Description of a Program for Nonlinear Programming: the Centroid Program

J. MOTTL
Geofyzika, Brno, Ječná 29 a, Czechoslovakia

1. INTRODUCTION

Let us consider the problem of non-linear programming in the form

$$g_e(\mathbf{x}) \leqslant 0, \quad e = 1, 2, \ldots, M, \tag{1}$$

$$f(\mathbf{x}) \rightarrow \min, \tag{2}$$

where g and f are non-linear functions of the independent variable $\mathbf{x} = \{x_1, x_2, \ldots, x_t, \ldots, x_n\}$.

2. A SOLUTION BY THE PROCEDURE OF A GRADUAL APPROACH OF THE FEASIBLE REGION

This approach was introduced first in Ref. 1. Let us describe the version followed in Ref. 2.

From functions (1) let the penalty function

$$F = \sum_e g_e^+ \tag{3}$$

be formed, where

$$g_e > 0 \rightarrow g_e^+ = g_e, \quad g_e \leqslant 0 \rightarrow g_e^+ = 0, \tag{4}$$

and by an arbitrary searching program S the lowermost point will be searched for upon the F surface with the ordinate $F = 0$ that defines ($g_e \leqslant 0$, $e = 1, 2, \ldots, M$) the feasible region.

After reaching this feasible point designated as $\mathbf{x}_{j-1}$ from (2) a suitable limitation is made in the form

$$g_{M+1(j)}(\mathbf{x}) = f(\mathbf{x}) + [-f(\mathbf{x}_j) + \bar{\Delta}], \tag{5}$$

where $\bar{\Delta}$ is the chosen positive constant. Let this limitation be introduced into the system forming the penalty function F according to (3), and again the feasible point is being searched for, designated now $\mathbf{x}_{j-2}$. After it is found let it substitute for $\mathbf{x}_j$ into (5), thus forming the new function $g_{M+1(j)}$, etc.

It is thus obvious that the feasible point $\mathbf{x}_j$ is being found, then the f function is raised by $\bar{\Delta}$ and again the feasible point $\mathbf{x}_{j+1}$ is being searched for. The last point from the sequence of feasible points found in this way is then obviously the extreme that is being searched for.

3. THE SEARCHING PROGRAM S

The entire problem is thus reduced, from the geometrical point of view, into a problem of repeated finding of the lowermost point (with $F = 0$) on the F surface.

In the following treatment let us start from the assumption that the penalty function F is formed by one depression of global dimensions. This depression further consists of partial depressions of smaller dimensions (see Fig. 1 for $n = 1$).

In agreement with this assumption we divide the searching process S into two stages. During the first stage of the global searching the lowermost point is found of the approximately medial surface of the global depression (point A). This requires a procedure that, during this search, is not sensitive to undulations of the F surface. After finding point A we pass into a detailed completion of search by the program that is oriented on to a local search, i.e. on to an economical search on non-undulated surfaces. It can be, for instance, an arbitrary gradient program (for a realisation the program ROMO was used).[2] This is motivated by the fact that the global program would work, during this stage, uneconomically.

The first global searching procedure is thus to be defined. Let us choose the following approach.

(1) The entire process during one step j (i.e. reaching the feasible region in the sense of Section 2) will consist of i-steps (iterations).

