JOURNAL OF MATHEMATICAL ANALYSIS AND APPLICATIONS 46, 533–541 (1974)

A Modified Homogeneous Algorithm for Function Minimization

D. H. JACOBSON AND L. M. PELS*

Department of Applied Mathematics, University of the Witwatersrand, Johannesburg, South Africa

Submitted by George Leitmann

Recently, an algorithm for function minimization was presented, based upon an homogeneous, rather than upon a quadratic, model. Numerical experiments with this algorithm indicated that it rapidly minimizes the standard test functions available in the literature. Although it was proved that the algorithm produces function values which continually descend, no proof of convergence was supplied.

In this paper, the homogeneous algorithm is modified primarily by replacing the cubic interpolation routine by Armijo's step size rule. Although not quite as fast as the original version on the standard test functions, this modified form has the advantage that a proof of convergence follows from a general theorem of Polak.

1. INTRODUCTION

In [1] a new algorithm for function minimization was presented. This algorithm, unlike most other algorithms for function minimization, is based upon a homogeneous model rather than on a quadratic. A consequence of this is that the algorithm converges under certain conditions [1] to the minimum of a homogeneous function (a quadratic function is homogeneous of degree 2) in $n + 2$ steps. On general functions the algorithm performed well and in almost all cases it was markedly superior to that of Fletcher and Powell [2]. Although it was proved in [1] that the algorithm descends at each iteration, it was not proved that the algorithm actually converges to the minimum of the function it is attempting to minimize.

In this paper we present a modified version of the homogeneous algorithm in which the step size selection by cubic interpolation [2] is replaced by Armijo's rule [3]. Although the resulting algorithm is somewhat slower than the original version, it is more robust in that a proof of convergence is obtained by using a general theorem of Polak [4].

* Partially supported by the South African Council for Scientific and Industrial Research.

2. Preliminaries

A function $F(x)$, $F: R^n \to R^1$, having a unique minimum at $x = \beta$ is said to be homogeneous of degree γ if it satisfies the equation

$$F(x) = \frac{1}{\gamma}(x - \beta)^T g(x) + \bar{\omega}, \tag{1}$$

where

$$g(x) \triangleq \frac{\partial F(x)}{\partial x} \tag{2}$$

and where $\bar{\omega}$ is the minimum value of the function. Upon defining $v \triangleq x^T g(x)$,

$$y \triangleq \begin{bmatrix} g(x) \\ F(x) \\ -1 \end{bmatrix}, \qquad \alpha \triangleq \begin{bmatrix} \beta \\ \gamma \\ \omega \end{bmatrix}, \qquad \text{and} \qquad \omega \triangleq \gamma\bar{\omega}, \tag{3}$$

Equation (1) can be written as

$$y^T \alpha = v. \tag{4}$$

If $n + 2$ points x_i $(i = 1,..., n + 2)$ and the associated values $F(x_i)$ and $g(x_i)$ $(i = 1,..., n + 2)$ are available, then the following relation holds

$$Y\alpha = V, \tag{5}$$

where

$$Y \triangleq \begin{bmatrix} y_1^T \\ \vdots \\ y_{n+2}^T \end{bmatrix} \qquad \text{and} \qquad V \triangleq \begin{bmatrix} v_1 \\ \vdots \\ v_{n+2} \end{bmatrix} \tag{6}$$

and where

$$y_i = \begin{bmatrix} g(x_i) \\ F(x_i) \\ -1 \end{bmatrix} \qquad \text{and} \qquad v_i = x_i^T g(x_i) \qquad (i = 1,..., n + 2) \tag{7}$$

Clearly, if $Y \in R^{(n+2)\times(n+2)}$ is invertible, α is uniquely obtained from (5) so that, for homogeneous functions, the minimizing point β, the degree of homogeneity γ, and the scaled minimum value of the function ω are found.

In order to find the minimum of a nonhomogeneous function, successive estimates of the minimum are obtained with (5) and the "most recent" $n + 2$ estimates x_i, x_{i-1},..., x_{i-n-1}. See [1] for a complete description of the original algorithm.

3. Modified Algorithm

We wish to minimize a general, not necessarily homogeneous, function $F(x), F\colon R^n \to R^1$, with respect to x. The modified version of the homogeneous algorithm appropriate for this task is as follows.

Step 1. Assume x_0, η_1, η_2, and N to be given; set $\gamma_0 = 2$, $\omega_0 = 0$, and $i = 0$.

Step 2. Compute $p_0 \triangleq -g(x_0)$ and use Armijo's subprocedure to calculate ρ_0.

Step 3. Set $x_1 = x_0 + \rho_0 p_0$.

Step 4. Set $\alpha_0{}^T = [x_1, \gamma_0, \omega_0]$, $P_0 = I$, and $j = 1$.

