Operations Research

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Technical Note—A Modified Conjugate Gradient Algorithm

Avinoam Perry,

OPERATIONS RESEARCH
Vol. 26, No. 6, November-December 1978
0030-364X/78/2606-1073 $01.25

Technical Notes

A Modified Conjugate Gradient Algorithm

AVINOAM PERRY
Bell Laboratories, Holmdel, New Jersey

(Received April 1976; accepted December 1977)

We present a modification of the Fletcher-Reeves conjugate gradient algorithm. Computational experience indicates that this modification yields an improved algorithm.

WE PRESENT a modification of the Fletcher-Reeves (FRCG) conjugate gradient algorithm [3]. This modification results in a more stable computational performance than the ones exhibited by either FRCG or PRCG (Polak-Ribiere conjugate gradient algorithm [7, 10]). We attribute this stable behavior to the modification proposed here, which incorporates one of the most important features of the quasi-Newton algorithm into the conjugate gradient procedure (this feature is missing in FRCG and PRCG) without increasing storage requirements.

We shall present the motivation for the modification and support these arguments by a sample of encouraging computational results.

1. DERIVATION OF THE MODIFIED CONJUGATE GRADIENT ALGORITHM

Let $p_k \equiv x_{k+1} - x_k$ and $q_k \equiv g_{k+1} - g_k$, where $g_k \equiv \nabla f(x_k)$ and the unconstrained optimization problem is minimize $f(x)$, $x \in R^n$. Let $d_k \equiv (1/\alpha_k) p_k$, where α_k is a scalar minimizing the one-dimensional function $h(\alpha) = f(x_k + \alpha d_k)$, $(\alpha \geqq 0)$. Conjugate directions in R^n have the property:

$$p_k' F_k p_{k+1} = \alpha_k^2 d_k' F_k d_{k+1} = 0 \tag{1}$$

where F_k is the Hessian of $f(x_k)$. If $f(x)$ is quadratic and F_k is constant (i.e., $F_k = F$), then (1) is equivalent to the requirement

$$q_k' p_{k+1} = 0. \tag{2}$$

A conjugate gradient direction at stage $k+1$ is constructed by taking a linear combination of the negative gradient at stage $k+1$ and the direction vector at stage k.

$$d_{k+1}=-g_{k+1}+\beta_k d_k. \quad (3)$$

Equation (2) implies that $\beta_k=q_k'g_{k+1}/q_k'd_k$ and (3) becomes

$$d_{k+1}=-g_{k+1}+(q_k'g_{k+1}/q_k'd_k)\cdot d_k=-D_{k+1}g_{k+1} \quad (4)$$

where

$$D_{k+1}=(I-p_kq_k')/p_k'q_k. \quad (4a)$$

If $f(x)$ is quadratic and α is computed with perfect accuracy, we have $q_k'g_{k+1}=(g_{k+1}-g_k)'g_{k+1}=g_{k+1}'g_{k+1}$ and $q_k'd_k=(g_{k+1}-g_k)'d_k=(g_{k+1}-g_k)'(-g_k+\beta_{k-1}d_{k-1})=g_k'g_k$, and (4) becomes

$$d_{k+1}=-g_{k+1}+(g_{k+1}'g_{k+1}/g_k'g_k)d_k. \quad (5)$$

Equation (5) is the well-known Fletcher-Reeves conjugate gradient direction [3]. Relaxing the assumption regarding the orthogonality of consecutive gradient vectors but assuming perfect line search, we have: $d_k'g_{k+1}=0$ and (4) becomes

$$\begin{aligned} d_{k+1}&=g_{k+1}-(q_k'g_{k+1}/g_k'd_k)d_k=-g_{k+1}-(q_k'g_{k+1}/g_k'(-g_k+\beta_{k-1}d_{k-1}))d_k \\ &=-g_{k+1}+(q_k'g_{k+1}/g_k'g_k)d_k. \end{aligned} \quad (6)$$

Equation (6) is the well-known Polak-Ribiere conjugate gradient direction [7, 10].

The conclusions reached upon the above derivations are that the Fletcher-Reeves procedure produces conjugate directions if and only if $f(x)$ is quadratic and line search is of perfect accuracy. If any one of these conditions is violated, then the directions produced by FRCG are not conjugate. The Polak-Ribiere method does not require $f(x)$ to be quadratic, but, nevertheless, the directions produced by PRCG are conjugate to each other only if line-search accuracy is perfect. In the absence of the above conditions the performances of the above algorithms tend to deteriorate if the algorithm is not restarted periodically. Our computational experience verifies the results obtained by McGuire and Wolfe [8] and Powell [12].