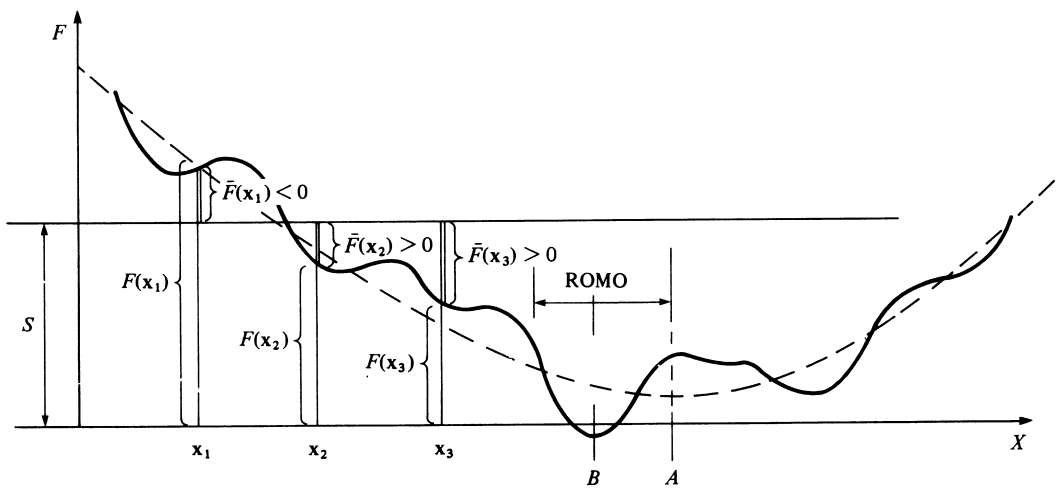

Figure 1

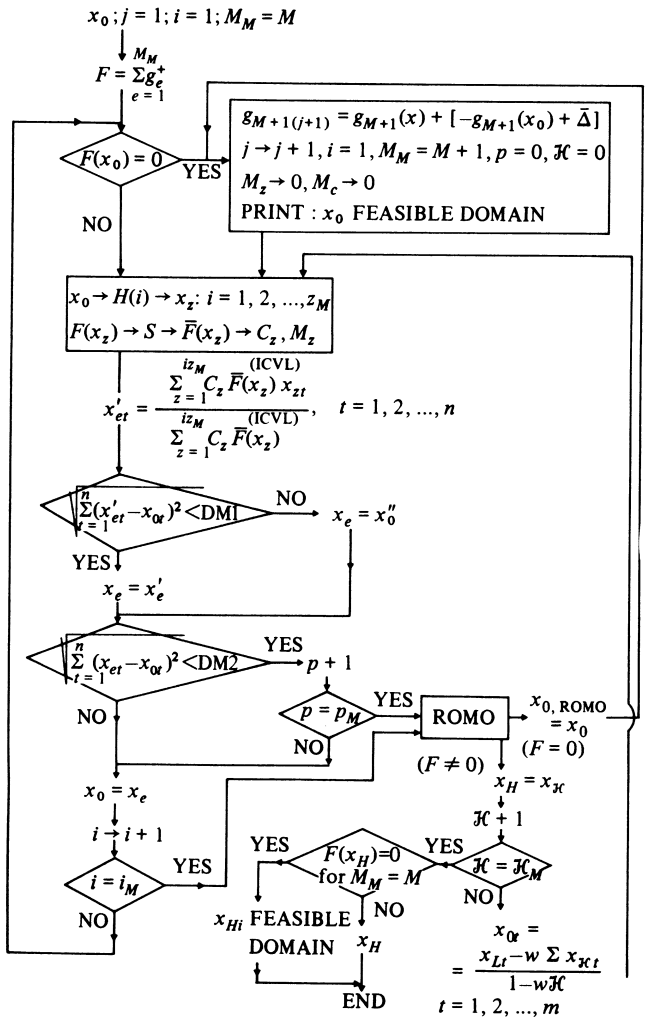

Figure 2

We will start with the $i = 1$ step. Let us choose arbitrarily the starting point $\mathbf{x}_0$ and let us distribute $z_m = 2n+1$ points $\mathbf{x}_z$ around it in such a way that in each direction $t = 1, 2, ..., n$ two points will always be placed on both sides of $\mathbf{x}_0$, their distance from $\mathbf{x}_0$ being $H(i)$. (This designation indicates that H will change as a function of the i-step – it will usually decline.) H is chosen globally as a rule, i.e. $H(i = 1)$ will equal the half of the searching-space dimension that is in agreement with the character of the first step of searching. The maximal number of i-iterations will be $i = i_M$ in limits (i_M is to be specified by the user).

(2) Let us now form a quantity

$$S = \frac{C_S}{iz_m} \sum_{z=1}^{iz_m} F(\mathbf{x}_z), \tag{6}$$

where C_S is a chosen parameter, its value being $C_S \geqslant 1$, F is a penalty function in the sense of (3), iz_m is a number of points $\mathbf{x}_z$ generated during steps 1 to i (in the $i = 1$ step it is thus z_m).

