

IRINEL DRAGAN

Un algorithme lexicographique pour la résolution des programmes polynomiaux en variables entières

Revue française d'informatique et de recherche opérationnelle,
tome 2, n° V3 (1968), p. 81-89

http://www.numdam.org/item?id=RO_1968__2_3_81_0

© AFCET, 1968, tous droits réservés.

L'accès aux archives de la revue « Revue française d'informatique et de recherche opérationnelle » 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/>

UN ALGORITHME LEXICOGRAPHIQUE POUR LA RESOLUTION DES PROGRAMMES POLYNOMIAUX EN VARIABLES ENTIERES. ⁽¹⁾

par Irinel DRAGAN ⁽²⁾

Résumé. — On développe un algorithme pour la résolution des programmes polynomiaux en nombres entiers ; l'algorithme représente l'extension d'un algorithme pour la résolution des programmes linéaires en variables binaires. Le point de départ des recherches a été un article de E. L. Lawler et M. D. Bell (Op. Res. T. 14, 6, 1966, p. 1098-1112).

Soit l'ensemble

$$B_n^p = \{ x \mid x = (x_i), \quad (i = 1, \dots, n), \quad x_i \in \{ 0, 1, \dots, p-1 \} \};$$

soit le système de polynômes à coefficients entiers $P_h(x)$, ($h = 0, 1, \dots, s$) ;
soit l'ensemble

$$X_0 = \{ x \mid x \in B_n^p, \quad P_j(x) \geq 0, \quad (j = 1, \dots, s) \}.$$

Nous développerons un algorithme pour la résolution du problème (A) :
trouver $\hat{x} \in B_n^p$, tel que

$$P_0(\hat{x}) = \sup_{x \in X_0} P_0(x).$$

1. QUELQUES DEFINITIONS ET PROPRIETES PRELIMINAIRES

Supposons que l'ensemble B_n^p est ordonné lexicographiquement ;
c'est-à-dire pour chaque couple de vecteurs $x \in B_n^p$, $y \in B_n^p$, on posera
 $x \leq y$ si et seulement si $n(x) \leq n(y)$, où

$$n(x) = p^{n-1}x_1 + p^{n-2}x_2 + \dots + px_{n-1} + x_n.$$

(1) Présenté au Congrès européen des associations ES, TIMS, IMS, IASPS, Amsterdam, 2-7 sept. 1968.

(2) Université « A. I. Cuza », Jassy, Roumanie.

La fonction $f(x)$, définie sur l'ensemble B_n^p , sera appelée lexicographique monotone non-décroissante (en abrégé l.m.n.), si $x \leq y$ entraîne

$$f(x) \leq f(y).$$

Théorème 1 : Les fonctions linéaires homogènes

$$u_1(x) = x_1, \quad u_2(x) = px_1 + x_2, \quad u_3(x) = p^2x_1 + px_2 + x_3, \dots, \\ u_n(x) = p^{n-1}x_1 + p^{n-2}x_2 + \dots + px_{n-1} + x_n,$$

définies sur l'ensemble B_n^p , sont des fonctions l.m.n. à valeurs non-négatives.

Soit $y \in B_n^p$, $z \in B_n^p$, deux éléments successifs, $y \leq z$, $y \neq z$; alors, il existe un indice k , ($1 \leq k \leq n$), tel que

$$y_1 = z_1, \dots, y_{k-1} = z_{k-1}, y_k = z_k - 1, \\ y_{k+1} = \dots = y_n = p - 1, z_{k+1} = \dots = z_n = 0.$$

Il en résulte

$$u_i(z) - u_i(y) = \begin{cases} 0 & \text{pour } i < k, \\ 1 & \text{pour } i \geq k, \end{cases}$$

pour chaque $i = 1, \dots, n$, ce qui suffit pour prouver la monotonie des fonctions $u_i(x)$, ($i = 1, \dots, n$). Évidemment, on a $u_i(x) \geq 0$, ($i = 1, \dots, n$), pour chaque $x \in B_n^p$.

Théorème 2 : Un polynôme $P[u_1(x), \dots, u_n(x)]$, à coefficients entiers non-négatifs, est un polynôme l.m.n. à coefficients entiers.