Step 5. If $\| g(x_{i+1}) \| = 0$, stop; otherwise, go to step 6.

Step 6. Calculate y_{i+1} and v_{i+1}. If $| y_{i+1}^T P_i e_j | \leqslant \eta_1$, set $x_0 = x_{i+1}$ and go to step 1; otherwise, use Eqs. (8) and (9) to calculate P_{i+1} and α_{i+1} and go to step 7.

Step 7. Set $i = i + 1$. If $j = n + 2$, reset $j = 1$; otherwise set $j = j + 1$.

Step 8. If $|(x_i - \beta_i)^T g(x_i)| < \eta_2$, set $x_0 = x_i$ and go to step 1; otherwise, go to step 9.

Step 9. Set $p_i = \sigma_i(x_i - \beta_i)$, where $\sigma_i = -\text{sign}[(x_i - \beta_i)^T g(x_i)]$.

Step 10. If $\| p_i \| + | \gamma_i | \leqslant N$, use Armijo's subprocedure to calculate ρ_i; otherwise, set $x_0 = x_i$ and go to step 1.

Step 11. Set $x_{i+1} = x_i + \rho_i p_i$; go to step 5.

In the above steps (see [1]),

$$P_{i+1} = P_i - \frac{P_i e_j (y_{i+1}^T P_i - e_j{}^T)}{y_{i+1}^T P_i e_j},$$

$$\alpha_{i+1} = \alpha_i + \frac{P_i e_j (v_{i+1} - y_{i+1}^T \alpha_i)}{y_{i+1}^T P_i e_j},$$

and

$$P_{i+1}^{-1} = P_i^{-1} + e_j(y_{i+1}^T - e_j P_i^{-1}) \tag{10}$$

and

$$V_{i+1} = V_i + e_j(v_{i+1} - e_j{}^T V_i), \qquad V_0 = \alpha_0 \tag{11}$$

where e_j is a unit vector whose components are zero except for the jth which is unity.

The *Armijo subprocedure* is as follows

Step 1. Set $k(x_i) = 0$ and $\rho_i(k(x_i)) = 1$.

Step 2. If

$$F(x_i + \rho_i(k(x_i))\, p_i) - F(x_i) + \frac{\rho_i(k(x_i))}{|\gamma_i| + 2} \,|\, p_i^T g(x_i)| \leqslant 0, \tag{12}$$

set $\rho_i = \rho_i(k(x_i))$ and return[1]; otherwise, set $k(x_i) = k(x_i) + 1$ and go to step 3.

Step 3. Set $\rho_i(k(x_i)) = \rho_i(k(x_i))/2k(x_i)$ and go to step 2.

4. Computational Results

The modified algorithm was tested on three functions, given in Tables I–III. In each case computation was terminated when $\| g(x_i)\| \leqslant 10^{-4}$. N was chosen to be 10^{12}.

TABLE I

Rosenbrock's Function[a] (Fletcher and Powell [2])

Algorithm	Number of Evaluations of $F(x)$
Homogeneous Algorithm	69
Modified Algorithm	80
Fletcher and Powell's Algorithm	167

[a] $F(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2$. The starting point was $(-1.2, 1)$.

TABLE II

Quartic with Singular Hessian [2][a]

Algorithm	Number of Evaluations of $F(x)$
Homogeneous Algorithm	64
Modified Algorithm	75
Fletcher and Powell's Algorithm	80.

[a] $F(x) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4$. The starting point was $(3, -1, 0, 1)$.

[1] In our Theorem, this $k(x_i)$ is referred to as $\hat{k}(x_i, p_i, \gamma_i)$.

TABLE III

Four-Dimensional Banana [5][a]

Algorithm	Number of Evaluations of $F(x)$
Homogeneous Algorithm[b]	192
Modified Algorithm[b]	270
Fletcher and Powell's Algorithm[b]	648
Homogeneous Algorithm[c]	154
Modified Algorithm[c]	263
Fletcher and Powell's Algorithm[c]	595

[a] $F(x) = 100(x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_3^2 - x_4)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)(x_4 - 1)$.

[b] The starting point was $(-1.2, 1, -1.2, 1)$.

[c] The starting point was $(-3, 1, -3, 1)$.

Note. Both the homogeneous algorithm and Fletcher and Powell's Algorithm require evaluation of $g(x)$ each time $F(x)$ is calculated (this is required by the cubic interpolation step size routine). In our modified algorithm, evaluation of $g(x)$ is not required by the Armijo rule, so that, although our function evaluation counts are higher than those of the homogeneous algorithm, the net computational work done is less.

5. Polak's Algorithm Model [4]

The algorithm model searches for a point with a given desirable property ϕ. We have

(i) an operator $A: R^n \to \{\text{all subsets of } R^n\}$ and

(ii) a stop rule $c: R^n \to R^1$.

