

Astérisque

JEAN-LOUIS NICOLAS

Algorithmes d'optimisation en nombres entiers

Astérisque, tome 38-39 (1976), p. 169-182

<http://www.numdam.org/item?id=AST_1976__38-39__169_0>

© Société mathématique de France, 1976, tous droits réservés.

L'accès aux archives de la collection « Astérisque » (<http://smf4.emath.fr/Publications/Asterisque/>) implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques

<http://www.numdam.org/>

ALGORITHMES D'OPTIMISATION EN NOMBRES ENTIERS

par

Jean-Louis NICOLAS

§ 1.- INTRODUCTION

Le problème général de programmation mathématique (ou d'optimisation) est le suivant : on cherche à maximiser ou minimiser une fonction de k variables réelles :

$$f(x_1, x_2, \dots, x_k)$$

lorsque ces variables sont soumises à des contraintes :

$$g_\alpha(x_1, x_2, \dots, x_k) \leq \text{ou} = A_\alpha ; \quad \alpha \in I.$$

Quand la fonction f et les contraintes g_α sont linéaires, la programmation est dite linéaire. Si on limite les variables à ne prendre que des valeurs entières, on a un problème de programmation mathématique en nombres entiers.

Nous allons, dans ce papier, envisager quelques algorithmes pour calculer les solutions du problème :

$$\left\{ \begin{array}{l} g(x_1, x_2, \dots, x_k) \leq C \\ \max f(x_1, \dots, x_k) \end{array} \right.$$

ou f et g sont des fonctions de \mathbb{N}^k dans \mathbb{R} , et C un nombre réel.

Ramanujan ([11]) dit qu'un nombre N est hautement composé si tout nom-

bre $n < N$ a strictement moins de diviseurs que N .

Appelons $d(n)$ le nombre de diviseurs de n . On sait que la fonction d est multiplicative (si m et n sont premiers entre eux, $d(mn) = d(m) d(n)$) et si la décomposition en facteurs premiers de n est $n = \prod p_i^{\alpha_i}$, alors $d(n) = \prod (\alpha_i + 1)$.

On a donc : N est hautement composé si et seulement si :

$$n < N \implies d(n) < d(N).$$

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

$$n = 2^{x_1} 3^{x_2} \dots p_k^{x_k} \dots \quad \text{avec } x_i \geq 0,$$

n^* est solution du problème de programmation suivant :

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

Désignons par $g(n)$ l'ordre maximum d'un élément du groupe des permutations S_n . Cette fonction g intervient dans plusieurs problèmes d'informatique théorique. On peut montrer (cf : [3] ou [4]) que :

$$g(n) = \max_{n_1 + n_2 + \dots + n_k = n} [p.p.c.m.(n_1, n_2, \dots, n_k)]$$

et que :

$$g(n) = \max_{\ell(j) \leq n} j$$

où la fonction ℓ est ainsi définie :

$$\text{si } n = \prod p_i^{\alpha_i}, \alpha_i \geq 1, \quad \text{alors } \ell(n) = \sum p_i^{\alpha_i} \quad \text{et } \ell(1) = 0,$$

(ainsi $\ell(180) = \ell(4.9.5) = 4 + 9 + 5 = 18$).

Le problème de calculer $g(n)$ est un problème de programmation :

$$\begin{cases} \ell(2^{x_1}) + \ell(3^{x_2}) + \dots + \ell(p_k^{x_k}) \leq n \\ \max j = x_1 \log 2 + x_2 \log 3 + \dots + x_k \log p_k \end{cases}$$

avec $\ell(p^x) = p^x$ si $x \geq 1$ et $\ell(p^0) = 0$.

§.II.- CLASSIFICATION DES SOLUTIONS

Etant donné deux fonctions $f(x_1, \dots, x_k)$ et $g(x_1, \dots, x_k)$ définies sur \mathbb{N}^k , croissantes sur chaque coordonnée x_i , à valeurs réelles, on considère le problème d'optimisation en nombres entiers :

$$\mathcal{P} \begin{cases} g(x_1, \dots, x_k) \leq C \\ \max f(x_1, \dots, x_k) \end{cases}$$

pour différentes valeurs de $C \in \mathbb{R}$.

