RAIRO. RECHERCHE OPÉRATIONNELLE

J. ABADIE

A. HAGGAG

Brève communication. Performance du gradient réduit généralisé avec une méthode quasi newtonienne pour la programmation non linéaire

RAIRO. Recherche opérationnelle, tome 13, nº 2 (1979), p. 209-216

<http://www.numdam.org/item?id=RO_1979__13_2_209_0>

R.A.I.R.O. Recherche opérationnelle/Operations Research
(vol. 13, n° 2, mai 1979, p. 209 à 216)

PERFORMANCE DU GRADIENT RÉDUIT GÉNÉRALISÉ AVEC UNE MÉTHODE QUASI NEWTONIENNE POUR LA PROGRAMMATION NON LINÉAIRE (*)

J. ABADIE, A. HAGGAG

Résumé. — *L'objet de cet article est d'exposer une variante de la méthode du Gradient Réduit Généralisé* (GRG) *pour la programmation non linéaire incluant une méthode quasi newtonienne pour calculer les directions de descente successives. On présente des expériences numériques, portant sur* 21 *problèmes-test, comparant les résultats des inclusions des options* BFGS *et Fletcher avec celle des gradients conjugués de Fletcher et Reeves. L'option* BFGS *apporte une économie de temps de* 40 % *en moyenne (et divise le temps par* 2,4 *lorsque le degré de liberté est au moins* 3).

Abstract. — *The purpose of this paper is to discuss a variant of the Generalized Reduced Gradient* (GRG) *algorithm for non-linear programming. This variant includes a quasi-Newton method to calculate successive search directions. Numerical experiences on* 21 *test-problems, exhibit the performance of the inclusions of* BFGS *or Fletcher switch option compared with the conjugate gradient of Fletcher and Reeves. The* BFGS *option brings a* 40% *saving in execution time over the original* GRG *(and divides execution time by* 2.4 *when the degree of freedom is at least* 3).

INTRODUCTION

La méthode GRG, classée première dans les comparaisons de Colville [12, 13, 14] est encore à ce jour, semble-t-il, la méthode la plus performante pour la résolution des programmes non linéaires [26], [28]. La recherche des directions de descente successives était faite, dans le code GRGA, par la méthode des gradients conjugués. De grands progrès ont été depuis réalisés dans l'optimisation sans contrainte, montrant à l'évidence que la conjugaison introduite par les méthodes quasi-newtoniennes était bien meilleure. Il nous a paru intéressant de nous en assurer avec cette variante GRGAH, dans le cadre d'un code existant.

Considérons le programme non linéaire :

$$\text{sous les contraintes}\quad \left.\begin{array}{ll} \operatorname{Min} F(X), & \\ f_i(X)=0, & i=1, \ldots, m, \\ \alpha \leqq X \leqq \beta, & \end{array}\right\} \tag{1}$$

où X est un vecteur de $\mathbf{R}^n$, ainsi que α et β qui définissent un parallélotope P, et où F, f_i sont des fonctions réelles continûment dérivables dans un ouvert contenant P. Si certaines contraintes étaient des inégalités, on les mettrait sous

(*) Reçu novembre 1978.
([1]) Électricité de France, Clamart.

la forme standard (1) en introduisant des variables d'écart. Les contraintes, notées $f(X)$ pour exprimer les composantes $f_i(X)$, $i=1, \ldots, m$, définissent dans $\mathbf{R}^n$ une variété V de dimension $n-m$. Nous notons par $F'(X)$ la dérivée de $F(X)$, dont nous faisons une matrice-ligne; la même notation s'applique à $f_i'(X)$. Enfin, $f'(X)$ désigne la matrice dont l'élément (i, j) est

$$\frac{\partial f_i}{\partial X_j}.$$

Le principe général de la méthode du Gradient Réduit Généralisé (GRG) [1, 2] est le suivant.