Desirable points are those with property ϕ.

The algorithm model is as follows

Step 1. Select an $x_0 \in R^n$.

Step 2. Set $i = 0$,

Step 3. Compute a point $y \in A(x_i)$.

Step 4. Set $x_{i+1} = y$.

Step 5. If $c(x_{i+1}) \geqslant c(x_i)$, stop; otherwise, set $i = i + 1$ and go to step 2.

THEOREM (POLAK'S). *Suppose* (i) $c(\cdot)$ *is either continuous at all nondesirable points* $x \in R^n$ *or else* $c(x)$ *is bounded from below in* R^n *and* (ii) *for every* $x \in R^n$ *which is not desirable,* $\exists \epsilon(x) > 0$ *and* $\delta(x) < 0$, *such that*

$$c(x'') - c(x') \leqslant \delta(x) < 0 \qquad \text{for all } x'' \in A(x') \quad \text{and} \quad x' \in R^n,$$

where $\| x' - x \| \leqslant \epsilon(x)$. *Then either the sequence* $\{x_i\}$ *is finite and its next to last element is desirable or else it is infinite and every accumulation point is desirable.*

Proof. See Polak [4].

To apply this theorem to our algorithm, we make the following definitions and assumptions:

(i) x_i is desirable iff $\| g(x_i)\| = 0$.

(ii) Let $F(\cdot)$ correspond to $c(\cdot)$ in the model.

Let $F(\cdot)$ be continuously differentiable in R^n and assume that there exists an $x_0 \in R^n$ such that the set $\{x \mid F(x) \leqslant F(x_0)\}$ is compact. This compactness assumption, together with the fact that because of Armijo's rule $F(x_{i+1}) < F(x_i)$ (as we shall see below), implies that $\{x_i\}$ is bounded and therefore has accumulation points.

(iii) Let $N > 0$ be such that

$$N - 2 \geqslant \sup_x \| g(x)\| \qquad \text{for } x \in \{x \mid F(x) \leqslant F(x_0)\}. \tag{13}$$

MAIN THEOREM. *Let* $\{x_i\}$ *be a sequence in* R^n *generated by the algorithm. Then either the sequence is finite and terminates at a desirable point or else it is infinite and every accumulation point* x^* *of* $\{x_i\}$ *is desirable.*

Proof. If the sequence is finite, the test at Step 5 ensures that the last point is desirable. In the case of an infinite sequence, we need to prove that conditions (i) and (ii) of Polak's theorem are satisfied. Clearly, (i) is satisfied by the assumption that $F(\cdot)$ is continuous.

To prove (ii) satisfied, we note that either

(a) $p_i = \sigma_i(x_i - \beta_i)$ or

(b) $p_i = -g(x_i)$.

When (a) occurs, $|(x_i - \beta_i)^T g(x_i)| \geqslant \eta_2$, and choosing $\epsilon_1 > 0$ [this can always be done because of (13)] so that $\epsilon_1 \| g(x_i)\|^2 \leqslant \eta_2$ gives

$$-p_i^T g(x_i) \geqslant \eta_2 \geqslant \epsilon_1 \| g(x_i)\|^2.$$

For (b),

$$-p_i^T g(x_i) = \| g(x_i)\|^2.$$

In either case, there is an $\epsilon > 0$ such that

$$-p_i^T g(x_i) \geqslant \epsilon \| g(x_i) \|^2. \tag{14}$$

Note also that $\| p_i \| + | \gamma_i | \leqslant N$.

Define

$$A(x) \triangleq \{ y = x + \rho[\hat{k}(x, p, \gamma)]\, p \mid p \in D(x) \},$$

where $\rho[\hat{k}(x, p, \gamma)]$ is the largest ρ, $0 < \rho \leqslant 1$, generated by the Armijo subprocedure, to satisfy

$$F(x + \rho[k(x, p, \gamma)]\, p) - F(x) - \frac{\rho[k(x, p, \gamma)]\, p^T g(x)}{| \gamma | + 2} \leqslant 0 \tag{15}$$

and where

$$D(x) \triangleq \{ p \mid \| p \| + | \gamma | \leqslant N \text{ and } -p^T g(x) \geqslant \epsilon \| g(x) \|^2 \}. \tag{16}$$

For x nondesirable we define

$$\Delta[\lambda(x, p, \gamma)] \triangleq F(x + \lambda(x, p, \gamma)\, p) - F(x) - \frac{1}{| \gamma | + 2} \lambda(x, p, \gamma)\, p^T g(x)$$

$$= - \lambda(x, p, \gamma) \left\{ p^T [g(x) - g(\xi)] - \left(1 - \frac{1}{| \gamma | + 2}\right) p^T g(x) \right\} \tag{17}$$

$$\leqslant - \lambda(x, p, \gamma) \{ p^T [g(x) - g(\xi)] + (1 - \tfrac{1}{2})\, \epsilon \| g(x) \|^2 \} \tag{18}$$

for $\xi \in [x, x + \lambda(x, p, \gamma)\, p]$ and for all $p \in D(x)$.

