JOURNAL OF OPTIMIZATION THEORY AND APPLICATIONS: Vol. 13, No. 6, 1974

New Approach to Comparison of Search Methods Used in Nonlinear Programming Problems

O. I. LARICHEV[1] AND G. G. GORVITS[2]

Communicated by H. Y. Huang

Abstract. This paper is concerned with the problem of investigating the properties and comparing the methods of nonlinear programming. The steepest-descent method, the method of Davidon, the method of conjugate gradients, and other methods are investigated for the class of essentially nonlinear valley functions.

Key Words. Nonlinear programming, numerical methods, unconstrained minimization, function minimization.

1. Statement of the Problem

About 100 methods are available today for the computer-aided search for the extremum of a nonlinear function of many variables; new methods are forthcoming. To solve his problem, an engineer has to select one of these methods. The problem is made unwieldy by the fact that there is no yardstick for the applicability of methods to real-life situations.

Nonlinear programming methods can be compared in two ways. The first approach can be termed *analytical* and is to prove that the methods under consideration converge for a certain subclass of nonlinear functions. Then, a comparison criterion is suggested, which in most cases is the convergence rate. Analytical expressions for the criterion are compared, and the relative advantages of the methods are determined.

In spite of its obvious superiority, this approach is hard to apply, because of the mathematical difficulties involved. The convergence of of many methods has been proved for only sufficiently simple, quadratic

[1] Research Scientist, Institute of Control Problems, USSR Academy of Sciences, Moscow, USSR.

[2] Research Scientist, Institute of Control Problems, USSR Academy of Sciences, Moscow, USSR.

or convex, functions (Ref. 1) or with some stringent constraints imposed on the functions. The convergence rate for strictly unimodal functions (Ref. 2) has not been found for any method.

The second approach can be termed *experimental.* The performances of methods in terms of some criterion (such as the number of function evoluations, the convergence rate, the number of iterations, *etc.*) are compared by using certain test functions. The literature has reported 10 to 20 test functions extensively used in such comparisons (Refs. 3–5).

For all the practical value of the experimental approach, the results are often dependent on the test functions themselves. Furthermore, for the same test function, the relative advantage of a method depends on the initial point selected. Both facts impair an objective evaluation of the methods.

This paper proposes a *middle-way approach* to the comparison of nonlinear programming methods. The objective of the paper is to compare some nonlinear programming methods for a broad class of unimodal functions, a class incorporating all known test functions.

The problems discussed are unconstrained minimization problems. Penalty functions (Ref. 6) help in reducing constrained minimization problems to these problems.

Section 2 describes the nonlinear programming methods, and Section 3 discusses the class of functions under consideration. Sections 4–7 analyze the properties of the methods for two-dimensional functions belonging to the class under consideration. Section 8 presents the basic ideas of our approach to the comparison of methods, and Sections 9–12 give the results of the comparison. Section 13 contains the experimental comparison of the methods for test functions of various dimensionality, and Section 14 summarizes the conclusions.

2. Search Methods

This section will describe the search algorithms which can be regarded as belonging to one class, because all of them have the following properties.

(a) The problem is to find the local minimum of the function $f(x)$ of the n-vector x. A point x^* is to be found such that $g(x^*) = 0$, where $g(x)$ is the gradient of $f(x)$. The search is iterative.

(b) At each iteration, a direction leading to a decrease of the function is selected, and a minimum along that direction is determined.

(c) The values of $f(x)$ and $g(x)$ alone are used in the search.

Assume that the procedure used in the one-dimensional search is the same in all the algorithms and that it leads to the accurate determination of the minimum; in other words, the least positive root α is determined by the equation

$$f_\alpha(x_i + \alpha p_i) = 0, \tag{1}$$

where $f(x)$ is the function to be minimized, x_i is the initial point along the ith direction, and p_i is the ith direction of minimization.

With this assumption, the search algorithms are different in the directions selected. Therefore, the assumed criterion for the efficiency of a method can be defined as the number of minimization directions required to reach the vicinity of the extremum.

The paper will be concerned with the following search methods:

(i) steepest-descent method (SD, Ref. 2),
(ii) accelerated method of parallel tangents (APT, Ref. 2),
(iii) conjugate-gradient method (CG, Ref. 5),
(iv) Davidon's method (D, Ref. 7),
(v) general form of variable metric algorithms (VM, Ref. 8).

Let us indicate how these methods select the search directions:

(SD) $$p_i = -g_i, \qquad g_i = \operatorname{grad} f(x_i); \tag{2}$$

(APT) $$p_{2i-2} = -g_{2i-2}, \tag{3}$$

$$p_{2i} = -g_{2i}, \tag{4}$$

$$p_{2i-1} = x_{2i-1} - x_{2i-3}, \tag{5}$$

with the first two steps made along the antigradient;

(CG) $$p_{i+1} = -g_{i+1} + (g_{i+1}^T g_{i+1}/g_i^T g_i) p_i; \tag{6}$$

(D) $$p_{i+1} = -H_{i+1} g_{i+1}, \tag{7}$$

$$H_{i+1} = H_i + A_i + B_i, \tag{8}$$

$$A_i = \sigma_i \sigma_i^T / \sigma_i^T y_i, \tag{9}$$

$$B_i = -H_i y_i y_i^T H_i / y_i^T H_i y_i, \tag{10}$$

$$y_i = g_{i+1} - g_i, \tag{11}$$

$$\sigma_i = x_{i+1} - x_i = \alpha_i p_i, \tag{12}$$

where α_i is the stepsize obtained by minimizing the function along the direction selected and where H_i is a symmetric and positive-definite matrix, if H_0 is also symmetric and positive definite.

The Davidon method belongs to the class of variable metric algorithms, which have the form

$$p_i = -H_i{}^T g_i ,$$

where the matrix H_i is constructed in a specified way. Let us consider those variable metric algorithms (Ref. 8) for which

$$\text{(VM)} \quad H_i = H_{i-1} + \Delta H_{i-1} , \tag{13}$$

$$\Delta H_i = \rho(\Delta x_{i-1} y_{i-1}^T / y_{i-1}^T \Delta g_{i-1}) - H_{i-1} \Delta g_{i-1} z_{i-1}^T / z_{i-1}^T \Delta g_{i-1} , \tag{14}$$

$$\Delta x_{i-1} = x_i - x_{i-1} , \tag{15}$$

$$\Delta g_{i-1} = g_i - g_{i-1} , \tag{16}$$

$$y_{i-1} = c_1 \Delta x_{i-1} + c_2 H_{i-1}^T \Delta g_{i-1} , \tag{17}$$

$$z_{i-1} = k_1 \Delta x_{i-1} + k_2 H_{i-1}^T \Delta g_{i-1} . \tag{18}$$

By varying the parameters ρ, c_1, c_2, k_1, k_2, we will have different algorithms. For $\rho c_1 = k_2 = 1$ and $c_2 = k_1 = 0$, we obtain the Davidon method; for $\rho c_1 = k_1 = 1$ and $c_2 = k_2 = 0$, we obtain the McCormick method; and for $\rho c_1 = k_1 = 0$ and $c_2 = k_2 = 1$, we obtain the Pearson method.

For quadratic functions, the D-method, the CG-method, and the VM-method converge within n steps, where n is the number of variables; the APT-method converges within $2n - 1$ steps, and the SD-method converges at the rate of geometric progression.

