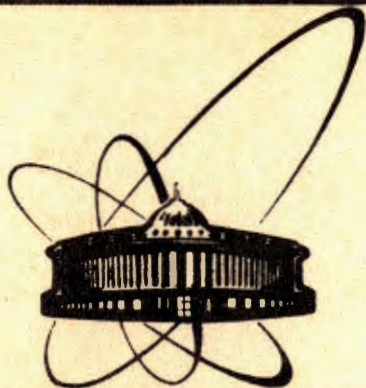


92-252



СООБЩЕНИЯ
ОБЪЕДИНЕННОГО
ИНСТИТУТА
ЯДЕРНЫХ
ИССЛЕДОВАНИЙ
ДУБНА

É11-92-252

V. P. Gergel*, V. A. Grishagin*, V. V. Kukhtin,
E. A. Rogatneva*, R. G. Strongin*, I. N. Vysotskaya*

A SOFTWARE SYSTEM FOR MULTIEXTREMAL
OPTIMIZATION

*Scientific Research Institute of Applied
Mathematics and Cybernetics, Nizhnij Novgorod
State University, Russia

1992

The system for multiextremal optimization (SYMOP) described here is a program package [1] oriented for using in the computer-aided design (CAD) as some invariant software tool of making efficient design decisions. Besides, it is possible to use this system independently for solving various optimization problems and systems of nonlinear algebraic equations.

The SYMOP package is designed to solve numerically the multicriteria multiextremal multidimensional optimization problems with complicated constraints. These problems can be considered as the mathematical models of decision making problems in various applications. For example, these may be identification problems (defining the functional dependencies by experimental data), approximation problems (choosing the optimal parameters of given functions), etc.

As we can see the SYMOP system differs from packages of similar purpose. On the one hand, the SYMOP is oriented upon solving complicated decision-making problems. On the other hand, a wide set of optimization problems can be solved by this system. Also, the decision making process performed by the SYMOP system is similar to the techniques used in the CAD to choose an optimal decision.

The system written in FORTRAN-IV has been implemented on the IBM and VAX type ^{of} computers. It needs 200 Kb of internal and 2 Mb of external memories as minimum.

1. The purpose and fields of application

This package can be applied to the class of general multicriteria optimization problems characterized by the following:

- model functionals can depend on model parameters nonmonotonously (multiextremally); the efforts needed for functional computations can be quite substantial:

- the optimal design decision can be defined by several (usually antagonistic) characteristics subdivided into two categories - the criteria of object efficiency and the criteria of fulfilling the assigned performances usually called constraints;

- the category of characteristics can vary during an optimal decision search (the efficiency criteria are converted into constraints and, vice versa, the constraints are converted into criteria) in accordance with changes of the design decision notion.

In complexity the specific optimization problems may sufficiently differ in the number of criteria and constraints, parameters to be varied and in the computational complexity of model functionals. But there is a common property in all these problems, they are global [2-4].

1.1 A mathematical model of optimization object. An object model is defined through a vector-function $W(y)$ where each coordinate function is positive

$$W_i(y) \geq 0, \quad 1 \leq i \leq n, \quad (1)$$

and a decrease of its values corresponds to a better decision.

The value of the vector parameter y describing a design decision has to belong to the region of the parameter variation

$$D = \{ y \in R^N : a_i \leq y_i \leq b_i, \quad 1 \leq i \leq N \}. \quad (2)$$

This region D is a N -dimensional hyperinterval.

A specific kind of functionals of the vector-function $W(y)$ and a boundary of the region D are defined through the nature of the objects being designed. The SYMOP system is tailored to some specific object through adding the associated programs to compute the values of $W_i(y)$, $1 \leq i \leq N$.

