RAIRO. INFORMATIQUE THÉORIQUE

G. ROBIN

Méthodes d'optimisation pour un problème de théorie des nombres

RAIRO. Informatique théorique, tome 17, n° 3 (1983), p. 239-247

<http://www.numdam.org/item?id=ITA_1983__17_3_239_0>

R.A.I.R.O. Informatique théorique/Theoretical Informatics
(vol. 17, n° 3, 1983, p. 239 à 247)

MÉTHODES D'OPTIMISATION POUR UN PROBLÈME DE THÉORIE DES NOMBRES (*)

par G. ROBIN (1)

Communiqué par J. E. PIN

Résumé. — *D'après Ramanujan, N est dit hautement composé s'il a plus de diviseurs que tous les nombres qui le précèdent. Nous proposons deux algorithmes de construction de tels nombres. Un algorithme de programmation dynamique nous permet de déterminer la liste de tous les nombres hautement composés inférieurs à un nombre donné. Le deuxième algorithme utilise les multiplicateurs de Lagrange généralisés, le théorème d'Everett, la méthode des bénéfices de J. L. Nicolas et permet de déterminer une tranche de nombres hautement composés.*

Mots clés : Programmation en nombres entiers, programmation dynamique, calculs et programmation en théorie des nombres.

Abstract. — *With Ramanujan, we shall say that N is a highly composite number if N has more divisors than all the numbers less than N. In this paper we show how dynamic programming, generalized Lagrange multipliers method, Everett 'theorem, Nicolas' benefit method allow us to construct first the table of all the highly composite numbers $\leq N_0$, N_0 given and then an interval of highly composite numbers between two given numbers.*

Key Words: Integer programming, non linear-programming, dynamic programming, explicit machine computation and programs in number theory, multiplicative theory.

1. NOMBRES HAUTEMENT COMPOSÉS ET NOMBRES HAUTEMENT COMPOSÉS SUPÉRIEURS

Les nombres hautement composés et les nombres hautement composés supérieurs ont été définis par Ramanujan [11]. Ils ont été ensuite étudiés par Alaoglu, Erdös [1], Erdös [2] et Nicolas [8].

On sait que tout entier n s'écrit $n = \prod_{k=1}^{\infty} p_k^{x_k}$ où $p_1 = 2$, $p_2 = 3$, p_k le k-ième nombre premier et $x_k \geq 0$.

On a alors $d(n) = \prod_{k=1}^{\infty} (x_k + 1)$.

DÉFINITION : *N est dit hautement composé* $\Leftrightarrow \forall n < N$, $d(n) < d(N)$.

(*) Reçu juillet 1982, révisé décembre 1982.
(1) U.E.R. des Sciences de Limoges, Département de Mathématiques, 123, rue Albert-Thomas, 87060 Limoges Cedex.

R.A.I.R.O. Informatique théorique/Theoretical Informatics, 0399-0540/1983/239/$ 5.00

Considérons le problème de programmation mathématique :

$$(P) \qquad \begin{cases} \text{Max } d(n), \\ \text{tel que } n \leqq N_0, \end{cases}$$

équivalent au problème de programmation en nombres entiers x_k :

$$(P') \qquad \begin{cases} \text{Max} \sum_{k=1}^{\infty} \log(x_k+1), \\ \text{avec} \quad \sum_{k=1}^{\infty} x_k \log p_k \leqq \log N_0. \end{cases}$$

Si N_0 est un nombre hautement composé, N_0 est la seule solution optimale de (P). Si N_0 n'est pas hautement composé, soit N le plus grand nombre hautement composé $< N_0$ alors la valeur du programme est $v(P) = d(N)$ et (P) a comme solutions optimales tous les nombres N', appelés largement composés, tels que $N \leqq N' < N_0$ avec $d(N) = d(N')$.

REMARQUE : N_0 est aussi solution optimale du problème :

$$(P_1) \qquad \begin{cases} \text{Min } n, \\ \text{tel que } d(n) \leqq D \end{cases}$$

pour $D = d(N_0)$.