En effet, la somme de plusieurs fonctions l.m.n. et le produit de plusieurs fonctions l.m.n. à valeurs non-négatives sont des fonctions l.m.n.

Il est bien clair que chaque polynôme à coefficients entiers $P(x)$ s'exprime d'une manière univoque comme un polynôme à coefficients entiers des fonctions $u_i(x)$; c'est le résultat des transformations linéaires non singulières

$$(T) \quad x_1 = u_1, \quad x_2 = u_2 - pu_1, \quad x_3 = u_3 - pu_2, \dots, \\ x_n = u_n - pu_{n-1},$$

et

$$(T^{-1}) \quad u_i = u_i(x), \quad (i = 1, \dots, n).$$

Soit le polynôme $P(x)$ de degré q ; si le polynôme correspondant en u contient tous les termes de degré inférieur ou égal à q et si leurs coefficients sont positifs, le polynôme $P(x)$ sera appelé un polynôme complet.

Évidemment, chaque polynôme complet est un polynôme l.m.n., mais la réciproque de cette affirmation n'est pas vraie.

Théorème 3 : Soit les polynômes à coefficients entiers $P_h(x)$, ($h = 0, 1, \dots, s$), et le polynôme à coefficients entiers $P(x)$ qui est un polynôme complet dont le degré est égal au degré maximal rencontré dans le système de polynômes considéré ; alors, il existe un nombre entier C non-négatif et des polynômes l.m.n. à coefficients entiers $Q_h(x)$, ($h = 0, 1, \dots, s$), tels que

$$P_h(x) = CP(x) - Q_h(x), \quad (h = 0, 1, \dots, s).$$

Soit $\bar{P}(u)$ et $\bar{P}_h(u)$, ($h = 0, 1, \dots, s$), les résultats de l'application de la transformation linéaire (T) sur les polynômes $P(x)$ et $P_h(x)$, ($h = 0, 1, \dots, s$) ; le polynôme $\bar{P}(u)$ contient tous les termes en u dont le degré est au plus égal au degré maximal rencontré dans le système de polynômes $\bar{P}_h(u)$, et ces termes possèdent des coefficients positifs. Alors, il existe des nombres entiers non-négatifs C tels que tous les coefficients des variables u_i dans les polynômes

$$\bar{Q}_h(u) = C\bar{P}(u) - \bar{P}_h(u), \quad (h = 0, 1, \dots, s),$$

soient des nombres entiers non-négatifs. Soit C le plus petit nombre entier qui possède cette propriété ; alors, les polynômes correspondants $Q_h(x) = \bar{Q}_h[u(x)]$, ($h = 0, 1, \dots, s$), seront des polynômes l.m.n. à coefficients entiers, selon le théorème 2. Mais,

$$Q_h(x) = CP(x) - P_h(x), \quad (h = 0, 1, \dots, s),$$

donc le théorème 3 est démontré.

D'une manière abrégée, les décompositions des fonctions $P_h(x)$, ($h = 0, 1, \dots, s$), seront obtenues à la suite des opérations suivantes : a) la transformation linéaire (T) réalisée sur les polynômes $P(x)$ et $P_h(x)$, ($h = 0, 1, \dots, s$) ; b) la recherche du nombre entier C ; c) le calcul des différences $Q_h(x) = CP(x) - P_h(x)$, ($h = 0, 1, \dots, s$). Il faudra remarquer que ces décompositions dépendent d'un polynôme quelconque appartenant à la classe des polynômes complets. Un choix convenable pourrait améliorer l'efficacité de l'algorithme lexicographique.

