REVUE FRANÇAISE D'AUTOMATIQUE, INFORMATIQUE, RECHERCHE OPÉRATIONNELLE. RECHERCHE OPÉRATIONNELLE

M. GONDRAN

Un outil pour la programmation en nombres entiers : « La méthode des congruences décroissantes »

Revue française d'automatique, informatique, recherche opérationnelle. Recherche opérationnelle, tome 7, n° V3 (1973), p. 35-54

<http://www.numdam.org/item?id=RO_1973__7_3_35_0>

R.A.I.R.O.
(7e année, septembre 1973, V-3, p. 35 à 54)

UN OUTIL POUR LA PROGRAMMATION EN NOMBRES ENTIERS : « LA METHODE DES CONGRUENCES DECROISSANTES »

par M. GONDRAN [1]

Résumé. — *L'auteur montre que « la méthode des congruences décroissantes » permet d'obtenir une bonne évaluation de la solution d'un programme à variables entières ou mixtes.*

1. INTRODUCTION

Les deux principales approches de la programmation en nombres entiers (méthode des troncatures et procédure d'exploration arborescente), ne donnent pas dans le cas général de bons résultats.

Les *méthodes de troncature* sont généralement très longues à converger : cela tient principalement au fait que les troncatures ne sont pas assez fortes. Or, une troncature est déterminée par son orientation et sa pénétration dans le domaine.

Une troncature forte sera donc une troncature, qui pour une orientation donnée, donne une bonne pénétration du domaine. Elle sera donc obtenue si on peut déterminer une bonne évaluation de la solution optimale d'un programme en nombres entiers.

Dans les *méthodes d'exploration* :

— si on utilise une exploration séquentielle (PSES; Branch and Bound, méthode LIFO), on ne peut généralement montrer l'optimalité d'une solution sans augmenter considérablement l'arborescence,

— si on utilise une exploration dirigée (PSEP), on doit souvent construire une arborescence importante : la raison essentielle est que l'évaluation par

(1) Département Traitement de l'information et études mathématiques de l'EDF à Clamart.

défaut (cas de minimisation) de la solution optimale aux sommets de l'arborescence n'est pas assez bonne.

Ainsi une difficulté majeure de ces deux approches réside dans le problème suivant : « Obtenir en *peu de calculs* une *bonne évaluation* par défaut (cas de la minimisation) de la solution optimale d'un programme en nombres entiers ».

Un premier type d'évaluation par défaut est la solution continue de ce programme. En tenant compte que certaines variables sont entières, on obtient une amélioration qui sera appelée « pénalités » (cf. en particulier Driebeck [2], Beale et Small [1], Tomlin [13], Roy, Benayoun et Tergny [12]).

Malheureusement l'évaluation donnée par ces « pénalités » n'est pas bien bonne.

Un second type d'évaluation par défaut qui donne une évaluation bien meilleure est la solution du problème asymptotique introduit par Gomory dans [5].

Malheureusement le nombre des calculs nécessaires est alors prohibitif. Nous proposons ici une méthode qui alliera *presque* les avantages des deux méthodes précédentes : peu de calculs et bonne évaluation.

En effet « la méthode des congruences décroissantes » permet de trouver, en peu de calculs, une borne inférieure (cas de la minimisation) de la fonction objective d'un problème asymptotique et, dans certains cas (imprévisibles) une solution de ce problème asymptotique.

De plus cette borne inférieure sera toujours meilleure que l'évaluation donnée par la méthode des pénalités.

Nous donnons aux paragraphes 2 et 3 une amélioration notable de cette méthode que nous avons introduit dans [7] et utilisée dans un article récent [9] (cf. aussi Frehel [4]).

Au paragraphe 4, nous en donnerons une généralisation tandis qu'au paragraphe 5 nous montrerons certains domaines d'applications.

2. METHODE DES CONGRUENCES DECROISSANTES

La méthode suivante se propose d'obtenir une bonne évaluation par défaut de la solution optimale du programme P

$$(P) \qquad \begin{cases} \min z = \sum_{i=1}^{n} c_i x_i \\ \text{avec } x \in X \end{cases}$$

ou

$$X = \{ x \in Z^n \mid \sum_{i=1}^{n} g_i x_i \equiv g_0 \ (\mathrm{mod}\ D),\ 0 \leqslant x_i \leqslant \alpha_i \} \tag{2-1}$$

$$\text{avec } (c_1, \ldots, c_n) \in \mathbf{R}^{+n},\ (g_1, \ldots, g_n) \in \mathbf{Z}_D^n,\ g_0 \in Z_D,$$

$$\alpha = (\alpha_1, \ldots, \alpha_n) \in \mathbf{N}^n.$$

On rappelle que l'on note Z/Z_D ou plus simplement Z_D le groupe des entiers modulo D.

On fera deux remarques sur ce problème.

2.1. Remarque

En (2.1) nous ne prenons qu'une égalité linéaire sur le groupe $\mathbf{Z}_D$, c'est-à-dire que nous considérons le cas où le groupe est cyclique. Nous généraliserons ce cas au paragraphe 4. Mais on peut déjà remarquer que, comme nous ne cherchons qu'une évaluation par défaut du problème asymptotique, nous pouvons nous restreindre à une seule égalité linéaire, donc à un groupe cyclique.

2.2. Remarque

La condition $x_i \leqslant \alpha_i$ n'est jamais restrictive puisque l'on a toujours :

$$x_i \leqslant r_i = \frac{D}{\mathrm{pgcd}\,(D, g_i)} = \text{ordre du sous-groupe engendré par } g_i \text{ dans } \mathbf{Z}_D \tag{2-2}$$

([5]), [7]).