Although periodic restarts improve the performance of PRCG and more so FRCG, the noise injected into these algorithms because of inaccurate line search causes the overall performance to vary greatly in an inconsistent manner. It is possible to relax the assumptions implied by (5) and (6) by resorting to (4), which generates conjugate directions regardless of these assumptions. However, the success of the quasi-Newton methods is attributed to the fact that the conjugate directions requirement is replaced by the requirement $q_k'S_{k+1}=p_k'$, where S_{k+1} is an updated positive definite matrix approximating the inverse Hessian of $f(x_k)$ at stage k, such that $d_{k+1}=-S_{k+1}g_{k+1}$. For nonquadratic functions the conjugate directions property is replaced by the ceahp approximation

$$q_k'd_{k+1}=0. \quad (7)$$

Equation (7) is satisfied by the quasi-Newton methods of the Huang family [6] only if line-search accuracy is perfect. Whenever line-search accuracy is less than perfect, (7) is replaced by

$$q_k' d_{k+1} = -q_k' S_{k+1} g_{k+1} = -p_k' g_{k+1}. \tag{8}$$

Note that if (7) is to be enforced under any circumstances, then S_{k+1} must become semidefinite so that $q_k' S_{k+1} = 0$.

The above results imply that a better algorithm can be constructed if the conjugate directions requirement in (7) is replaced by (8). Since the conjugate direction in (4) yields the result in (7), one can add a rank one correction to D_{k+1} in (4a) such that the new direction yields the result in (8). Applying these conclusions to the conjugate gradient equation results in the modification of (4) to

$$d_{k+1} = -g_{k+1} + ((q_k - \alpha_k d_k)' g_{k+1} / q_k' d_k) \cdot d_k = -S_{k+1} g_{k+1} \tag{9}$$

where $S_{k+1} = D_{k+1} + p_k p_k' / p_k' q_k$, so that (8) is satisfied.

Since (8) is equivalent to (7) whenever line search is perfectly accurate, it follows that, under such circumstances, (9) is equal to (6) and the modfication becomes effective whenever $p_k' g_{k+1} \neq 0$.

2. COMPUTATIONAL EXPERIENCE

Experiments with the new modified conjugate gradient algorithm (PMCG) as well as Fletcher-Reeves (FRCG), Polak-Ribiere (PRCG), and the quasi-Newton method BFGS [1, 2, 5, 14] involved six well-known test problems, denoted here as functions 1 through 6 (see the appendix for a detailed description of each function and its source). All problems were solved by each one of the algorithms under two different line-search accuracy measures. The line-search technique applied is the well-known quadratic interpolation method [4]. To ensure successful implementation of the line search procedure, a unimodal region was secured before the first interpolation was performed. Line-search accuracy was measured by the index $\delta = |p_{k+1}' g_{k+2}| / |p_{k+1}' g_{k+1}|$. Performance of a given algorithm was measured by total number of stages and total number of function evaluations (in parentheses). We use the term "stage" to define the step carrying a point x_k along a direction d_k to a new point $x_{k+1} = x_k + \alpha_k d_k$. It follows that the total number of stages per algorithm is equal to the total number of gradient evaluations.

The statistics "total number of function evaluations" include the number of gradient evaluations multiplied by n. (If one is interested in number of function evaluations not including gradient evaluations, the total number of stages multiplied by n should be subtracted from the total number of function evaluations.) Each time a direction vector pointed upward rather than downward, it was replaced by the direction of steepest descent. Although we do not provide statistics regarding the number of times per experiment this phenomenon took place, we note that this procedure was a significant factor in determining the overall performance of FRCG and PRCG whenever line-search accuracy measures were light.

The stopping rule applied throughout was $\nabla|f(x^*)|\leqq 10^{-5}$. If an experimental run exceeded 200 stages before reaching such a point x^* the run was terminated and its function value was reported instead.

All computer programs were coded in APL using interactive mode [9] and were run on the CDC6400 computer at Northwestern University.

In the following tables we present our computational results under two measures of line-search accuracy. These measures are denoted as mode 1 ($\delta\leqq 0.1$) and mode 2 ($\delta\leqq 0.001$), where δ is the index of line-search accuracy defined above. The one-dimensional search was terminated whenever the above constraint became satisfied. Our computational study is divided into two parts. In the first part no restarts were initiated, and in the second all algorithms were restarted after a multiple of $(n+1)$ stages (where n is the dimension of the function in question). In order to minimize unnecessary details, we combined the data on functions 2, 3, 4 into one table and provided separate tables for functions 1, 5, and 6.

3. CONCLUDING REMARKS

On the basis of our computational experience, we arrived at the following conclusions:

(a) For the two-dimensional function minimization problem there are no significant performance differences among PRCG, PMCG, and BFGS.

(b) For the two-dimensional function minimization problem all of the above algorithms perform better in most cases when not restarted.

(c) For problems of dimensions greater than two, BFGS without restarts performed better than all the conjugate gradient procedures.

(d) All conjugate gradient procedures are expected to exhibit significant improvement in their performances when restarted periodically.

(e) Although PMCG does not always perform better than FRCG or PRCG, it should be preferred to these two procedures because it is more stable in most cases and its performance does not depend so heavily on random noise.

(f) There is no significant change in the performance of PMCG as the line search accuracy is changed.

(g) When a conjugate gradient algorithm is needed to solve the function minimization problem, we recommend PMCG with periodic restarts and minimum line search.

ACKNOWLEDGMENT

I wish to express my sincere appreciation to L. S. Lasdon, M. A. Saunders, and D. F. Shanno, whose comments on an earlier draft resulted in a significant improvement in the presentation.

