V. DUMITRU
Department of Economic Cybernetics
Academy of Economic Studies

Some Results in Nonlinear Programming Problems Solving*

The paper is a synthesis of the results obtained by the author during 1970—1975 in solving nonlinear programming problems. Together with some results of a theoretical character, some conclusions linked with the efficiency of algorithms for optimizing the functions of variables and, in this connection, with the efficiency of solving non linear problems with constraints by the SUMT technique are given. Special attention is granted to the mixed type algorithms.

Nonlinear mathematical programming is an extremely topical field involving many researchers. Considerable efforts are being made and new articles and books containing results on some or other of the multiple aspects of the relevant problems are constantly published. An important place is held by the nonlinear programming problems solving on the computer with its specific aspects: the time necessary for preparing the problem, the computing machine time necessary for solving the problem, the accuracy and reliability in solving a problem, a.s.o. The interest for finding an answer to these problems is great and the amount of articles and books published in this field has constantly been increasing. Among the works of wider range, we shall mention: [11], [16], [24], [27], [35], [45], [49], [54], [65], [67], [79], [83], [86],[87],[114], [134], [142].

In applied mathematics in general and particularly in mathematics of problems solving on the computer is of very great importance for the researcher of gaining experience by numerical tests on the computer.

Because of the extremely large variety of nonlinear problems we are still far from the stage of "automatedly" solving any problem. The actual solving

*) A paper submitted to the Nineth International Symposium of Mathematical Programming, Budapest, Hungary, August 23—27, 1976.

of a problem depends on the computing technique, on the computation programs and still, to a greater extent, on the "art" of solver.

The author of this paper was directly confronted with this problem when asked to solve some nonlinear optimization problems necessary for carrying out scientific research contracts.

In this field, which I will call "computer nonlinear programming", one of the investigated problems, whose importance has been outlined also by Ph. Wolfe [54], is that of a comparative systematic and unitary study of the behaviour of theirs the most important known algorithms, for efficiency in problems solving. For this research we have selected the algorithms for finding the extremum points for functions of n variables, taking into account that the sequential optimization technique (SUMT) is used in solving a large class of problems of nonlinear mathematical programming. Our investigation covers 26 algorithms, programmed and studied with regard to their behaviour in solving 10 tests problems. The results are given in [39], [40], [41], [42] and especially [45].

A. During the efficiency studies some results of a theoretical character have been obtained; some algorithms have been modified or new procedure has been given which, in a way or another, has improved their behaviour in solving the problems.

An important place in our study is held by the algorithms for optimizing the functions of n variables which do not use the information provided by the gradient estimation. There are sufficient reasons for tackling the subject. The valuation of the differential, if possible, may be expensive both for man and machine, as far as time is concerned. Moreover, the class having the greatest complexity and the most difficult nonlinear programming problems is considered to be the one in which the functions defining the problem are not easy to infer and most often their values are given by tables by simulation or by recurring formulas.

During these studies, particular attention has been granted to Spendley-Hest-Himsworth [132], succesive simplexes algorithm for optimizing a nonlinear function of n variables, later on developed by Nedler and Mead, for which various modifications are suggested with a view to increasing its convergence speed or stability. Our interest for this algorithm has on the one hand been generated by the remarkable characteristics of the set of points making up a simplex and, on the other hand, by the fact that the simplex algorithm is believed to have more chances, in its own class, of bringing about a global optimum. Six new variants of the original algorithm have been proposed in all in [39] and [45]. One of the variants regards a weighted computation of the simplex centroid by means of which the new optimization line is found, while another one creates a mixed algorithm by combining the simplex algorithm with the cyclic optimization algorithm on the axes, which is a convergent algorithm for the class of pseudo-convex functions (respectively pseudo-concave). Moreover, the newly created algorithm reduces the computation effort as compared to the algorithm of cyclic optimization on the axes and increases the reliability in obtaining the solution for each of the algorithms it is made up of.

Studies have been conducted for all the algorithms described in paragraph B, as regards their efficiency in solving test problems known for their

degree of difficulty. Through the large number of tests carried out, the numerical results given in [39]—[45] may contribute to solving the decision-making problem we are faced with, when we have to choose the appropriate algorithm. As the current point is getting nearer the optimum point, the variation graphs of the computation effort can be used for establishing a strategy of combined use of several algorithms, with a view to obtaining a maximum efficiency in solving the problem.

Similarly, gradient methods in the unconstrained optimization have been studied. As is known, in these methods the optimum point is successively approximated by the formula:

$$x^{k+1} = x^k + \lambda_k d^k \tag{1}$$

where d^k is the direction of the gradient or a direction linked in a some way with this direction. Special attention has been granted to the conjugate directions algorithms, which have the property of minimizing a quadratic form into a finite number of iterations and especially to the Fletcher-Reeves and Davidon-Fletcher-Powell algorithms.

For the Fletcher-Reeves algorithm, a direction formula has been studied in the hypothesis that the unidimensional optimum is determined, as usually in practice, with error:

$$d^k = g^k - \sum_{i=0}^{k-1} \gamma_i^k d^i, \qquad \text{where } \gamma_i^k = \frac{(g^k)^T(g^{i+1} - g^i)}{(d^i)^T(g^{i+1} - g^i)} \tag{2}$$

and $g^k = \nabla f(x^k)$. This formula, which is valid for the maximum problem, appears as a generalization of the formula direction of Fletcher-Reeves or modified Fletcher-Reeves algorithm. The idea of finding d^k as a combination of all previous directions appears with Beckman [12]. Assuming that the optimization on direction is achieved accurately enough, with Beckman all γ_i^k $i = 0,...$ $k - 2$ are null. The attempt to bring in a several-terms correction, as regards important errors occurring in approximating the unidimensional optimum, has been suggested by the fact that the greatest computation effort is made when finding this optimum. The results of the efficiency tests carried out by the author are, however, unconvincing. The digital tests with a two-term formula

$$d^k = g^k - \frac{g^k(g^k - g^{k-1})}{d^{k-1}(g^k - g^{k-1})}\, d^{k-1} - \frac{g^k(g^{k-1} - g^{k-2})}{d^{k-2}(g^{k-1} - g^{k-2})}\, d^{k-2} \tag{3}$$

show, in some cases, an increased efficiency in comparison with the one-term formula. But, in most cases, the results are unfavourable. It is as if the error caused by the fact that the function is not a quadratic form is conveyed to a greater extent by a formula with several terms an is prevalent on the error brought in by an insufficient optimization on direction.

As regards Davidon-Fletcher-Powell algorithm, a procedure of using the algorithm in solving a problem of local maximum of a function of class C^2, when the unidimensional optimization subroutine provides maximum on direction, is described in [44].

The Davidon-Fletcher-Powell algorithm as described in [32] and [51], just like in all the other references known by the author, is trying to find a

local minimum of a function $f(x) \in C^2$, $x \in R$ through a number of Newton-type approximations given by the iterative process (1) where

$$d^k = - H_k^{-1} \nabla f(x^k). \tag{4}$$

The matrix H^{-1}, whose computation in each point x^k would be difficult for various reasons, being approximated only with the aid of the first order differentials, by

$$H_k^{-1} = H_{k-1}^{-1} - \frac{H_{k-s}^{-1} v_k v_k^T H_{k-1}^{-1}}{v_k^T H_{k-1}^{-1} v_k} + \frac{s_k s_k^T}{s_k^T v_k} \tag{5}$$

where

$$s_k = x^k - x^{k-1} \tag{6}$$

$$v_k = \nabla f(x^k) - \nabla f(x^{k-1}) \tag{7}$$

and λ_k provides the minimum on direction d^k, that is

$$f(x^k + x_k d^k) = \min_{\lambda > 0} f(x^k + \lambda d^k). \tag{8}$$

The unit matrix, or any other symmetrical and positive definite matrix approximation, can be taken as H_0^{-1}. It is shown, in this case, [51] that H_0^{-1} remains positive definite, that is an essential condition for the stability of the process.