2.3. Automorphismes [8]

Considérons les automorphismes de $\mathbf{Z}_D$. Ils sont de la forme :

$$\Phi_\lambda(g) \equiv \lambda g (\mathbf{Z}_D) \tag{2-3}$$

où λ est un nombre premier avec D. Le nombre d'automorphismes est donc égal à $\varphi(D)$, l'indicateur d'Euler de D.

On rappelle que si $D = p_1^{\beta_1} \ldots p_n^{\beta_n}$, alors

$$\varphi(D) = D\left(1 - \frac{1}{p_1}\right) \ldots \left(1 - \frac{1}{p_n}\right). \tag{2-4}$$

Pour tout automorphisme Φ_λ de $\mathbf{Z}_D$, l'équation de groupe (2-1) est équivalente à :

$$\sum_{i=1}^{n} \Phi_\lambda(g_i) x_i \equiv \Phi_\lambda(g_0) \qquad (\mathbf{Z}_D) \tag{2-5}$$

on a donc $\varphi(D)$ équations de ce type.

On notera par $|g|_D$ le représentant de $g \in \mathbf{Z}_D$ dans $[0, D-1]$.

Posons alors :

(2-6) $$f_i = |\Phi_\lambda(g_i)|_D \quad \text{pour } i = 0, 1, \dots, n$$

l'équation (2-5) s'écrit alors :

(2-7) $$\sum_{i=1}^{n} f_i x_i = f_0 + x_{n+1} D$$

avec x_{n+1} entier *positif* puisque le premier membre est positif.

2.4. Algorithme

Déterminons alors la solution du programme linéaire P_c

$$(P_c)\left\{\begin{array}{ll} \min z = \sum_{i=1}^{n} c_i x_i & \\ \text{avec} \quad \sum_{i=1}^{n} f_i x_i = f_0 + x_{n+1} D & \\ 0 \leqslant x_i \leqslant \alpha_i & \text{pour } i \text{ de } 1 \text{ à } n \\ x_{n+1} \geqslant 0 & \end{array}\right.$$

On peut supposer que les indices sont tels que :

(2-8) $$\frac{c_1}{f_1} \leqslant \frac{c_2}{f_2} \leqslant \dots \leqslant \frac{c_n}{f_n}.$$

Posons alors $F_0 = 0$ et $F_i = F_{i-1} + f_i \alpha_i$ pour i de 1 à n.

Soit i_0 l'indice tel que :

(2-9) $$F_{i_0-1} \leqslant f_0 < F_{i_0}$$

(si i_0 n'existe pas, alors P_c n'a pas de solution et à plus forte raison le programme P).

Alors la solution de P_c est :

(2-10) $$x_i^* = \alpha_i \quad \text{pour } i \in I = \{ i | i \leqslant i_0 - 1 \}$$

(2-11) $$x_{i_0}^* = \frac{f_0 - F_{i_0-1}}{f_{i_0}}$$

(2-12) $$x_i^* = 0 \quad \text{pour } i \in I = \{ i | i > i_0 \}$$

(2-13) $$z^* = \sum_{i \in I} c_i \alpha_i + c_{i_0} \frac{f_0 - F_{i_0-1}}{f_{i_0}}$$

Si $x_{i_0}^*$ donné par (2-11) est entier la solution de P_c est entière et c'est la solution de P.

Dans le cas contraire, nous allons faire un changement de variables qui va permettre de transformer le programme P en un programme presque du même type mais dans un groupe d'ordre plus petit.

Pour cela on élimine x_{i_0} de l'équation (2-7) et on pose :

(2-14) $$y_i = \alpha_i - x_i \quad , \text{ pour } i \in I$$

(2-15) $$y_{i_0} = x_{n+1}$$

(2-16) $$y_i = x_i \qquad \text{pour } i \in \bar{I}$$

Le programme P s'écrit alors :

(2-17) $$\min z = z^* + \sum_{i \in I} \left(\frac{c_{i_0}}{f_{i_0}} f_i - c_i \right) y_i + \frac{c_{i_0} D}{f_{i_0}} y_{i_0} + \sum_{i \in \bar{I}} \left(c_i - \frac{c_{i_0}}{f_{i_0}} f_i \right) y_i$$

(2-18) soumis à $0 \leqslant y_i \leqslant \alpha_i$, y_i entier, pour i de 1 à n.

(2-19) $$x_{i_0} = \frac{1}{f_{i_0}} \left(f_0 - F_{i_0 - 1} + D y_{i_0} + \sum_{i \in I} f_i y_i - \sum_{i \in \bar{I}} f_i y_i \right)$$

(2-20) $$0 \leqslant x_{i_0} \leqslant \alpha_{i_0} \quad \text{et } x_{i_0} \text{ entier.}$$

Les conditions (2-19) et (2-20) sont alors équivalentes à :

(2-21) $$0 \leqslant f_0 - F_{i_0 - 1} + D y_{i_0} + \sum_{i \in I} f_i y_i - \sum_{i \in I} f_i y_i \leqslant \alpha_{i_0} f_{i_0}$$

et

(2-22) $$\sum_{i \in I} (-f_i) y_i + D y_{i_0} + \sum_{i \in I} f_i y_i \equiv f_0 - F_{i_0 - 1} \pmod{f_{i_0}}$$

En supprimant la condition (2-21) le programme défini par (2-17), (2-18) et (2-22) est un programme du même type que P mais dans un *groupe* $Z_{f_{i_0}}$ *d'ordre plus petit que* D ($f_{i_0} = |\Phi(g_{i_0})|_D \leqslant D - 1$).

On refera sur ce programme la procédure effectuée sur P.

Au bout d'un nombre fini d'itérations ($< D$) la solution donnée par (2-11) est entière puisque le dénominateur peut décroître jusqu'à 1.