APPENDIX

The following functions were used in our computational study.

1: $100(x_2-x_1^2)+(1-x_1)^2$ [13]

2: $(x_2-x_1^2)^2+(1-x_1)^2$ [15]

3: $(x_2-x_1^2)^2+100(1-x_1)^2$ [15]

4: $100(x_2-x_1^3)^2+(1-x_1)^2$ [15]

5: $100(x_2-x_1^2)^2+(1-x_1)^2+90(x_4-x_3^2)^2$ $+(1-x_3)^2+10.1[(x_2-1)^2+(x_4-1)^2]$ $+19.8(x_2-1)(x_4-1)$ [16]

6: $(x_1+10x_2)^2+5(x_3-x_4)^2+(x_2-2x_3)^4+10(x_1-x_4)^4$ [11]

Algorithm	No Restarts		Restarts	
	Mode 1	Mode 2	Mode 1	Mode 2
Function 1, $x_0=-1.2, 1$				
FRCG	65 (583)	0.2145	33 (346)	29 (426)
PRCG	24 (221)	21 (285)	30 (346)	29 (443)
PMCG	23 (212)	16 (227)	30 (324)	27 (414)
BFGS	25 (214)	18 (250)	33 (359)	29 (443)
Functions 2, 3, 4, $x_0=-1.2, 1$				
FRCG	39 (398)	26 (315)	36 (322)	25 (265)
PRCG	41 (425)	24 (276)	35 (318)	25 (268)
PMCG	34 (324)	23 (253)	35 (319)	24 (260)
BFGS	33 (327)	22 (268)	33 (315)	26 (281)
Function 5, $x_0=-3, -1, -3, -1$				
FRCG	8.032	2.115	70 (806)	134 (1725)
PRCG	193 (1914)	$1.5\cdot10^{-6}$	130 (1589)	82 (1113)
PMCG	177 (1750)	167 (1736)	94 (1006)	67 (944)
BFGS	38 (373)	39 (539)	69 (793)	54 (769)
Function 6, $x_0=3, -1, 0, 1$				
FRCG	$4.61\cdot10^{-6}$	$3.12\cdot10^{-7}$	63 (758)	$3.98\cdot10^{-8}$
PRCG	112 (1190)	75 (831)	56 (687)	78 (1204)
PMCG	85 (846)	79 (862)	53 (639)	60 (837)
BFGS	19 (192)	18 (257)	14 (170)	22 (340)

REFERENCES

1. C. G. Broyden, "The Convergence of a Class of Double Rank Algorithms, Parts I and II," *J. Inst. Math. Appl.* **7,** 76–90, 222–236 (1971).
2. R. Fletcher, "A New Approach to Variable Metric Algorithms," *Comp. J.* **13,** 317–322 (1970).
3. R. Fletcher and C. M. Reeves, "Function Minimization by Conjugate Gradients," *Comp. J.* **7,** 149–154 (1964).
4. R. L. Fox, *Optimization Methods for Engineering Design*, pp. 51–55, Addison Wesley, 1973.
5. Goldfarb, D., "A Family of Variable Metric Methods Derived by Variational Means," *Math. Comp.* **24,** 23–26 1970.
6. H. Y. Huang, "Unified Approach to Quadratically Convergent Algorithms for Function Minimization," *J.O.T.A.* **5,** 405–423 (1970).
7. R. Klessig and E. Polak, "Efficient Implementation of the Polak-Ribiere Conjugate Gradient Algorithm," *SIAM J. Control* **10,** 524–549 (1972).
8. M. F. McGuire and P. Wolfe, "Evaluating a Restart Procedure for Conjugate Gradients," Report RC-4382, IBM Research Center, Yorktown Heights, 1973.
9. A. Perry, "UCNLP-An Interactive Package of Programs for Unconstrained Nonlinear Optimization Purposes," a Working Paper, November 1975, revised May 1976.
10. E. Polak and G. Ribiere, "Note sur la Convergence des Methodes de Directions Conjugées," *Rev. Francaise Inf. Rech. Oper.* **16 RI,** 35–43 (1969).
11. M. J. D. Powell, "Recent Advances in Unconstrained Optimization," *Math. Prog.* **1,** 26–57 (1971).
12. M. J. D. Powell, "Restarts Procedures for the Conjugate Gradient Method," *Math. Prog.* **2,** 241–254 (1977).
13. H. H. Rosenbrock, "Automatic Method for Finding the Greatest or Least Value of a Function," *Comp. J.* **3,** 175–184 (1960).
14. D. F. Shanno, "Conditioning of Quasi-Newton Methods for Function Minimization," *Math. Comp.* **24,** 647–656 (1970).
15. B. F. Whitte and W. R. Holst, Paper submitted at the 1964 Spring Joint Computer Conference, Washington, D.C., 1964.
16. C. F. Wood, Westinghouse Research Labs., 1968.

Published in Operations Research on December 01, 1978 as DOI: 10.1287/opre.26.6.1073.
This article has not been copyedited or formatted. The final version may differ from this version.