3. Class of Functions

The behavior of the methods is studied for the class of differentiable unimodal functions. One specific feature of this type of functions is valleys. The valley is described best in geographical terms (rather than mathematical terms): its surface is represented by a mountainous terrain with a river flowing in the gorge. The valley bottom can be regarded as a river bed, and the valley generating line as the direction of flow. A valley can be characterized by the steepness of the walls, the width of the bottom, and the slope along the river, i.e., the rate at which the valley bottom decreases along the generating line. The valley bottom and the river bed can be either straight or curving.

The search is known to be most difficult in the case of a narrow, gently sloping and curving valley with steep walls. This is exactly the case for most of the now widely-used test functions. Usually, the values of these functions vary widely along some directions (normal to the valley bottom) and weakly along some other directions (along the valley bottom). The contour lines of these functions look like *bananas*, *pears*, *open rings*, and so on. For these functions, the number of bends in a valley is not too great.

Let us refer to the class of nonlinear unimodal functions with curving valley without many bends as class V, and let us consider only one element in this class. The search for an extremum of a function $f(x) \in V$ can be staged as follows.

(i) *Descent into the valley.* At this stage, the value of the function and the gradient norm fall sharply. Numerical experiments lead to the conclusion that, as a rule, the descent takes no more than n steps.

(ii) *Turn along the valley.* Depending on the method employed, this stage is more or less successive *learning* or change of the descent direction into the direction along the valley bottom. In the case of successful *approximation of the valley*, this stage is characterized by large angles made by adjacent directions of minimization.

(iii) *Advance along the bottom of the narrow, gently sloping valley.* This stage involves insignificant decrease of the function. If the ith direction of search runs along the bottom of the narrow, gently sloping valley, then the ith point and the $(i+1)$th point lie either on opposite walls or both on the bottom. If both points lie on the walls, then we can approximately write

$$g_{i+1} \approx -g_i \,. \tag{19}$$

If the ith direction of search is at a large angle with respect to the generating line of the valley (that is, $p_i = -g_i$), then the $(i+1)$th point stays at the valley bottom and

$$\| g_{i+1} \| \ll \| g_i \| . \tag{20}$$

(iv) *Search in the vicinity of the extremum.* As is well known, the function is nearly quadratic at this stage.

To have a physical picture of what the methods do, the behavior of all algorithms was analytically studied at each stage for two-dimensional functions of the class V. However, the qualitative conclusions will be shown to be equally valid for functions of higher dimensionality.

4. Certain Properties of the Steepest-Descent Method

In accordance with (2), the selection of the direction in the SD-method does not depend on former information and depends only on the point where the descent starts.

This nonadaptive approach may prove successful for those points where the nature of the function changes abruptly, so that previous information does not help us in getting closer to the minimum. At the very beginning of the search, a step along the antigradient is also convenient because, without knowledge of the behavior of the function, another selection is risky and can entail considerable departure from the extremum.

Subsequent to the descent stage, for the straight or smoothly curving portions of the valley bottom, the SD-method leads to search trajectories which are sawtooth curves. The progress toward the minimum is very slow. This practically leads to *cycling*.

5. Certain Properties and Modifications of the *APT*-Method

The APT-method (Ref. 2) is intended for searching the minimum of a function having concentric ellipsoidal surfaces of equal level. In the two-dimensional case and for a quadratic function, two steps are taken along the antigradient; then, one step is taken along the direction connecting the last point found and the initial point of the search. Consequently, a descent to the valley bottom is made from two points on the walls, and then a step along the bottom is made. In the case of a quadratic function of two variables, this direction leads to the minimum point.

In the case of a curving valley, three steps do not suffice to find an extremum; therefore, another iteration is needed. Therefore, the search direction along the antigradient leading to the valley should be used and then the last two points on the valley bottom should be connected to obtain a new direction of search.

For valley functions, the APT-method is known to converge poorly (Ref. 2), because the selected directions do not always yield a reduction of the function. Therefore, it is reasonable that, if $p_i^T g_i \geqslant 0$, one resets

$$p_i = -g_i . \tag{21}$$

Also, we have experimentally established that, for unimodal functions of the class V, the method converges much quicker if *restart* is intro-

duced; this implies that, at a certain point, the entire former information is forgotten and the search restarts anew.

Below, we will discuss the APT-method with restart after every $n + 1$ steps and with correction of the direction by (21).

6. Some Properties of the *CG*-Method

By virtue of (6), p_{i+1} (a subsequent direction in the CG-method) can be represented as the sum of two vectors: the vector of the preceding direction p_i and the gradient in the $(i + 1)$th point, both vectors being taken with certain coefficients. Because the coefficient of p_i is greater than zero, p_{i+1} can turn at an angle not exceeding $\pi/2$ relative to p_i.

Besides, the coefficient of p_i depends only on the norms of g_i and g_{i+1} but not on their directions; g_{i+1} is included in (6) with a constant coefficient of 1.

In motion along the bottom of a straight valley, g_{i+1} corrects p_i, so that p_{i+1} is a good approximation to the valley direction. When the valley turns, the coefficient of p_i does not decrease; in other words, there is a trend to maintain the previous direction. This fact indicates that the method is *inertial* and ill-adapted to variations in the nature of the function.

The poor performance of the method was revealed by numerical verification in the case of the Rosenbrock function (Ref. 5). It was suggested (Ref. 5) that the method should be restarted periodically after τ iterations.

To preserve quadratic convergence, τ should be at least to n. On the other hand, τ should be as close to n as possible to improve flexibility. In Ref. 5, the authors selected $\tau = n + 1$. Selection of $\tau = n$ is equally justified.

7. Certain Properties of the Davidon Method

Theorem 7.1. Each subsequent direction in the Davidon method can be represented as a linear combination of two vectors, the antigradient vector at the initial point of the direction and the vector of the preceding search direction.

Proof. Using (7)–(12), one can write

$$H_{i+1}g_{i+1} = (H_i + A_i + B_i)g_{i+1} = H_i g_{i+1} + B_i g_{i+1}. \tag{22}$$

It is evident that

$$A_i g_{i+1} = 0, \tag{23}$$

because

$$\sigma_i{}^T g_{i+1} = 0 \tag{24}$$

in the precise determination of the minimum along the direction. Hence, it also follows that

$$g_i{}^T H_i g_{i+1} = 0. \tag{25}$$

From (22), it follows that

$$\begin{aligned} H_{i+1}g_{i+1} &= H_i g_{i+1} - H_i(g_{i+1}-g_i)(g_{i+1}^T - g_i{}^T)H_i\, g_{i+1}/(g_{i+1}^T - g_i{}^T)H_i\,(g_{i+1}-g_i) \\ &= H_i g_i[g_{i+1}^T H_i g_{i+1}/(g_{i+1}^T H_i g_{i+1} + g_i{}^T H_i g_i)] \\ &\quad + H_i g_{i+1}[g_i{}^T H_i g_i/(g_{i+1}^T H_i g_{i+1} + g_i{}^T H_i g_i)]. \end{aligned} \tag{26}$$

Denote

$$g_i{}^T H_i g_i = l_i\,, \tag{27}$$

$$g_{i+1}^T H_i g_{i+1} = m_i\,. \tag{28}$$

Then,

$$H_{i+1}g_{i+1} = H_i g_i m_i/(l_i + m_i) + H_i g_{i+1} l_i/(l_i + m_i). \tag{29}$$