Puisqu'à chaque itération *on n'a pas perdu la condition d'intégrité des variables*, l'algorithme précédent donne une solution *entière* $\bar{x}$ et une *évaluation par défaut* $\bar{z}$ de la solution optimale du programme P.

Par contre à chaque itération *on a perdu la contrainte* $0 \leqslant x_{i_0} \leqslant \alpha_{i_0}$. Si pourtant on a $0 \leqslant \bar{x}_i \leqslant \alpha_i \; \forall_i \in I$, alors $\bar{x}$ est une *solution* optimale de P.

2.5. Exemple

$$\text{(2-23)} \qquad \min z = 5x_1 + 3x_2$$

avec

$$\text{(2-24)} \qquad \begin{gathered} 4x_1 + x_2 \equiv 3 \pmod 7 \\ x_1, x_2 \text{ entiers positifs.} \end{gathered}$$

Appliquons la méthode des congruences décroissantes en choisissant à chaque itération l'automorphisme identité.

L'équation (2-24) s'écrit :

$$\text{(2-25)} \qquad 4x_1 + x_2 = 3 + 7x_3$$

la solution du programme continu est :

$$\text{(2-26)} \qquad x^* = \left(\frac{3}{4}, 0\right) \quad , \quad z^* = \frac{15}{4}.$$

Le changement de variable, $y_1 = x_3, y_2 = x_2$, avec

$$\text{(2-27)} \qquad x_1 = \frac{1}{4}(3 + 7y_1 - y_2)$$

donne le nouveau programme

$$\text{(2-28)} \qquad \min z = \frac{15}{4} + \frac{1}{4}(35y_1 + 7y_2)$$

avec

$$\text{(2-29)} \qquad \begin{gathered} y_1 + y_2 \equiv 3 \pmod 4 \\ y_1, y_2 \text{ entiers positifs.} \end{gathered}$$

L'équation (2-29) s'écrit

$$\text{(2-30)} \qquad y_1 + y_2 = 3 + 4y_3$$

La solution du programme continu est :

$$\text{(2-31)} \qquad \underset{\text{entière}}{y^* = (0, 3)} \quad , \quad z^* = 9.$$

De (2-31) et (2-27), on déduit alors

$$\bar{x} = (0, 3) \quad , \quad \bar{z} = 9.$$

Comme $\bar{x} \geqslant 0$, c'est même la solution optimale de (2-23), (2-24).

3. QUELQUES CHOIX IMPORTANTS

Dans la procédure présentée au paragraphe 2, deux choix sont à faire à chaque étape. Le premier est le choix de l'automorphisme Φ_λ qui donnera l'équation (2-7); le second, moins visible, est le classement des indices dans (2-8) en cas d'égalité des rapports. Nous allons étudier ces deux choix.

3.1. Choix de l'automorphisme

Nous devons choisir un automorphisme parmi les $\varphi(D)$ possibles. Le but de la méthode étant d'obtenir la meilleure évaluation possible du programme P, sans trop de calculs, on aura donc intérêt à choisir, à chaque itération, l'automorphisme qui donne la plus grande valeur au programme P_c. Dans le cas où il n'existe pas de contraintes de bornes sur les variables, cet automorphisme Φ_λ est obtenu par la résolution de :

$$(\hat{P})\begin{cases} \max\limits_{\lambda} \; c_i \dfrac{|\lambda g_0|_D}{|\lambda g_i|_D} \\ \text{sous les contraintes} \quad \dfrac{c_i}{|\lambda g_i|_D} \leqslant \dfrac{c_j}{|\lambda g_j|_D} \\ \qquad\qquad\qquad\qquad\quad \operatorname{pgcd}(\lambda, D) = 1 \end{cases}$$

qui peut aussi s'écrire :

$$\max_{\lambda} \qquad \min_{j} \; |\lambda g_0|_D \frac{c_j}{|\lambda g_j|_D}$$
$$\operatorname{pgcd}(\lambda, D) = 1$$

La recherche de $\hat{\lambda}$ solution de $\hat{P}$ est trop longue dans le cas général et nous sommes amenés à définir une bonne heuristique. Il semble bien alors qu'une excellente heuristique est de donner la plus grande valeur possible à $f_0 = |\lambda g_0|_D$.

On recherchera donc les automorphismes qui donnent les meilleures valeurs à f_0.

Soit $\delta = \operatorname{pgcd}(D, |g_0|_D)$. On a alors le théorème suivant :

Théorème 1. *Il existe toujours un automorphisme tel que $f_0 = D - \delta$.*

Démonstration constructive.

D'après le théorème de Bezout, il existe une infinité de couple $(\lambda, \mu) \in Z^2$ tel que :

(3-3) $$-\lambda\, |g_0|_D + \mu D = \delta.$$

L'algorithme d'Euclide nous permet de déterminer un de ces couples soit (λ_1, μ_1). Les autres couples sont de la forme :

(3-4) $$\lambda = \lambda_1 + \alpha D'$$
$$\mu = \mu_1 + \alpha g'_0 \qquad \text{pour tout } \alpha \in Z$$

où $D' = D/\delta$ et $g'_0 = |g_0|_D/\delta$ avec pgcd $(\lambda_1, D') = 1$.

λ_1 et D' étant premiers entre eux, un théorème (1) de Lejeune-Dirichlet montre qu'il existe une infinité de nombres premiers de la forme $\lambda_1 + \alpha D'$. Nous pouvons alors en déduire qu'il existe un $\alpha = \alpha_0$ compris entre 0 et $\delta - 1$ tel que le $\lambda = \lambda_0$ correspondant soit premier avec δ.

Comme de plus, il est évident que λ_0 est premier avec D', λ_0 est premier avec D.

Si on choisit alors l'automorphisme Φ_{λ_0}, il donne la plus grande valeur à f_0 puisque l'on a :

$$|\Phi_{\lambda_0}(g_0)|_D = D - \delta.$$

Le théorème est donc démontré.