Let us then form quantities

$$\bar{F}(\mathbf{x}_z) = S - F(\mathbf{x}_z). \tag{7}$$

Their geometrical meaning follows from Fig. 1. Finally the centre of balance will be formed from ordinates $\bar{F}(\mathbf{x}_z)$ in the form:

$$x'_{et} = \frac{\sum_{z=1}^{iz_m} C_z \bar{F}(\mathbf{x}_z) x_{zt}}{\sum_{z=1}^{iz_m} \bar{F}(\mathbf{x}_z) C_z} \text{ (ICVL)}, \quad t = 1, 2, \ldots, n. \tag{8}$$

(3) After finding point $\mathbf{x}'_e(i = 1)$ the step $i = 2$ takes place (generally $i \to i+1$) during which we will try to get the found position of $\mathbf{x}'_e(i)$ more precise (in such a sense that it would approach the lowermost point of the global depression F). The ICVL parameter must be chosen as

Table 1

Example	*IM*	*IP*	ICVL	CVL	CS	VH	*C1(I)*	*DELTP*	*XO*	*XE*	*XFD*
							Goldstein and Price				
1	20	20	1	0.2	1.6	1.5–0.2	1–1	10^4	−1, 1	0.66, −1.12	−0.0025, −1.003
2	20	20	1	0.2	1.6	1.5–0.2	1–1	10^4	2, 2	−0.029, −0.43	−0.006, −0.992
3	20	20	1	0.01	1.4	1.8–0.3	3–1	10^4	−2, −2	0.47, −1.08	0.0045, −0.909
4	20	20	1	0.2	1.6	1.5–0.2	1–1	10^4	2, −2	1.25, −0.37	0.0016, −1.006
5	20	20	1	0.2	1.6	1.5–0.2	1–1	10^4	1, 1	0.95, −1.06	−0.01, −1.014
6	20	20	1	0.2	1.6	1.5–0.2	1–1	10^4	−2, −2	−0.21, 0.28	0.0095, −1.008
							Branin				
1	20	20	1	0.2	1.6	6–0.2	2–1	1	−5, 15	0.12, 14.24	0.216, 14.96
2	20	20	1	0.2	1.2	6–0.2	10–1	1	10, 10	−0.03, 5.7	0.069, 6.81
3	20	20	1	0.2	1.2	6–0.2	10–1	1	5, 5	−0.0007, 5.83	−0.001, 5.99
4	20	20	1	0.2	1.2	6–0.2	10–1	1	0, 0	0.0059, 5.99	0.0059, 5.99
							Shekel				
1	20	10	9	1	1.05	2.6–0.3	10–1	2	5, 5, 5, 5	3.94, 4.004, 3.996, 3.904	3.99, 4.00, 3.99, 3.99
2	20	10	9	1	1.05	2.6–0.3	10–1	2	8, 6, 2, 0	4.02, 4.03, 3.97, 3.98	3.99, 3.99, 4.00, 4.00
3	20	10	9	1	1.05	4–0.4	10–1	2	8, 6, 2, 0	3.98, 3.92, 4.02, 4.02	3.99, 4.00, 3.99, 4.00
4	20	10	9	1	1.05	2.6–0.3	10–1	2	0, 0, 8, 0	1.06, 1.06, 0.96, 1.06	0.997, 0.999, 1.002, 0.999
							Hartmann				
1	20	10	9	1	1.05	2.6–0.7	1–1	2	0, 0, 1	0.059, 0.56, 0.67	0.116, 0.553, 0.852
2	20	10	9	1	1.05	2.6–0.7	1–1	2	1, 0, 1	0.726, 0.515, 0.679	0.1004, 0.552, 0.852
3	20	10	9	1	1.05	2.6–0.7	1–1	2	0, 1, 1	0.0394, 0.432, 0.699	0.117, 0.560, 0.853