1. Choisir un point initial $X^0 \in P \cap V$ (dans la suite, pour simplifier la présentation et sauf mention expresse du contraire, on supposera pour tout j, $\alpha_j = -\infty$ et $\beta_j = +\infty$; nous renvoyons le lecteur à [1, 7] pour le traitement complet), puis séparer les variables en variables basiques (dépendantes, en nombre m, formant un vecteur y) et non basiques (indépendantes, en nombre $n-m$, formant un vecteur x) de sorte qu'en posant

$$X^0 = (y^0, x^0) \qquad \text{et} \qquad f'(X^0) = \left(\frac{\partial f}{\partial y^0}, \frac{\partial f}{\partial x^0}\right) = (B, N),$$

la matrice B (la base) soit régulière (condition de non-dégénérescence). La même séparation est appliquée à

$$F'(X^0) = \left(\frac{\partial F}{\partial y^0}, \frac{\partial F}{\partial x^0}\right) = (C_B, C_N).$$

D'après le théorème des fonctions implicites, il existe un voisinage Ω de x^0 et une fonction continue unique y telle que

$$f(y(x), x) = 0, \qquad \forall x \in \Omega, \quad y(x^0) = y^0.$$

La dérivée existe et est continue dans un voisinage de x^0, en outre

$$\frac{dy}{dx^0} = -B^{-1} N.$$

Lorsqu'on exprime la fonction économique selon les seules variables indépendantes, on a $\Phi(x) = F(y(x), x)$; la méthode consiste à faire une itération du problème sans contrainte :

$$\operatorname{Min} \Phi(x). \tag{2}$$

2. Calculer le gradient réduit, qui est la dérivée de $\Phi(x)$:

$$g^T = \Phi'(x) = C_N - C_B B^{-1} N,$$

où $u = -C_B B^{-1}$ est une estimation du multiplicateur de Lagrange. Ici se placent les tests d'arrêt (par exemple, g suffisamment proche de zéro).

3. Déterminer la direction de recherche

$$d = \begin{pmatrix} d_B \\ d_N \end{pmatrix} = \begin{pmatrix} k \\ h \end{pmatrix};$$

a) choisir une direction h tel que le produit scalaire $g^T h$ soit strictement négatif, et un pas λ dans cette direction pour faire progresser les variables indépendantes :

$$x^{i+1} = x^i + \lambda h^i;$$

b) la direction des variables dépendantes est donnée par

$$k = -B^{-1}(Nh).$$

4. Calculer les variables dépendantes en résolvant le système de m équations à m variables y :

$$f(x^{i+1}, y) = 0,$$

ce que l'on fait par une méthode pseudo-newtonienne [5, 6] :

$$\tilde{y}^{j+1} = \tilde{y}^j - B^{-1} f(x^{i+1}, \tilde{y}^j),$$

où $\tilde{y}^1 = y^0 + \lambda k$ et où B^{-1} est calculé une fois pour toute au début de l'itération extérieure en X^0. Par conséquent, les itérations intérieures de la méthode pseudo-newtonienne ont une convergence linéaire à condition que le pas λ soit assez petit.

Remarque : Le choix des indices des colonnes de B est important.

Lorsque des variables sont à une borne, elles ne doivent pas appartenir à la base, la même règle s'applique lorsque $\tilde{y}^j$ atteint une borne. On pourra voir Abadie et Guigou [7] et Abadie [1, 3] pour plus de détails concernant le choix de base et les changements de base.

5. Améliorer la valeur de la fonction économique $\Phi(x)$ et retourner en 2.

L'itération du problème (2) était effectuée dans le code GRGA [2] selon la méthode des gradients conjugués de Fletcher et Reeves [20] qui peut se décrire comme suit :

$$(a)\qquad h^0 = -g^0$$

$$(b)\qquad h^i = -g^i + \frac{|g^i|^2}{|g^{i-1}|^2} h^{i-1}, \qquad i=1, \ldots, LC,$$

où $|\ |$ désigne la norme euclidienne et LC le nombre de variables libres (degré de liberté, ici $n-m$); ceci constitue un cycle, à la fin duquel on reprend en (*a*). On note qu'à chaque itération i une recherche unidimensionnelle est nécessaire pour déterminer le pas λ dans la direction. S'il y a changement de base ou si une variable atteint une borne on reprend aussi en (*a*). Le lecteur pourra consulter les références citées [1, 7] pour voir comment la méthode est modifiée lorsque des X_j sont à une borne α_j, β_j finie. La méthode de Fletcher et Reeves améliore nettement les résultats par rapport à la méthode du gradient où $h=-g$, et a constitué en son temps un progrès décisif. Il est cependant bien connu que les méthodes quasi newtoniennes sont en général bien supérieures à la méthode des gradients conjugués; pour un exposé complet *voir* Avriel [10] ou Himmelblau [24]. Himmelblau [23] a d'ailleurs comparé 15 algorithmes sur 14 problèmes-test sans contrainte : la méthode de Fletcher et Reeves arrive 13^e, les deux premières étant celle de Fletcher [18] et celle de Davidon, Fletcher et Powell [15, 19] dite DFP; cependant une autre méthode quasi-newtonienne n'a pas été testée par Himmelblau, celle de Broyden, Fletcher, Goldfarb et Shanno [11, 18, 21, 27] dite BFGS, alors qu'actuellement beaucoup pensent qu'elle est la meilleure (*voir* par exemple Dennis et Moré [16]).