Si $\delta = 1$, il n'existe qu'un automorphisme tel que $f_0 = D - \delta$.

Si $\delta > 1$, l'équation (3-4) montre qu'il peut exister au plus δ valeurs possibles (mod D) pour λ.

Pratiquement on considérera tous ces automorphismes et on choisira celui qui maximise $\hat{P}$. Pour améliorer encore la résolution de $\hat{P}$ nous pouvons être amenés à chercher les automorphismes tel que $f_0 = D - k\delta$ $(k \geqslant 1)$.

On a alors le théorème suivant :

Théorème 2. *Il existe un automorphisme tel que $f_0 = D - k\delta$ si et seulement si* pgcd $(k\lambda_1, D') = 1$

Démonstration constructive.

Un automorphisme Φ_λ est tel que $f_0 = D - k\delta$ si il existe $\mu \in Z$ tel que :

(3-5) $$-\lambda\,|g_0|_D + \mu D = k\delta.$$

L'équation (3-3) pouvant s'écrire :

(3-6) $$-k\lambda_1\,|g_0|_D + k\mu_1 D = k\delta$$

on en déduit alors la forme générale des λ vérifiant (3-5), soit :

(3-7) $$\lambda = k\lambda_1 + \beta D' \quad \text{pour tout } \beta \in Z.$$

---

(1) Je remercie J. Pouget de m'avoir suggéré l'utilisation du théorème de Lejeune-Dirichlet.

Alors si pgcd $(k\lambda_1, D') \neq 1$, (3-7) entraîne pgdc $(\lambda, D) \neq 1$ et il n'existe pas d'automorphisme tel que $f_0 = D - k\delta$.

Si pgcd $(k\lambda_1, D') = 1$, alors l'application du théorème de Lejeune-Dirichlet montre qu'il existe au moins un $\beta = \beta_0$ compris entre 0 et $\delta - 1$ tel que le $\lambda = \lambda_0$ correspondant soit premier avec D. L'automorphisme Φ_{λ_0} répond alors à l'énoncé.

Si $\delta > 1$, l'équation (3-7) montre qu'il peut exister au plus δ valeurs possibles (mod D) pour λ. Pratiquement on considérera alors tous ces automorphismes et on choisira celui qui maximise $\hat{P}$.

3.2. Exemple

Reprenons ici l'exemple 2.5.

L'algorithme d'Euclide nous donne $\lambda_1 = 2$ avec pgdc $(\lambda_1, D) = 1$.

L'automorphisme Φ_2 transforme alors (2-24) en :

$$x_1 + 2x_2 = 6 + 7x_3$$

d'où la solution continue de P_c

$$\bar{x} = (0, 3) \quad , \quad \bar{z} = 9.$$

C'est la solution optimale du problème donné.

3.3. Exemple

$$\min z = 2x_1 + 3x_2 + x_3 \tag{3-7}$$

avec

$$12x_1 + 18x_2 + 7x_3 \equiv 2 \pmod{20} \tag{3-8}$$

$$x_i \geqslant 0$$

Itération 1. L'algorithme d'Euclide nous donne $\lambda_1 = 9$. L'équation (3-4) s'écrit :

$$\lambda = 9 + 10\alpha \tag{3-9}$$

on en déduit deux automorphismes possibles Φ_9 et Φ_{19} puisque pgcd $(9, 20) =$ pgcd $(19, 20) = 1$.

— L'automorphisme Φ_9 transforme (3-8) en

$$8x_1 + 2x_2 + 3x_3 = 18 + 20x_4 \tag{3-10}$$

d'où la solution continue

$$x^* = \left(\frac{9}{4}, 0, 0\right) \quad , \quad z^* = \frac{9}{2}. \tag{3-11}$$

— L'automorphisme Φ_{19} transforme (3-8) en :

$$8x_1 + 2x_2 + 13x_3 = 18 + 20x_4$$

d'où la solution continue

$$x^* = \left(0, 0, \frac{18}{13}\right) \quad , \quad z^* = \frac{18}{3}.$$

On choisit donc l'automorphisme Φ_9.

En posant $y_1 = x_4, y_2 = x_2, y_3 = x_3$ et

$$x_1 = \frac{1}{8}(18 + 20y_1 - 2y_2 - 3y_3) \tag{3-12}$$

le nouveau programme s'écrit :

$$\min z = \frac{9}{2} + \frac{1}{4}(20y_1 + 10y_2 + y_3) \tag{3-13}$$

avec

$$4y_1 + 2y_2 + 3y_3 \equiv 2 \pmod 8 \tag{3-14}$$

$$y_i \geqslant 0.$$

Itération 2. L'algorithme d'Euclide nous donne $\lambda_1 = 3$. L'équation (3-4) s'écrit :

$$\lambda = 3 + 4\alpha \tag{3-15}$$

on en déduit deux automorphismes possibles : Φ_3 et Φ_7 puisque pgcd $(3,8) =$ pgcd $(7,8) = 1$.

— L'automorphisme Φ_7 transforme (3-14) en :

$$4y_1 + 6y_2 + 5y_3 = 6 + 8y_4$$

d'où la solution continue :

$$y^* = \left(0, 0, \frac{6}{5}\right) \quad , \quad z^* = \frac{51}{10}.$$

— L'automorphisme Φ_3 transforme (3-14) en

$$4y_1 + 6y_2 + y_3 = 6 + 8y_4 \tag{3-16}$$

d'où la solution continue :

$$y^* = (0, 0, 6) \quad , \quad z^* = 6. \tag{3-17}$$

On choisit donc l'automorphisme Φ_3.

(3-18) Alors les équations (3-17) et (3-12) donnent $\bar{x} = (0, 0, 6)$. Comme $\bar{x} \geqslant 0$ c'est une solution optimale du problème.