Table 2

```
  N     M     IM      KPM      KAPM      IP     ICVL
  2     4     20      3        1         20     1

DELTP          CVL            A              DM 1           DM 2           CZU            CS
  0.100000E+05   0.200000E+00   0.100000E+01   0.150000E+01   0.200000E+00   0.000000E+00   0.160000E+01

Vector VH
  0.150000E+01 0.140000E+01 0.130000E+01 0.120000E+01 0.110000E+01 0.100000E+01 0.120000E+01 0.140000E+01
  0.120000E+01 0.100000E+01 0.900000E+00 0.800000E+00 0.700000E+00 0.600000E+00 0.800000E+00 0.600000E+00
  0.500000E+00 0.300000E+00 0.200000E+00 0.200000E+00

Vector XO
 -0.100000E+01 0.100000E+01

Vector C1
  0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01
  0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01
  0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01

  KMV     KMC     IZM     KAPMR
  10      4       6       0
H              HVL            HVLC           D              Ømega
  0.120000E+00   0.150000E+00   0.200000E-01   0.300000E+00   0.200000E+00

Vector XL                      Vector XE                      FP = 0.147188E+04
    0.000000E+00   0.000000E+00    0.729797E+00   -0.107127E+01    1-2   -0.0026   -1.0034

Feasible domain                Vector XE                      FD = 0.151167E+04
   -0.100000E+01   0.100000E+01    0.664570E+00   -0.112653E+01    1-2   -0.2184   -1.2117

Feasible domain                FO = 0.346684E+04  Initial vector XO  FG = 0.148861E+04
    0.500000E+00   0.100000E+01    1-2    0.6646   -1.1265          1-2   -0.2226   -1.1218

Vector XE                      FG = 0.184944E+04              FP = 0.147188E+04
    0.848572E+00  -0.321094E-01    1-2    0.4440   -1.0613          1-2    0.0063   -0.9986

Vector XE                      FP = 0.147320E+04              FD = 0.148912E+04
    0.902176E+00  -0.516874E+00    1-2    0.0508   -0.9451          1-2    0.2705   -0.8564

Vector XE                      FD = 0.149517E+04              FG = 0.148316E+04
    0.874667E+00  -0.814761E+00    1-2   -0.2369   -0.8600          1-2    0.1810   -0.8471

Vector XE                      FG = 0.148265E+04              FP = 0.147188E+04
    0.806532E+00  -0.988190E+00    1-2   -0.0870   -0.8543          1-2   -0.0003   -1.0044

                               FP = 0.147315E+04              FD = 0.150465E+04
                                    1-2    0.0550   -0.9478          1-2   -0.2270   -1.2009

                               FD = 0.158982E+04              FG = 0.148937E+04
                                    1-2    0.3055   -1.1128          1-2   -0.2265   -1.1309

                               FG = 0.147304E+04              Vector XH
                                    1-2   -0.0485   -1.0477

                                                              Feasible domain
                                                                -0.256074E-02  -0.100337E+01
```

an odd number. C_z is the weight parameter. The meaning of both parameters will be explained later.

This precision will obviously grow with the increase of number of points $\mathbf{x}_z$ that will be considered in (8). It will further be influenced by a choice of a position in the $i+1$ step newly added points $\mathbf{x}_z$.

We will proceed so that the new point $\mathbf{x}_0(i+1)$ will be made identical with the found point $\mathbf{x}'_e(i)$, and further $z_m = 2n+1$ new points $\mathbf{x}_z$ around it that will be added to the set of points $\mathbf{x}_z$ from previous steps i to (8) (and at forming S according to (6) as well). In (8) and (6) in the $i+1$ step the summation will thus be from $z = 1$ to iz_m: the maximal number of considered points will be $i_p z_m$ in limits – i_p is specified by the user – the earlier generalised points are to be left out. As far as $H(i+1)$ is concerned it is recommended to choose $H(i+1) < H(i)$: gradually larger respect is thus achieved to the more detailed course of the F surface.

It follows from definitions (6), (7) and (8) that points of coordinates $\mathbf{x}'_e(i)$ should have a tendency to get closer to the region being examined with the smallest ordinates of the penalty surface F (see again the geometrical interpretation according to Fig. 1) as the number of steps i increases.

In order to further assist this tendency weight coefficients are to be introduced into the expression (8); through those a differentiated weight is given to information F from various points $\mathbf{x}_z$.