Our computation programs have used

$$H_0^{-1} = I. \tag{9}$$

If we have to solve an $f(x)$ maximum problem, obviously we shall minimize $-f(x)$. However, there are instances in which, from a practical point of view, things are not so very simple. For example, the subroutine that carries out the unidimensional optimization (8) may be programmed only for the maximum and thus its conversion for solving minimum probems shall require some intervention. Such problems are either complicated — even impossible if they exist only as an object program — or undisirable, because our entire system of optimization programs is built up for the maximum. The problem arises how to use the D—F—P algorithm for solving a problem of maximum under such conditions.

It seems, an this is how the author himself has worked so far, that the only thing to change is direction d^k, so that (4) becomes

$$d^k = H_k^{-1} \nabla f(x^k), \tag{4'}$$

and λ_k shall determine the maximum on direction d^k:

$$f(x^k + \lambda_k d^k) = \max_{\lambda > 0} f(x^k + \lambda d^k), \tag{8'}$$

while formulas (1), (5), (6) and (7) remain unaltered because, if d^k computed with (4) is a good direction for the minimum, $-d^k$ must be a good direction for the maximum.

This reasoning is correct as long as H_k^{-1} remains, under these conditions, positively defined, because only in this case we have

$$(d^k)^T \nabla f(x^k) = \nabla^T f(x^k) H_k^{-1} \nabla f(x^k) > 0. \tag{10}$$

Experimental tests show that by starting from $H_k^{-1} = I$, H_k^{-1} becomes negatively defined, even at the n^{th} iteration, in a maximizing process. As H_k^{-1} has become a negatively defined matrix, λ_k must be allowed to become negative and, in any case, an attempt to locate the optimum on direction d^k with a positive step will be a failure, which will uselessly increase the number of estimations of the function.

This remark led us to the conclusion that in the problem of maximum, in which the optimization subroutine also provides the maximum on direction, it is more efficient to approximate the inverse of Hessian matrix by the same formula (5) but as a negative definite matrix and to continue to use (4) instead of using the direction with a changed sign, given by the algorithm for the minimum. To secure the stability of the procedure, prove is made that if H_k^{-1} is negative definite then H_{k+1}^{-1} remains negative definite. The proof follows the same pattern as in the case when H_k^{-1} remains positive definite, when the algorithm for the minimum is used. The suggested procedure is stable and computes the value of the function several times less than in the case when the direction of unidimensional maximization was the direction, taken with a changed sign, as determined by the algorithm for the minimum.

As is known, the Davidon-Fletcher-Powell algorithm is a very efficient one. However, it has the disadvantage that in finding inverse of the Hessian matrix (5) a large store is necessary, which may be prohibitive for big problems. Of course, the approximation can be built, at each iteration, in the external store. However, this procedure is extremely slow and the time gained by the very good convergence speed of the algorithm will be by far surpassed by the low working speed of the procedure, caused by the transfer, at each iteration, of a big quantity of information from the internal store into the external store of the computer and vice versa. In such cases, we suggest in [43] the use of a mixed procedure, by combining a vectorial algorithm — which one operates with in the internal store until an approximation of the Hessian matrix inverse on the external store socket is achieved — with a Newton-Raphson matrix type algorithm; the optimization on the Newton type direction beeing done after a certain number of iterations, generally n. In order to obtain this result, three new mixed algorithms were suggested and studied, which we called: the gradient-hessian (or Cauchy-Newton) algorithm, the Fletcher-Reeves-Newton I algorithm [41] and the Fletcher-Reeves-Newton II algorithm [43].

In the Cauchy-Newton algorithm, n iterations are made on the gradient's direction during which an approximation of inverse of the Hessian matrix is found, as an application of the Sherman-Morrison formula [128] after which the optimization is made on the Newton direction given by formula (4). The Fletcher-Reeves-Newton I algorithm has the same steps as the Cauchy-Newton algorithm with the difference that instead of an optimization on the gradient's direction, optimizations are made on the Fletcher-Reeves directions. The study of these algorithms has led us to the conclusion that such a method is efficient, in the sense of the convergence rate, although it remains less efficient than that of the Davidon-Fletcher-Powell method. Having esta-

blished this, we must draw up an algorithm which can store the elements necessary for computing the approximation of the inverse in the external store with a smaller transfer of information as the first two algorithms using, just as the Davidon-Fletcher-Powel algorithm, a recurring formula for the approximation of the inverse, require, as we have mentioned, a big transfer of information at each iteration. This is done by the Fletcher-Reeves-Newton II algorithm, where the first n iterations are made after n mutually conjugate directions, for example Fletcher-Reeves directions and the approximation of the inverse can be built only with the aid of matrix D_N whose column vectors are the mutually conjugated vectors d^k, on which the n optimizations are made, normalized, by the factor

$$\nu_k = \frac{\lambda_k}{\sqrt{d^k g^k}} \tag{11}$$

The building up of matrix D_N requires at each iteration only the transfer of the normalized d^k vector.

It is pointed out in [43] that the approximation of the Hessian matrix inverse of a function of class C^2 is $H_n^{-1} = D_N D_N^T$ and if the function to be optimized is a quadratic form $f(x) = \frac{1}{2} x^T Ax + bx$, in this case $H_n^{-1} = A^{-1}$. Obviously, the method can be used only for finding the inverse of a symmetrical and define matrix A, under the conditions in which the initial point is given as to allow the creation of the n mutually conjugated directions. In this case $f(x) = \frac{1}{2} x^T Ax$ is minimized. If A is non-singular but simmetrical and we don't even know- — as it happened in practice — that it is defined, in this case $f(x) = \frac{1}{2} x^T A^T Ax$ is minimized and $A^{-1} = D D^T A^T$.

Moreover, the procedure can be used in finding the generalized inverse A^+ of a matrix A of the type $(m \times n)$, $m > n$, range $A = n$, in which case $A^+ = D D^T A^T$. The computation of the generalized inverse is a current problem in applications and equivalent to the problem of minimizing deviations by the least squares method.

It is seen from the way of building the algorithm that any other algorithm of conjugate directions can be used in the n iterations of the inverse construction, instead of the Fletcher-Reeves algorithm, so that this procedure defines a class of mixed algorithms.

As is known, most non-linear programming algorithms generate a series of convergent points in an infinite number of steps towards the optimum point. The inference of some criteria for estimating the error upon ending the computation process is an extremely important problem in solving nonlinear problems. Some relative results as to the character of the extremum points and to the extremum values of the pair of quadratic form are inferred in [45]: Let:

$f(x) = \frac{1}{2} x^T Ax + b^T x$ and $f_2(x) = \frac{1}{2} x^T A_x^{-1} + b^T x$, be two quasi conjugate quadratic forms.

Thus, the following propositions are proved:

1) If one of the forms, be it $f_1(x)$ admits an extremum point x_1^*, the other quadratic form $f_2(x)$ also admits an extremum point x_2^* of the same nature and we have $x_2^* = A^2 x_1^*$.

2) The extremum values $f_1(x_1^*)$ and $f_2(x_2^*)$ are strictly positive if x_1^* is a maximum point, and strictly negative if x_1^* is a minimum point, and we have the inequalities:

$$-\frac{1}{2}\frac{1}{\lambda 1} b^T b \leqslant f_1(x_1^*) \leqslant -\frac{1}{2}\frac{1}{\lambda n} b^T b \tag{12}$$

$$-\frac{1}{2}\lambda_n b^T b \leqslant f_2(x_2^*) \leqslant -\frac{1}{2}\lambda_1 b^T b \tag{13}$$

where λ_1 is the smallest and λ_n the largest characteristic value of the definite matrix A.

The above formulas, however, do not practically solve the problem. The computation of the characteristic values is a difficult task and if the characteristic values λ_1 and λ_n are very different, the interval yielded by the above formulas is large.

It is not difficult to prove that the inequalities satisfied by the values of the mentioned quadratic forms in the extremum points no longer depend on the knowing of λ_1 and λ_n but only on the interval in which the values characteristic of A are placed:

If $-\infty < \lambda_I < -1$ or $< \lambda_i < 1$

$$\text{then } f_1(x_1^*) < f_2(x_2^*) = -\frac{1}{2} bAb \tag{14}$$

If $-1 < \lambda_i < 0$ or $\lambda_i > 1$

$$\text{then } f_1(x_1^*) > f_2(x_2^*) = -\frac{1}{2} bAb. \tag{15}$$

The above formulaas may be useful in the cases where the condition of applicability is found with the aid of the known theorem: Any characterstic value λ_i of a matrix A does not surpass in the modulus any of its canonical norms.