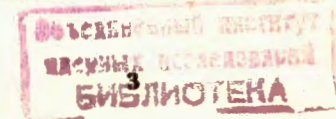
1.2. The conditions of the design decision efficiency. When choosing an optimal decision, various requirements are assigned to the values of coordinate functions $W_i(y)$, $1 \leq i \leq n$, from (1). Some of them need to be minimized. By declaring the set F of numbers of such functions one can define criteria of object efficiency:

$$f(y) = (f_1(y), \dots, f_n(y)) = W_F(y), \quad (3)$$

where

$$f_j(y) = W_{j_i}(y), \quad j_i \in F.$$

And for the remaining part of the coordinate functions $W_i(y)$, $i \in F$, there may be assigned the sufficient efficiency levels (admissions). Then the conditions of admissibility for the decision



y can be given through inequalities

$$g_j(y) = W_j(y) - q_j \leq 0, \quad j_1 \in G = \{1, \dots, n\} \setminus F, \quad 1 \leq j \leq m, \quad m = n - s. \quad (4)$$

The inequalities (4) define a *feasible region of design decision*

$$Q = \{ y \in D: q(y) \leq 0 \}$$

The above procedure of stating a general multicriteria problem is to a large extent defined by the experience gained in designing the like objects. Choosing some unique statement of the optimization problem may come across many difficulties and, besides, in the course of searching for an optimal decision the idea of an efficient variant may undergo some changes. For convenience, all basic controlling actions of the user are put together and shown in Fig 1. At this, all the actions under the double hyphen line do not interrupt acquisition of data to be used in the optimization algorithms.

2. Algorithms for solving a general multicriteria problem

In the SYMOP system the methods for estimating efficient decisions are based upon successively reducing an initial multidimensional multicriteria problem with nonlinear nonconvex constraints to a set of scalar one-dimensional problems. This technique is secured by convolution schemes reducing the vector criteria to scalar ones [5-6], by special modifications of penalty function method [7] and by the dimension reducing mappings [4].

The above approach has been studied theoretically [7-8] and provides a convergence to efficient decisions only. A formal description of each reducing scheme used within this approach is given below.

2.1. Aggregating the efficiency criteria. The criteria $f_i(y)$, $1 \leq i \leq s$, from (3) are usually antagonistic. To characterize the various decisions of the multicriteria optimization problem we make use of the concept of the point $y^* \in Q$ that cannot be improved

$$\forall y \in Q: f_i(y) \leq f_i(y^*), \quad 1 \leq i \leq s \rightarrow f_i(y) = f_i(y^*), \quad 1 \leq i \leq s.$$

This is an *efficient (Pareto's optimal) decision* [5-6] and it may be considered to be a *partial decision* for the multicriteria

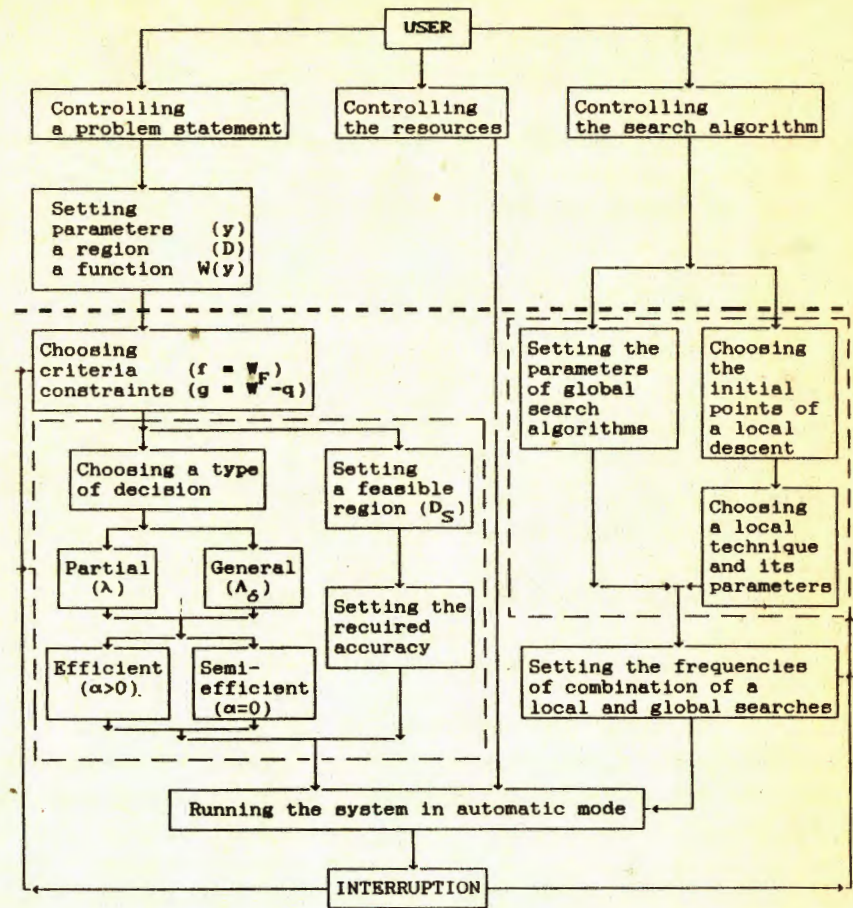


Fig. 1. The system-user interaction in the course of solving a general multicriteria problem

problem. Then the set $\Pi(Q) \subset Q$ of all efficient points is accepted to be the *general decision* of this problem.