3.4. Choix de i_0

Théoriquement on n'a pas besoin de s'occuper du choix du i_0 parmi les indices possibles. Mais pratiquement on cherche en même temps que l'évaluation par défaut du programme P une solution optimale de P, si possible. On aura donc intérêt à chercher toutes les solutions possibles du problème continu.

Pour simplifier l'exposé qui va suivre, supposons que les bornes des variables x_i ne jouent pas dans l'optimisation.

Si les variables sont indicées comme en (2-8), une solution optimale de P_c est :

$$x^* = \left(\frac{f_0}{f_1}\ 0, \ldots, 0\right).$$

Soit $I_0 = \left\{ i \mid \frac{c_i}{f_i} = \frac{c_1}{f_1} \right\}$.

On cherchera donc d'abord s'il existe un vecteur x_{I_0} entier positif tel que :

(3-19) $$\sum_{i \in I0} f_i x_i = f_0.$$

S'il existe une solution entière à ce problème, on aura la solution optimale du problème P.

Sinon considérons deux cas

cas 1. Soit $f_{i_0} = \min\limits_{i \in I_0} f_i$. Si $f_{i_0} \mid f_i$ pour tout $i \in I_0$.

On applique la méthode des congruences décroissantes en passant dans le groupe $Z_{f_{i_0}}$.

cas 2. Si f_{i_0} / f_i pour au moins un $i \in I_0$, on change l'équation (2-7) en considérant un autre automorphisme. On aura alors intérêt à choisir cet automorphisme parmi ceux qui donnent une grande valeur à f_0 autre que $D - \delta$, c'est-à-dire tel que $f_0 = D - 2\delta$, $D - 3\delta$, etc. (cf. théorème 2 pour déterminer de tels automorphismes). On résoud alors le programme P_c avec cette nouvelle équation à la place de (2-7).

3.5. Exemple

$$\min z = 2x_1 + 3x_2 + 3x_3 \tag{3-20}$$

avec

$$7x_1 + 6x_2 + 3x_3 \equiv 4 \quad (\text{mod } 9) \tag{3-21}$$
$$x_i \geqslant 0.$$

Itération 1. L'algorithme d'Euclide nous donne $\lambda_1 = 2$ avec pgcd $(\lambda, D) = 1$. L'automorphisme Φ_2 transforme alors (3-21) en :

$$5x_1 + 3x_2 + 6x_3 \equiv 8 + 9x_4 \tag{3-22}$$

d'où la solution continue :

$$x^* = \left(\frac{8}{5}, 0, 0\right) \quad , \quad z^* = \frac{16}{5}. \tag{3-23}$$

$$y_1 = x_4, y_2 = x_2, y_3 = x_3 \text{ et}$$

$$x_1 = \frac{1}{5}(8 + 9y_1 - 3y_2 - 6y_3), \tag{3-24}$$

d'où le nouveau programme

$$\min z = \frac{16}{5} + \frac{1}{5}(18y_1 + 9y_2 + 3y_3) \tag{3-25}$$

avec

$$y_1 + 3y_2 + y_3 \equiv 3 \quad (\text{mod } 5) \tag{3-26}$$
$$y_i \geqslant 0.$$

Itération 2. $\Phi = \Phi_3$.

$$3y_1 + 4y_2 + 3y_3 = 4 + 5y_4 \tag{3-27}$$

$$y^* = \left(0, 0, \frac{4}{3}\right) \quad , \quad z^* = 4.$$

$$t_1 = y_1, t_2 = y_2, t_3 = y_4 \text{ et}$$

$$y_3 = \frac{1}{3}(4 + 5t_3 - 3t_1 - 4t_2), \tag{3-28}$$

d'où le nouveau programme :

$$\min z = 4 + (3t_1 + t_2 + t_3) \tag{3-29}$$

avec

$$t_2 + t_3 \equiv 1 \quad (\text{mod } 3) \tag{3-30}$$
$$t_i \geqslant 0.$$

Itération 3. $\Phi = \Phi_2$.

(3-31) $$2t_2 + 2t_3 = 2 + 3t_4$$

Ici $\frac{c_2}{f_2} = \frac{c_3}{f_3}$. donc on cherche toutes les solutions entières positives de :

(3-32) $$2t_2 + 2t_3 = 2.$$

On trouve donc :

$t^* = (0, 0, 1)$ ce qui donne $\bar{x} = (-2, 0, 3)$

$t^* = (0, 1, 0)$ ce qui donne $\bar{x} = (1, 1, 0)$.

On a donc une solution optimale *(1, 1, 0)*.

3.5. Evaluation du nombre de calculs

Le nombre d'étapes dans « la méthode des congruences décroissantes » paraît être approximativement en log D. Le nombre de calcul est donc proportionnel à n log D.

La méthode s'appliquera avec succès pour D très grand et n moyen.

4. METHODES DES CONGRUENCES DECROISSANTES GENERALISEES

Au paragraphe 2, nous avons supposé que le groupe était cyclique.

Considérons ici le programme Q

$$(Q)\begin{cases} \min z = \sum_{i=1}^{n} c_i x_i \\ \text{avec } x \in X \end{cases}$$

ou

(4-1) $$X = \left\{ x \in Z^n \,\middle|\, \sum_{i=1}^{n} g_i x_i \equiv g_0 \text{ (dans } G_\Delta), 0 \leqslant x_i \leqslant \alpha_i \right\}$$

avec $(c_1, \ldots, c_n) \in \mathbf{R}^{+n}$, $(\alpha_1, \ldots, \alpha_n) \in \mathbf{N}^n$,

G_Δ = groupe somme directe des groupes cycliques $\mathbf{Z}_{\varepsilon_i}$ (i de 1 à n)

$$(g_1, \ldots, g_n) \in G_\Delta^n \quad , \quad g_0 \in G_\Delta.$$

4.1. Transformation de Q

L'équation linéaire (4-1) dans le groupe G_Δ se décompose en n équations linéaires dans les groupes cycliques $\mathbf{Z}_{\varepsilon_j}$:

$$\sum_{i=1}^{n} g_i^j x_i \equiv g_0^j \pmod{\varepsilon_j} \text{ pour } j \text{ de } 1 \text{ à } n. \tag{4-2}$$

Si $g_0 \neq 0$ (dans G_Δ), il existe un j_0 tel que $g^{j_0} \not\equiv 0(Z_{\varepsilon_{j_0}})$.