EXEMPLE : Soit les polynômes

$$P(x) = 31x_1 + 6x_2 + x_3,$$

$$P_0(x) = -x_1 + x_2 + 2x_3, \quad P_1(x) = 18x_1 + 5x_2 + x_3 - 16,$$

définis sur l'ensemble B_3^5 , parmi lesquels $P(x)$ est un polynôme complet. Après la transformation linéaire (T) on obtiendra

$$\bar{P}(u) = u_1 + u_2 + u_3,$$

$$\bar{P}_0(u) = -6u_1 - 9u_2 + 2u_3, \quad \bar{P}_1(u) = -7u_1 + u_3 - 16.$$

Le nombre entier C doit satisfaire l'inégalité

$$C \geq \max \{-6, -9, 2, -7, 0, 1\},$$

donc $C = 2$ et on a

$$Q_0(x) = 2P(x) - P_0(x) = 63x_1 + 11x_2,$$

$$Q_1(x) = 2P(x) - P_1(x) = 44x_1 + 7x_2 + x_3 + 16.$$

2. LA RESOLUTION D'UN PROBLEME AUXILIAIRE

D'abord nous montrerons comment agit l'algorithme lexicographique pour la résolution du problème auxiliaire (B) : trouver le premier élément \tilde{x} de l'ensemble

$$Z = \{x \mid x \in B_n^p, \quad P_h(x) \geq 0, \quad (h = 0, 1, \dots, s)\},$$

où $P_h(x)$ sont des polynômes à coefficients entiers, définis sur l'ensemble B_n^p .

On commence par une étape préparatoire. On choisit un polynôme à coefficients entiers, $P(x)$, qui satisfait les conditions suivantes : 1) $P(x)$ est un polynôme complet ; 2) la liste (L) des valeurs du polynôme $P(x)$ pour tous les vecteurs de l'ensemble B_n^p se trouve à notre disposition ; 3) le degré du polynôme $P(x)$ est égal au degré maximal trouvé dans le système de polynômes $P_h(x)$, $(h = 0, 1, \dots, s)$. On applique le procédé développé dans le paragraphe précédent et on obtiendra le nombre entier non-négatif C et les polynômes l.m.n. à coefficients entiers $Q_h(x)$, $(h = 0, 1, \dots, s)$, tels que

$$P_h(x) = CP(x) - Q_h(x), \quad (h = 0, 1, \dots, s).$$

Maintenant, il faudra résoudre le problème (B) : trouver le premier élément \tilde{x} de l'ensemble

$$Z = \{x \mid x \in B_n^p, \quad Q_h(x) \leq CP(x), \quad (h = 0, 1, \dots, s)\}$$

dans lequel $P(x)$ et $Q_h(x)$ sont des polynômes l.m.n. à coefficients entiers et C un nombre entier non-négatif. L'étape préparatoire est finie et il faudra commencer les itérations de l'algorithme.

Évidemment, le système d'inéquations qui détermine l'ensemble Z peut être remplacé par l'inéquation unique

$$\max_h Q_h(x) \leq CP(x);$$

c'est l'inéquation pour laquelle nous chercherons le premier élément de l'ensemble des solutions, en connaissant la liste (L) des valeurs de la fonction $P(x)$ pour tous les éléments de l'ensemble B_n^p . Il est bien clair que si les fonctions $Q_h(x)$ sont des fonctions l.m.n., la fonction

$$M(x) = \max_h Q_h(x)$$

sera aussi une fonction l.m.n.

Le vecteur $z \in B_n^p$ sera appelé « vecteur de départ » pour la résolution du problème (B), si pour chaque $x \leq z$, $x \neq z$, on a $x \notin Z$; le vecteur nul est considéré comme un vecteur de départ.

Théorème 4 : Si le vecteur $z \in B_n^p$ est un vecteur de départ pour la résolution du problème (B) et le vecteur $\tilde{z} \in B_n^p$ est le premier élément qui satisfait l'inéquation

$$(E) \quad M(z) \leq CP(x),$$

nous aurons les conclusions suivantes : a) si \tilde{z} n'existe pas, le problème (B) sera insoluble ; b) si $\tilde{z} = z$, la solution du problème (B) sera $\tilde{x} = \tilde{z}$; c) si $\tilde{z} \neq z$, le vecteur \tilde{z} sera un nouveau vecteur de départ et $\tilde{z} \succ z$.