Note that in addition to the above efficient points there appear points $y^1 \in \Pi^1(Q)$ that cannot be concurrently improved for

each partial efficiency criterion $f_i(y)$, $1 \leq i \leq s$, i.e.

$$\forall y \in Q \rightarrow \exists i, 1 \leq i \leq s : f_i(y) \geq f_i(y^i).$$

These are semi-efficient decisions and it is known that $\Pi(Q) \subset \Pi^1(Q)$.

The efficient decision $y^* \in Q$ can be calculated by using the convolution scheme [5-6]. In this case

$$y^* = y_\lambda^* (\omega) = \arg \min_{y \in Q} \phi_\lambda(y), \quad (5)$$

where

$$\phi_\lambda(y) = \max \lambda_i f_i(y) + \alpha \sum_{i=1}^s f_i(y),$$

$$\lambda \in \Lambda = \{ \lambda \in R^N : \sum_{i=1}^s \lambda_i = 1, \lambda_i \geq 0, 1 \leq i \leq s \}.$$

The positive coefficient α has to be sufficiently small (if $\alpha=0$, then the point y^* from (5) is a semi-efficient decision). The coordinates λ_i , $1 \leq i \leq s$, of the vector $\lambda \in \Lambda$ may be considered as measures of partial criterion importance.

The decision of the scalar optimization problem (5) is a partial decision of the multicriteria optimization problem corresponding to the weight vector λ . If we take all vectors $\lambda \in \Lambda$, then we may obtain the general decision

$$\Pi(Q) = \bigcup_{\lambda \in \Lambda} y_\lambda^*.$$

As an approximate estimate of the general decision there is taken a finite set of the points $\Pi_\delta(Q) \subset \Pi(Q)$ calculated by solving the problem (5) for each vector λ from some assigned δ -grid $\Delta_\delta \subset \Lambda$.

2.2. Decomposing the constrained optimization problems. Through the use of the penalty function technique [7], the problem (5) may be reduced to the unconstrained optimization problems

$$\min_{y \in D} F(y), \quad F(y) = \phi_\lambda(y) + \beta P(y), \quad (6)$$

where $\beta > 0$ is a penalty coefficient and $P(y)$ is a penalty function. In this case, $F(y)$ may be of the form

$$P(y) = \sum_{l=1}^m \gamma_l Q_l(y),$$

$$\gamma \in \Gamma = \{ \gamma \in R^m : \sum_{l=1}^m \gamma_l = 1, \gamma_l > 0, 1 \leq l \leq m \},$$

$$Q_l(y) = |g_l(y)| + g_l(y), \quad 1 \leq l \leq m.$$

The penalty coefficient β has to be sufficiently large (practically it may be obtained by solving problem (6) with several

gradually increasing values of β). The coefficients γ_l , $1 \leq l \leq m$, balance the partial constraints $g_l(y)$, $1 \leq l \leq m$, in the neighborhood of the current estimation of the efficient decision [7-8].

2.3. Reducing to the one-dimensional problems. It is known that the segment $[0, 1]$ of the real axis x can be uniquely and continuously mapped onto the N -dimensional hyperinterval D from (2). The mappings of this kind are called Peano space filling curves or Peano curves. If the minimizing function $F(y)$ from (6) is continuous, then the continuity of the curve $y(x)$ will provide the following equality

$$\min_{y \in D} F(y) = \min_{x \in [0, 1]} F(y(x)) \quad (7)$$

and then the multidimensional problem of minimizing the function $F(y)$ will be reduced to the problem of minimizing the one-dimensional function $F(y(x))$. If the function $F(y)$ is of the Lipschitz type, then the function $F(y(x))$ will satisfy the Gelder condition with index N^{-1} , i.e.

$$\forall x_1, x_2 \in [0, 1] \rightarrow |F(y(x_1)) - F(y(x_2))| \leq L |x_1 - x_2|^{1/N}$$

(in particular, $L^1 = 4L\sqrt{N}$ where L is the Lipschitz constant of $F(y)$). The algorithms of approximate computations of images $y(x)$ are discussed in [4].