Considérons alors l'équation j_0 et un automorphisme Φ_λ de $\mathbf{Z}_{\varepsilon_{j_0}}$. Cette équation devient :

$$\sum_{i=1}^{n} \Phi_\lambda(g_i^{j_0}) x_i \equiv \Phi_\lambda(g_0^{j_0}) \qquad (\mathbf{Z}_{\varepsilon_{j_0}}) \tag{4-3}$$

c'est-à-dire en posant :

$$f_i = \left|\Phi_\lambda(g_i^{j_0})\right|_{\varepsilon_{j_0}} \qquad \text{pour } i = 0, 1, \ldots, n \tag{4-4}$$

$$\sum_{i=1}^{n} f_i x_i = f_0 + x_{n+1}\varepsilon_{j_0} \tag{4-5}$$

avec x_{n+1} entier *positif* puisque le premier membre est positif.

On considère alors, comme en (2-4), le programme P_c que l'on résoud en continu.

On fait alors le changement de variables (2-14), (2-15), (2-16). Le programme Q s'écrit alors :

$$\min z = z^* + \sum_{i \in I} \left(\frac{c_{i0}}{f_{i0}} f_i - c_i\right) y_i + \frac{c_{i0}\varepsilon_{j0}}{f_{i0}} y_{i0} + \sum_{i \in I} \left(c_i - \frac{c_{i0}}{f_{i0}} f_i\right) y_i \tag{4-6}$$

soumis à

$$0 \leqslant y_i \leqslant \alpha_i \quad , \quad y_i \text{ entier, } i \text{ de } 1 \text{ à } n \tag{4-7}$$

$$0 \leqslant f_0 - F_{i0-1} + \varepsilon_{j0} y_{i0} + \sum_{i \in I} f_i y_i - \sum_{i \in I} f_i y_i \leqslant \alpha_{i0} f_{i0} \tag{4-8}$$

$$\sum_{i \in I} (-f_i) y_i + \varepsilon_{j0} y_{i0} + \sum_{i \in I} f_i y_i \equiv f_0 - F_{i0-1} \pmod{f_{i0}} \tag{4-9}$$

et pour tout $j \neq j_0$.

$$\begin{aligned}\sum_{i \in I} (f_i g_{i0}^j - f_{i0} g_i^j) y_i + g_{i0}^j \varepsilon_{j0} y_{i0} + \sum_{i \in I} (f_{i0} g_i^j - f_i g_{i0}^j) y_i \\ \Xi f_{i0} g_0^j + g_{i0}^j (F_{i0-1} - f_0) - \sum_{i \in I} f_{i0} g_i^j \alpha_i \pmod{f_{i0}\varepsilon_j}.\end{aligned} \tag{4-10}$$

En supprimant la condition (4-8) le programme défini alors par (4-6), (4-7), (4-9) et (4-10) est un programme du même type que Q.

Si $\Delta = \prod_{j=1}^{n} \varepsilon_j$ était l'ordre du groupe G_Δ initial, l'ordre du nouveau groupe est $\Delta' = (f_{i0}) \prod_{j \neq j_0} (f_{i_0}\varepsilon_j) = \frac{\Delta \cdot (f_{i0})^n}{\varepsilon_{j0}}$. (Pratiquement l'ordre du groupe sera plus petit car l'équation (4-10) peut être divisée par pgcd (g_i^j, f_{i0}).)

On refera sur ce programme la procédure effectuée sur Q. Mais on ne pourra donc pas dire comme en 2-4 que l'ordre du groupe G_Δ' est plus petit que G_Δ et que la procédure s'arrête en un nombre fini d'étapes de ce fait.

Pratiquement, la procédure s'arrêtera pourtant en un nombre fini d'étapes car rapidement on aura $g_0 \equiv 0(Z_\Delta)$.

Puisque à chaque étape on n'a pas perdu la condition d'intégrité des variables, « la méthode des congruences décroissantes généralisée » donne une solution *entière* $\bar{x}$ et une *évaluation par défaut* $\bar{z}$ de la solution optimale de Q. Si de plus $0 \leqslant \bar{x}_i \leqslant \alpha_i$, $\bar{x}$ est une *solution optimale* de Q.

4.2. Quelques choix

A l'intérieur de la procédure précédente, il reste quelques choix à faire à chaque étape : choix de l'équation j_0, choix de l'automorphisme Φ_λ et choix de i_0 comme dans le paragraphe 3.

Ces choix sont des heuristiques permettant d'améliorer en moyenne la méthode.

Pour Φ_λ et i_0 nous ferons comme au paragraphe 3.