Pour justifier le théorème 4 il faudra remarquer d'abord que si \tilde{x} existe nous aurons $\tilde{x} \succ z$. Alors, les conclusions du théorème résultent de la manière suivante : a) si \tilde{x} existait nous aurions eu $M(z) \leq M(\tilde{x}) \leq CP(\tilde{x})$, donc \tilde{z} existerait aussi ; b) conclusion évidente ; c) si \tilde{x} satisfaisait

$$z \leq \tilde{x} \leq \tilde{z}, \tilde{x} \neq \tilde{z},$$

nous aurions eu $M(z) \leq M(\tilde{x}) \leq CP(\tilde{x})$, donc \tilde{z} n'aurait pas été le premier vecteur qui satisfait l'inéquation (E).

En tenant compte du théorème 4 nous pourrions formuler les opérations d'une itération quelconque de l'algorithme. Supposons qu'on connaît un vecteur de départ z_r et on commence l'itération r ; pour $r = 1$ on prend $z_1 = 0$. L'itération r contient les opérations suivantes : 1. Calculer les nombres $Q_h(z_r)$, ($h = 0, 1, \dots, s$), et le nombre $M(z_r)$; 2. extraire de la liste (L) le premier vecteur z_{r+1} qui satisfait l'inéquation $M(z_r) \leq CP(x)$; 3. appliquer les conclusions du théorème 4 pour établir si le problème (B) a été résolu ou non et passer à l'itération $r + 1$ dans le dernier cas, en remplaçant z_r par z_{r+1} .

L'algorithme conduit à la résolution du problème (B) après un nombre fini d'itérations, car tous les vecteurs de départ obtenus à la fin des itérations de l'algorithme sont distincts et l'ensemble B_n^p est fini.

On peut dire que l'algorithme lexicographique cherche la solution du problème (B) parmi les vecteurs de départ construits par lui-même.

EXEMPLE : Soit le polynôme $P_1(x) = 18x_1 + 5x_2 + x_3 - 16$; trouver le premier élément $\tilde{x} \in B_3^5$, qui satisfait l'inéquation $P_1(x) \geq 0$. En prenant le polynôme complet $P(x) = 31x_1 + 6x_2 + x_3$, dont la table des valeurs est donnée dans le tableau II, on réalise l'étape préparatoire de la résolution ; on a $C = 1$, $Q_1(x) = 13x_1 + x_2 + 16$. Maintenant, en tenant compte de la liste (L) on construit le tableau des vecteurs de départ. Après 4 itérations (tableau I) on obtiendra la solution du problème : $\tilde{x} = (0, 3, 1)$; cette solution est le 17^e vecteur de l'ensemble B_3^5 . L'algo-

rithme n'a pas utilisé pendant la résolution que 4 vecteurs de départ, les autres vecteurs ont été éliminés par l'algorithme.

TABLEAU I

r	z_r	$Q_1(z_r)$	$M(z_r)/C$
1	0 0 0	16	16
2	0 2 4	18	18
3	0 3 0	19	19
4	0 3 1	19	19

3. L'ALGORITHME LEXICOGRAPHIQUE

Soit le programme polynomial en nombres entiers : trouver le vecteur $\hat{x} \in B_n^p$, tel que

