, x_n$ tels que

$$x_1 + 2x_2 + 3x_3 + \dots + n\,x_n = C$$

et qui maximisent le produit :

$$(x_1+1)(x_2+1)\dots(x_n+1) .$$

RÉFÉRENCES

[1] L. ALAOGLU and P. ERDÖS. - On highly composite and similar numbers. Trans. Amer. Math. Soc. 56, (1944), p. 448-469.

[2] G. HADLEY. - Non linear and dynamic programming. Reading, Palo Alto London, Addison-Wesley publishing Company, (1964) (Addison-Wesley series in management science and economics).

[3] J. L. NICOLAS. - Des exemples de programmation non linéaire en théorie des nombres. Séminaire de théorie des nombres Delange-Pisot-Poitou, 14e année, 1972-73, n° 10, 11 pages.

[4] J. L. NICOLAS. - Répartition des nombres hautement composés de Ramanujan. Can. J. Maths, vol. XXIII, n° 1, (1971) p. 116-130.

[5] J. L. NICOLAS. - Sur un problème d'optimisation en nombres entiers de T. S. Saaty. A paraître dans R.A.I. R.O. , 1975.

[6] S. RAMANUJAN. - Highly composite numbers. Proc. London Math. Soc., Series 2, t. 14, (1915), p. 347-400 ; and Collected papers, p. 78-128 - Cambridge at the University Press, (1927).

[7] T. L. SAATY. - Optimization in integers and related extremal problems. Mc Graw-Hill, (1970).

-:-:-:-

Jean-Louis NICOLAS
Département de
Mathématiques
U. E. R. des Sciences de
Limoges
123, rue Albert Thomas
87100 LIMOGES