Dans l'ensemble des vecteurs $x = (x_1, x_2, \dots, x_k) \in \mathbb{N}^k$ vérifiant $g(x) \leq C$, le maximum de f est atteint en un ou plusieurs points qui sont solutions du problème \mathcal{P} et que l'on classe en deux catégories :

- on dira que $x^* \in \mathbb{N}^k$ est une solution minimale de \mathcal{P} si :

$$\forall x \in \mathbb{N}^k, \quad g(x) < g(x^*) \implies f(x) < f(x^*).$$

- on dira que x^* est une solution large de \mathcal{P} si :

$$\forall x \in \mathbb{N}^k, \quad g(x) \leq g(x^*) \implies f(x) \leq f(x^*).$$

Pour résoudre le problème \mathcal{P} pour toutes les valeurs de C réelles, il est inutile de considérer toutes ces valeurs. Par exemple, si $g(x)$ est une forme linéaire $\sum a_i x_i$ à coefficients a_i entiers, on peut se restreindre à étudier les cas où C est entier. On distinguera donc deux sortes de bornes pour le problème \mathcal{P} :

- On dit que C est une borne intéressante de \mathcal{P} s'il existe au moins une solution minimale x^* avec $g(x^*) = C$.

- On dit que C est une borne largement intéressante de \mathcal{P} s'il existe au moins une solution large x^* de \mathcal{P} avec $g(x^*) = C$.

La connaissance des bornes intéressantes et largement intéressantes du problème \mathcal{P} , ainsi que des solutions correspondantes, permet de résoudre complètement le problème \mathcal{P} pour toutes les valeurs de C.

§.III.- PROGRAMMATION SANDWICH

On considère les trois problèmes suivants avec deux variables, notées x et y, et contrainte linéaire :

$$\mathcal{P}_1 \begin{cases} x + ay \leq A \\ \max f(x,y) \end{cases} \quad \mathcal{P}_2 \begin{cases} x + by \leq B \\ \max f(x,y) \end{cases} \quad \mathcal{P}_3 \begin{cases} x + cy \leq C \\ \max f(x,y) \end{cases}$$

avec $0 < a < b < c$.

PROPOSITION 1.- Si (x^*, y^*) est une solution minimale (resp. large) de \mathcal{P}_1 et \mathcal{P}_3 , c'est une solution minimale (resp. large) de \mathcal{P}_2 .

Démonstration : On a les identités :

$$i) \quad x^* + cy^* - (x + cy) = (x^* + by^*) - (x + by) + (c - b)(y^* - y)$$

$$ii) \quad x^* + ay^* - (x + ay) = (x^* + by^*) - (x + by) + (a - b)(y^* - y) .$$

Supposons que (x^*, y^*) soit une solution minimale de \mathcal{P}_1 et \mathcal{P}_3 , on a :

$$x + ay < x^* + ay^* \implies f(x,y) < f(x^*, y^*)$$

et

$$x + cy < x^* + cy^* \implies f(x,y) < f(x^*, y^*) .$$

Soit maintenant (x,y) tel que $x + by < x^* + by^*$. Si $y \leq y^*$, i) donne $x + cy < x^* + cy^*$ ce qui entraîne $f(x,y) < f(x^*, y^*)$.

Si $y \geq y^*$, ii) donne $x + ay < x^* + ay^*$, ce qui entraîne $f(x,y) < f(x^*, y^*)$.

On voit ainsi que (x^*, y^*) est une solution minimale de P_2 . Le cas d'une solution large se traiterait de même en remplaçant partout les inégalités strictes par des inégalités larges.

On démontrerait de même la proposition :

PROPOSITION 2. - Soient A_1 et A_2 deux bornes intéressantes (resp. largement intéressantes) consécutives de P_1 ayant une solution minimale (resp. large) : (x^*, y^*) et (x^{**}, y^{**}) . On suppose que $x^* + cy^* = C_1$ et $x^{**} + cy^{**} = C_2$ sont deux bornes intéressantes (resp. largement intéressantes) consécutives de P_3 . On pose $x^* + by^* = B_1$ et $x^{**} + by^{**} = B_2$. Alors B_1 et B_2 sont deux bornes intéressantes (resp. largement intéressantes) consécutives de P_2 avec comme solution unique (x^*, y^*) et (x^{**}, y^{**}) .

Des propositions 1 et 2, on déduit immédiatement le corollaire :

Corollaire : Si les tables des solutions strictes (resp. larges) des problèmes P_1 et P_3 coïncident jusqu'à un certain rang, c'est aussi la table des solutions strictes (resp. larges) du problème P_2 .