As regards the method of solving a large scale nonlinear programming problem having a special structure with the aid of the linear parametric programming, we think it is of particular importance, considering the efficiency of the linear programming codes. We also think that it is possible to create some computing systems on this basis. In this respect, we mention a result obtained in collaboration with V. Ionescu [38] and with which our nonlinear programming studies have started. It refers to the solving with the aid of the parametric linear programming of the following class of nonlinear problems:

$$max\ x_0 = \sum_{i=1}^{n} C_j(y)xj, \tag{16}$$

$$\sum_{j=1}^{n} a_{ij}(y)x_j = b_i(y),\ i = 1, ..., n \qquad (17)$$
$$x_j \geqslant 0 \qquad j = 1, ..., n$$
$$y \geqslant 0$$

where

$c_j(y)$, $a_{ij}(y)$ and $b_i(y)$ are polynomials of a certain degree in y.

An algorithm for solving the above problem is given in 96 for the case in which two vectors of the matrix and the free term are polynomials of the first degree in y. The problem is also dwelt on in [37], in case that $c(y)$ are polynomials of the second degree and a_{ij} and b_i are constant. We consider the case in which one of the column vectors of the matrix of coefficients and also the vector of free terms are second degree polynomials.

B. The results on the efficiency of the algorithms are given detailedly in [39] [40], [41] [42] and [45].

The studies on the efficiency aimed, in the first place, at examining the behaviour of various algorithms in finding the optimum of functions having different degrees, of nonlinearity with a view of finding for instance, the capacity of the algorithm to solve problems when the iterative process takes place in an area in which the function has a deep curve, the Hessian becomes singular, the function is quadratic but with a wide range of the characteristic values etc. In general, we have had in view the capability of the algorithm of solving the problem with certain accuracy in a given time limit for the machine. To make possible a comparison of the results obtained by various researchers — since the computation time depends on the capability of the computer- we have assumed that the computation time is proportional to the number $NE = NF + n \cdot NG$, where NF is the number of function evaluations NG of the gradient and n is the number of variables. By accepting this formula, we must also accept the objection one may raise that the computation time also depends on the number of logical operations strictly necessary for the algorithm. The answer is given by the fact that the above computation time is prevalent for large problems.

From the viewpoint of general conclusions, we think our test may bare several aspects:

a) The study of the efficiency must be conducted in a methodological way throughout the optimization process. The results given in [45], where NE has been found for $d = \| x^k - x^T \| < \varepsilon$, ($\varepsilon = 1, ..., 10^{-6}$), render an image of the algorithm behaviour other than those given in [41], where the computation effort was estimated only at the end of the optimization process.

b) The efficiency of the procedure depends, at least in the vicinity of the optimum point, on the rounding error brought in by the computer. If we intend to obtain high precision results, a double-precision cmputation is absolutely necessary. One example will make things clear: when using the Fletcher-Reeves algorithm for solving Rosenbrock's problem in simple precision, with an equivalent number of estimation of function $NE = 412$, the algorithm reaches a point at distance from the optimum equal to $d = 0.15 \times 10^{-4}$, after which the precision cannot be improved any more. For $NE = 3093$, the distance is $d = 0.12 \times 10^{-4}$ and the points obtained will oscillate around the point reached for $NE = 412$. In double precision, for

$NE = 254$ the disstance is $d = 0.33 \times 10^{-4}$ and the process comes to an end after satisfying the stop criterion $\| g^k \| < 10^{-6}$ having $d = 0.62 \times 10^{-7}$ for $NE = 294$. This aspect of the problem becomes particularly conspicuous when comparing simple — and double — precision graphs of efficiency estimation. In simple precision the curve of efficiency estimation tends, under a certain value of accuracy, to become parallel with the abscissa on which the computation effort is measured. The tests were carried out with IBM/360 and IBM/370 computers where the word length is 32 bits.

c) The efficiency of the procedure is considerably influenced also by the strategy of the length of the search step in the unidimensional optimization subroutine. In general, we got the best results when, at the beginning of the optimization process, the constant first step was normalized by $\| d^k \|$. When the current point got near the optimum point, the first step was taken equal to a fraction of the total length of the step achieved with the previous iteration or as a means of the total lengths of steps for the last n unidimensional optimizations.

d) The efficiency is also closely linked with the choice of the initial point.

With regard to the algorithms efficiency in each of the surveyed classes we think that, based on the results obtained in solving the considered test problems, we can conclude the following: i) The most efficient results in the class of algorithms not using the gradient were obtained by Davies, Swann and Campey 's algorithm (DSC) and by Powell's algorithm. In general, they have the best convergence speed, even in the vicinity of the optimum point. There are some exceptions, too: Powell's algorithm for Fiacco and McCormic function [1] indicates as an optimum the stationary but non optimal point $x = (0, 0)$, which is strange, since we do not deal with a gradient method. Likewise, the algorithm converges relatively slowly for Wood [2] and Powell [3] functions.

Hook and Jeeves algorithm has a very good behaviour, but it requires a large number of function estimations for satisfying the stop criterion. Rosenbrock's algorithm holds the next place.

The efficiency of cyclical optimization algorithm is generally low. However, there are some exceptions also in this case: Witte and Holst's cubic problem[4] and the ten-variable square problem[5]. In this latter case the hiperelipsoid has its centre in the origin and the axes on the direction of the co-ordinate axes and, obviously, for such a problem the algorithm will have a good behaviour. The optimization on the diagonal increases its convergence speed.

---

1) $f(x) = (x_1^4 - 3)^2 + x_2^4$;

2) $f(x) = 100\ (x_1^2 - x_2)^2 + (1 - x_1)^2 + 90(x_3^2 - x_4)^2 + (1 - x_3)^2 + 10.1[(x_2 - 1)^2 + (x_4 - 1)^2] + 19.8(x_2 - 1)\,(x_4 - 1)$,

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

4) $f(x) = 100(x_2 - x_1^2) + (1 - x_1)^2$

5) $f(x) = \sum_{k=1}^{10} (110 - 10k)x_k^2$ or $f(x) = \sum_{k=1}^{10} (110 - 10k)^2 x_k^2$

Save for Rosenbrock[1] and Beale[2] functions, the Nelder and Mead's simplex algorithm has-as it appears from our experiment — a low convergence speed, by no means comparable with Powell's algorithm, as it is mentioned in the literature.

However, the convergence is considerably dependent on the configuration and dimension of the original simplex. The algorithm has the disasvantage that the simplex may degenerate and, in this case, iterations will stop at a point lying far from the optimum. With regard to the reliability in solving the problem, we obtained better results when, from the very beginning or at each m iterations the simplex was extended by means of a unidimensional optimization subroutine (maximal simplex), but in this case, the number of estimations of the function was very large.

The simplex weighted centroid, in agreement with formula given in [39], has a better behaviour in all cases except Fiacco and McCormic's function; when using also the unidimensional optimization subroutine the results are still better save for Rosenbrock's function. The disadvantage of weighting is conspicuous when the vertices of the simplex are laid on a level surface.

According to the results obtained with the carried out tests, the combination of the simplex algorithm with the cyclic algorithm into the simplex-cyclo mixed algorithm is favourable, at least from the viewpoint of the reliability in solving the problem and the accuracy. The number of function evaluations is rather large, especially because the cyclic algorithm.

ii) We studied the efficiency in the class of algorithms using the information provided by the gradient for: Cauchy's algorithm, subrelaxed algorithm, gradient-diagonal mixed algorithm, Fletcher-Reeves algorithm, modified Fletcher-Reeves algorithm, parallel tangents algorithm (PARTAN), Davidon-Fletcher-Powell algorithm, Cauchy-Newton mixed algorithm and Fletcher-Reeves-Newton II algorithm.

The numerical results have been published in [43] and [45] in two variants: one using subroutine STEP that search for the optimum on direction through successive interpolations with a quadratic, without using derivatives and the other using subroutine STEPG that interpolates cubicaly and uses the gradient.

Our numerical results, on a large number of uniformly effected tests, have generally confirmed the results obtained by other researchers. The conjugate directions algorithms have a more efficient behaviour in solving the problems also for non quadratic functions. Among them a special place is held by Davidon-Fletcher-Powell algorithm and Fletcher-Reeves algorithm and the variants.