These coefficients C_z can be defined by differently chosen hypotheses. A common feature of these hypotheses should be the fact that a relatively larger weight should be given to information from those points $\mathbf{x}_z$ where this

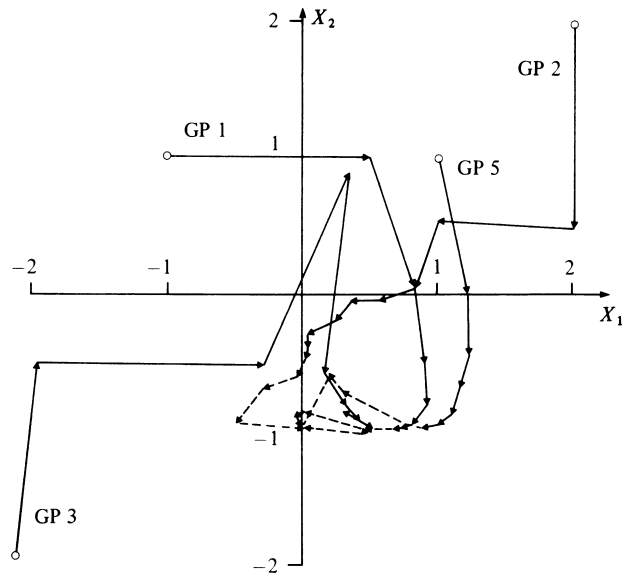

Figure 3

or further independent information indicates the relatively higher probability of these points being situated closer to the region being searched for with the smallest ordinates of the penalty surface F.

Among this information $\bar{F}(\mathbf{x}_z)$, $F(\mathbf{x}_z)$ and number i of the iteration step could all be included.

To illustrate, let us present examples of possible hypotheses of this type.

(*a*) C_z will be relatively larger at points $\mathbf{x}_z$ where more equations $g_e(\mathbf{x}_z) \leqslant 0$ are satisfied (i.e. for more e than at other points) it obviously indicates a higher probability of these points being situated closer to the feasible region being searched for that is defined by the following relation:

$$g_e(\mathbf{x}_z) \leqslant 0, \quad e = 1, 2, \ldots, M.$$

(*b*) C_z will be relatively larger at those points $\mathbf{x}_z$ where ordinates $F(\mathbf{x}_z)$ are smaller.

(*c*) At all points that were generated by later iteration steps i, C_z will be relatively larger than at previous steps i.

For a practical procedure the hypothesis was chosen according to (*b*) and (*c*) that leads to the expression for C_z in the form:

$$C_z = \frac{C1(i)}{F^{\tilde{c}}_{(\mathbf{x}_z)} + a}. \tag{9}$$

Constants $C1(i)$: $i = 1, 2, \ldots, i_M$; $\tilde{c}$, a are to be chosen by the user. The set of constants $C1(i)$ enables realisation of point (*c*) hypothesis, i.e. that the weight parameters C_z will be greater at $\mathbf{x}_z$ points generated in later i-steps contrary to those C_z values at $\mathbf{x}_z$ points generated in previous i-steps: this is why $C1$ is a function of i. They are to be chosen arbitrarily (as the positive numbers), but with decreasing magnitudes as i increases (the selected method of inserting into the memory requires the input in a reversed order, though). The constant ICVL in (8) permits one to relatively intensify the influence of greater values F. It must be chosen as an odd number (in the opposite case the originally negative values of $\bar{F}$ would be transformed into positive ones).

Constant $\tilde{c}$ is an exponent. If it is chosen larger the increasing measure of respecting the point (*b*) hypothesis is being provided. Constant a is to prevent the expression (9) at $F(\mathbf{x}_z) \to 0$ from becoming infinitely large, and is chosen as an arbitrary small positive number.

As far as the constant C_S in expression (6) is concerned, it is obvious that the impact of interest will be for $C_S = 1$. For decreasing C_S the process is obviously accelerated (i.e. distances $\mathbf{x}'_e(i)$, $\mathbf{x}'_e(i+1)$ are getting larger – there is, of course, some possibility for oscillation tendencies). For increasing C_S the opposite is true.