Considering that $g_{i+1}^T H_i g_i = 0$, one can write

$$H_i g_{i+1} = t_1 g_{i+1} + t_2 H_i g_i\,. \tag{30}$$

Multiplying (30) by g_{i+1}^T gives

$$m_i = t_1 g_{i+1}^T g_{i+1}\,, \tag{31}$$

$$t_1 = m_i/g_{i+1}^T g_{i+1}\,. \tag{32}$$

Premultiplying (30) by $g_i{}^T$ gives

$$g_i{}^T H_i g_{i+1} = t_1 g_i{}^T g_{i+1} + t_2 g_i{}^T H_i g_i\,, \tag{33}$$

$$t_2 = -t_1 g_i{}^T g_{i+1}/g_i{}^T H_i g_i = -(m_i/g_{i+1}^T g_{i+1})(g_i{}^T g_{i+1}/l_i). \tag{34}$$

Consequently,

$$\begin{aligned} H_{i+1}g_{i+1} &= H_i g_i m_i/(m_i + l_i) \\ &\quad + [g_{i+1}m_i/g_{i+1}^T g_{i+1} - H_i g_i(m_i/l_i)(g_{i+1}^T g_i/g_{i+1}^T g_{i+1})]\, l_i/(m_i + l_i) \\ &= H_i g_i\{m_i/(m_i + l_i) - [m_i/(m_i + l_i)](g_{i+1}^T g_i/g_{i+1}^T g_{i+1})\} \\ &\quad + g_{i+1} m_i l_i/(m_i + l_i) g_{i+1}^T g_{i+1}\,, \end{aligned} \tag{35}$$

and then

$$p_{i+1} = [m_i/(m_i + l_i)g_{i+1}^T g_{i+1}][p_i(g_{i+1}^T g_{i+1} - g_{i+1}^T g_i) - g_{i+1} g_i^T H_i g_i]$$
$$= k_{i+1}[p_i(g_{i+1}^T g_{i+1} - g_{i+1}^T g_i) + g_{i+1}(p_i^T g_i)], \tag{36}$$

where

$$k_{i+1} = m_i/(m_i + l_i)\, g_{i+1}^T g_{i+1}, \qquad k_{i+1} > 0. \tag{37}$$

According to Eqs (7)–(12), the matrix H_i includes all the preceding matrices H_j, $j = 0, 1, \ldots, i - 1$, or allows for all the preceding values of the gradient.

The coefficient of the vector p_i depends on only two values of the gradient (g_i and g_{i+1}) and can be negative if

$$g_{i+1}^T g_i > 0. \tag{38}$$

Let us consider the variations of the coefficients of the vectors p_i and g_{i+1} in minimizing a valley function. The coefficient of p_i can change sign depending on the vectors g_i and g_{i+1} and the angle between them. In the descent to the valley or with an abrupt change of the valley direction, the condition (38) holds. According to (36), this can lead to an abrupt change of the subsequent search direction. This fact gives the method *flexibility* and its ability to respond quickly to changes in the function shape.

At the third stage of the search, in the narrow valley, the coefficient of p_i is positive and nearly maximal if (19) is true. Since

$$p_{i+1}^T p_i = k_{i+1} p_i^T p_i (g_{i+1}^T g_{i+1} - g_{i+1}^T g_i),$$

then $p_{i+1}^T p_i > 0$; consequently, neighboring directions of minimization make an acute angle.

The coefficient of the vector g_{i+1} is always positive and depends only on the angle between the vectors $H_i g_i$ and g_i. If the ith direction of minimization does not deviate greatly from the antigradient (which is possible in the descent to the valley or in a change of the valley direction), then this coefficient is maximal. This fact greatly affects the subsequent direction of minimization by making it *turn* along the valley bottom. In moving along the straight valley, this coefficient is small, because the gradient is nearly normal to the bottom of the narrow valley.

8. Approach to the Comparison of Methods

It is clearly impossible to devise a searching method which is optimal for the entire class of unimodal functions. For each method, a

function can be found for which it converges well, and yet another function can be found for which the method does not converge. For each particular problem, an algorithm can be originated so that it takes into account all of its properties and converge faster than most effective methods. Therefore, in the experimental approach, the methods are compared through several typical functions. The specific features of these functions, such as the availability of narrow, gently sloping curving valleys, were allowed for in the test functions chosen by Rosenbrock, Powell, Fletcher, and others skilled in the art of searching. Traditionally, the legitimacy of each search method is verified through these test functions.

As already noted, the estimation of the value of such methods is relative because, to a considerable degree, it depends on the choice of the nominal point initially selected. For a specific method and test function, the initial point can be *lucky* or *unlucky*.

We now suggest another approach to compare the algorithms. Let the minimization directions from point A be defined for two methods of nonlinear programming (Fig. 1). The direction AD with its minimum at the point D is given in the first method. The point C which is on the straight line AD is defined in the second method through two successive steps AB and BC.

If $f(D) \leqslant f(C)$, then we may think that the first method gives a better advance toward the extremum in the neighborhood of point A than the second method does.

Indeed, after the minimization in one direction, the first method guarantees a function value at least as good as that given by the second method after the minimization along two directions. A unimodal function can have several extremums along the segments AB, BC, AD. However, the procedure of search along a direction is developed so as to reach the nearest of them. Thus, an explicit or implicit assumption is made that, in the neighborhood of each point in a chosen direction, the function under study is strictly unimodal. But it is exactly when this assumption is not valid, that *stumbling blocks* for all of the methods under consideration appear.

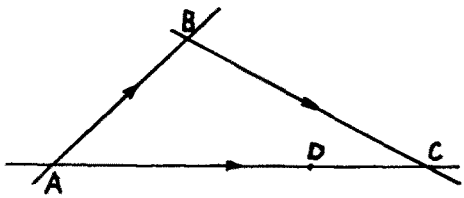

Fig. 1. Choice of the minimization directions at point A.

If the above assumption is true, then the nearest minimum is along the directions AB, BC, and AD, and the function cannot have more than one minimum along the segment AC.

This way of comparing nonlinear programming methods may be termed *local*, rather than *global*. As will be shown later, the conditions under which one method is superior to another depend on the search stage. Therefore, the proposed way of comparison reveals the *local superiority* of one method over another, which is true only for a specified stage of the search.

9. Comparison of SD-Method and APT-Method

The first two steps of these methods coincide. Indeed,

$$p_0^{\mathrm{SD}} = p_0^{\mathrm{APT}} = -g_0\,, \qquad p_1^{\mathrm{SD}} = p_1^{\mathrm{APT}} = -g_1\,.$$

Therefore, if both algorithms begin the descent from the same point, they do it identically. Beginning with the third step, the methods choose directions in different ways. Since the APT-method works with restart, we may think of SD and APT as equivalent at the stage of turn. Now, let us proceed to the third search stage, the movement along the narrow, gently sloping valley. Let us see when one method is better than another. Comparing the methods by the above approach, we define the conditions under which the second APT direction can pass through the final point of the second and third SD-steps.