2.4. The method of solving the one-dimensional problems. To minimize the one-dimensional problem from the right part of (7), we use a global search algorithm based on informational-statistical methods [4]. Mainly these methods are characterized by the fact that they produce the minimizing sequences converging to the optimal points only.

A brief description of the multidimensional general algorithm of global search (the MGAGS algorithm) is given below.

According to this algorithm, the first two calculations (or trials) of the function values $F(y(x))$ are performed in the points $x^0 = 0$ and $x^1 = 1$. The point x^{k+1} of the next trial is selected in the following way:

1) order the points x^0, \dots, x^k of the previous trials by an ascending value, i.e.

$$0 = x_0 < x_1 < \dots < x_k < \dots < x_k = 1;$$

2) find the values

$$M = \max_{1 \leq i \leq k} |z_i - z_{i-1}| / \rho_i, \quad m = \begin{cases} 1, & M = 0, \\ rM, & M > 0, \end{cases} \quad (8)$$

where $z_i = F(y(x_i))$, $0 \leq i \leq k$, $\rho_i = (x_i - x_{i-1})^{1/N}$, $1 \leq i \leq k$, and the number $r > 1$ is a parameter of the method;

3) calculate the characteristics $R(i)$ for the intervals (x_{i-1}, x_i) , where

$$R(i) = m\rho_i + (z_i - z_{i-1}) / m\rho_i - 2(z_i + z_{i-1});$$

4) find among the intervals (x_{i-1}, x_i) , $1 \leq i \leq k$, the interval (x_{i-1}, x_i) with maximum characteristic;

5) set

$$x^{k+1} = \frac{x_i + x_{i-1}}{2} - \frac{\text{sign}(z_i - z_{i-1})}{2r} \left\{ \frac{r(z_i - z_{i-1})}{m} \right\}^N$$

where $\text{sign}(u)$ denotes the sign of u .

The stop condition terminates the optimization process, if $(x_i - x_{i-1}) < \varepsilon$. The value t is computed in the above step 4, and $\varepsilon > 0$ is the given accuracy requirement. In addition, the optimization may terminate if either a given number of values for the vector-function $W(y)$ is calculated or the needed extremum estimation

$$z_k^* = \min (F(y(x_i)) : 0 \leq i \leq k)$$

is obtained.

General converging capabilities of the MGAGS algorithm may be summarized by the below statement (see [4]).

Theorem. If the MGAGS algorithm is used for minimizing the Lipschitz function $F(y)$ (with constant L), then for any accumulation point y^i of the minimizing sequence $\{y^k\}$ generated by this algorithm the following will be true:

1) if there is another accumulation point y'' of the sequence $\{y^k\}$, then $F(y^i) = F(y'')$, i.e. a simultaneous convergence to the different function values is impossible and, hence, if minimizing function is not a constant the method will produce a nonuniform net in the region D from (2);

2) $z^k = F(y^k) \geq F(y^i)$, $k = 1, 2, \dots$, i.e. the algorithm cannot generate convergence to the points, where the function value exceeds the result of some realized trial;

3) if at some step we obtain $m > 16L\sqrt{N}$, m from (8), then y^i will be a global minimum point of $F(y)$ and, moreover, the set of all accumulation points of the sequence $\{y^k\}$ will coincide with the set of global minimum points for $F(y)$.

2.5. The information support of a decision making process. We would like to emphasize that solving a multiextremal problem is

usually an extremely hard job. Here we suggest the following way to solve a set of them.

Solving the problems (7) means calculating the values of the vector-function $W(y)$ from (1) in points $y \in D$ chosen by an optimization method. The results of these trials provide us with the search information set

$$\Omega_k = \{ (y^i, W^i)^T : 0 \leq i \leq k \}.$$

Instead of vectors $y^i \in D$, $0 \leq i \leq k$, the associated preimages $x^i: y^i = y(x^i)$, $1 \leq i \leq k$, may be stored in the set Ω_k .