At this stage it will be useful to present the complete block scheme of the procedure.

It is obvious from Fig. 2 that we continue performing iterations i maximally to the limiting value of i_M (specified by a particular user) when the stage of detailed searching begins by means of the ROMO method. This stage may happen eventually even earlier if the distance of two sequentially found new points $\mathbf{x}'_e(i)$ and $\mathbf{x}'_e(i+1)$ will be smaller than the chosen small limit of the DM 2. (It thus indicates an obvious slowing down of the iteration process typical for the proximity of the bottom of the depression.)

Besides this, on the other hand the maximum distance of $\mathbf{x}'_e(i)$ and $\mathbf{x}'_e(i+1)$ points is also limited on DM 1 in order not to generate unwanted oscillations of iteration steps again. (In case the distance of $\mathbf{x}'_e(i+1)$ from $\mathbf{x}'_e(i)$ is larger than DM 1, new $\mathbf{x}''_e(i+1)$ is to be chosen on the straight line connecting $\mathbf{x}'_e(i+1)$ with $\mathbf{x}'_e(i)$ at the distance DM 1 from $\mathbf{x}'_e(i)$; thus:

$$\mathbf{x}''_{et}(i+1) = \mathbf{x}_{et}(i) + \alpha F'_t, \quad t = 1, 2, \ldots, n, \tag{10}$$

where

$$F'_t = \mathbf{x}'_{et}(i+1) - \mathbf{x}'_{et}(i); \quad \alpha = \frac{\text{DM 1}}{\sqrt{\sum_{t=1}^{n} (F'_t)^2}}.$$

The program finally prints the vector $\mathbf{x}_H$ (see Fig. 2). This is a point with the lowest value F of the barrier surface. It is tested on feasibility from the point of view of limitations (1) only (i.e. without limitation (5)). By this process the accuracy of calculation is preserved even if greater values of DELTP are chosen.

4. VARIATIONS

The question arises whether the program could be at least approximately modified for searching the more general character of the F surface than was assumed in Section 3 (Fig. 1), i.e. the F surface not with one only but with a limited (smaller) number of depressions of a more global character (thereafter more generally undulated again). Let us use the procedure applied in Ref. 3 when individual global depressions will be sequentially being searched through. The first $\mathbf{x}_0$ is to be chosen in the centre of the searching region. If success is not achieved from this point (i.e. neither by the global or the detailed ROMO stage) the point $\mathbf{x}_{H,\,\text{ROMO}} = \mathbf{x}_{\mathscr{H}}$ is the result that from the geometrical point of view represents the achieved lowermost point of the unsuccessful $\mathscr{H}$-depression (the global one). In this case the new starting point $\mathbf{x}_0 = \mathbf{x}_{\mathscr{H}+1}$ is chosen for the global searching as the centre of gravity of the entire searching region (of the H dimension) weakened by cavities (their dimensions being $H_0 < H$: their ratio $\omega = H_0/H$ is chosen usually between 0.2 and 0.25). These cavities are in the vicinity of all $\mathscr{H}$ achieved lowermost points of

Table 3

N	M	IM	KPM	KAPM	IP	ICVL
1	2	20	2	1	20	1

DELTP	CVL	A	DM 1	DM 2	CZU	CS
0.150000E+02	0.100000E+01	0.100000E+01	0.100000E+02	0.150000E+00	0.000000E+00	0.120000E+01

Vector VH
0.100000E+02 0.100000E+02 0.900000E+01 0.800000E+01 0.900000E+01 0.800000E+01 0.700000E+01 0.600000E+01
0.800000E+01 0.700000E+01 0.600000E+01 0.500000E+01 0.400000E+01 0.300000E+01 0.200000E+01 0.100000E+01
0.100000E+01 0.800000E+00 0.500000E+00 0.200000E+00

Vector XO
0.200000E+01

Vector C1
0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01
0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01
0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+01

KMV	KMC	IZM	KAPMR
5	3	4	0

H	HVL	HVLC	D	Ømega
0.100000E+01	0.100000E+01	0.200000E+00	0.200000E+01	0.200000E+00