Let

$$p_2^{\mathrm{APT}} = \gamma(p_2^{\mathrm{SD}} + p_3^{\mathrm{SD}}), \tag{39}$$

where

$$p_2^{\mathrm{APT}} = -(\alpha_0 g_0 + \alpha_1 g_1)\alpha_2^{\mathrm{APT}}, \tag{40}$$

$$p_2^{\mathrm{SD}} = -\alpha_2^{\mathrm{SD}} g_2^{\mathrm{SD}}, \tag{41}$$

$$p_3^{\mathrm{SD}} = -\alpha_3^{\mathrm{SD}} g_3^{\mathrm{SD}}. \tag{42}$$

Here, γ is a proportionality coefficient, $\gamma > 0$.

Therefore,

$$p_2^{\mathrm{APT}} = \gamma(-\alpha_2^{\mathrm{SD}} g_2^{\mathrm{SD}} - \alpha_3^{\mathrm{SD}} g_3^{\mathrm{SD}}), \tag{43}$$

that is, point x_4 determined by the SD-method lies on the straight line p_2^{APT}, in addition to points x_0 and x_2. Since these three points are obtained when descending into the valley along the antigradient and are on the

same straight line, we are at the straight section of the valley. Thus, the APT-method chooses the direction along the valley bottom and is superior to the SD-method when advancing along straight sections of the valley bottom.

Let us elucidate the conditions under which SD is locally better than APT. Let the second SD-direction pass through the final point of the second and third APT-steps. Then,

$$p_2^{\mathrm{APT}} + p_3^{\mathrm{APT}} = \gamma p_2^{\mathrm{SD}}, \tag{44}$$

where

$$p_3^{\mathrm{APT}} = -\alpha_3^{\mathrm{APT}} g_3^{\mathrm{APT}}. \tag{45}$$

Therefore,

$$p_2^{\mathrm{APT}} = \alpha_3^{\mathrm{APT}} g_3^{\mathrm{APT}} - \gamma \alpha_2^{\mathrm{SD}} g_2^{\mathrm{SD}}, \tag{46}$$

that is, the gradients at points x_2 and x_3 form an acute angle, which can happen with a sharp turn of the valley.

Thus, along the straight sections and with gently sloping valley turns, APT advances to the extremum faster than SD. But, in a sharp turn, SD may happen to be superior to APT. Since we consider a valley with few turns, this situation cannot arise too often. Besides, the introduction of a guaranteed relaxation and restart after the $(n+1)$th step according to (21) for APT improves the method considerably at the valley turns, because the importance of the gradient in the selection of a direction increases. In the neighborhood of the extremum, APT converges in $2n-1$ steps and SD converges at the rate of geometric progression.

Consequently, in dealing with functions belonging to the class V, the APT-method is on the whole more effective than the SD-method: at the first and second stages, these methods are equivalent; and, at the highly important stage of advancement along the valley bottom, APT is superior to SD.

10. Comparison of APT-Method and CG-Method

The first direction is chosen identically by both methods. They are considered to be equivalent when descending. Unlike the CG-method, the second APT-direction always forms the angle $\pi/2$ with the first direction. Nevertheless, the methods are roughly the same when turning, because they both use a restart.

Now, let us proceed to the third search stage. Let us consider the conditions under which CG can be locally superior to APT when advancing along the valley. Let the first direction of CG pass through the final point of the first and second APT-steps.

Because

$$p_0^{CG} = p_0^{APT} = -g_0\,, \tag{47}$$

$$p_1^{APT} = -g_1\,, \tag{48}$$

$$p_1^{CG} = -(g_1{}^T g_1/g_0{}^T g_0)g_0 - g_1\,, \tag{49}$$

$$p_2^{APT} = -\alpha_0 g_0 - \alpha_1^{APT} g_1\,, \tag{50}$$

$$\alpha_1^{APT} p_1^{APT} + \alpha_2^{APT} p_2^{APT} = \gamma p_1^{CG}, \tag{51}$$

then

$$-\alpha_1^{APT} g_1 - \alpha_2^{APT}(\alpha_0 g_0 + \alpha_1^{APT} g_1) = -\gamma\alpha_1^{CG}[(g_1{}^T g_1/g_0{}^T g_0)g_0 + g_1]; \tag{52}$$

and because

$$g_0{}^T g_1 = 0, \tag{53}$$

then

$$\alpha_1^{APT} + \alpha_2^{APT}\alpha_1^{APT} = \gamma\alpha_1^{CG}, \tag{54}$$

$$\alpha_0\alpha_2^{APT} = \gamma\alpha_1^{CG}(g_1{}^T g_1/g_0{}^T g_0). \tag{55}$$

Consequently,

$$\alpha_0\alpha_2^{APT}(g_0{}^T g_0/g_1{}^T g_1) = \gamma\alpha_1^{CG}, \tag{56}$$

$$\alpha_1^{APT} + \alpha_2^{APT}\alpha_1^{APT} = \alpha_0\alpha_2^{APT}(g_0{}^T g_0/g_1{}^T g_1). \tag{57}$$

From new on, the index APT can be omitted.

By virtue of (20), we have

$$\|g_1\| \ll \|g_0\|. \tag{58}$$

Besides, it is natural to suppose that the advance in descending is smaller than the advance along the straight valley section, that is,

$$\alpha_0\|g_0\| < \alpha_1\|g_1\|. \tag{59}$$

From (57), it follows that

$$\alpha_1(1 + \alpha_2) = \alpha_0\alpha_2\|g_0\|^2/\|g_1\|^2, \tag{60}$$

$$(1 + \alpha_2)/\alpha_2 = (\alpha_0/\alpha_1)\|g_0\|^2/\|g_1\|^2. \tag{61}$$

From (58), Ineq. (59) may change sign, and we obtain

$$\alpha_0 \| g_0 \|^2 > \alpha_1 \| g_1 \|^2,$$

which does not contradict (61). Thus, (61) may hold when searching along the straight valley sections, that is, CG is superior to APT when searching in the valley.

Then, it is necessary to ascertain whether APT is locally superior to CG. Let the first APT direction pass through the final point of the first and second steps of CG. Then,

$$\gamma \alpha_1^{\mathrm{APT}} p_1^{\mathrm{APT}} = \alpha_1^{\mathrm{CG}} p_1^{\mathrm{CG}} + \alpha_2^{\mathrm{CG}} p_2^{\mathrm{CG}}; \tag{62}$$

and, multiplying both parts by $p_0 = -g_0$, we obtain

$$\alpha_1^{\mathrm{CG}} p_1^{T\mathrm{CG}} p_0 + \alpha_2^{\mathrm{CG}} p_2^{T\mathrm{CG}} p_0 = 0. \tag{63}$$

Such a relation of three successive directions of minimization can take place when the valley turns abruptly. In this case, it must be remembered that the methods coincide at the null step (the step along the antigradient). CG always makes the first step from the quadratic approximation of the given valley section. The first step along the antigradient in the APT-method can give better results than CG only when the valley direction changes sharply and CG deviates from this direction considerably. If the valley direction does not change sufficiently often, which is the case for functions of class V, then CG is superior to APT at the stage of advancing along the valley bottom.

At the last stage, the performance of both methods is almost equal. Hence, CG is generally more effective than APT for functions of class V, since CG is superior to APT at the important stage of advancing along the valley bottom.