And to solve a next new problem (7) related to the same general multicriteria problem, the set Ω_k can be recalculated to the following set:

$$\omega_k = \{ (x_\nu, z_\nu)^T : 0 \leq \nu \leq k \}. \quad (9)$$

The trial points x_ν , $0 \leq \nu \leq k$, from the set ω_k are ordered by the ascending values and $z_\nu = F(y(x_\nu))$, $0 \leq \nu \leq k$. The values z_ν , $0 \leq \nu \leq k$, are estimated with the help of the values W^i , $0 \leq i \leq k$, from the set Ω_k (without repeating any hard computations of the values W^i).

The algorithm can start solving a new problem immediately from a $(k+1)$ -th step making use of the set ω_k from (9). This, undoubtedly, will speed up the problem solution.

3. The SYMOP system configuration

According to the technique of reducing a general multicriteria problem to the solution of some simpler problems, the general idea of the system's structure is the principle of hierarchically interconnected software modules.

On the highest level of this hierarchy (Fig. 2) the modules are functionally subdivided into the following groups (subsystems):

- the optimization subsystem that contains modules implementing the optimization algorithms necessary in solving general multicriteria problems;
- the system monitor responsible for controlling the SYMOP system;
- the service program library including needed auxiliary modules.

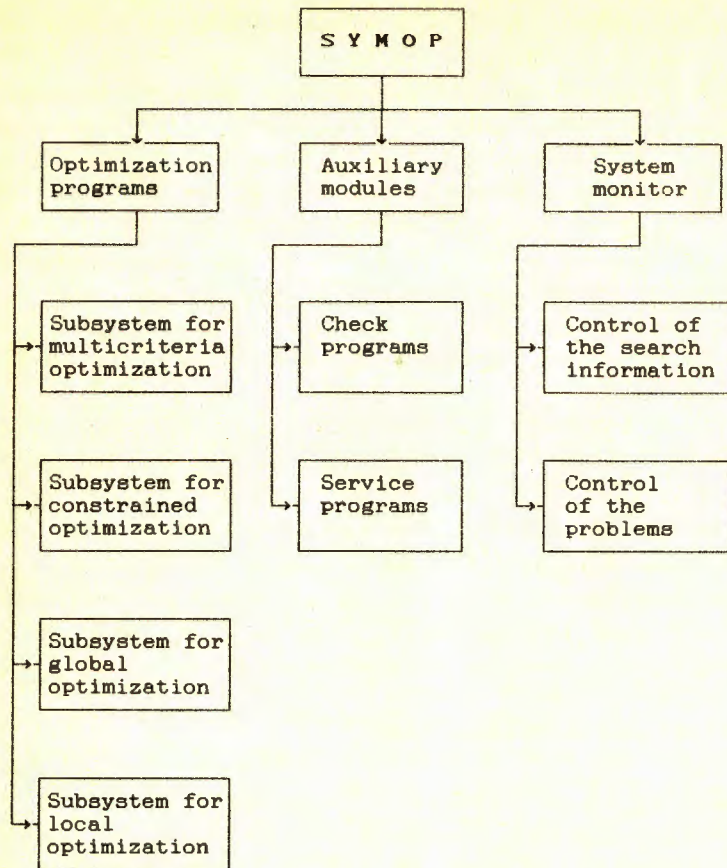


Fig. 2. General scheme of the SYMOP system

3.1. A structure of the optimization subsystem. First of all, this subsystem, through its modules, is responsible for reducing multicriteria problems to some sequence of constrained optimization

problems. And similarly, the subsystem of constrained optimization contains the modules that by the penalty function method reduce constrained optimization problems to a set of unconstrained problems. The modules responsible for solving one-dimensional scalar unconstrained problems constitute a global optimization subsystem. The local optimization subsystem contains the modules implementing the algorithms of a local search.

Fig. 2 shows interaction between the optimization subsystems. Upon generating a general criterion (a convolution of the criteria $f_i(y)$, $1 \leq i \leq s$), the multicriteria optimization subsystem will apply to the constrained optimization subsystem in order to solve a scalar optimization problem. In its turn, this subsystem, upon estimating the penalty coefficient β from (6) and choosing the weight coefficients r_i , $0 \leq i \leq m$, will apply to the global optimization subsystem to solve an one-dimensional unconstrained problem. And finally, the global optimization subsystem will apply to the local optimization subsystem.