Pour D quelconque, l'unique solution optimale de (P_1) est un nombre hautement composé.

Considérons le problème de relaxation lagrangienne associé au programme (P) [4] :

$$(PR)_{\lambda>0} : \quad \text{Max } \varphi_\lambda(n) = \log d(n) - \lambda \log n.$$

La définition de Ramanujan des nombres hautement composés supérieurs peut s'interpréter de la façon suivante ([9], [10]).

DÉFINITION : *N est dit hautement composé supérieur $\Leftrightarrow \exists \lambda > 0$ tel que N est solution optimale de* $(PR)_\lambda$.

D'après le théorème d'Everett ([3, 5, 7], p. 77), si N^* est optimal pour $(PR)_\lambda$ alors N^* est optimal pour (P) avec $N_0 = N^*$. Ainsi les nombres hautement composés supérieurs sont-ils hautement composés.

La résolution de $(PR)_\lambda$ est facile :

Pour p premier, α entier, soit :

$$\lambda_{p,\alpha} = \frac{1}{\log p} \log \frac{\alpha+1}{\alpha}$$

On conjecture que tous ces nombres sont distincts et on peut montrer que pour trois nombres premiers distincts p, q, r, on ne peut avoir l'existence de α, β, γ tels que $\lambda_{p,\alpha} = \lambda_{q,\beta} = \lambda_{r,\gamma}$ [8]. Soit Λ l'ensemble de ces nombres.

Pour $\lambda > 0$ et q premier soit :

$$a_q = \left[\frac{1}{q^\lambda - 1} \right] \quad \text{et} \quad N_\lambda = \prod_{q \text{ premier}} q^{a_q}.$$

Alors : si $\lambda \notin \Lambda$, N_λ est la seule solution optimale de $(PR)_\lambda$;

si $\lambda = \lambda_{p,\alpha}$ pour une seule valeur de p, N_λ et $N_{\lambda/p}$ sont les deux solutions optimales;

si $\lambda = \lambda_{p,\alpha} = \lambda_{q,\beta}$ $(p \neq q)$, il y a 4 solutions optimales N_λ, $N_{\lambda/p}$, $N_{\lambda/q}$ et $N_{\lambda/pq}$.

En ordonnant Λ en une suite décroissante, il est donc facile de construire la suite croissante des nombres hautement composés supérieurs.

Au paragraphe 2 nous proposons un algorithme pour construire tous les nombres hautement composés $\leqq M$, M donné. Pour cela nous résolvons (P) par une méthode de programmation dynamique pour toutes les valeurs $N_0 \leqq M$.

Au paragraphe 3 nous construisons une tranche de nombres hautement composés c'est-à-dire nous recherchons toutes les solutions optimales du programme (P) pour tout $N_0 \in [N'_0, N''_0]$. Pour cela nous résolvons $(PR)_\lambda$ pour une valeur de λ que nous déterminons en fonction de N'_0. Ensuite nous utilisons les remarques d'Everett ([3], p. 414), reprises sous la notion de bénéfice par Nicolas ([9, 10]) et que nous avons déjà utilisées dans [12]. Cette méthode de bénéfice nous permet de remplir les « gaps » entre deux solutions de $(PR)_\lambda$.

Au paragraphe 4 nous montrons que la connaissance des 5 000 premiers nombres hautement composés invalide une conjecture d'Alaoglu et Erdös [1] et nous déterminons le plus petit nombre ayant au moins 10^{1000} diviseurs.

2. CONSTRUCTION D'UNE TABLE DE NOMBRES HAUTEMENT COMPOSÉS

2.1. Désignons par N_k le plus petit nombre hautement composé ayant k facteurs premiers et proposons nous de construire pour K donné la table de tous les nombres hautement composés $\leqq N_{K+1}$. Le lemme suivant permet de

majorer les exposants de ces nombres dans leur décomposition en facteurs premiers.