11. Comparison of CG-Method with D-Method

Theorem 11.1. The processes of search for an extremum by the D-method and the CG-method when moving from the same initial point coincide along the first two directions for any nonlinear function.[3]

Under (6)–(12), the first directions of motion defined by both methods coincide, specifically,

$$p_0 = -g_0 . \tag{64}$$

[3] As shown in Ref. 9, both methods coincide completely for a quadratic function; starting from the same point, they choose the same directions.

Therefore, after the minimization along the first direction, the same vector is used to find the second direction. According to (7)–(12), we have

$$-p_1^{\mathrm{D}} = H_1 g_1 = (H_0 + A_0 + B_0)g_1 = g_1 + B_0 g_1$$
$$= (g_0^T g_0 g_1 + g_1^T g_1 g_0)/(g_0^T g_0 + g_1^T g_1); \tag{65}$$

and, according to (6), we have

$$-p_1^{\mathrm{CG}} = (g_1^T g_1/g_0^T g_0)g_0 + g_1 = (g_0^T g_0 g_1 + g_1^T g_1 g_0)/g_0^T g_0 . \tag{66}$$

Hence, $p_1^{\mathrm{D}} = K p_1^{\mathrm{CG}}$, where K is a positive number.

The subsequent directions of motion defined by D and CG may prove to be different. Let us consider the conditions under which the periodic *restarting* used in the CG-method allows one to choose luckier directions than those defined by the D-method. Let the jth directions start from the same point in both methods, and let the jth direction for CG coincide with the antigradient. CG has an advantage when its jth direction passes through the final point of the $(j+1)$th direction defined by D. We have

$$\gamma g_j = \alpha_j H_j g_j + \alpha_{j+1} H_{j+1} g_{j+1} . \tag{67}$$

Multiplying (67) by σ_{j-1}^T, we obtain

$$\sigma_{j-1}^T \sigma_j + \sigma_{j-1}^T \sigma_{j+1} = 0. \tag{68}$$

From (68), it follows that two out of three subsequent minimization directions p_{j-1}, p_j, p_{j+1} form an obtuse angle. Then, we can draw the conclusion that the choice of the antigradient direction may be better if the search direction changes greatly when D is used.

We define the conditions under which the D-method provides selection of better directions than the CG-method. Let the jth directions start from the same point in both methods. D has an advantage when the jth direction defined by this method passes through the final point of the $(j+1)$th direction defined by CG. We have

$$\gamma H_j g_j = \alpha_j g_j + \alpha_{j+1}[(g_{j+1}^T g_{j+1}/g_j^T g_j)g_j + g_{j+1}],$$

that is,

$$\gamma H_j g_j = [(\alpha_j g_j^T g_j + \alpha_{j+1} g_{j+1}^T g_{j+1})/g_j^T g_j]g_j + \alpha_{j+1} g_{j+1} . \tag{69}$$

Since $p_{j-1}^T g_j = 0$ and $g_j^T g_{j+1} = 0$, we can write

$$g_{j+1} = \lambda p_{j-1} , \tag{70}$$

where λ, $\lambda \neq 0$, is a constant. Substituting Eqs. (36) and (70) into

Eq. (69), we obtain

$$\gamma k_j[p_{j-1}(g_j^T g_j - g_{j-1}^T g_j) - (g_{j-1}^T H_{j-1} g_{j-1}) g_j] = -\lambda\alpha_{j+1} p_{j-1} - [(\alpha_j g_j^T g_j + \alpha_{j+1} g_{j+1}^T g_{j+1})/g_j^T g_j] g_j . \tag{71}$$

On the right-hand and left-hand sides of Eq. (71), there are the same vectors g_j and p_{j-1}. The coefficients of these vectors on the right-hand side of Eq. (71) are unknown and cannot be compared exactly with those on the left-hand side.

However, consider the conditions under which Eq. (71) holds. The vector g_j with negative coefficients enters both sides of the equation. Moving along the narrow valley with condition (19), the coefficient of p_{j-1} on the left-hand side is greater than zero. At the turn of the valley, it can be negative. The sign of the coefficient of p_{j-1} on the right-hand side depends on the sign of λ. Let us consider the succession of vectors p_{j-1}, $-g_j$, $-g_{j+1}$. If $-p_{j-1}^T g_{j+1} < 0$, which is characteristic of the valley turn, then $\lambda > 0$, and the coefficient of p_{j-1} is negative. Along the straight sections,

$$-p_{j-1}^T g_{j+1} > 0, \qquad \lambda < 0,$$

and the coefficient of p_{j-1} is positive.

Hence, on both sides of Eq. (71), the coefficients of the same vectors have the same signs when searching in the narrow valley. The squared lengths of vectors on the left are related by

$$g_j^T g_j (g_{j-1}^T H_{j-1} g_{j-1})^2 / p_{j-1}^T p_{j-1} (g_j^T g_j - g_{j-1}^T g_j)^2 \cong g_j^T g_j (g_{j-1}^T p_{j-1})^2 / 4 p_{j-1}^T p_{j-1} (g_j^T g_j)^2 = (1/4)(g_{j-1}^T p_{j-1} / \| g_j \| \cdot \| p_{j-1} \|)^2. \tag{72}$$

When moving in the valley, this value is small, because of the assumption that the vector p_{j-1} is directed along the ravine, and g_{j-1} is perpendicular to it.

On this assumption, the advance in the descent is much smaller than the motion along the straight ravine, i.e.,

$$\alpha_j \| g_j \| \ll \alpha_{j+1} \| (g_{j+1}^T g_{j+1} / g_j^T g_j) g_j + g_{j+1} \|. \tag{73}$$

Thus,

$$\alpha_j \sqrt{(g_j^T g_j)} \ll \alpha_{j+1} \sqrt{[(g_{j+1}^T g_{j+1})^2 / g_j^T g_j + g_{j+1}^T g_{j+1}]}. \tag{74}$$

But

$$\alpha_{j+1} \sqrt{[(g_{j+1}^T g_{j+1})^2 / g_j^T g_j + g_{j+1}^T g_{j+1}]} = \alpha_{j+1} \sqrt{(g_{j+1}^T g_{j+1})} \sqrt{(g_{j+1}^T g_{j+1} / g_j^T g_j + 1)} \cong \alpha_{j+1} \sqrt{(g_{j+1}^T g_{j+1})}, \tag{75}$$

because of (20). Consequently,

$$\alpha_j \sqrt{(g_j{}^T g_j)} \ll \alpha_{j+1} \sqrt{(g_{j+1}^T g_{j+1})}, \tag{76}$$

despite the fact that (20) holds.

Let

$$\alpha_j/\alpha_{j+1} = o\,(\sqrt{(g_{j+1}^T g_{j+1})}/\sqrt{(g_j{}^T g_j)}). \tag{77}$$

Let us consider the squared lengths of the vectors on the right-hand side of Eq. (71). Their ratio is

$$\begin{aligned} &g_j{}^T g_j[(\alpha_j g_j{}^T g_j + \alpha_{j+1} g_{j+1}^T g_{j+1})/g_j{}^T g_j]^2/(\alpha_{j+1}^2 g_{j+1}^T g_{j+1}) \\ &\quad = (g_j{}^T g_j/g_{j+1}^T g_{j+1})(\alpha_j/\alpha_{j+1} + g_{j+1}^T g_{j+1}/g_j{}^T g_j)^2 \\ &\quad = (g_j{}^T g_j/g_{j+1}^T g_{j+1})(g_{j+1}^T g_{j+1}/g_j{}^T g_j)[(\alpha_j/\alpha_{j+1})\sqrt{(g_j{}^T g_j/g_{j+1}^T g_{j+1})} \\ &\qquad + \sqrt{(g_{j+1}^T g_{j+1}/g_j{}^T g_j)}]^2 \approx g_{j+1}^T g_{j+1}/g_j{}^T g_j\,. \end{aligned}$$

But $\| g_{j+1} \|/\| g_j \|$ is a small value. Therefore, in the advance along the narrow valley, Eq. (71) holds.