Notre nouveau code GRGAH [9, 22] a recours à ces méthodes quasi newtoniennes qui offrent plusieurs options; de plus la recherche unidimensionnelle est évitée ou raccourcie.

DÉTERMINATION DE LA NOUVELLE DIRECTION DE RECHERCHE

La direction de recherche h est donnée par

$$h^i = -H^i g^i \qquad (H^0 = I \text{ c'est-à-dire } h^0 = -g^0),$$

où H est une matrice définie positive d'ordre $n-m$. On notera que dans les problèmes pratiques un grand nombre de bornes est atteint par x à l'optimum, et même bien avant dans le cours des itérations. Si n_b est le nombre de ces bornes atteintes, l'ordre de H est $n-m-n_b$ qui est généralement petit. En outre, il n'est pas nécessaire d'effectuer une recherche unidimensionnelle, c'est-à-dire de déterminer un pas λ qui minimise $\Phi(x^i+\lambda d^i)$; cette recherche est extrêmement coûteuse en appels de fonction et l'un des avantages des méthodes quasi newtoniennes est de l'éviter. De plus, le pas initial $\lambda=1$ est un choix normal [17], alors qu'aucun choix ne s'impose dans la méthode de Fletcher et Reeves. En fait, si avec le pas choisi on obtient un nouveau point qui améliore la valeur de la fonction de façon satisfaisante (grâce à un test d'acceptabilité, voir Fletcher [18]), on s'en contente et on continue une nouvelle itération; pour plus de détails sur le choix du pas et pour une discussion complète voir Haggag [22].

FORMULE DE MISE A JOUR

La mise à jour de H à chaque itération est assurée par la formule générale de Broyden [11] dépendant d'un paramètre φ selon les valeurs duquel on obtient les formules usuelles de DFP ou de BFGS ou toute autre combinaison. Comme la formule de mise à jour retenue permet de conserver la symétrie de H on ne garde en mémoire de l'ordinateur qu'une demi-matrice triangulaire stockée sous forme de tableau unidimensionnel.

On pose pour simplifier

$$H=H^i, \qquad H^+=H^{i+1}, \qquad p=x^i-x^{i-1}, \qquad q=g^i-g^{i-1}.$$

Lorsque φ décrit $(0,1)$ l'expression générale suivante :

$$H^+=(1-\varphi)\left[H+\frac{pp^T}{q^Tp}-\frac{Hqq^TH^T}{q^THq}\right]+\varphi\left[\left[I-\frac{pq^T}{q^Tp}\right]H\left[I-\frac{qp^T}{q^Tp}\right]+\frac{pp^T}{q^Tp}\right]$$

donne les formules de la famille de Broyden [11]; pour $\varphi=0$ on retrouve DFP et lorsque $\varphi=1$ on a BFGS. Une des propriétés de ces méthodes est que les matrices H successives restent définies positives.

La première version de GRGAH appelée GRGAF passe de l'une à l'autre des formules précédentes selon le résultat d'un test (Fletcher [18]) si $p^Tq<q^THq$ on prend $\varphi=0$, sinon $\varphi=1$. La deuxième version GRGABFGS utilise seulement la formule BFGS.