Exemple :

$$P_1 \begin{cases} 2x + 3y \leq A \\ \max(x+1)(y+1) \end{cases} \quad P_2 \begin{cases} (\log 2)x + (\log 3)y \leq B \\ \max(x+1)(y+1) \end{cases} \quad P_3 \begin{cases} 5x + 8y \leq C \\ \max(x+1)(y+1) \end{cases}$$

Les solutions P_1 et P_3 se calculent facilement. Comme elles coïncident jusqu'à $x=6, y=2$, ce sont aussi les solutions de P_2 . En remplaçant dans P_1 et P_3 la contrainte par $px + qy$ où $\frac{p}{q}$ représente les convergents successifs dans le développement en fractions continues de $\frac{\log 2}{\log 3}$, on peut obtenir une table des solutions de P_2 aussi longue que l'on veut.

Sur le plan théorique, G. BESSI a caractérisé les solutions de P_2 et sa méthode, tout à fait effective, fournit un autre algorithme de calcul.

D'autre part, les propositions ci-dessus sont un embryon d'une théorie à développer de l'approximation d'un problème de programmation par un autre.

§ IV.- PROGRAMMATION DYNAMIQUE

On veut résoudre le problème suivant :

Trouver des variables entières et positives vérifiant $\sum_{i=1}^k a_i x_i \leq A, a_i > 0,$
 a_i entiers, qui maximisent $f(x) = \sum_{i=1}^k f_i(x_i).$

On suppose ici qu'il n'y a qu'une seule contrainte linéaire à coefficients entiers et que la fonction à optimiser est séparable. Remarquons d'abord que le nombre N de solutions possibles est fini : on doit avoir $0 \leq x_i \leq A/a_i$ pour chaque i. L'algorithme de programmation dynamique (cf. [2], ch.10) est un procédé d'énumération des solutions possibles.

On définit pour $j = 1, 2, \dots, k$ et $0 \leq M \leq A$

$$\Delta_j(M) = \max_{\sum_{1 \leq i \leq j} a_i x_i \leq M} \left(\sum_{i=1}^j f_i(x_i) \right).$$

On voit que $\Delta_k(A)$ donnera la solution du problème. D'autre part, les $\Delta_j(M)$ se calculent par récurrence :

$$\Delta_1(M) = \max_{x_1 \leq M/a_1} f_1(x_1)$$

et :

$$\begin{aligned} \Delta_j(M) &= \max_{0 \leq x_j \leq \frac{M}{a_j}} [f_j(x_j) + \Delta_{j-1}(M - a_j x_j)] \\ &= \max [f_j(0) + \Delta_{j-1}(M), f_j(1) + \Delta_{j-1}(M - a_j), \dots]. \end{aligned}$$

Remarquons que dans ([4]) pour faire une table de la fonction g(n), définie dans l'introduction, jusqu'à n = 8000, l'algorithme utilisé était un algorithme de programmation dynamique.

§ V.- LES MULTIPLICATEURS DE LAGRANGE

Si l'on devait résoudre en nombres réels le problème :

$$\rho \left\{ \begin{array}{l} g(x) = g(x_1, \dots, x_k) = C \\ \max f(x) = f(x_1, \dots, x_k) \end{array} \right.$$

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

$$\frac{\frac{\partial f}{\partial x_1}}{\frac{\partial g}{\partial x_1}} = \frac{\frac{\partial f}{\partial x_2}}{\frac{\partial g}{\partial x_2}} = \dots = \frac{\frac{\partial f}{\partial x_k}}{\frac{\partial g}{\partial x_k}} = \rho \quad ,$$

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 :

$$(1) \quad d(f(x) - \rho g(x)) = 0$$

au point $x = (x_1, \dots, x_k)$ solution du problème ρ

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_k sont entières. Mais on peut l'adapter en remplaçant la condition (1) par une condition plus restrictive :

$$f - \rho g \text{ a un maximum en } x .$$

Cela nous donne le théorème :

THÉORÈME 1.- (des multiplicateurs de Lagrange en nombres entiers) - Soient $E \in \mathbb{R}^n$; f et g deux fonctions de E dans \mathbb{R} . On suppose que pour $\rho \in \mathbb{R}^+$, $f - \rho g$ 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 :

$$\left\{ \begin{array}{l} g(x) \leq C \\ \max_{x \in E} f(x) \end{array} \right.$$