The results obtained give one a qualitative comparison of the methods under study.

(i) By virtue of Theorem 7.1, at the first search stage both methods coincide.

(ii) At the second search stage, CG can prove to be superior.

(iii) At the third search stage, D possesses an unquestionable advantage. This method is fit for the advance along the valley bottom providing that the valley bottom generating line is not very tortuous.

(iv) At the fourth search stage, both methods are of equal worth, since they are quadratically convergent.

12. Variable Metric Algorithms

In Ref. 8, a group of methods is considered that depends on certain parameters; for special values of those parameters, the Davidon, Pearson, McCormick, and other algorithms are obtained. The group belongs to a broader class of *variable metrics algorithms* (Ref. 10).

We show that, when $n = 2$, Algorithms I–VII of Ref. 8 choose the directions identically and they coincide in the case of precise minimization along the search direction.

Let us consider a class of algorithms of the form (13)–(18). Transform the expression (13) as follows:

$$-p_i = H_i^T g_i = H_{i-1}^T g_i + \Delta H_{i-1}^T g_i = H_{i-1}^T g_i + \rho y_{i-1} \Delta x_{i-1}^T g_i / y_{i-1}^T \Delta g_{i-1}$$
$$- (k_1 \Delta x_{i-1} + k_2 H_{i-1}^T \Delta g_{i-1}) \Delta g_{i-1}^T H_{i-1}^T g_i / z_{i-1}^T \Delta g_{i-1} . \tag{78}$$

Since $\Delta x_{i-1}^T g_i = 0$, then

$$\rho y_{i-1} \Delta x_{i-1}^T g_i / y_{i-1}^T \Delta g_{i-1} = 0$$

and

$$-p_i = H_{i-1}^T g_i - (\Delta g_{i-1}^T H_{i-1}^T g_i / z_{i-1}^T \Delta g_{i-1})(k_1 \Delta x_{i-1} + k_2 H_{i-1}^T g_i - k_2 H_{i-1}^T g_{i-1})$$
$$= (1 - k_2 \Delta g_{i-1}^T H_{i-1}^T g_i / z_{i-1}^T \Delta g_{i-1}) H_{i-1}^T g_i - (\Delta g_{i-1}^T H_{i-1}^T g_i / z_{i-1}^T \Delta g_{i-1})$$
$$\times (k_1 \alpha_{i-1} p_{i-1} + k_2 p_{i-1})$$
$$= (1/z_{i-1}^T \Delta g_{i-1})[(z_{i-1}^T \Delta g_{i-1} - k_2 \Delta g_{i-1}^T H_{i-1}^T g_i) H_{i-1}^T g_i$$
$$- (k_1 \alpha_{i-1} + k_2) \Delta g_{i-1}^T H_{i-1}^T g_i p_{i-1}]. \tag{79}$$

Also, we can write

$$H_{i-1}^T g_i = a g_i + b p_{i-1} , \tag{80}$$

where a and b are constants. Let us find a and b. We have

$$g_i^T H_{i-1}^T g_i = a g_i^T g_i , \qquad a = g_i^T H_{i-1}^T g_i / g_i^T g_i ;$$

and

$$g_{i-1}^T H_{i-1}^T g_i = a g_{i-1}^T g_i + b g_{i-1}^T p_{i-1} ,$$
$$b = (g_{i-1}^T H_{i-1}^T g_i \, g_i^T g_i - g_i^T H_{i-1}^T g_i \, g_{i-1}^T g_i)/(g_i^T g_i)(g_{i-1}^T p_{i-1}).$$

Let $B = 1/z_{i-1}^T \Delta g_{i-1}$. Substituting (80) into (79), we have

$$-p_i = B\{(z_{i-1}^T \Delta g_{i-1} - k_2 \Delta g_{i-1}^T H_{i-1}^T g_i)(g_i^T H_{i-1}^T g_i / g_i^T g_i) g_i$$
$$+ (z_{i-1}^T \Delta g_{i-1} - k_2 \Delta g_{i-1}^T H_{i-1}^T g_i)$$
$$\times [(g_{i-1}^T H_{i-1}^T g_i \, g_i^T g_i - g_i^T H_{i-1}^T g_i \, g_{i-1}^T g_i)/(g_i^T g_i)(g_{i-1}^T p_{i-1})] p_{i-1}$$
$$- \Delta g_{i-1}^T H_{i-1}^T g_i (k_1 \alpha_{i-1} + k_2) p_{i-1}\}. \tag{81}$$

Consider the coefficient of the vector g_i. We have

$$(k_1 \Delta x_{i-1}^T \Delta g_{i-1} + k_2 \Delta g_{i-1}^T H_{i-1} \Delta g_{i-1} - k_2 \Delta g_{i-1}^T H_{i-1}^T g_i)(g_i^T H_{i-1}^T g_i / g_i^T g_i)$$
$$= -(p_{i-1}^T g_{i-1})(k_1 \alpha_{i-1} + k_2)(g_i^T H_{i-1}^T g_i / g_i^T g_i). \tag{82}$$

Transform the coefficient of the vector p_{i-1} as follows:

$$-(p_{i-1}^T g_{i-1})(k_1\alpha_{i-1} + k_2)(g_{i-1}^T H_{i-1}^T g_i\, g_i^T g_i - g_i^T H_{i-1}^T g_i\, g_{i-1}^T g_i)/(g_i^T g_i)(p_{i-1}^T g_{i-1})$$
$$- (k_1\alpha_{i-1} + k_2)(g_i^T H_{i-1}^T g_i - g_{i-1}^T H_{i-1}^T g_i)$$
$$= -[(k_1\alpha_{i-1} + k_2)/g_i^T g_i]\, g_i^T H_{i-1}^T g_i (g_i^T g_i - g_{i-1}^T g_i). \tag{83}$$

Substituting (82) and (83) into (81), we have

$$p_i = B(g_i^T H_{i-1}^T g_i / g_i^T g_i)(k_1\alpha_{i-1} + k_2)[(g_i^T g_i - g_{i-1}^T g_i)\, p_{i-1} + p_{i-1}^T g_{i-1} g_i]. \tag{84}$$

Let the choice of x_0 and H_0 be independent of the parameter set. Then, the direction p_0 is the same for all algorithms. Therefore, x_1 and g_1 are the same, and the vector p_1 is the same, except for a multiplying factor. Consequently, x_2 and g_2 are the same, and so on.

Thus, all algorithms of the type (13)–(18), regardless of the values of the parameters, give the same succession of points at $n = 2$. Since D is a particular case of such an algorithm, then everything that has been said about its properties is characteristic of all the algorithms of the class (13)–(18).

The results of the comparison stated above have a qualitative local character, and the analysis is given for two-dimensional functions. In this connection, it is interesting to compare the conclusions made with the results of an experimental comparison of the methods for well-known test functions.