With respect to the algorithms surveyed in this class, our research has shown the following:

Cauchy's algorithm is one of a very slow speed convergence, and its use is more inefficient as the necessary precision or the number of variables is

---

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

2) $f(x) = \sum_{i=1}^{3} \{C_i - x_1(1 - x_2^i)\}_j^2 \quad c_1 = 1.5\ c_2 = 2.25,\ c_3 = 2.625$

larger. Its subrelexed use is obviously advantageous, diminishing approximately by half the computation effort, especially at the beginning of the optimization process.

The combination of Cauchy's algorithm with an optimization "on the diagonal" into a so-called "gradient-diagonal" algorithm has a spectacular effect from the efficiency viewpoint. We find again an approximately 20 times increase of the convergence speed for Rosenbrock's function, for Witte and Holst's problem or for Fiacco's problem, as mentioned in [139]. However, the speed increases only a few times for Wood or Powell's problem. With these examples, the "gradient-diagonal" algorithm is comparable with Fletcher-Reeves algorithm, that is it has a speed convergence in the class of conjugate direction algorithms. Also in [139] it is mentioned that the experience existing at the time indicated as the best strategy for the diagonal movement its carrying out after performing seven optimizations on gradient direction. However, it is natural to think that the optimal strategy rather depends on the number of variables than on the number of optimizations. Consequently, for each problem we carried out tests with a diagonal movement after $p = n + 1$, $p = n + 3$, $p = n + 5$ optimizations on gradient direction. The results of the tests prove that the idea of relating the diagonal movement to the number of variables is viable and that the best strategy would consist in performing the optimization on diagonal after $n + 1$ optimizations on gradient direction where n stands for the number of variables.

We should suggest the continuation of the experimental research on the gradient-diagonal algorithm by means of a formula for finding the direction given by $d^k = x^k - x^{k-n}$, $k \equiv 0 \pmod{n}$, $k = 0,1, \ldots$, that is an algorithm similar to the cyclical algorithm with optimization on diagonal.

By comparing Fletcher-Reeves algorithm and its modified variant [1], tests show a slight superiority of the modified variant for Fiacco, Beale, Box and Powell's problems. For Rosenbrock's function the same results are obtained with subroutine STEPG, while better resullts with the modified variant are obtained by means of subroutine STEP. The cubic function of Witte an Holst shows nonconclusive results. With respect to Wood's problem, the modified Fletcher-Reeves algorithm, as well as Davidon-Fletcher-Powell algorithm, reaches during the optimization process the singular Hessian, known point from whose area it goes out only by a big computation effort. Beside this situation and the results obtained with subroutine STEPG for Witte and Holst' cubic function, Davidon-Fletcher-Powell algorithm is the most efficient one tested by us.

The parallel tangents algorithm (PARTAN) is the least efficient one in the class of conjugate directions algorithms. Cauchy-Newton mixed algorithm (hessian-gradient) is definitely superior, though it is less efficient than Fletcher-Reeves algorithm. The mixed Fletcher-Reeves-Newton II algorithm has been tested both for studying its efficiency and for checking its capacity of accurately determining the inverse.

With regard to the efficiency of the procedure, it seems that, if the optimization on Newton-type direction $d^{n+1} = -H_n^{-1} \cdot g^{n+1}$ is made at the

1) $d^k = -g^k + \dfrac{g^k g^k - g^k g^{k-1}}{d^{k-1} g^{k-1}} d^{k-1}$

beginning of the process and we are far from the optimum point, this direction is less good than the $n + 1$ Fletcher-Reeves direction and, consequently, the algorithm appears to be — at least in this part of the process — less efficient than Fletcher-Reeves algorithm.

In order to improve the efficiency, we carried out tests in which the movement on Newton-type direction is effected only if the current point is close enough to the optimum point, for instance if $\| g^k \| < 1$; in this case the algorithm proves to be more efficient than Fletcher-Reeves algorithm, and the convergence in the vicinity of the optimum point is sometimes even faster than for Davidon-Fletcher-Powell algorithm. However, as a whole, even this strategy is an idea less efficient than Davidon-Fletcher-Powell procedure.

The capability of the algorithm of providing, for the nonquadratic function, the Hessian matrix inverse in the vicinity of the optimum point was tested for all the test problems and in all cases we obtained matrices very close to those obtained by means of Davidon-Fletcher-Powell algorithm. As an example, we give in the appended table the approximation obtained for the Hessian matrix inverse through both methods, as far as Rosenbrock, Witte and Holst, Beale and Wood's problems are concerned. The algorithm has also been tested for its capability of providing the inverse of a positive definite matrix with a wide range of its eigen values. In all cases, after n, double-precision, iterations we obtained the inverse with an error less than 10^{-5} for each element.

iii) On studying the behaviour of the optimization process on various portions of our iteration, it occurred to us to use combined algorithms. As against the mixed algorithm, which continuously makes up several algorithms, the combined algorithm uses distinctly, in various stages of the process, the algorithms having the best convergence speed. The carried out tests have not allowed so far the definition of such a strategy.

iiii) With regard to the research of algorithm efficiency in solving constrained problems, Powell, Fletcher-Reeves, Davidon-Fletcher-Powell and Rosen algorithms, as well as Box's complex method were studied in a comparative manner; the first three algorithms are used for solving a constrained problem within the SUMT method.

From the results concerning the tests on this problem we can draw a conclusion whose truthfulness could be assumed a priori: the most efficient unconstrained optimization algorithms are also the most efficient in solving a problem through the SUMT method, and a specialized algorithm is more efficient than a general algorithm.

C) The following programs and subroutines [45] have been written, tested and investigated from the viewpoint of the solving capability of the various classes of problems, in collaboration with Florica Luban, Sigrid Moga and Radu Şerban, members of the Operational Research Laboratory, directed by the author, at the Cybernetics Department of The Academy of Economic Studies in Bucharest:

P. 1. The main program for calling subroutines SIMIXC, POWELL, CYCO and HOOKJV.

P. 2.. The main program for calling subroutines RSBR and RSBWS.

P. 3. The program for calling subroutines SMPX and SMPXW.

P. 4. The main program for calling unconstrained optimization subroutines using the gradient.

P. 5. The program for solving the constrained problems through Box's COMPLEX method (in collaboration with Sigrid Moga and M. Falk).

P. 6. Rosen's program for solving linearly constrained convex problems (in collaboration with Radu Şerban).

S. 1. The unidimensional optimization subroutine STEP.

S. 2. The unidimensional optimization subroutine STEPG.

S. 3. Subroutine CYCO (in collaboration with Florica Luban) programs the cyclical optimization algorithm on the axes and, possibly, on diagonal.

S. 4. Subroutine HOOKJV (in collaboration with Radu Şerban) programs Hooke and Jeeves algorithm.

S. 5. Subroutine POWELL (in collaboration with Florica Luban) programs Powell's algorithm.

S. 6. Subroutine SIMIXC (in collaboration with Florica Luban) programs the succesive simplex mixed type algorithm, with cyclical optimization on co-ordinates axes.

S. 7. Subroutine RSBR (in collaboration with Sigrid Moga) programs Rosenbrock's algorithm.

S. 8. Subroutine RSBRWS (in collaboration with Sigrid Moga) programs Davies, Swann and Campey algorithm.

S. 9. Subroutine SMPX (in collaboration with Sigrid Moga) programs the successive simplex algorithm, with or without weighted centroid.

S.10. Subroutine SMPXWS (in collaboration with Sigrid Moga) programs the unidimensional optimization successive simplex algorithm, with or without weighted centroid.

S.11. Subroutine CAUCHY (in collaboration with Sigrid Moga) programs Cauchy's algorithm.

S.12. Subroutine GRDSR (in collaboration with Sigrid Moga) programs the subrelaxed gradient algorithm.

S.13. Subroutine DIAGL (in collaboration with Sigrid Moga) programs the gradient-diagonal mixed type algorithm.

S.14. Subroutine FLRV programs the Fletcher-Reeves algorithm.

S.15. Subroutine FLRVMO programs an algorithm on conjugated directions in which the direction is computed according to a modified Fletcher-Reeves formula.