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

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

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

Application : Construction des bornes hautement intéressantes : soit le problème de programmation en nombres entiers :

$$(2) \quad \left\{ \begin{array}{l} g(x) = \sum_{i=1}^k a_i x_i \leq A ; \quad a_i \geq 0 ; \quad x_i \geq 0 ; \quad x_i \text{ entier} \\ \max f(x) = \sum_{i=1}^k f_i(x_i) \quad , \quad f_i \text{ concave} \end{array} \right.$$

La différence entre ce problème et le problème de programmation dynamique du § IV est : les a_i peuvent ne pas être entiers mais les fonctions f_i doivent être concaves, c'est-à-dire :

$$f_i(n+1) - f_i(n) \leq f_i(n) - f_i(n-1) \quad \text{pour } n \text{ entier } \geq 1 .$$

Etant donné ρ réel, positif, on cherche si la fonction $f(x) - \rho g(x)$ a un maximum absolu en x^* .

Pour qu'une fonction séparable $\Phi(x) = \sum_i \Phi_i(x_i)$ ait un maximum en $x = (x_1, x_2, \dots, x_k)$, il faut et il suffit que pour tous les i , Φ_i ait un maximum en x_i . Pour que $f(x) - \rho g(x)$ ait un maximum en x^* il faut que $f_i(x_i) - \rho a_i x_i$ ait un maximum en x_i^* , ce qui entraîne pour tout i :

$$f_i(x_i^* + 1) - \rho a_i (x_i^* + 1) \leq f_i(x_i^*) - \rho a_i x_i^*$$

et :

$$f_i(x_i^* - 1) - \rho a_i (x_i^* - 1) \leq f_i(x_i^*) - \rho a_i x_i^*$$

ce qui donne :

$$(3) \quad f_i(x_i^* + 1) - f_i(x_i^*) \leq \rho a_i \leq f_i(x_i^*) - f_i(x_i^* - 1) .$$

Comme les fonctions f_i ont été supposées concaves, les inégalités (3) entraînent :

$$(4) \quad \frac{f_i(1) - f_i(0)}{a_i} \geq \frac{f_i(2) - f_i(1)}{a_i} \geq \dots \geq \frac{f_i(x_i^*) - f_i(x_i^* - 1)}{a_i} \geq \rho \geq \frac{f_i(x_i^* + 1) - f_i(x_i^*)}{a_i} .$$

Le nombre ρ étant fixé, les inégalités (4) déterminent la valeur de x_i^* (si les inégalités sont strictes). On pose alors : $A = \sum a_i x_i^*$ et on dit que A est une borne de Lagrange du problème (2) associée à ρ . En faisant varier ρ , on obtiendra diverses solutions du problème (2) pour des bornes de Lagrange $A_{n_1}, A_{n_2}, \dots, A_{n_j}$ qui forment une sous-suite de l'ensemble des bornes intéressantes.

Exemples : On trouvera des exemples de cette méthode en théorie des nombres dans [11], où Ramanujan définit les nombres hautement composés supérieurs et dans [3], chap. 3 où l'on calcule de façon très simple $g(n)$ pour une suite infinie d'entiers n . Citons ici un exemple plus concret :

Le stockage des pièces de rechange d'un sous-marin. Dans le livre de Hadley [2], p. 362, un problème de stockage des pièces de rechange pour un sous-marin et traité par la programmation dynamique, revient à minimiser la fonction :

$$f = \sum_{j=1}^3 f_j = \sum_{j=1}^3 \pi_j \phi_j ,$$

avec $\pi_j = \$ 800 ; 600 ; 1300$ et :

$$\phi_j(x) = \begin{cases} \mu_j P(x-1, \mu_j) - x P(x, \mu_j) & \text{si } x \geq 1 \\ \mu_j & \text{si } x = 0 \end{cases}$$

avec $\mu_j = 4 ; 2 ; 1$.

La fonction $P(x, \mu)$ est la fonction cumulative de Poisson :

$$P(x, \mu) = \sum_{n=x}^{\infty} \frac{\mu^n}{n!} e^{-\mu}$$

et la contrainte est $x_1 + 2x_2 + 2x_3 \leq A = 10$.

La fonction $f = \sum_{j=1}^3 f_j$ est séparable et les fonctions f_j sont convexes.

Les valeurs de la fonction $P(x, \mu)$ sont données par les tables (cf : [13]).