Vector XL
0.100000E+02

Feasible domain
0.200000E+01

Vector XE
0.117424E+02

Vector XE
0.107255E+02

Vector XE
0.102632E+02

Vector XE
0.110375E+02

Vector XE
0.107675E+02

Vector XE
0.108411E+02

Vector XE
0.117235E+02

Vector XE
0.124955E+02

Vector XE
0.119103E+02

Vector XE
0.113679E+02

Vector XE
0.118476E+02

Vector XE
0.123304E+02

Vector XE
0.125891E+02

Vector XE
0.129257E+02

Vector XE
0.134454E+02

Vector XE
0.138437E+02

Vector XE
0.141326E+02

Vector XE
0.143552E+02

Vector XE
0.145340E+02

Vector XE
0.146743E+02

FO = 0.178434E+01
1–1 14.6743

FG = 0.150951E+01
1–1 14.4743

FP = 0.150951E+01
1–1 14.4743

FD = 0.114373E+02
1–1 12.4743

FG = 0.150951E+01
1–1 14.4743

FP = 0.150951E+01
1–1 14.4743

FD = 0.114373E+02
1–1 12.4743

FG = 0.150951E+01
1–1 14.4743

FP = 0.150951E+01
1–1 14.4743

FD = 0.114373E+02
1–1 12.4743

FG = 0.150951E+01
1–1 14.4743

Vector XH

Feasible domain
0.144743E+02

Exit 33377

unsuccessful global depressions $\mathbf{x}_{\mathscr{H}}$ reached in the particular step j. If $\mathbf{x}_L$ is the centre of the searching region the expression for the new starting point in the $\mathscr{H}+1$ step is as follows:

$$\mathbf{x}_{0,t} = \frac{\mathbf{x}_{L,t} - \omega \sum_{\mathscr{H}} \mathbf{x}_{\mathscr{H},t}}{1 - \omega\mathscr{H}}, \quad t = 1, 2, \ldots, n. \qquad (11)$$

The maximum number of iterations is $\mathscr{H}_M$.

5. TESTING OF THE METHOD

Four test problems from Ref. 7 were chosen for testing, and several arbitrarily selected starting points were used for each.

The course of the tests is presented in Table 1, where: *IM* (i_M) is the limiting number of iterations i, *IP* (i_p) is the limiting number of Z_m of points x_z preserved in the memory during the calculation of (8), ICVL is the constant in (8) (an odd number has to be chosen); CVL ($\tilde{C}$) is the constant in (9); CS is the constant in (6); VH are individual distances between points $\mathbf{x}_Z$ and the central point $\mathbf{x}_0$ in the direction of individual coordinate axes. The set VH($i = 1$) to VH(i_M) is chosen with the declining character; $C1$ is the set of $C1(i)$ in (9) for $i = 1$ to i_p; DELT is the constant $\bar{\Delta}$ in (5), XO is the starting point. [Note: in Table 1 the first and last values are given only of sequences VH(i) and $C1(i)$.] XE is the final point of iterations in the sense of (8), i.e. the result of the global phase of search; XFD is the final extreme made more precise by the local search by means of Ref. 2 (in Fig. 3 this

phase is designated by the dashed lines). Entries of this subroutine are KMV, KMC, IZM, KAPMR, H, HVL, HVLC, D, Ømega. Their meaning is given in Ref. 2.

(1) Problem of Goldstein and Price (Ref. 7, page 14).

The search domain is $2 \leqslant x_{1,2} \leqslant 2$. Four local minima exist, the global minimum being at $(0, -1)$. Results of the tests are presented in Table 1. The corresponding course of the processes is illustrated in Fig. 3 and the printout of one solution is in Table 2.

The problem is characterised by the fact that functional values reach the order of 10^4–10^5 for small changes of the independent variable $(-2, 2)$. Under these conditions the following appeared suitable to be set: CVL < 1, ICVL = 1, CS > 1.

(2) Problem of Branin (Ref. 7, page 13).

The search domain is $-5 \leqslant x_1 \leqslant 10$, $0 \leqslant x_2 \leqslant 15$. Altogether, three minima exist. Results are also presented in Table 1. The character of the problem is similar to that of problem (1).