LEMME : *Si* $N = \prod_{p \text{ premier}} p^{v_p(N)}$ *hautement composé est* $< N_{k+1}$ *alors :*

$$v_{p_l}(N) \geqq v_{p_{l+1}}(N), \qquad \forall l \leqq k, \tag{1}$$

$$q^{a_q+1} \geqq p_{k+1} \quad \Rightarrow \quad v_q(N) \leqq 2a_q \qquad (a_q \geqq 1), \tag{2}$$

$$q^{a_q} \leqq p_k \quad \Rightarrow \quad v_q(N) \geqq a_q. \tag{3}$$

Si $N = 2^{v_p(2)} \times \ldots \times r_2^3 r_1^3 R_1^2 \times \ldots \times q_3^2 q_2^2 q_1^2 Q_1 Q_2 \times \ldots \times p_k$:

$$v_s(N) > 6 \quad \Rightarrow \quad q_3^3 < p_{k+2}^2/s \tag{4}$$

$$v_s(N) \geqq 8 \quad \Rightarrow \quad r_2^2 < p_{k+1}/s. \tag{5}$$

Démonstration : Elle est facile ; donnons celle de la relation (4) par exemple. Soit $N' = N p_{k+1} p_{k+2}/q_1 q_2 q_3 s$; nous avons $d(N') \geqq (192/189) d(N)$ donc $N' > N$ par suite $p_{k+1} p_{k+2} > q_1 q_2 q_3 s$ et donc $q_3^3 < p_{k+2}^2/s$.

2.2. Le lemme précédent permet de construire une suite $\beta = (\beta_p)$ avec $\beta_p = 0$ si $p \geqq p_{K+2}$ telle que si N est hautement composé $\leqq N_{K+1}$ alors $v_p(N) \leqq \beta_p$.

Soit $\mathcal{N} = \{ N / v_p(N) \leqq \beta_p, \forall p \text{ premier} \}$.

Considérons le programme :

$$(Q)_{N_0} \begin{cases} \text{Max } d(n), \\ \text{avec } n \in \mathcal{N}, \\ n \leqq N_0, \end{cases}$$

qui peut s'écrire aussi :

$$(Q')_{N_0} \begin{cases} \text{Max} \sum_{k=1}^{K+1} \log(x_k + 1) \\ \text{avec} \sum_{k=1}^{K+1} x_k \log p_k \leqq \log N_0 \quad \text{et} \quad x_k \leqq \beta_{p_k}, \quad k = 1, \ldots, K+1. \end{cases}$$

LEMME : $N \leqq N_{K+1}$ *est hautement composé* $\Leftrightarrow \exists M \in \mathcal{N}$, $M \leqq N_{K+1}$, *tel que* N *soit la plus petite solution optimale de* $(Q)_M$.

Démonstration : Si $N \leqq N_{K+1}$ est hautement composé alors N est solution de $(Q)_N$. Inversement soit N la plus petite solution optimale de $(Q)_M$ et $n < N$; soit

N' le plus grand nombre hautement composé $\leqq n$ alors $d(n) \leqq d(N')$. Comme $N' \leqq N_{K+1}$, $N' \in \mathcal{N}$ donc $d(N') < d(N)$ et donc $d(n) < d(N)$.

D'après ce lemme on aura la liste des nombres hautement composés inférieurs à N_{K+1} en résolvant tous les problèmes $(Q)_M$ pour $M \in \mathcal{N}$ et en ne conservant que les plus petites solutions optimales $\leqq N_{K+1}$.

2.3. Programme dynamique

Pour $k=1, \ldots, K+1$ soit $N^{(k)}=(N_w^{(k)})_{w=1,\ldots,l_k}$ la suite croissante des nombres de $\mathcal{N}$ tels que :

(a) le plus grand nombre premier divisant N est $\leqq p_k$;

(b) $d(N_w^{(k)}) < d(N_{w+1}^{(k)})$ pour $w=1, \ldots, l_k-1$;

(c) $\forall w, \forall n$ tel que $N_w^{(k)} < n < N_{w+1}^{(k)}$ $d(n) \leqq d(N_w^{(k)})$.