On en déduit la table des valeurs de $\frac{f_i(x_i - 1) - f_i(x_i)}{a_i}$ qui vaut

$$\frac{\pi_i}{a_i} [f_i(x_i - 1) - f_i(x_i)] = \frac{\pi_i}{a_i} P(x, \mu_i) :$$

(5)

	a_i	π_i	$x_i=1$	$x_i=2$	$x_i=3$	$x_i=4$	$x_i=5$	$x_i=6$	$x_i=7$
i=1	1	800	785,36	726,72	609,52	453,20	296,96	171,92	88,56
i=2	2	600	259,38	178,20	96,96	42,84			
i=3	2	1300	410,80	171,73	52,13				

On obtient alors le tableau des bornes de Lagrange :

ρ	x^*	$A = g(x^*)$	$f(x^*)$
	0 0 0	0	5700
	785,36 1 0 0	1	4914
	726,72 2 0 0	2	4188
	609,52 3 0 0	3	3578
(6)	453,20 4 0 0	4	3125
	410,80 4 0 1	6	2285
	296,96 5 0 1	7	1989
	259,38 5 1 1	9	1470
	178,20 5 2 1	11	1114
	171,92 6 2 1	12	942
	171,73 6 2 2	14	599

§ 6.- LA MÉTHODE DES "BÉNÉFICES"

L'inconvénient principal de la méthode exposée au paragraphe précédent est de ne pas fournir toutes les bornes intéressantes : dans l'exemple du sous-marin, le tableau (6) ne donne pas la solution pour $A = 5, 8, 10, 13$. Nous allons voir comment on peut y remédier. Les hypothèses sont toujours celles du problème (2).

Soit A une borne de Lagrange associée à un nombre ρ et la solution x^* correspondante du problème (2).

Soit $x = (x_1, \dots, x_k)$ un vecteur à coordonnées entières. On définit le bénéfice de x , $\text{bén}(x)$, par la formule :

$$(7) \quad \text{bén}(x) = f(x^*) - \rho g(x^*) - (f(x) - \rho g(x)).$$

On pose :

$$\text{bén } x = \sum_{i=1}^k \text{bén}(x_i)$$

avec :

$$\text{bén } x_i = f_i(x_i^*) - f_i(x_i) - \rho(g_i(x_i^*) - g_i(x_i)) .$$

Remarque : Le bénéfice, défini par (7), dépend non seulement de la solution x^* de Lagrange, mais aussi de ρ . D'autre part, il résulte du § 5, que pour tout i , on a : $\text{bén}(x_i) \geq 0$.

THÉORÈME 2.- (de majoration des bénéfiques, cf. [6], proposition 2).

Soient $y = (y_1, \dots, y_k)$ et y' deux solutions possibles du problème (2) et soit z une solution exacte du même problème avec la valeur $A = g(z)$ et vérifiant l'une ou l'autre condition :

- i) $g(y) \leq g(z) \leq g(y')$
- ii) $f(y) \leq f(z) \leq f(y')$.

Soit x une solution de Lagrange du problème (2) associée à ρ , par rapport à laquelle on définit les bénéfiques. Alors, on a :

$$(8) \quad \text{bén } z \leq \text{bén } y + \rho [g(y') - g(y)] = \text{bén } y' + f(y') - f(y).$$

Démonstration : Comme z est une solution exacte, l'hypothèse (i) implique :

$$g(y) \leq g(z) = A \Rightarrow f(y) \leq f(z).$$

De même, ii) entraîne :

$$f(y') \geq f(z) \Rightarrow g(y') \geq A = g(z).$$

Dans les deux cas on a :

$$\begin{aligned} \text{bén } z &= f(x) - f(z) - \rho [g(x) - g(z)] \\ &\leq f(x) - f(y) - \rho [g(x) - g(y')]. \end{aligned}$$

A l'aide de la définition des bénéfiques (7), on obtient (8).

Dans la pratique, on cherchera des solutions possibles y, y', y'', \dots entre deux solutions de Lagrange consécutives, et on majorera avec le théorème précédent le bénéfice d'une solution intéressante éventuelle. On éliminera alors les valeurs de $\xi_i = z_i - x_i$ qui auraient un trop grand bénéfice, il ne restera plus qu'un petit nombre de valeurs de $\xi = z - x$ à essayer.