RÉSULTATS NUMÉRIQUES

Le travail que nous présentons consiste à comparer les trois variantes de GRG : GRGA, GRGAF, GRGABFGS obtenues respectivement en associant GRG avec les méthodes des gradients conjugués de Fletcher et Reeves, et les méthodes quasi-newtoniennes DFP et BFGS. Nous utilisons deux séries de problèmes-test, ceux de Colville [12] et ceux de Himmelblau [24]. Colville [13] avait effectué une étude comparative avec 30 méthodes sur 8 problèmes, les derniers résultats [14] montraient que la méthode classée première était la méthode GRG [5, 6, 8], de nombreuses améliorations ont été apportées depuis pour augmenter la sûreté de fonctionnement et la facilité d'utilisation. La machine utilisée est une IBM 370/168 de la Direction des Études et Recherches d'Électricité de France avec OPT=2 correspondant à un temps standard de Colville [12] de 3,9 secondes.

(*a*) Problèmes de Colville :

	GRGA	GRGABFGS		GRGA	GRGABFGS		GRGA	GRGABFGS
1 (*cl*)	53	52	3 (*f*)	75	34	6	154	85
2 (*f*)	286	217	3 (*nf*)	86	42	7 (*cl*)	270	150
2 (*nf*)	367	317	4 (*s*)	69	35	8	171	90

Temps d'exécution en millisecondes.
(*cl*), contraintes linéaires; (*s*), sans contraintes; *f* (*nf*), point de départ faisable (non faisable).

Afin de comparer les deux variantes GRGA, GRGABFGS nous calculons le facteur f_{ap}, défini dans [8] qui est le quotient

$$\frac{(a, p)}{\min_a (a, p)},$$

où (a, p) est l'évaluation correspondant à la résolution du problème p par l'algorithme a. Nous choisissons d'entendre ici par évaluation le temps d'exécution. Le facteur temps moyen (moyenne arithmétique des f_{ap}) est de 1,69 pour GRGA alors qu'il est de 1,00 pour GRGABFGS. C'est-à-dire que le temps de résolution moyen d'un problème par GRGABFGS est 0,59 fois le temps de GRGA.

En conséquence, l'option BFGS apporte une amélioration substantielle sur celle des gradients conjugués.

(*b*) Problèmes d'Himmelblau

Nous avons également résolu les 21 problèmes d'Himmelblau (qui contiennent les problèmes de Colville). Nous obtenons avec les trois variantes le classement suivant :

1. GRGABFGS;
2. GRGAF;
3. GRGA,

avec respectivement les facteurs temps moyen de 1,04; 1,27; 1,62.

On note que l'amélioration apportée par les variantes quasi-newtoniennes se remarque d'autant plus que le degré de liberté est significatif (le degré de liberté est la quantité $n-m-n_b$ définie plus haut).

C'est ainsi que pour ceux des 21 problèmes où le degré de liberté est au moins trois, on trouve respectivement pour les trois temps moyens ci-dessus 1,00; 1,47; 2,41.

CONCLUSION

L'introduction d'une des méthodes quasi-newtoniennes les plus performantes pour les problèmes sans contrainte et son adaptation dans le code du Gradient Réduit Généralisé qui est réputé le plus rapide et le plus fiable depuis des années [4, 25, 26, 28] rendent cette nouvelle variante encore plus performante.

BIBLIOGRAPHIE

1. J. Abadie, *Application of the* GRG *Algorithm to Optimal Control Problems*, in J. Abadie, ed., Integer and Nonlinear Programming, North-Holland, Amsterdam, 1970, p. 191-211.
2. J. Abadie, *Méthode du Gradient Réduit Généralisé : le code* GRGA, Note HI 1756/00, Électricité de France, Paris, février 1975.
3. J. Abadie, *The* GRG *Method for Non-linear Programming*, p. 335-362, *in* H. J. Greenberg, ed., Design and Implementation of Optimization Software, Sijthoff and Noordhoff, Alphen aan den Rijn, The Netherlands, 1978.
4. J. Abadie, *Advances in Non-linear Programming*, in K. B. Haley, ed., Operational Research 78, North-Holland, Amsterdam, 1978, p. 900-930.
5. J. Abadie et J. Carpentier, *Généralisation de la méthode du gradient réduit de Wolfe au cas de contraintes non linéaires*, Note HR 6678, Électricité de France, Paris, (octobre 1965).
6. J. Abadie J. Carpentier, *Generalization of the Wolfe Reduced Gradient Method to the Case of Nonlinear Constraints*, in R. Fletcher, ed., Optimization, Academic Press, London, 1969, p. 37-47.
7. J. Abadie et J. Guigou, *Gradient Réduit Généralisé*, Note HI 069/02, Électricité de France, Paris, avril 1969.
8. J. Abadie et J. Guigou, *Numerical Experiments with the* GRG *Method*, in J. Abadie, ed., Integer and Non-linear Programming, North-Holland, Amsterdam, 1970.
9. J. Abadie et A. Haggag, *Méthode quasi-newtonienne dans une variante du Gradient Réduit Généralisé* (GRGAH), Note HI 2458/00, Électricité de France, Paris, août 1977.
10. M. Avriel, *Nonlinear Programming*, Prentice-Hall, Englewood Cliffs, New Jersey, 1976.
11. C. G. Broyden, *A new Double-Rank Minimization Algorithm*, Notices Amer. Math. Soc., vol. 16, 1969, p. 670.
12. A. R. Colville, *A Comparative Study on Non-Linear Programming Codes*, Rep. 320-2949, N. Y. Scientific Center, IBM Corp, Yorktown Heights, New York, 1968.