Pour j_0 nous prendrons :

(4-11) $$\varepsilon_{j0} = \min_j \{ \varepsilon_j \mid g_0^j \equiv 0 \text{ dans } \mathbf{Z}_{\varepsilon_j} \}.$$

4.3. Exemple

(4-12) $$\min z = 3x_1 + x_2$$

avec

(4-13) $$x_1 + 2x_2 \equiv 2 \quad (\text{mod } 3)$$

(4-14) $$4x_1 + 3x_2 \equiv 4 \quad (\text{mod } 6)$$

$$x_1, x_2 \text{ entiers positifs.}$$

Itération 1. (4-13) s'écrit :

(4-15) $$x_1 + 2x_2 = 2 + 3x_3$$

d'où la solution

$$x^* = (0, 1, 0) \quad , \quad z^* = 1.$$

En posant $y_1 = x_1, y_2 = x_3$ et

$$x_2 = \frac{1}{2}(2 - y_1 - 3y_2) \tag{4-16}$$

Le nouveau programme s'écrit :

$$\min z = 1 + \frac{1}{2}[5y_1 + 3y_2] \tag{4-17}$$

avec

$$y_1 + y_2 \equiv 0 \quad (\text{mod } 2) \tag{4-18}$$

$$5y_1 + 9y_2 \equiv 2 \quad (\text{mod } 12) \tag{4-19}$$

$$y_1, y_2 \text{ entiers positifs.}$$

Itération 2. En appliquant Φ_5 à (4-19) on obtient :

$$y_1 + 9y_2 = 10 + 12y_3 \tag{4-20}$$

d'où la solution :

$$y^* = \left(0, \frac{10}{9}\right), \; z^* = \frac{8}{3}.$$

En posant $t_1 = y_1$, $t_2 = y_3$ et

$$y_2 = \frac{1}{9}(10 + 12t_2 - t_1) \tag{4-21}$$

le nouveau programme s'écrit :

$$\min z = \frac{8}{3} + \frac{1}{3}(7t_1 + 6t_2) \tag{4-22}$$

avec

$$t_1 + 6t_2 \equiv 1 \quad (\text{mod } 9) \tag{4-23}$$

$$8t_1 + 12t_2 \equiv 8 \quad (\text{mod } 18) \tag{4-24}$$

$$t_1, t_2 \text{ entiers positifs.}$$

Itération 3. En appliquant Φ_8 à (4-23) on obtient :

$$8t_1 + 3t_2 = 8 + 9t_3 \tag{4-25}$$

d'où la solution

$$t^* = (1, 0) \quad , \quad z^* = 5.$$

En posant $u_1 = t_3$, $u_2 = t_2$ et

(4-26) $$t_1 = \frac{1}{8}(8 + 9u_1 - 3u_2)$$

le nouveau programme s'écrit :

(4-27) $$\min z = 5 + \frac{3}{8}(7u_1 + 3u_2)$$

avec

(4-28) $$7u_1 + 3u_2 \equiv 0 \quad (\text{mod } 8)$$

(4-29) $$9u_1 + 9u_2 \equiv 0 \quad (\text{mod } 18)$$

$$u_1, u_2 \text{ entiers positifs.}$$

Ici on a $g_0 \equiv 0$ dans G_Δ. Fin de l'algorithme.

En utilisant (4-21), (4-16) on trouve donc :

$$\bar{x} = (1, 2), \bar{z} = 5.$$

Comme $\bar{x} \geqslant 0$, c'est même la solution optimale de (4-12), (4-13), (4-14).

5. APPLICATIONS DE « LA METHODE DES CONGRUENCES DECROISSANTES »

5.1. Cas d'un programme tout entier

Dans le cas où la matrice des contraintes est entière et où les variables sont toutes entières, la solution du problème asymptotique donne une excellente évaluation de la solution optimale du programme initial. « La méthode des congruences décroissantes » ou, « la méthode des congruences décroissantes généralisée », qui donne une bonne évaluation du problème asymptotique donnera donc une bonne évaluation du programme initial.

De plus, comme nous le montrons dans [10], elle permet de résoudre exactement le problème asymptotique par une procédure arborescente.

5.2. Cas d'un programme mixte

Considérons le programme linéaire en variables mixtes :

$$(P_m) \begin{cases} \min z = cx \\ \text{avec } Ax = a \\ \qquad x \geqslant 0 \quad , \quad \text{certains } x_j \text{ entiers,} \end{cases}$$

où les A_{ij} et a_i sont entiers.

Transformons ce programme comme dans Gomory Johnson [6] (cf. Guignard [11]).

Soit I_0 une base optimale du programme continu lié à P_m, I les variables hors bases correspondant aux variables entières et J les variables hors bases correspondant aux variables continues.

Si la solution continue $x_{I_0} = (A^{I_0})^{-1}\, a = \bar{x}_{I_0}$ vérifie les contraintes d'intégrité, on a une évaluation exacte (et une solution) du programme P_m.

Sinon il existe un $i_0 \in I_0$ tel que $\bar{x}_{i_0}$ n'est pas entier.

Alors si $D = |\det(A^{I_0})|$, le tableau simplicial donne :

(5-1) $$x_{i0} = \frac{\alpha_{i0}}{D} - \sum_{j \in I} \frac{\beta_{i0,j}}{D} x_j - \sum_{j \in J} \frac{\beta_{i0,j}}{D} x_j$$

avec α_{i_0}, $\beta_{i0,j}$ entiers puisque A_{ij} et a_i sont entiers.