S.16. Subroutine TGPAR (in collaboration with Sigrid Moga) programs the parallel tangents algorithm.

S.17. Subroutine DVFLPW (in collaboration with Sigrid Moga) programs the Davidon-Fletcher-Powell algorithm.

S.18. Subroutine CAUHES (in collaboration with Florica Luban) programs the gradient hessian mixed type algorithm and Newton-Fletcher-Reeves I algorithm.

S.19. Subroutine FRNII (in collaboration with Florica Luban) programs Fletcher-Reeves-Newton II algorithm.

Approximating * the Inverse Hessian Matrix Using the Fletcher-Reeves-Newton Method, Comparing With Davidon-Fletcher-Powell Method

<table>
<tr><td colspan="2">$min\ f(x) = 100\ (x_2 - x_1^2)^2 + (1\text{-}x_1)^2$(Rosenbrock)</td><td colspan="2">$min\ f(x) = 100(x_2 - x_1^3)^2 + (1 - x_1)^2$ (Witte & Holst)</td></tr>
<tr><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)
$k = 31$, $\|g\|^2 = 0.72 \times 10^{-15}$
$f = 0.36 \times 10^{-18}$; $d = 0.4 \times 10^{-10}$
$nf = 68$; $ne = 204$
$H_k^{-1} = \begin{pmatrix} 0.50000 & 0.99999 \\ 0.99999 & 2.0050 \end{pmatrix}$</td><td>DAVIDON-FLETCHER-POWELL
$k = 29$; $\|g\|^2 = 0.1 \times 10^{-17}$
$f = 0.41 \times 10^{-20}$; $d = 0.1 \times 10^{-9}$
$nf = 69$; $ne = 207$
$H_k^{-1} = \begin{pmatrix} 0.50002 & 1.0004 \\ 1.0004 & 2.0058 \end{pmatrix}$
$k = 30$; $\|g\|^2 = 0.$
$f = 0$; $d = 0.$
$nf = 72$; $ne = 216$</td><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)
$k = 16$; $\|g\|^2 = 0.28 \times 10^{-9}$
$f = 0.71 \times 10^{-13}$; $d = 0.94 \times 10^{-7}$
$nf = 38$; $ne = 114$
$H_k^{-1} = \begin{pmatrix} 0.50019 & 1.5003 \\ 1.5003 & 4.5050 \end{pmatrix}$
$k = 17$; $\|g\|^2 = 0.11 \times 10^{-15}$
$f = 0.2 \times 110^{-17}$; $d = 0.45 \times 10^{-8}$
$nf = 41$; $ne = 123$</td><td>DAVIDON-FLETCHER-POWELL
$k = 24$; $\|g\|^2 = 0{,}4 \times 10^{-16}$
$f = 0.12 \times 10$; $d = 0{,}22 \times 10^{-10}$
$nf = 108$; $ne = 324$
$H_k^{-1} = \begin{pmatrix} 0.50000 & 1.5000 \\ 1.5000 & 4.5050 \end{pmatrix}$
$k = 35$; $\|g\|^2 = 0.4 \times 10^{-25}$
$f = 0.24 \times 10^{-28}$; $d = 0.21 \times 10^{-13}$
$nf = 114$; $ne = 342$</td></tr>
<tr><td colspan="2">$min\ f(x) = \sum_{i=1}^{3} [ci - x_1(1 - x_2^i)]^2$, $c_1 = 1.5$; $c_2 = 2.25$; $c_3 = 2.625$
(Beale)</td><td colspan="2">$min\ f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2 + 90(x_4 - x_3^2)^2 + (1 - x^3)^2 + 10.1(x_2 - x_1)^2 + (x_4 - x_1)^2] + 19.8(x_2 - x_1)(x_4 - x_1)$ (Wood)</td></tr>
<tr><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)
$k = 11$; $\|g\|^2 = 0.16 \times 10^{-12}$
$f = 0.16 \times 10^{-14}$; $d = 0.84 \times 10^{-8}$
$nf = 29$; $ne = 87$
$H_k^{-1} = \begin{pmatrix} 3.1222 & 0.77439 \\ 0.77439 & 0.21376 \end{pmatrix}$
$k = 12$; $\|g\|^2 = 0.58 \times 10^{-21}$
$f = 0.91 \times 10^{-21}$, $d = 0.7 \times 10^{-10}$
$nf = 31$; $ne = 93$</td><td>DAVIDON-FLETCHER-POWELL
$k = 9$; $\|g\|^2 = 0.16 \times 10^{-20}$
$f = 0.18 \times 10^{-22}$; $d = 0.2 \times 10^{11}$
$nf = 27$; $ne = 81$
$H_k^{-1} = \begin{pmatrix} 3.1238 & 0.7746 \\ 0.7746 & 0.2137 \end{pmatrix}$
$k = 10$; $\|g\|^2 = 0.$
$f = 0$; $d = 0$
$nf = 28$; $ne = 84$</td><td>FLETCHER-REEVES-NEWTON ($\mu < 1$)
$k = 29$; $\|g\|^2 = 0.27 \times 10^{-13}$
$f = 0.23 \times 10^{-16}$; $d = 0.47 \times 10^{-8}$
$nf = 65$; $ne = 325$; $H_k^{-1} =$
$= \begin{Bmatrix} 0.1427 & 0.2836 & -0.1352 & -0.2713 \\ & 0.5686 & -0.2712 & -0.5440 \\ & & 0.1427 & 0.2835 \\ & & & 0.5686 \end{Bmatrix}$
$k = 30$; $\|g\|^2 = 0.37 \times 10^{-19}$
$f = 0.23 \times 10^{-22}$; $d = 0.3 \times 10^{-11}$
$nf = 76$; $ne = 380$</td><td>DAVIDON-FLETCHER-POWELL
$k = 37$; $\|g\|^2 = 0.16 \times 10^{-12}$
$f = 0.37 \times 10^{-15}$; $d = 0.13 \times 10^{-7}$
$ne = 163$; $ne = 815$ $H_k^{-1} =$
$= \begin{Bmatrix} 0.1427 & 0.2837 & -0.1353 & -0.2714 \\ & 0.5688 & -0.2713 & -0.5442 \\ & & 0.1428 & 0.2836 \\ & & & 0.5689 \end{Bmatrix}$
$k = 38$; $\|g\|^2 = 0.4 \times 10^{-17}$
$f = 0.59 \times 10^{-20}$; $d = 0{,}4 \times 10^{-10}$
$nf = 165$; $ne = 825$</td></tr>
</table>

* STEPG, double precision.

S.20. Subroutine VEM (in collaboration with Sigrid Moga) unites in a sole subroutine, vectorial methods for determining the directors of unidimensional optimizations.

S.21. Subroutine MAM (in collaboration with Florica Luban) unites in a sole subroutine, metrix methods of determining the direction unidimensional optimizations.