3.2. A structure of the system monitor. Modules of the monitor are spread among the following parts.

Programs for controlling the search information are responsible for a dynamic memory allocation, storing and processing the search information (see Section 2.5). These programs can also interrupt the optimization process and store the search information in the system archives. Later on the monitor can restore the search information and proceed the optimization.

Programs for controlling the decision-making process provide setting the optimization problem statement through choosing the sets F from (3), G and q from (4). Also these programs check the execution of the optimization subsystems, accumulate a statistics of the system implementation, support a dynamic loading of the modules into the internal memory.

Programs for installing the SYMOP system set the initial values of parameters and adapt the SYMOP system to the associated configuration of the user's computer.

For a system-user interaction the system monitor has some suitable dialogue programs (see Section 4).

3.3. A structure of the auxiliary programs. The library of auxiliary programs holds software modules responsible for reliability of the system execution. There are some service programs in this library.

The names of instructions should be written in capital letters.

4.3. A dialogue mode of the system. This mode is a main one in the SYMOP system. It is a convenient form of a user-system interaction and needed when complicated optimization problems are solved and if the parameter values vary upon analysis of the available search information.

In a dialogue mode there are possible two forms of interaction. In case of the first form, control of the system is provided with its language (see Section 4.2). Through it there is obtained a uniformity in the system control facilities in various modes. This form of dialogue mode is oriented upon an experienced user.

In the second form of interaction, running the SYMOP system is controlled by a dialogue monitor. A dialogue scenario is described with the help of a "finite automat" form under which a running of the SYMOP system is represented as a sequence of transfers from one state of the dialogue to another. In the SYMOP there are distinguished the following kinds of states:

- the "notification" state, when the monitor sends some text message to the user and then the system goes on running in the next dialogue state;

- the "alternative" state, when a user answers "Yes" or "No" only to the dialogue monitor choosing one of the two transfers;

- the "choice" state, when a user is given a list of possible states and chooses one of them;

- the "action" state, when one of the system's modules is executed. In this state a user can change values of parameters. When the associated module is over, a message with the results obtained can be displayed. A jump to the next dialogue state is then done.

The dialogue monitor provides running the SYMOP system according to the declared scenario. In the course of computation, a transfer from one scenario to another is possible.

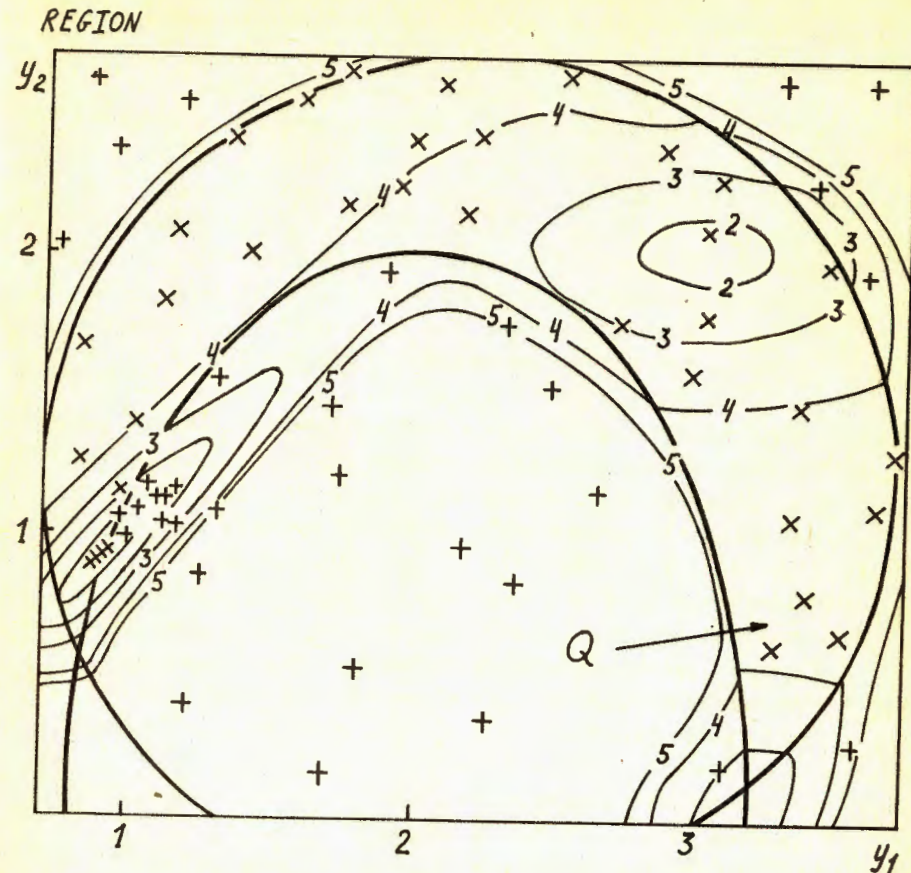


Fig. 3. Results of the solution for the test function $f(x)$
(the variation region, the feasible region,
the level curves, the trial points)