$$P_0(\hat{x}) = \sup_{x \in X_0} P_0(x),$$

TABLEAU II. — Liste (L) pour le polynôme $P(x) = 31x_1 + 6x_2 + x_3$

x	$P(x)$	x	$P(x)$	x	$P(x)$	x	$P(x)$	x	$P(x)$
0 0 0	0	1 0 0	31	2 0 0	62	3 0 0	93	4 0 0	124
0 0 1	1	1 0 1	32	2 0 1	63	3 0 1	94	4 0 1	125
0 0 2	2	1 0 2	33	2 0 2	64	3 0 2	95	4 0 2	126
0 0 3	3	1 0 3	34	2 0 3	65	3 0 3	96	4 0 3	127
0 0 4	4	1 0 4	35	2 0 4	66	3 0 4	97	4 0 4	128
0 1 0	6	1 1 0	37	2 1 0	68	3 1 0	99	4 1 0	130
0 1 1	7	1 1 1	38	2 1 1	69	3 1 1	100	4 1 1	131
0 1 2	8	1 1 2	39	2 1 2	70	3 1 2	101	4 1 2	132
0 1 3	9	1 1 3	40	2 1 3	71	3 1 3	102	4 1 3	133
0 1 4	10	1 1 4	41	2 1 4	72	3 1 4	103	4 1 4	134
0 2 0	12	1 2 0	43	2 2 0	74	3 2 0	105	4 2 0	136
0 2 1	13	1 2 1	44	2 2 1	75	3 2 1	106	4 2 1	137
0 2 2	14	1 2 2	45	2 2 2	76	3 2 2	107	4 2 2	138
0 2 3	15	1 2 3	46	2 2 3	77	3 2 3	108	4 2 3	139
0 2 4	16	1 2 4	47	2 2 4	78	3 2 4	109	4 2 4	140
0 3 0	18	1 3 0	49	2 3 0	80	3 3 0	111	4 3 0	142
0 3 1	19	1 3 1	50	2 3 1	81	3 3 1	112	4 3 1	143
0 3 2	20	1 3 2	51	2 3 2	82	3 3 2	113	4 3 2	144
0 3 3	21	1 3 3	52	2 3 3	83	3 3 3	114	4 3 3	145
0 3 4	22	1 3 4	53	2 3 4	84	3 3 4	115	4 3 4	146
0 4 0	24	1 4 0	55	2 4 0	86	3 4 0	117	4 4 0	148
0 4 1	25	1 4 1	56	2 4 1	87	3 4 1	118	4 4 1	149
0 4 2	26	1 4 2	57	2 4 2	88	3 4 2	119	4 4 2	150
0 4 3	27	1 4 3	58	2 4 3	89	3 4 3	120	4 4 3	151
0 4 4	28	1 4 4	59	2 4 4	90	3 4 4	121	4 4 4	152

où

$$X_0 = \{ x \mid x \in B_n^p, \quad P_j(x) \geq 0, \quad (j = 1, \dots, s) \},$$

et $P_h(x)$, ($h = 0, 1, \dots, s$), sont des polynômes à coefficients entiers définis sur l'ensemble B_n^p .

La résolution du problème (A) se ramène à la résolution d'un nombre fini de problèmes auxiliaires de la forme (B_k) : trouver le vecteur \hat{x}_k , premier élément de l'ensemble

$$X_k = \{ x \mid x \in B_n^p, \quad P_0(x) \geq \lambda_k, \quad P_j(x) \geq 0, \quad (j = 1, \dots, s) \}$$

où

$$\lambda_k = \begin{cases} -\infty & k = 0, \\ P_0(\hat{x}_{k-1}) + 1 & k > 0. \end{cases}$$

En effet, la résolution successive des problèmes auxiliaires (B_0) , (B_1) , (B_2) , ..., revient à trouver le premier programme \hat{x}_0 du problème (A), puis le premier programme \hat{x}_1 « meilleur » que \hat{x}_0 , le premier programme \hat{x}_2 « meilleur » que \hat{x}_1 , et ainsi de suite.

Chaque vecteur \hat{x}_k est un vecteur de départ pour la résolution du problème (B_{k+1}) ; le vecteur nul est le vecteur de départ avec lequel on commence la résolution du problème (B_0) . Chaque problème (B_k) est un problème du type considéré dans le paragraphe précédent. Alors, on peut donner la description d'une itération r de l'algorithme lexicographique pour la résolution du problème (A). On connaît un vecteur de départ z_r pour la résolution d'un problème auxiliaire (B_k) . Il faudra appliquer les opérations suivantes :

a) Trouver le vecteur z_{r+1} , premier vecteur qui satisfait l'inéquation

$$\max_h Q_h(z) \leq CP(x),$$

où la fonction $Q_0(x)$ est construite à l'aide du nombre λ_k ; si le vecteur z_{r+1} n'existe pas, le programme (A) est résolu, car ou bien nous aurons $\hat{x} = \hat{x}_{k-1}$ (pour $k > 0$), ou bien \hat{x} n'existe pas (pour $k = 0$) ; si le vecteur z_{r+1} existe, passer à l'opération suivante ;

b) Établir si $z_{r+1} \neq z_r$ ou $z_{r+1} = z_r$; dans le premier cas, commencer l'itération $r + 1$ avec le nouveau vecteur de départ z_{r+1} ; dans le second cas, passer à l'opération suivante ;

c) Calculer le nombre

$$\lambda_{r+1} = P_0(z_r) + 1,$$

ajouter $\lambda_{r+1} - \lambda_r$ à la fonction $Q_0(x)$ et passer à l'itération $r + 1$ avec le vecteur de départ z_{r+1} .