REFERENCES

0. ABADIE J. (Ed.) *Nonlinear Programming.* North-Holland, Amsterdam, 1967.
1. ABADIE J., (Ed.): *Integer and Nonlinear Programming.* North-Holland, Amsterdam, 1970.
2. ABADIE J., GUIGOU J.: *Numerical Experiments with the GRG Method, in Integer and Nonlinear Programming.* North-Holland, Amsterdam 1970, Abadie J. (Ed.).
3. ADACHI N.: *On the Uniqueness of Search Directions in Variable-metric Algorithms.* JOTA, 11, 6, 1973.
4. ALLRAN R. R., JOHNSEN S.E.J.: *An Algorithm for Solving Nonlinear Programming Problems Subject to Nonlinear Inequality Constraints.* The Computer Journal, vol. 13, 1970, No.2.
5. AMUNATEGUI J., DUMITRU V.: *Computer Experiments with an Unconstrained Nonlinear Program.* Economic Computation and Ec. Cybernetics, 3, 1970.
6. AOKI M.: *Introduction to Optimization Techniques.* Macmillan, New York, 1971.
7. ARMACOST R. L., FIACCO A.V., *Computational Experience in Sensitivity Analysis for Nonlinear Programming.* Mathematical Programming, Vol. 6, No.3, June 1974.
8. AUSLENDER A.: *Optimistion. Methodes numériques.* Masson, Paris-New York-Barcelone-Milan, 1976.
9. BALAKRISHNAN A. V. (ed.): *Techniques of Optimization.* Academic Press, New York, 1972.
10. BEALE, E.M.L.: *Numerical Methods, in Nonlinear Programming.* J. Abadie, ed., North-Holland, Amsterdam 1967.
11. BEALE E.M.L.: *Mathematical Programming in Practice.* Pitman & Sons, London 1968.
12. BECKMAN F. S.: *The Solution of Linear Equations by the Conjugate Gradient Method* In Mathematical Methods for Digital Computers. Vol. I—II. Ralston A., Wilf H., Eds. J. Wiley, New York, 1967.
13. BLUM E., OETTLI W.: *An Extension of the Method of Feasible Directions.* In Optimization Techniqes, IFIP Technical Conference, Goos, G., Hartmanis, J., (Eds), Berlin-Heidelberg-New York, 1975.
14. BOX M. J.: *New Method of Constrained Optimization and a Comparison with Other Methods.* Computer J., 8 1965.
15. BOX M. J.: *A Comparison of Several Curent Optimization Methods and the Use of Transformations in Constrained Problems.* The Comp. J. 9, 1966, 67—77.
16. BOX M. J., DAVIES D., SWANN W. H.: *Non-Linear Optimization Techniques.* Monograph, Oliver and Boyd, Edinburgh, 1969.
17. BRACKEN I., MC CORMICK G. P.: *Selected Applications of Nonlinear Programming* Wiley, New York, 1968.
18. BROYDEN C. G.: *The Convergence of a Class of Double-Rank Minimization Algorithms.* 1. General considerations JIMA, 6,76—90, 1970 2. The néw algorithm JIMA, 222—231.
19. CANON M. D., CULLUM C. D., POLAK E.; *Theory of Optimal Control and Mathematical Programming.* McGraw-Hill, New York, 1970.
20. CARROLL C. W.: *The Created Response Surface Technique for Optimizing Nonlinear Restrained Systems.* Operations Res. 9(2), 1961.
21. CAUCHY A.: *Méthode générale pour la résolution des systèmes d'équations simultanées.* Compt. Rend., 25: 536—538 (1847).
22. CEA J.: *Optimisation, théorie et algorithmes.* Dunod, Paris, 1971.

23. COLLUM JANE: *An Algorithm for Minimizing a Differentiable Function that Uses only Function Values.* In Techniques of Optimization, A. V. Balakrishnan. Ed., Academic Press, New York, London, 1972.
24. COLVILLE A. R.: *A Comparative Study on Nonlinear Programming Codes.* IBM Scientific Center Technical Report No. 320—2949, New York, 1968.
25. COTTLE R. W.: *The Principal Pivoting Method of Quadratic Programming.* In Mathematics of the Decision Sciences, part 1, G. B. Dantzig and A. F. Veinott, (Eds), Amer. Math. Soc., Providence, R. I., 1968.
26. CURRY H. B.: *The Method of Steepest Descent for Non-Linear Minimization Problems.* Quart. Appl. Math. 2(3), 1944.
27. DANCEA IOAN: *Metode de optimizare.* Ed. Dacia, Cluj-Napoca, 1976.
28. DANIEL J. W.: *Convergent Step-Sizes for Gradient-Like Feasible Direction Algorithms for Constrained Optimization.* In Nonlinear programming, ed. J. B. Rosen, O. L. Mangasarian and K. Ritter, Academic Press, 1970.
29. DANILIN YU. M.; *Dual Direction Methods for Function Minimization.* In Optimization Techniques, IFIP Technical Conference, G. Goos & J. Hartmanis (Eds.), Springer-Verlag, Berlin-Heidelberg-New York, 1975.
30. DANTZIG G.: *Linear Programming and Extensions.* Princeton Univ. Press, Princeton, N. J., 1963.
31. DANTZIG G. B., VEINOTT A. F., (Eds.: *Mathematics of the Decision Sciences.* Amer. Math. Soc., Providence, R. I., 1968.
32. DAVIDON W. C.: *Variable Metric Method for Minimization.* A.E.C. Research and Development Report, ANL-5990, 1959.
33. DAVIES D., SWANN W. H.: *Review of Constrained Optimization.* In Optimization, Fletcher, R. (Ed.), Academic Press, London, 1968.
34. DAVIES D.: *Some Practical Methods of Optimization.* In Integer and Nonlinear Programming. Abadie, J., (Ed.), North-Holland, Amsterdam, 1970.
35. DESCHENEAU J. C.: *Comparative Studies in Non-Linear Mathematical Programming Techniques and Their Application to Optimal Control Problems.* M. Sc. Thesis, Computing Science Department, U. Of Alberta, Edmonton, 1973.
36. DRAGOMIRESCU M., MALIȚA M.: *Programare neliniară.* Ed Științifică, București, 1972.
37. DRĂGAN I.: *Asupra unei clase de probleme de programare parametrică.* St. cerc. mat.,17, 3, 1965.
38. DUMITRU V., IONESCU V.: *Une méthode de solution numérique d'une catégorie de problèmes de programmation non linéaire.* Ec. Comp. & Ec. Cyb. 1, 1966.
39. DUMITRU V.: *Functions Optimization without Evaluating Derivatives.* Ec. Comp. & Ec. Cyb. Part. I, 2, 1973, Part. II, 1, 1974.
40. DUMITRU V., MOGA S.: *Testing the Efficiency of the Sequential Optimizing Technique.* Ec. Comp. & Ec. Cyb. 4, 1973.
41. DUMITRU V.: Gradient Methods for Unconstrained Optimization, Ec. Comp. & Ec. Cyb, 4, 1974.
42. DUMITRU V., ȘERBAN R.: *Speed Convergence and the Choice of Unidimensional Extremum Point.* Ec. Comp. & Ec. Cyb., 1, 1975.
43. DUMITRU V.: *Conjugate Directions Method, Hessian Approximation and a New Class of Mixed Type Algorithms for Optimization.* Ec. Comp.& Ec. Cyb, 2, 1975.
44. DUMITRU V.: *On the Davidon-Fletcher-Powell Algorithm for Maximization.* Ec. Comp & Ec. Cyb, 3, 1975.
45. DUMITRU V.: *Programare neliniară: algoritmi, programe, rezultate numerice.* Appendix P. In collab. with Fl. Luban, S. Moga, R. Șerban, Ed. Academiei, București, 1975.
46. EVANS J. P., GOULD F. J., TOLLE J.W.: *Exact Penalty Functions in Nonlinear Programming.* Mathematical Programming, 4, 1973.
47. EVTUSHENKO YU. G.: *Algorithms for Solving Non-Linear Programming Problems.* In Optimization Techniques. IFIP Conference, G. Goos & J. Hartmanis Eds., Springer -Verlag, Berlin-Heidelberg-New York, 1975.
48. FAURE P., HUARD P.: *Résolution des programmes mathématiues à fonction nonlinéaire par la methode du gradient réduit.* Revue Francaise de Recherche Opérationnelle, 9, No. 36, 1965.
49. FIACCO A. V., McCORMICK G. P. *Nonlinear Programming, Sequential Unconstrained Minimization Techniques.* Wiley, New York, 1968.