In Figs. 2–8, the results of an experimental comparison of the methods are shown. The number of iterations is set along the abscissa, and the logarithm of the function value is set along the ordinate: the solid line characterizes the D-method, the dashed line refers to the CG-method, the dotted line refers to the APT-method, and the solid-dotted line refers to the SD-method. Figures 2–5 show the results of the comparison for the two-dimensional Box function (Ref. 3)

$$f(x_1, x_2) = \sum_{\nu} [\exp(-x_1\nu) - \exp(-x_2\nu) - \exp(-\nu) + \exp(-10\nu)]^2,$$

where the summation is over the values $\nu = 0.1, 0.2, \ldots, 0.9, 1.0$. The following initial points are employed: (0, 0), (0, 20), (5, 0), (2.5, 10).

Figure 6 shows the results of the comparison for the Rosenbrock function (Ref. 7), a parabolic valley,

$$f(x_1, x_2) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2.$$

The initial point is (1.2, 1.0).

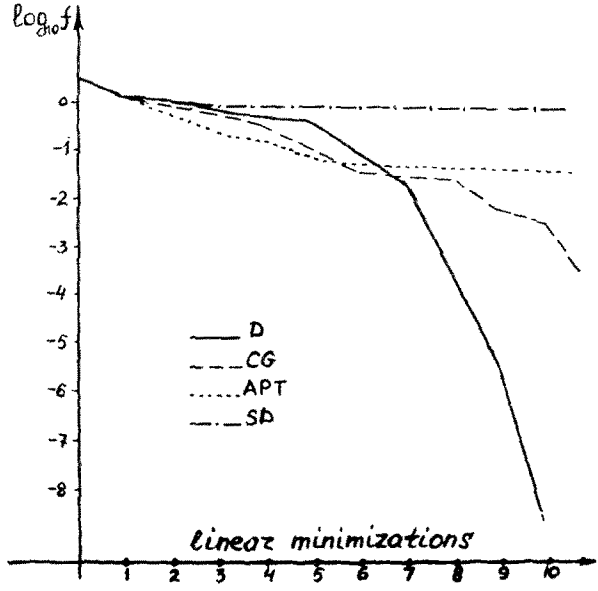

Fig. 2. Box's function, $x_0 = (0, 0)$.

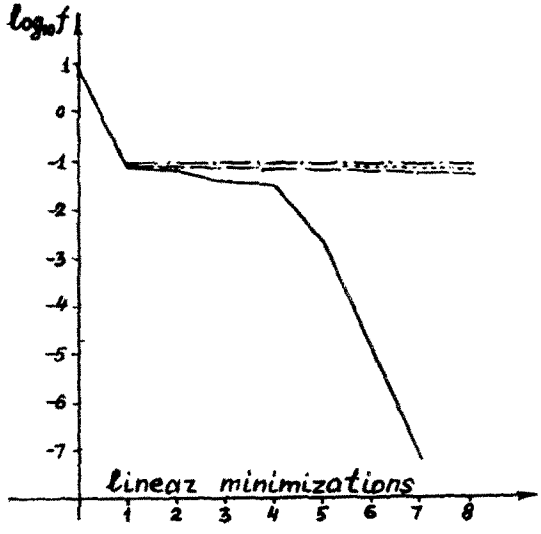

Fig. 3. Box's function, $x_0 = (0, 20)$.

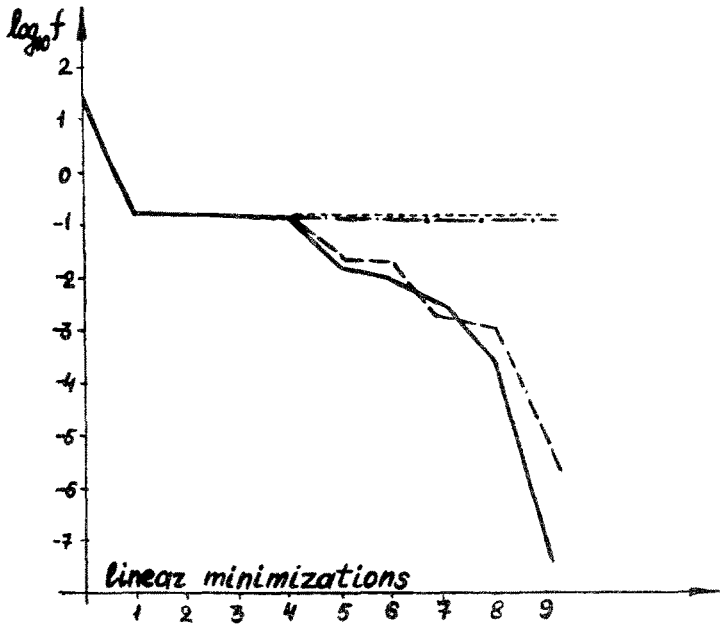

Fig. 4. Box's function, $x_0 = (5, 0)$.

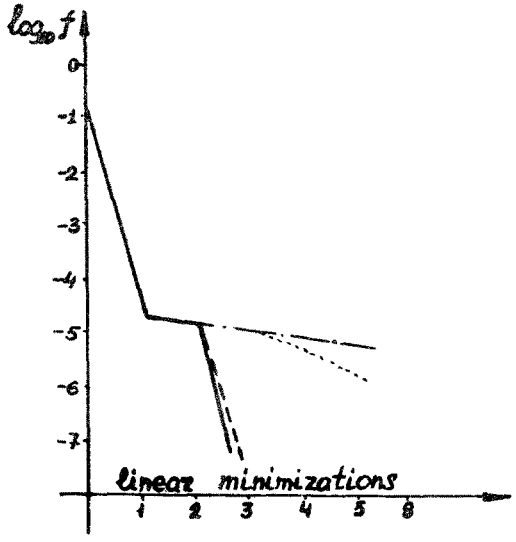

Fig. 5. Box's function, $x_0 = (2.5, 10)$.

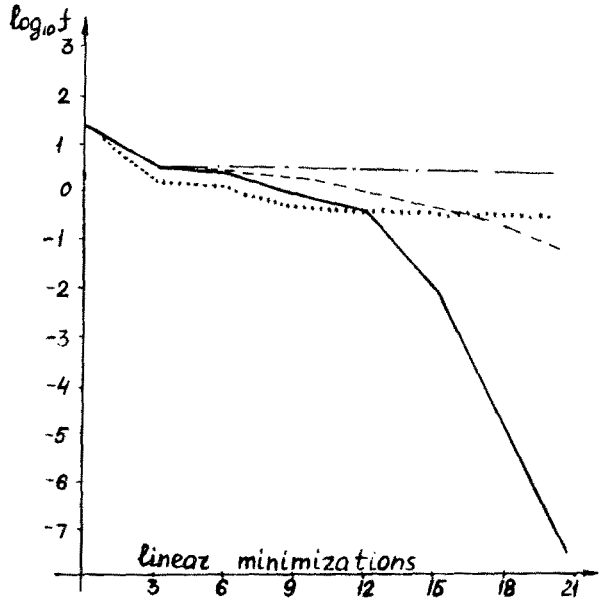

Fig. 6. Parabolic valley.