Consider now the expression

$$\tilde{\Delta}[\lambda] = - \lambda(x, \tilde{p}, \gamma) \left\{ \tilde{p}^T [g(x) - g(\tilde{\xi})] + \frac{\epsilon}{2} \| g(x) \|^2 \right\}, \tag{19}$$

where

$$\tilde{p} \in \tilde{D} \triangleq \{ p : \| p \| + | \gamma | \leqslant N \}$$

and

$$\tilde{\xi} \in [x, x + \lambda(x, \tilde{p}, \gamma)\, \tilde{p}] \qquad \text{for } x \in R^n.$$

Now since $\| \tilde{p} \|$ is bounded and $g(x)$ is continuous, we have that there exists a $\bar{\lambda}(x) > 0$ of form $1/2^i i!$, such that

$$\tilde{\Delta}[\bar{\lambda}(x)] \leqslant \delta(x) < 0 \qquad \text{for all } \tilde{p} \in \tilde{D}. \tag{20}$$

A fortiori this implies that

$$\Delta[\bar{\lambda}(x)] \leqslant \delta(x) < 0 \qquad \forall p \in D(x), \tag{21}$$

which implies the existence of $\rho[\hat{k}(x, p, \gamma)]$.

By continuity of $g(x)$,

$$-\bar{\lambda}(x)\{\hat{p}^T[g(x') - g(\xi')] + \tfrac{1}{2}\|g(x')\|^2\} \leqslant \delta(x)/2 \tag{22}$$

$$\forall \hat{p} \in \tilde{D} \qquad \text{and} \qquad \forall x' \in \{x' : \| x - x' \| \leqslant \epsilon(x)\} \triangleq B(x, \epsilon(x)).$$

This implies, again a fortiori, that

$$F(x' + \bar{\lambda}(x)\, p) - F(x') - \bar{\lambda}(x) \cdot \frac{1}{|\gamma| + 2}\, p^T g(x') \leqslant \frac{\delta(x)}{2}, \tag{23}$$

$\forall x' \in B(x, \epsilon(x))$ and $\forall p \in D(x')$.

Clearly, we have $\rho[\hat{k}(x', p, \gamma)] \geqslant \bar{\lambda}(x)$, where

$$F(x' + \rho[\hat{k}(x', p, \gamma)]\, p) - F(x') - \rho[\hat{k}(x', p, \gamma)]\, \frac{1}{|\gamma| + 2}\, p^T g(x') \leqslant 0, \tag{24}$$

which is

$$\begin{aligned} F(x' + \rho[\hat{k}(x', p, \gamma)]\, p) - F(x') &\leqslant \rho[\hat{k}(x', p, \gamma)]\, \frac{1}{|\gamma| + 2}\, p^T g(x') \\ &\leqslant \bar{\lambda}(x)\, \frac{1}{N + 2}\, p^T g(x') \\ &\leqslant -\frac{\epsilon \bar{\lambda}(x)}{N + 2}\, \| g(x')\|^2 \\ &\leqslant -\frac{\epsilon \bar{\lambda}(x)}{2(N + 2)}\, \| g(x)\|^2 \end{aligned}$$

$$\forall x' \in B(x, \epsilon(x)) \qquad \text{and} \qquad \forall p \in D(x'),$$

which satisfies the conditions of Polak's theorem.

6. Conclusion

The homogeneous algorithm presented in [1] has been modified by replacing the cubic interpolation subprocedure with Armijo's subprocedure. The performance of this modified algorithm is slightly inferior to that of the original algorithm on the standard test functions, but the modifications make it possible to prove convergence of the method with a general theorem of Polak.

References

1. D. H. Jacobson and W. Oksman, An algorithm that minimizes homogeneous functions of N variables in $N + 2$ iterations and rapidly minimizes general functions *J. Math. Anal. Appl.* **38** (1972), 535.

2. R. Fletcher and M. J. D. Powell, A rapidly convergent descent method for minimization *Comput. J.* **6** (1963), 163.
3. L. Armijo, Minimization of functions having continuous partial derivatives, *Pacific J. Math.* **16** (1966), 1–3.
4. E. Polak, Computational Methods in Optimization: A Unified Approach, Academic Press, New York, 1971.
5. A. R. Colville, A Comparative Study of Nonlinear Programming Codes, Technical Report No. 320-2949, IBM New York Scientific Center, New York, 1968.