On pose $D_w^{(k)} = \log N_w^{(k)}$ pour $w=1, \ldots, l_k$.

Algorithme

1. Soit $N^{(1)} = \{ 2, 2^2, \ldots, 2^{\beta_1} \}$.

2. Pour $w=1, \ldots, l_{k-1}$ et pour $\alpha=1, \ldots, \beta_{p_k}$, faire S.P.

3. La nouvelle suite $N^{(k)}$ est construite; faire $k=k+1$ si $k \leqq K+1$ aller en 2; sinon FIN.

Sous programme S.P.

S.P.1. $M = N_w^{(k-1)} \times p_k^{\alpha}$; $D = D_w^{(k-1)} + \log(\alpha+1)$.

Soit r tel que $N_r^{(k-1)} < M < N_{r+1}^{(k-1)}$.

Si $D \leqq D_r^{k-1}$ retour; sinon S.P.2.

S.P.2. Pour $i=r+1, \ldots, l_{k-1}$.

Si $D \geqq D_i^{(k-1)}$, $N_i^{(k-1)}$ est éliminé.

Sinon soit $N^{(k-1)}$ la nouvelle suite avec M et sans les $N_i^{(k-1)}$ éliminés; retour.

2.4. Exemple

Pour $K=105$ nous pouvons choisir le vecteur majoration suivant :

$$\beta = (18, 10, 8, 6, 4, 4, 4, 3, 2, \ldots, \beta_{p_{18}}=2, \beta_{p_{19}}=1, \ldots, \beta_{p_{106}}=1).$$

L'algorithme programmé sur un ordinateur MITRA 15 CII, en double précision, nous donne la liste des 5 284 premiers nombres hautement composés.

3. CONSTRUCTION D'UN INTERVALLE DE NOMBRES HAUTEMENT COMPOSÉS

3.1. Nous nous proposons de construire un intervalle de nombres hautement composés autour d'un nombre N_0 donné.

La première étape est de déterminer les nombres hautement composés supérieurs entourant N_0. D'après des résultats théoriques ([1, 8]), si p_k est le plus grand facteur premier d'un nombre hautement composé supérieur N, on a $p_k \sim \log N$ et d'autre part d'après le paragraphe 1 :

$$\lambda \sim \frac{\log 2}{\log p_k} \quad \text{par suite} \quad \lambda \sim \frac{\log 2}{\log \log N}.$$

On résoud donc d'abord $(\mathrm{PR})_\lambda$ avec $\lambda = \log 2/\log \log N_0$, ceci nous donne un nombre hautement composé supérieur. En construisant la table des nombres hautement composés supérieurs voisins on arrive facilement à encadrer N_0. Soient N^* et N^{**} les nombres hautement composés supérieurs tels que $N^* \leqq N_0 < N^{**}$ et λ^* la valeur commune du multiplicateur de Lagrange.

3.2. Définissons la notion de bénéfice ([8, 9])

Posons $B(n) = \text{Bénéfice}_{\lambda^*}(n/N^*) := \varphi_{\lambda^*}(N^*) - \varphi_{\lambda^*}(n)$ et considérons le problème avec pénalité suivant :

$$(\mathrm{R}) \qquad \begin{cases} \operatorname{Max} d(n), \\ B(n) \leqq B_0, \\ n \leqq M, \end{cases}$$

Si N est hautement composé, N vérifie (R) avec $B_0 = B(N)$ et $M = N$. Inversement la plus petite solution optimale de (R) est un nombre hautement composé.

La méthode des bénéfices consiste à résoudre (R) pour une valeur correcte de B_0.

Le problème est donc de connaître des majorations du bénéfice. Nous avons (*voir* [8]) :

LEMME : *Soit* $N_n \leqq M < N_{n+1}$, *trois nombres hautement composés. Posons :*

$$\delta_n := B(N_{n+1}) + \log(d(N_{n+1}))/d(N_n))$$

alors $B(M) \leqq \delta_n$.