13. A. R. COLVILLE, *Non-Linear Programming Study Results as of June* 1970 (private circulation).
14. A. R. COLVILLE, *A Comparative Study on Nonlinear Programming codes*, in H. W. KUHN, ed., Proceedings of the Princeton Symposium on Mathematical Programming, Princeton University Press, Princeton, New Jersey, 1970.
15. W. C. DAVIDON, *Variable Metric Method for Minimization*, Rep. ANL-5990, Rev. Argonne National Laboratoires, Argonne, Ill., 1959.
16. D. E. DENNIS et J. J. MORÉ, *Quasi Newton Methods, Motivation and Theory*, S.I.A.M. Review, vol. 19, (1), 1977, p. 46-89.
17. L. C. W. DIXON, *The Choice of Step Length, a Crucial Factor in the Performance of Variable Metric Algorithms*, in F. LOOTSMA, ed., Numerical Methods for Non-Linear Optimization, Academic Press, London, 1972, p. 149-170.
18. R. FLETCHER, *A New Approach to Variable Metric Algorithms*, Computer J., vol., 13, 1970, p. 317-322.
19. R. FLETCHER et M. J. D. POWELL, *A Rapidly Convergent Descent Method for Minimization*, Computer J., vol. 6, 1963, p. 163-168.
20. R. FLETCHER et C. M. REEVES, *Function Minimization by Conjugate Gradients*, Computer J., vol. 7, 1964, p. 149-154.
21. D. GOLDFARB, *A Family of Variable Metric Methods Derived by Variational Means*, Math. Comp., vol. 24, 1970, p. 23-26.
22. A. H. HAGGAG, *Études d'algorithmes d'optimisation non linéaires : une variante de* GRGA, These, C.N.R.S. n° TD493, 6-12-76, Université Pierre-et-Marie-Curie, Paris, 1976.
23. D. M. HIMMELBLAU, *A Uniform Evaluation of Unconstrained Optimization Techniques*, in F. A. LOOTSMA, ed., Numerical Methods for Nonlinear Optimization, Academic Press, London, 1972, p. 69-97.
24. D. M. HIMMELBLAU, *Applied Nonlinear Programming*, McGraw-Hill, New York, 1972.
25. F. A. LOOTSMA, *Performance Evaluation of Non-Linear Program Codes from the Viewpoint of a Decision Maker*, Paper presented at the IFIP WG 2.5 Working Conference on Performance Evaluation on Numerical Software, Baden (Austria), 11-15 December 1978, and at the 5th Conference on Mathematical Programming, Matrafüred (Hungary), 22-26 January 1979.
26. E. SANDGREN, *The Utility of Nonlinear Programming Algorithms*, Ph. D. Thesis, Purdue University, December 1977.
27. D. F. SHANNO, *Conditioning of Quasi-Newton Methods for Function Minimization*, Mathematics of Computation, vol. 24, 1970, p. 617-656.
28. R. L. STAHA, *Constrained Optimization via Moving Exterior Truncations*, Ph. D. Thesis, The University of Texas at Austin, May 1973.
29. K. SCHITTKOWSKI, *A Numerical Comparison of* 13 *Nonlinear Programming Codes with Randomly Generated Test Problems*, to appear in : L. C. W. DIXON and G. P. SZEGO, eds, Numerical Optimisation of Dynamical Systems, North-Holland Publishing Company, Amsterdam, 1979.