50. FIACCO A. V.: *Penalty Methods for Mathematical Programming in E with General Constraint Sets.* JOTA, Vol 6, No.3, 1970.
51. FLETCHER R., POWELL M.J.D.: *A Rapidly Convergent Descent Method for Minimization.* The Comp. J. 6 1963, 163—168.
52. FLETCHER R., REEVES C. M.: *Function Minimization by Conjugate Gradients.* Computer J., 7, 1964.
53. FLETCHER R.: *Function Minimization Without Evaluating-Derivatives — Review.* The Comp. J., 8,33—41, 1965.
54. FLETCHER R. (ED).: *Optimization.* Academic Press, New York, 1969.
55. FLETCHER R., MCCANN A. P.: *Acceleration Techniques for Nonlinear Programming in Optimization.* Fletcher Ed., Academic Press, 1969.
56. FLETCHER R.: *A New Approach to Variable Metric Algorithms.* The Comp. J .13, 1970, 317—322.
57. FLETCHER R.: *A Class of Methods for Nonlinear Programming with Termination and Convergence Properties.* In Integer and Nonlinear Programming, Amsterdam, 1970.
58. FLETCHER R.: *An Exact Penalty Function for Nonlinear Programming, with Inequalities.* Mathematical Programming, 5, 1973.
59. GEOFRRION A. M.: *Strictly Concave Parametric Programming.* Management Science, Part I 13, 1966, Part. II 13(5), 1967.
60. GEOFRRION A. M.: *Elements of Large-Scale Mathematical Programming,* Management Science, July 1970, Vol. 16, No. 11.
61. GOFFIN R., DE BEER A., KILSDONK A. C.M.: *Non-Linear Optimization Package OPTAC-II.* Information Systems and Automation Department. N. V. Philips, Eindhoven, 1974.
62. GREENSTADT J.: *Variations on Variable-Metric Methods.* Mathematics of Computation Vol. 24, No. 109, American Mathematical Society, Providence, 1970.
63. GUIGOU J.: *Expériences numériques comparatives concernant la méthode du gradient conjugué.* Bulletin E.D.F., Serie C, No.2, 1968.
64. GUIGOU J.: *Présentation et utilisation du code GREG.* Note HI 582/2. Electricité de France, Paris, 1971.
65. HAUER J. F., DESCHENEAU J. C.: *MORPAK-I: A Mathematical Optimization Routines Package for Unconstrained Optimization.* Technical Report TR 73-9, Computing Science Dept., U. of. Alberta, July 1973.
66. HESTENES M. R.: *Multiplier and Gradient Methods.* JOTA, Vol. 4, No. 5, 1969.
67. HIMMELBLAU D. M.: *Applied Nonlinear Programming.* McGraw-Hill, New York, 1972.
68. HOOKE J., JEEVES T. A.: *Direct Search Solution of Numerical and Statistical Problems.* JACM, 8, 1961, 212—229.
69. HUANG H. Y., LEVY A. V.: *Numerical Experiments on Quadratically Convergent Algorithms for Function Minimization.* JOTA, 6, 3, 1970.
70. HUANG N. Y., CHAMBLISS J. P.: *Quadratically Convergent Algorithms and One-dimensional Search Schemes.* JOTA, 11,2, 1973.
71. HUARD P.: *Résolution de P.M. à contraintes non-linéaires par le méthode des centres.* Note E.F.D. HR 5690, May 6, 1964.
72. HUARD P.: *Algorithmes d'optimistion et fonctions multiroques,* Bulletin E.D.F. Serie C, No.2, 1972.
73. IONESCU VL., LUPAȘ L.; *Tehnici de calcul în teoria sistemelor vol.* I—II, Ed. Tehnică, București, 1974.
74. JACOBY S.L.S., KOWALIK J.S., PIZZO, J.T.; *Iterative Methods for Nonlinear Optimization Problems.* Prentice-Hall, Englewood Cliffs, N. J., 1972.
75. KAPLAN A. A.: *Penalty Methods and Some Applications of Mathematical Programming.*- In Optimization Techniques, IFIP Technical Conference, G. Goos & J. Hartmanis Eds., Springer-Verlag, Berlin-Heidelberg-New York, 1975.
76. KELLEY J. W., JR.; *Cutting Plane Method for Solving Convex Programs.* J. Soc. Ind. Appl. Math. 8 (4), 1960.
77. KIEFER J.: *Optimum Sequential Search and Approximation Methods under Minimum Regularity Assumption,* SIAM J., 5, 1957.
78. KLESSING T., POLAK E.; *Efficient Implementations of the Polak-Ribière Conjugate Gradient Algorithm,* SIAM J. Control, 10, 3, 1972.
79. KOWALIK J., OSBORNE M. R.: *Methods for Unconstrained Optimization Problems.* American Elsevier, New York, 1968.

80. KOWALIK J. S., RAMAKRISHAN K.G.; *A Numerically Stable Optimization Methods Based on a Homogenous Function.* Submitted to "Mathematical Programming", May, 1975.
81. KREUSER J. L., ROSEN J. B.: *GPM/GPMNLC Extended Gradient Projection Method for Non-linear Constraints.* University of Wisconsin, Madison, 1971.
82. KÜNZI H. P., TZSCHACH M. G., ZENDAR C. A.: *Numerical Methods for Mathematical Optimization.* Academic Press, New York & London, 1968.
83. LASDIN L. S.: *Optimization Theory for Large Systems.* Macmillan, New York, 1970.
84. LEON A.: *A Comparison among Eight Known Optimizing Procedure.* In Recent Advances in Optimization Techniques. A Lavi & T. Vogl (Eds.) John Wiley, N. Y., 1966.
85. LEVITIN E. S., POLJAK B.T.: *Constrained Minimization Methods.* USSR Comp. Math. and Math. Phys.6, 1966, No.5.
86. LOOTSMA F. A.: *Boundary Properties of Penalty Functions for Constrained Minimization.* Philips Res. Repts. Suppl. 3, 1970.
87. LOOTSMA F. A., (Ed.: *Numerical Methods for Non-Linear Optimization.* Academic Press, London, 1972.
88. LOOTSMA F. A.: *A Survey of Methods for Solving Constrained-Minimization Problems Via Unconstrained Minimization.* In F. A. Lootsma (ed.) Numerical Methods for Nonlinear optimization. A. P., London, 1972, 314—348.
89. LOOTSMA F. A.: *The Algol 60 Procedure Minifun for Solving Non-Linear Optimization Problems.* Report 4761, Philips. Research Laboratories, Eidhoven, the Netherlands, 1972.
90. LOOTSMA F. A.: *Penalty Function Performance of Several Unconstrained Minimization Techniques.* Philips Res. Repts. 27 (1972a), 358—385.
91. LOOTSMA F. A.: *The Design of a Nonlinear Optimization Programme for Solving Technological Problems.* In Optimization and Optimal Control, Proceeding of a Conference Held et Oberwolfach, 1974, Bulirsch, R., Oettly, W. Stoer, S. (Eds.), Springer-Verlag, Berlin-Heidelber-New York, 1975.
92. MANGASARIAN O. L.: *Nonlinear Programming.* McGraw-Hill, New York, 1969.
93. MANGASARIAN O.L.: *Dual, Feasible Direction Algorithms.* In: Techniques of Optimization, A. V. Balakrishnan, Ed., Academic Press, New York & London, 1972.,
94. MARTOS B.: *Nonlinear Programming Theory and Methods.* Akadémici Kiado, Budapest 1975.
95. MARUȘCIAC I.: *Metode de rezolvare a problemelor de programare neliniară.* Editura Dacia, Cluj, 1973.
96. MAURIN H.: *Parametrization générale d'un programme linéaire.* Revue Francaise de Recherche Operationelle, 32, 1969.
97. MC CORMICK G. P., MYLANDER W. C., FIACCO A. V.,: *Computer Program Implementing the Sequential Unconstrained Minimization Technique for Nonlinear Programming.* Technical paper RAC-TP-151, Research Analysis Corporation, McLean, Virginia, USA, 1965.
98. MC CORMICK G. P., J. D. PEARSON: *Variable Metric Methods and Unconstrained Optimization.* In Optimization, Fletcher, R., (Ed.), Academic Press, London, 1969.
99. MIELE A., MOSELEY P. E., GRAGG E. E.: *Numerical Experiments on Hestenes' Method of Multipliers for Mathematical Programming Problems.* Rice University, Aero-Astronautics Report No. 85, 1971.
100. MIELE A., LEVY A. V., GRAGG E. E.: *Modifications and Extensions of the Conjugate Gradient — Restoration Algorithm for Mathematical Programming Problem.* JOTA, vol. 7, June, 1971.
101. MIELE A., CRAGG E. E., LEVY A. V.: *Use of the Augmented Penalty Function in Mathematical Programming Problems.* JOTA, vol. 8, August, 1971.
102. MIELE A., COGGINS G. M., LEVY A. V.: *Updating Rules for the Penalty Constant Used in the Penalty Function Method for Mathematical Programming Problems.* Rice University, Aero-Astronautics Report No. 90, 1972.
103. MIELE A., MOSELEY P. E., LEVY A. V., COGGINS G. M.: *On the Method of Multipliers for Mathematical Programming Problems.* JOTA, vol. 10, N0.1, 1972.
104. MIHOC GH., NĂDEJDE I.: *Programare matematică.* vol. I—II. Ed. Științifică, București, 1966.
105. MIHOC GH., ȘTEFĂNESCU A.: *Programare matematică.* Ed. didactică și pedagogică, București, 1973.