Ce lemme ne donne pas une bonne majoration sur l'intervalle $[N^*, N^{**}]$ mais si l'on commence à construire une sous famille de nombres hautement

composés de l'intervalle on peut ainsi obtenir une meilleure majoration. Dans la pratique on part avec $B_0 = \lambda^*/3$.

3.4. La méthode de construction

Pour p premier et $j \in \mathbb{Z}$ on pose :

$$\Delta(p, j) := \log((v_p(N^*) + 1/(v_p(N^*) + j + 1)) + \varepsilon j \log p,$$

alors :

$$B(N) = \sum_{p/N \times N^*} \Delta(p, v_p(N) - v_p(N^*)).$$

Soit :

$$Q = \{ p/\exists j \in \mathbb{Z}^*, \Delta(p, j) \leqq B_0 \},$$

$$Q = \{ q_1, q_2, \ldots, q_{k_0} \}.$$

$\forall k = 1, 2, \ldots, k_0$, soit J_k l'ensemble ordonné des entiers $a_{k,l}$ ($l = 1, \ldots, a_k$) tels que $\Delta(q_k, a_{k,l}) \leqq B_0$.

Le programme (R) est alors résolu par une méthode de programmation dynamique. Le lemme précédent permet de s'assurer si tous les nombres hautement composés sont obtenus.

On notera $N^k = (N_j^k)_{j=1, 2, \ldots, l_k}$ la suite obtenue lors de l'étape k de l'algorithme ; $D_j^{(k)} = \log d\ (N_j^{(k)})$; $B_j^{(k)} = B(N_j^{(k)})$.

3.5. Algorithme

1. Déterminer N^* et N^{**} deux nombres hautement composés supérieurs tels que $N^* \leqq N_0 < N^{**}$ et soit λ^* le multiplicateur de Lagrange.

2. On définit $N^{(1)} = \{ N^* q_1^{a_{1,1}}, \ldots, N^* q^{a_{1,\alpha_1}} \}$.

3. Pour $l = 1, \ldots, k-1$ et pour $\alpha = a_{k,1}, \ldots, a_{k,\alpha_k}$, faire S.P.

4. La nouvelle suite $N^{(k)}$ est construite; faire $k = k+1$; si $k \leqq k_0$ aller en 3, sinon aller en 5.

5. Nous connaissons $N^{(k_0)}$; calculer $A = \underset{i=1, \ldots, l_{k_0-1}}{\text{Max}} \delta(i)$. Si $A > B_0$ faire $B_0 = A$ et aller en 2. Sinon fin.

Sous programme S.P.

S.P.1. $M = N_l^{(k-1)} \times q_k^\alpha$; $D = D_l^{(k-1)} + \log(\alpha + 1)$; $B = B_l^{(k-1)} + \Delta(q_k, \alpha)$.

Si $B > B_0$ retour ; sinon S.P.2.

S.P.2. Soit r tel que $N_r^{(k-1)} < M < N_{r+1}^{(k-1)}$.

Si $D < D_r^{(k-1)}$ retour ; sinon S.P.3.

S.P.3. Pour $i = r+1, \ldots, l_{k-1}$.

Si $D \geqq D_i^{(k-1)}$, $N_i^{(k-1)}$ est éliminé.

Sinon soit $N^{(k-1)}$ la nouvelle suite avec M et sans les $N_i^{(k-1)}$ éliminés ; retour.

3.6. Exemple

Si $N_0 = e^{1000}$, N^* a 164 facteurs premiers. Pour $B_0 = \lambda^*/3$ (B_0 suffisant), on obtient tous les nombres hautement composés en faisant varier seulement 18 exposants ($k_0 = 18$) et parmi ceux-ci 14 varie d'une unité. Le temps de calcul sur un IRIS 80 CII est d'environ 15 secondes ; la capacité de l'ordinateur nous permet d'atteindre $N_0 = e^{35000}$

4. APPLICATIONS

4.1. Alaoglu et Erdös avaient conjecturé ([1], p. 467) que si N est hautement composé alors :

1° il existe un nombre premier p tel que Np soit hautement composé,

2° Il existe un nombre premier q tel que N/q soit hautement composé.