En posant :

(5-2) $$g_j \equiv \beta_{i0,j} \pmod{D} \text{ pour } j \in I$$

(5-3) $$g_0 \equiv \alpha_{i0} \pmod{D}$$

(5-4) $$J^+ = \{ j \in J \mid \beta_{i0,j} > 0 \}$$

(5-5) $$J^- = \{ j \in J \mid \beta_{i0,j} < 0 \}$$

(5-6) $$x^+ = \sum_{j \in J} \beta_{i0,j} x_j,\ \bar{x} = \sum_{i \in J-} \beta_{i0,j} x_j$$

La condition d'intégrité sur x_{i_0} transforme l'équation (5-1) en :

(5-7) $$\sum_{j \in I} g_j x_j + x^+ - x^- \equiv g_0 \qquad (\text{mod } D)$$

De plus dans le tableau simplicial la fonction économique s'écrit :

(5-8) $$z = \bar{z} + \sum_{j \in I \cup J} d_j x_j$$

ou :

(5-9) $$\bar{z} = c_{I_0}(A^{I_0})^{-1} a = c_{I_0} \bar{x}_{I_0}$$

(5-10) $$d_j = c_j - c_{I_0}(A^{I_0})^{-1} A^j \geqslant 0, \qquad j \in I \cup J.$$

Une évaluation par défaut de P_m pourra être une évaluation par défaut du programme Q_m obtenue par relaxation des contraintes de P_m :

(Q_m) $$\begin{cases} \min z = \bar{z} + \sum_{j \in I} d_j x_j + \sum_{j \in J^+} d_j x_j + \sum_{j \in J^-} d_j x_j \\ \text{avec} \qquad \sum_{j \in I} g_j x_j + x^+ - x^- \equiv g_0 \qquad (\text{mod } D) \\ \qquad x_j \geqslant 0, \qquad x_I \text{ entier} \end{cases}$$

5.3. Evaluation du programme Q_m

En remarquant que pour une solution optimale de Q_m on doit avoir x^+ *ou* x^- *nul, le minimum de* Q_m *sera obtenu pour le minimum des deux programmes suivants* :

$$(Q_m^+) \quad \begin{cases} \min z = \bar{z} + \sum_{j \in I} d_j x_j + d^+ x^+ \\ \text{avec} \quad \sum_{j \in I} g_j x_j + x^+ \equiv g_0 \pmod D \\ x_j, x^+ \geqslant 0, \quad x_j \text{ entiers} \end{cases}$$

et

$$(Q_m^-) \quad \begin{cases} \min z = \bar{z} + \sum_{j \in I} d_j x_j + d^- x^- \\ \text{avec} \quad \sum_{j \in I} g_j x_j + (-1)\, \bar{x} \equiv g_0 \pmod D \\ x_j, x^- \geqslant 0, \quad x_j \text{ entiers} \end{cases}$$

ou

$$(5\text{-}11) \qquad d^+ = \min_{j \in J^+} \left(\frac{d_j}{\beta_{i0,j}} \right), \qquad d^- = \min_{j \in J^-} \left(\frac{d_j}{\beta_{i0,j}} \right).$$

De plus les équations d'intégrité de Q_m^+ et Q_m^- montrent que x^+ *et* x^- *doivent être entiers.*

Les deux programmes Q_m^+ et Q_m^- pourront donc être évalués par la méthode des congruences décroissantes.

Le *minimum de ces deux évaluations* sera alors une bonne évaluation par défaut de Q_m, et par suite de P_m.

5.4. Remarque

La transformation du programme P_m et l'évaluation par défaut ont été obtenues en tenant compte de l'intégrité d'une des variables de base x_i. On pourrait obtenir d'autres évaluations en considérant les autres variables devant être entières.

On prendrait alors pour évaluation par défaut le minimum des évaluations trouvées.

REMERCIEMENT

Je tiens à remercier le rapporteur pour ses remarques pertinentes.

BIBLIOGRAPHIE

[1] BEALE (E. M. L.) and SMALL (R. E.), *Mixed Integer programming by a Branch and Bound Technique*, Proc. IFIP Congress 65, ed. W. H. Kalenich, vol. 2, 1966.

[2] DRIEBECK (N. J.), An Algorithm for the Solution of Mixed Integer Programming Problems, *Management Science*, 12, 1966, p. 497-520.

[3] FREHEL (J.), « Principe d'optimalité et énumération implicite ». Communication présentée à la journée combinatoire d'IBM en février 1972.

[4] FREHEL (J.), « Utilisation des conditions d'intégrité pour le calcul des fonctions d'évaluations ». Communication présentée à la journée combinatoire de l'AFCET le 1er juin 1972 (à paraître dans R.I.R.O.).

[5] GOMORY (R. E.), On the Relation between Integer and Non Integer Solutions to Linear Programs, *Proceedings of the Nacional Academy of Sciences*, vol. 53 1965, p. 260-265.

[6] GOMORY (R. E.) and JOHNSON (E. L.), Some Continuous Functions related to Corner Polyhedra, *Mathematical Programming*, vol. 3, 1972, n° 1, p. 23-85.

[7] GONDRAN (M.), Programmation linéaire en nombres entiers : optimisation dans un cône, *R.I.R.O.*, 4e année, R-2, 1970, p. 11-27.

[8] GONDRAN (M.), Programmation linéaire en nombres entiers, *Bulletin de la Direction des Etudes et Recherches d'EDF*, série C, 1970, n° 2, p. 5-32.

[9] GONDRAN (M.), Forte pénalité en programmation linéaire en nombre entiers méthode des congruences décroissantes, *Bulletin de la Direction des Etudes et Recherches d'EDF*, série C, 1972, n° 2.

[10] GONDRAN (M.), « Résolution d'un problème d'optimisation sur un groupe par une procédure arborescente ». Note EDF, HI 962/02 du 10 octobre 1972.

[11] GUIGNARD (M.), « Programmation mixte en nombres entiers : inégalités valides de Gomory-Johnson », Communication présentée à la journée combinatoire de l'AFCET du 2 décembre 1971.

[12] ROY (B.), BENAYOUN (R.) et TERGNY (J.), From SEP Procedure to the Mixed OPHELIE Program *in* : *Integer and Non Linear Programming*, Abadie (J.) Editor (chap. 20), North-Holland, 1970.

[13] TOMLIN (J. A.), Branch and bound method for integer and non-convex programming, *in* : *Integer and Non Linear Programming*, Abadie (J.) Editor (chap. 21), North-Holland 1970.