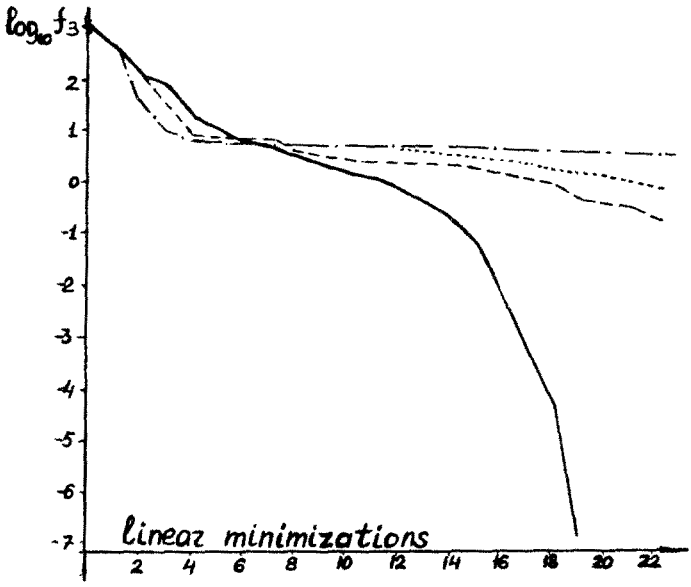

Fig. 7. Helical valley.

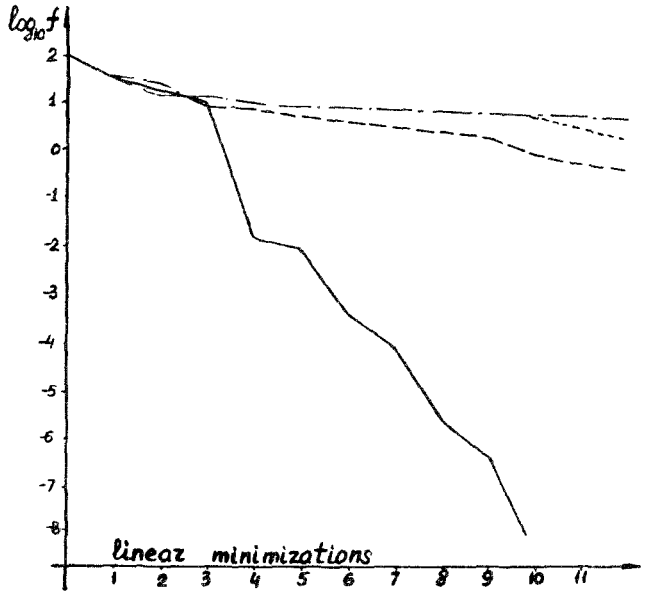

Fig. 8. Powell's function.

Figure 7 gives the results of the comparison for the Fletcher and Powell test function (Ref. 7), a helical valley,

$$f(x_1, x_2, x_3) = 100\{[x_3 - 100(x_1x_2)]^2 + [(x_1^2 + x_2^2)^{0.5} - 1]^2\} + x_3^2,$$

where

$$2\pi\theta = \begin{cases}\arctan(x_2/x_1), & x_1 > 0,\\ \pi + \arctan(x_2/x_1), & x_1 < 0,\end{cases}$$

$$-\pi/2 < 2\pi\theta < 3\pi/2, \qquad -2.5 < x_3 < 7.5.$$

The initial point is $(-1, 0, 0)$.

Figure 8 plots the results of the comparison for the Powell test function (Ref. 11)

$$f(x_1, x_2, x_3, x_4) = (x_1 + 10x_2)^2 + 5(x_3 - x_4)^2 + (x_2 - 2x_3)^4 + 10(x_1 - x_4)^4.$$

The initial point is $(3, -1, 0, 1)$.

13. Results

(i) The qualitative results of the analytical comparison of different methods are confirmed by the numerical compariosn for a number of test functions.

(ii) The figures show that, at the first stage of the search, APT and SD coincide, CG coincides with D and is better than APT.

(iii) Then, the *turn* stage follows, where SD, CG, and APT can be better than D. APT can be better than CG. For example, for the Box function [initial point (0, 0), see Fig. 2], CG and APT provide a smaller function value than D at the 2nd through 6th steps.

(iv) Then, the stage of moving in the valley follows; here, the advantages of D over CG, of CG over APT, and of APT over SD are unquestionable. But exceptions are also possible. Thus, Fig. 6 shows that, at the beginning of the search, APT succeeded in approximating the valley and outstripped CG. But its *turn* was worse; and, starting from the 16th step, CG does better than APT.

(v) Since it advances in the valley successfully, D is the first method to arrive in the neighborhood of the extremum, where the function diminishes abruptly.

(vi) At each stage of search, the methods can be ranked as in Table 1.

(vii) CG is better than D only at one stage, the *turn*. This superiority can be attributed to the property of restart. This fact allows one to suggest a new method of search possessing the best properties of both algorithms.

The D-method forms the core of this new method and is modified as follows: after the descent into the valley (that is, after n iterations), one resets $H_{n+1} = H_0$. This proposed method is found to be more effective than the CG-method and the D-method.

The results obtained also make it possible to understand why, for the same function, the performances of the methods can change, depending on the initial point of search. Indeed, the choice of the initial

Table 1. Algorithm rank table.

Stage of search	Descent	Turn	Advancement in the valley	Moving in the neighborhood of the extremum
SD-method	1	1	4	3
APT-method	2	2	3	2
CG-method	2	1	2	1
D-method	3	3	1	1

point defines the duration of the search stage. Because the performances of the methods differ at the different stages of search, it is possible to choose *lucky* points so that diverse methods would prove to be equally efficient.

References

1. POLYAK, B. T., *Minimization Methods for Functions of Several Variables (in Russian)*, Economics and Mathematical Methods, Vol. 3, No. 6, 1967.
2. WILDE, D. J., *Optimum Seeking Methods*, Prentice-Hall, Englewood Cliffs, New Jersey, 1964.
3. BOX, M. J., *A Comparison of Several Current Optimization Methods, and the Use of Transformations in Constrained Problems*, Computer Journal, Vol. 9, No. 1, 1966.
4. FLETCHER, R., *Function Minimization without Evaluating Derivatives, a Review*, Computer Journal, Vol. 8, No. 1, 1965.
5. FLETCHER, R., and REEVES, C. M., *Function Minimization by Conjugate Gradients.* Computer Journal, Vol. 7, No. 12, 1964.
6. FIACCO, A. V., and MCCORMICK, G. R., *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley and Sons, New York, New York, 1968.
7. FLETCHER, R., and POWELL, M. J. D., *A Rapidly Convergent Descent Method for Minimization*, Computer Journal, Vol. 6, No. 2, 1963.
8. HUANG, H. Y., *Unified Approach to Quadratically Convergent Algorithms for Function Minimization*, Journal of Optimization Theory and Applications, Vol. 5, No. 6, 1970.
9. MYERS, G. E., *Properties of the Conjugate-Gradient and Davidon Methods*, Journal of Optimization Theory and Applications, Vol. 2, No. 4, 1968.
10. ADACHI, N., *On Variable-Metric Algorithms*, Journal of Optimization Theory and Applications, Vol. 7, No. 6, 1971.
11. POWELL, M. J. D., *An Iterative Method for Finding Stationary Values of a Function of Several Variables*, Computer Journal, Vol. 5, No. 2, 1962.