Or dans la liste des 5 000 premiers nombres hautement composés,

1° il existe un nombre invalidant la première conjecture à savoir :

$$N = 2^{12}3^7\bar{5}^4(7\times 11\times 13)^3(17\times 19\times 23)^2 29\times \ldots \times 337.$$

2° il existe trois nombres invalidant la seconde :

$$N = 2^{10}3^6 5^4(7\times \ldots \times 19)^2 23\times \ldots \times 137,$$

$$N = 2^{10}3^7 5^5(7\times \ldots \times 13)^2(17\times \ldots \times 37)^2 41\times \ldots \times 479,$$

$$N = 2^{10}3^7 5^5(7\times \ldots \times 13)^2(17\times \ldots \times 41)^2 43\times \ldots \times 541.$$

Ces contre-exemples sont-ils en nombre finis ?

4.2. Recherche du plus petit nombre possédant au moins 10^{1000} diviseurs

D'après la remarque du paragraphe 1, le nombre cherché est un nombre hautement composé et la méthode du paragraphe 3 permet de le déterminer.

Ce nombre N possède 13 198 chiffres et sa décomposition en facteurs premiers est :

$$N = 2^{20}\times 3^{12}\times 5^8\times 7^7\times(11\times 13\times 17)^5\times(19\times 23)^4\times(29\times \ldots \times 71)^3$$
$$\times(73\times \ldots \times 421)^2\times 431\times \ldots \times 30\,113.$$

Ce nombre possède $21 \times 13 \times 9 \times 8 \times 6^3 \times 5^2 \times 4^{11} \times 3^{62} \times 2^{3175}$ diviseurs soit :

$$d(N) = 2^{3203} \times 3^{68} \times 5^2 \times 7 \times 13$$

$$\log_{10} d(N) = 1\,000{,}000\,30...$$

Par suite N possède en plus des 10^{1000} diviseurs demandés au moins $6{,}9 \times 10^{996}$ autres diviseurs.

BIBLIOGRAPHIE

1. L. ALAOGLU et P. ERDÖS, *On Highly Composite and Similar Numbers*, Trans. Amer. Math. Soc., vol. 56, 1944, p. 448-469.
2. P. ERDÖS, *On Highly Composite Numbers*, J. London Math. Soc., vol. 19, 1944, p. 130-133.
3. H. EVERETT, *Generalized Lagrange Multiplier Method for Solving Problems of Optium Allocation of Ressources*, Operation Research, vol. II, 1963, p. 399-417.
4. A. M. GEOFFRION, *Lagrangean Relaxation for Integer Programming*, Mathem. Prog. Stud., vol. 2, 1974, p. 82-114.
5. F. J. GOULD, *Extension of Lagrange Multipliers in Non Linear Programming*, S.I.A.M. J. Appl. Math., vol. 17, n° 6, novembre 1969.
6. G. HADLEY, *Non Linear and Dynamic Programming*, Reading Palo Alto London; Addison-Wesley pub Conf., 1964.
7. J. L. LAURIÈRE, *Elements de programmation dynamique*, Gauthier-Villars, Paris, 1979.
8. J. L. NICOLAS, *Répartition des nombres hautement composés de Ramanujan*, Can. J. Math., vol. III, n° 1, 1971, p. 116-130.
9. J. L. NICOLAS, *Problèmes d'optimisation en nombres entiers*, Astérisque, vol. 24-25, 1975, p. 325-333.
10. J. L. NICOLAS, *Algorithmes d'optimisation en nombres entiers*, Astérisque, vol. 38-39, 1976, p. 169-182.
11. S. RAMANUJAN, *Highly Composite Numbers*, Proc. London Math. Soc., t. 14, série 2, 1915, p. 347-400; collected papers, p. 78-128, Cambridge, 1927.
12. G. ROBIN, *Sur un problème d'optimisation en nombres entiers*, Math. Oper. Stat. Ser. Opt., vol. 11, n° 3, 1980, p. 403-420.