Évidemment, le vecteur de départ ne peut rester le même que pour deux itérations successives, qui ont réalisé le passage d'un problème auxiliaire à un autre ; d'où le théorème qui suit :

Théorème 5 : L'algorithme lexicographique résout le programme (A) après un nombre fini d'itérations.

Dans le cas où une itération r change le problème auxiliaire on peut commencer l'itération suivante avec le vecteur de départ $z_r + e_n$, car z_r n'est pas la solution du nouveau problème auxiliaire ; l'algorithme avec cette petite variation sera illustré par notre exemple.

4. EXEMPLE

Soit le programme linéaire en nombres entiers : trouver $\hat{x} \in B_3^5$, tel que

$$P_0(\hat{x}) = \sup_{x \in X_0} P_0(x),$$

où

$$P_0(x) = -x_1 + x_2 + 2x_3,$$

$$P_1(x) = 18x_1 + 5x_2 + x_3 - 16,$$

$$X_0 = \{ x \mid x \in B_3^5, \quad P_1(x) \geq 0 \}.$$

L'étape préparatoire de la résolution a été réalisée dans le premier paragraphe ; nous avons obtenu $C = 2$ et

$$Q_0(x) = 63x_1 + 11x_2,$$

$$Q_1(x) = 44x_1 + 7x_2 + x_3 + 16,$$

en prenant le polynôme complet $P(x) = 31x_1 + 6x_2 + x_3$.

L'application de l'algorithme donnera les résultats contenus dans le tableau III; le tableau a été construit à l'aide de la liste (L) donnée dans le second paragraphe. Nous avons résolu successivement les problèmes auxiliaires $(B_0), \dots, (B_5)$; pendant la résolution l'algorithme a utilisé seulement 31 vecteurs de l'ensemble B_3^5 qui contient 125 vecteurs ; les solutions des problèmes auxiliaires ont été :

$$\hat{x}_0 = (0, 3, 1), \hat{x}_1 = (0, 3, 2), \hat{x}_2 = (0, 3, 3), \hat{x}_3 = (0, 3, 4), \hat{x}_4 = (0, 4, 4)$$

et le problème (B_5) a été insoluble. Alors, le programme considéré a la solution $\hat{x} = (0, 4, 4)$

TABLEAU III

k	r	z_r	$Q_0(z_r) + \lambda_k$	$Q_1(z_r)$	$M(z_r)/C$	λ_k
0	1	0 0 0	—	16	8	
	2	0 1 2	—	25	12,5	
	3	0 2 1	—	31	15,5	
	4	0 2 4	—	34	17	
	5	0 3 0	—	37	18,5	
	6	0 3 1	—	38	19	
1	7	0 3 2	39	39	19,5	6
2	8	0 3 3	41	40	20,5	8
3	9	0 3 4	43	41	21,5	10
4	10	0 4 0	56	44	28	12
	11	0 4 4	56	48	28	
5	12	1 0 0	76	60	38	13
	13	1 1 1	87	68	43,5	
	14	1 2 1	98	75	49	
	15	1 3 0	109	81	54,5	
	16	1 4 0	120	88	60	
	17	2 0 0	139	104	69,5	
	18	2 1 2	150	113	75	
	19	2 2 1	161	119	80,5	
	20	2 3 1	172	126	86	
	21	2 4 0	183	132	91,5	
	22	3 0 0	202	148	101	
	23	3 1 2	213	157	106,5	
	24	3 2 2	224	164	112	
	25	3 3 1	235	170	117,5	
	26	3 4 1	246	177	123	
	27	4 0 0	265	192	132,5	
	28	4 1 3	276	202	138	
	29	4 2 2	287	208	143,5	
	30	4 3 2	298	215	149	
	31	4 4 1	309	221	154,5	