(3) Problem of Shekel (Ref. 7, page 12).

The search domain is $0 \leqslant x_j \leqslant 10, j = 1,2,3,4$. Results are presented in Table 1. This problem is characterised by the barrier function of a small, nearly constant value everywhere except in points of local minima. The global minimum is at (4; 4; 4; 4). In this case it appeared suitable to set CVL = 1, ICVL $\gg$ 1, CS $\sim$ 1, with respect to the structure of (6), (8) and (9).

(4) Problem of Hartmann (Ref. 7, page 13).

The search domain is $0 \leqslant x_j \leqslant 1$, $j = 1, 2, 3$. Results are displayed in Table 1. This problem's character is similar to that of problem (3).

All four selected problems are of an extreme character in comparison with common cases – in problem (1) extremely large functional values appear as well as their extremely large differences, while in problem (3) very small changes of functional values appear – that necessarily required the differential choice of control parameters CS, ICVL and CVL, which most influenced the convergence of the process.

Note. The parameters ICVL and CVL were introduced additionally to cope with the possible occurrence of problems of extreme kind. They will not be necessary for general cases. It was found that an increase of CS led to a more stable but slower process and vice versa.

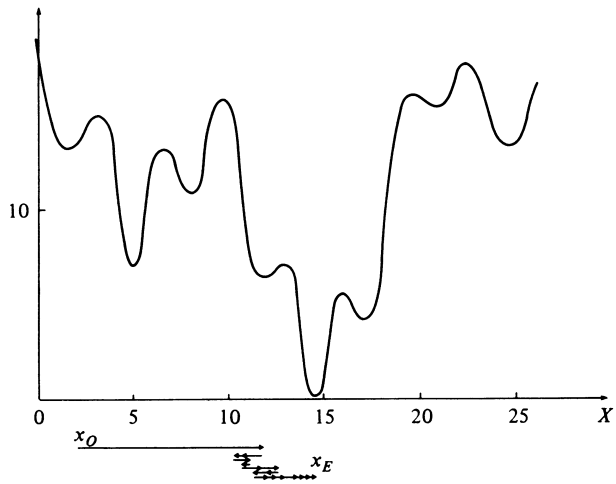

Figure 4

In the majority of cases the operating point approached the desired extreme in the course of the global phase of search. By means of the local search this extreme was then achieved in all cases – even in those ones when the remaining undulations were in its way. This property of the subroutine[2] follows from its initially apparent non-economy, i.e. that the search in the direction of gradient does not terminate at the moment the value of the penalty function starts to increase but that it continues through KMV chosen steps.

As far as the achievement of the global extreme is concerned, in problems (3) and (4) a relatively larger sensitivity manifested itself in the choice of the starting point, which was related to the fact that these problems differ relatively most from the assumed character of functions in Section 3, i.e. one global 'depression' with local undulations and furthermore with very small changes of functional values. Here, it would be necessary to increase the number of points $\mathbf{x}_z$, i.e. IP.

Finally, let me present an illustration problem; its character approaches the assumed one (a global valley with depressions), that intuitively illustrates the method.

REFERENCES

1. J. Huard, Resolution of mathematical programming with nonlinear constraints by the method of centres. In *Nonlinear Programming* (edited J. Abadie). North-Holland, Amsterdam (1967).
2. J. Mottl, Description of a program for nonlinear programming. *The Computer Journal* **22**, 3, 256–261 (1979).
3. J. Mottl, Description of a program for solving problems of nonlinear programming. *The Computer Journal* **21**, 3, 263–269 (1978).
4. J. Huard, A method of centres by upper bounding functions with applications in nonlinear programming (edited J. B. Rosen and O. L. Mangasarian). Academic Press, New York (1970).
5. H. Kunzzi, H. Tcharch and C. Zahnder, *Numerical Methods of Mathematical Optimization*. Academic Press, New York (1968).
6. M. Powell, *The Computer Journal* **7**, 155 (1964).
7. *Toward Global Optimization*, vol. 2. Edited by L. C. W. Dixon and G. Szöge. North-Holland, Amsterdam (1978).