Exemple : Problème du sous-marin : $\rho = 250$, la solution de Lagrange est $(x_1, x_2, x_3) = (5, 1, 1)$. Le tableau des bénéfiques peut s'obtenir à partir du tableau (5) en modifiant légèrement la théorie, pour l'adapter à un problème de minimisation :

	bén($\xi_i = -2$)	bén($\xi_i = -1$)		bén($\xi_i = 1$)	bén($\xi_i = 2$)
i=1	249	46,96		78,08	240
(22) i=2	-	18,76		143,6	451
i=3	-	321		157	553

On cherche une solution z telle que $g(z) = 10$. On a une solution possible $y = (6, 1, 1)$. On applique le théorème 2 avec la condition i) et $y = y'$.

On obtient :

$$\text{bén}(z) \leq \text{bén}(y) = 78,08\dots$$

Les composantes de $\xi = z - x$, si $\text{bén}(z) < 78$ doivent être : $\xi_1 = 0$ ou -1 , $\xi_2 = 0$ ou -1 , $\xi_3 = 0$. Dans tous les cas, on aurait $g(z) < g(x)$. La solution du problème de l'exemple N°2 pour $A = 10$ est donc $x = (6, 1, 1)$.

On pourrait montrer de même que pour $A = 13$, la solution est $(5, 2, 2)$.

§ VII.- QUELQUES PROBLÈMES

Utiliser la méthode des bénéfiques pour faire une table des nombres hautement composés de Ramanujan, ou pour calculer $g(n)$, définie dans l'introduction, pour de grandes valeurs de n sans avoir à calculer $g(m)$ pour tous les $m < n$. Exemple, on a comme solution de Lagrange, pour $\rho = 222$:

$$g(198\ 976) = 2^8 \cdot 3^5 \cdot 5^3 \cdot 7^3 \cdot 11^2 \cdot 13^2 ; 17^2 \cdot 19^2 \cdot 23^2 \prod_{29 \leq p \leq 1637} p$$

et pour $\rho = 224$, $g(200\ 633) = 1657\ g(198\ 976)$. Calculer $g(200\ 000)$.

Les multiplicateurs de Lagrange en nombres entiers s'adaptent au cas de plusieurs contraintes, en particulier, ils s'appliquent au problème :

"Trouver $\max d(n)$ lorsque $n \leq a$ et $\Omega(n) \leq b$ "

(pour $n = \prod_i p_i^{\alpha_i}$, on a $\Omega(n) = \sum_i \alpha_i$ et $d(n) = \prod_i (\alpha_i + 1)$).

Mais les bornes de Lagrange a et b vont dépendre de façon plus compliquée de deux multiplicateurs ρ et λ .

Adapter les méthodes des § V et § VI à des fonctions non séparables, ou non concaves (en utilisant l'enveloppe concave).

RÉFÉRENCES

- [1] L. ALAOGU and P. ERDOS - 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 - Ordre maximal d'un élément du groupe des permutations
et highly composite numbers, Bull.Soc.Math. France 97
(1969), p. 129-191.
- [4] J.L. NICOLAS - Calcul de l'ordre maximum d'un élément du groupe symé-
trique S_n , R.I.R.O. 3^e année, N° R-2/1969, p.43-50.
- [5] J.L. NICOLAS - Des exemples de programmation non linéaire en théorie
des nombres. Séminaire de théorie des nombres Delange-
Pisot-Poitou, 14^e année,1972-73, N°10, 11 pages.
- [6] J.L. NICOLAS - Répartition des nombres hautement composés de Ramanujan
Can.J.Maths, Vol XXIII, N°1, (1971), p. 116-130.
- [7] J.L. NICOLAS - Sur un problème d'optimisation en nombres entiers de
T.L. Saaty, R.A.I.R.O., 9^e année, vol.2, p. 67-82.
- [8] J.L. Nicolas - Problèmes d'optimisation en nombres entiers. Astéris-
que 24-25, (1975) p. 325-333.
- [9] S. PILLAI - Highly abundant numbers, Bull. Calcutta math. soc. t.35
1943, p. 141-156.
- [10] S. PILLAI - Highly composite number J. Indian Math.Soc. t.8, 1944,
p. 61-74.
- [11] S. RAMANUJAN - Highly composite numbers. Proc. London Math. Soc.,
Séries 2, t.14, (1915), p. 347-400 ; and Collected papers,
p.78-128, Cambridge at the University Press, (1927).
- [12] T.L. SAATY - Optimization in integers and related extremal problems.
Mc Graw-Hill, (1970).
- [13] Tables of individual and cumulative terms of Poisson's distribution,
Van Nostrand Company Inc.

* * *

Jean-Louis NICOLAS

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