106. MURRAY W. III: *Conditioning in Barrier and Penalty Functions Arising in Constrained Non-linear Programming.* The Princeton Mathematical Programming Symposium, August 1967.
107. MURTAGH B. A., SARGENT R.W., H.: *A Constrained Minimization Method with Quadratic Convergence.* In Optimization, Fletcher R., (Ed.), Academic Press, London, 1969.
108. MURTAGH B. A., SARGENT, R.W.H.: *Computational Experience with Quadratically Convergent Minimization Methods.* The Computer Journal, 13,2, 1970.
109. MYERS G.: *Properties of the Conjugate-Gradient and Davidon Mthods.* JOTA, 2,4, 1968.
110. NELDER J. A., MEAD R.: *A Simplex Method for Function Minimization.* The Comp.J. 7, 1964, 308—313.
111. OREN S. S., LUENBERGER D. G.: *The Self-Scaling Variagle Metric Algorithm* ((SSVM) Proc. 5th Hawaii Int. Conf. on Systems Sciences, pp. 130—132, 1972.
112. ORTEGA J. M., RHEINBOLDT W. C.: *Iterative solution of nonlinear equations of several variables.* Academic Press, New York, 1970.
113. PARISOT G. R.: *Résolution numerique approchée du probléme de programmation linéaire par application de la programmation logarithmique.* Thesis, Université de'Lille, 1961.
114. POLAK E.: *Computational Method in Optimization: A Unified Approach.* Academic Press, New York, 1971.
115. POLYAK R. A.: *A General Method for Solving Extremum Problems.* Dokl. Akad. Nauk SSSR, Tome 174—1967.
116. POWELL M.J.D.: *An Efficient Method for Finding the Minimum of a Function or Several Variables without Calculating Derivatives.* The Comp. J. 7, 1964, 155—162.
117. POWELL M.J.D.: *Rank one Methods for Unconstrained Optimization.* In Integer and Nonlinear Programming. Abadie, J. (Ed.), North-Holland, Amsterdam, 1970.
118. POWELL M.J.D.: *A Survey of Numerical Methods for Unconstrained Optimization.* SIAM Rewiew, 12, 1970, 79—97.
119. POWELL M.J.D.: *Recent Advances in Unconstrained Optimization.* Math. Progr. 1, 1971, 26—57.
120. ROCKAFELLER R. T.: *Convex Analysis.* Princeton University Press, Princeton, N. J., 1969.
121. RAMSAY J. O.: *A Family of Gradient Method for Optimization.* The Computer Journal, 13, 4, 1970.
122. RIBIERE G.: *Sur la méthode de Davidon-Fletcher-Powell pour minimisation des fonctions.* Management Science, 16, 9, 1970.
123. ROSEN J. B.: *The Gradient Projection Method.* Part. I, Linear Constraints, J. Soc. Ind. Appl. Math., 8, 1960.
124. ROSEN J. B., SUZUKI S.: *Construction of Nonlinear Programming Test Problems.* Comm. ACM, 8, 1965, 113.
125. ROSEN J. B., MANGASARIAN O. L., RITTER K., Eds.: *Nonlinear Programming.* Academic Press, New York, 1970.
126. ROSENBROCK H. H.: *An Automatic Method for Finding the Greatest or Least Value of a Function.* Computer J., 3, 1960, 175.
127. SARGENT R.W.H.: *Minimization without Constraints.* In M. Avriet, M. J. Rijckaert and D.J.Wilde (eds.) Optimization and Design Prentice-Hall, Englewood Cliffs, New Jersey, 1973, 37—75.
128. SCHERMAN J., MORRISON J.: *Adjustment of an Inverse Matrix Corresponding to Changes in a Given Column or a Given Row of the Original Matrix.* Ann. math. stat., 21, 1949.
129. SHAH B. V., BUEHLER R. J., KEMPTHORNE, C.: *Some Algorithms for Minimizing a Function of Several Variables.* SIAM J. vol. 12, pp. 74—92, March 1964.
130. SHANNO D. F.: *Conditioning of Quasi-Newton Methods for Function Minimization.* Math. of. Comp. 24, 1970, 647—656.
131. SPEDICATO E.: *Stability of Huang's update for the Conjugate Gradient Method.* JOTA, 11, 5, 1973.
132. SPENDLEY W., HEXT G. R., HIMSWORTH, F.R.: *Sequential Application of Simplex Designs.* In Optimization and Evolutionary Operation. Technometircs, 4,1962.
133. STAHA R. L.: *Constrained Optimization Via Moving Exterior Truncation.* Thesis, University of Texas. Austin, 1973.
134. STAHA R. L., HIMMELBLAU D. M.: *Evoluation of Constrained Nonlinear Programming Techniques.* Working paper, Univ. of Texas, Austin, 1973.

135. STOER J.: *On the Numerical Solution of Constrained Least-Squares Problems.* SIAM J. Num. Anal. 8, 1971.
136. TOPIKS D. M., VEINOTT A. F.: *On the Convergence of Some Feasible Direction Algorithms for Nonlinear Programming.* SIAM J. Control 5, 1967.
137. VERCONSTRE A. M.: *Etude comparative des méthodes de minimisation de Fletcher et Powell et de Davidon.* Bulletin E.D.F., Serie C., No.1, 1970.
138. WISMER D. A., (Ed.): *Optimization Methods for Large-Scale Systems... with Applications.* McGraw-Hill, New York, 1971.
139. WOLFE PH.: *Methods of Nonlinear Programming.* In Nonlinear Programming. Abadie, J. (Ed.), North-Holland, Amsterdam, 1967.
140. WOLFE PH.: *Convergence Theory in Nonlinear Programming.* In Optimization, Fletcher, R. (Ed.)., Academic Press, London, 1969.
141. WOLFE PH.: *Review of Nonlinear Programming.* In Integer and Nonlinear Programming. Abadie, J. (Ed.), North-Holland, Amsterdam, 1970.
142. ZADEH L. A., NEUSTADT, L. W., BALAKRISHNAN, A. V. (Eds.): *Computing Methods in Optimization Problems.* Academic Press, New York, 1969.
143. ZADEH N.: *A Note on the Cyclic Coordinate Ascent Method.* Management Science-A., 16, 9, 1970.
144. ZANGWILL W.I.: *Nonlinear Programming: A Unified Approach.* Prentice-Hall, N. J., 1969.
145. ZOUTENDIJK G.: *Nonlinear Programming, Computational Methods.* In Integer and Nonlinear Programming. Abadie J., (Ed.), North-Hollan, Amsterdam, 1970.

QUELQUES RESULTATS DANS LA RESOLUTION DES PROBLEMES DE PROGRAMMATION NON—LINEAIRE A L'AIDE DE L'ORDINATEUR

L'auteur présente une synthèse des résultats obtenus pendant la période 1970—1975 dans la résolution des problèmes de programmation non-linéaire. Il présente, quelques résultats théoriques et surtout, certaines conclusions sur l'efficacité des algorithmes pour optimiser les fonctions de n variables et, en même temps, sur l'efficacité de la résolution des problèmes non-linéaires à restrictions, au moyen de la technique SUMT. On accorde un intérêt spécial aux algorithmes de type mixte.

НЕКОТОРЫЕ РЕЗУЛЬТАТЫ ПОЛУЧЕННЫЕ ПРИ РАЗРЕШЕНИИ ЗАДАЧ НЕЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ ЭВМ

В работе представлен синтез результатов, полученных автором в периоде 1970—1975 гг. при разрешении задач нелинеинного программирования. Вместе с некоторыми результатами теоретического характера представлены, в особенности, некоторые выводы связонные с эффективностью алгоритмов, применяемых для оптимизации функций от n переменных и в связи с этим, — выводы относительно эффективности решения нелинейных задач с ограничениями с помощью метода SUMT. Особое внимание уделено аглоритмам смешанного типа.

13. V. 1977