An example of executing a dialogue mode is given below where the same problem from Section 4.2 is described (the user's answers are marked with the sign "=").

- The SYMOP system is running. Will You install the system ?
= No

- There are available the following kinds of entering the system

- 1 - Solving a new optimization problem
- 2 - Optimizing the problem with the search information available

- Choose the variant

= 1

- Describe the model of your optimization problem

The next parameters may be changed:

- N - Amount of parameters
- NCF - Amount of functionals

= N = 2, NCF = 3

- State the multicriteria optimization problem

The next parameters may be changed:

- F - List of criterion numbers
- G - List of constraint numbers
- Q - Vector of admissions

= F = 1, G = 2,3, Q = 0.0,0.0

- Choose the search region

The next parameters may be changed:

- A - Left boundary
- B - Right boundary

= A = 0.7,0.0, B = 3.7,2.7

- Assign the method parameters

The next parameters may be changed:

- R - Method parameter
- EPS - Accuracy parameter
- KINC - Amount of iterations

= R = 1.5, EPS = 0.00001, KINC = 200

5. Numerical results

To demonstrate the SYMOP system capabilities, we shall show the results of minimizing a two-dimensional multiextremal test problem of the following kind (see [10]):

- the criterion is:

$$F(y) = - (1.5y_1^2 \exp(1-y_1^2) - 20.25(y_1-y_2)^2) + (0.5y_1-0.5)^4 (y_2-1)^4 \exp[(0.5y_1-0.5)-(y_2-1)^4].$$

- the constraints are:

$$g_1(y) = (y_1-2.2)^2 + (y_2-1.2)^2 - (1.5)^2,$$

$$g_2(y) = 1 - (y_1-2)^2 / (1.2)^2 + y_2^2 / 4,$$

- the search region is

$$D = \{ y \in R^2 : 0.7 \leq y_1 \leq 3.7, 0 \leq y_2 \leq 2.7 \}.$$

This problem has been solved by the SYMOP system. In this experiment we have set the method parameter (r) and the accuracy (ϵ) as follows: $r = 2, 0$, $\epsilon = 10^{-5}$. During the minimization the system has made 108 iterations of the MGAGS algorithm. The global minimum estimation is

$$y_1^* = 0.939, y_2^* = 0.949, \Phi(y^*) = -1.488.$$

The numerical results are shown in Fig. 3. First of all we can see the boundary of the feasible region Q (bold lines) and level curves of the function $F(y)$. Note the function has three local optima and the global minimum point, denoted by $*$, belongs to the boundary of Q . Then we can see the trial points generated by the system. These points are marked with the sign "+" and the points falling into the feasible region Q are denoted with the sign "x". As we can note the trial points are concentrated in the feasible region and in the neighborhood of the global minimum point.

The SYMOP system approbation is performed for the system of digital data transmission and the semiconductor circuit design.

References

1. Strongin, R.G. and Gergel, V.P. (1978). *The system of multiextremal optimization*. In: Samarskii, A.A. (ed). Packages of application programs. The software for optimization problems. Nauka, Moscow, pp. 39-50. (in Russian)
2. Dixon, L.C.W. and Szegö, G.P. (1978). *Towards global optimization*, 2. North-Holland, Amsterdam.
3. Horst, R. and Tuy, H. (1990). *Global optimization*. Springer Verlag.
4. Strongin, R.G. (1978). *Numerical methods for multiextremal problems*. Nauka, Moscow. (In Russian)
5. Wierzbicki, A.P. (1980). *A methodological guide to multiobjective optimization*. In: Lect. Notes Contr. and Inform. Sci., 22. Springer Verlag, pp. 99-123.
6. Krasnoshechov, P.S., Morozov, V.V. and Fedorov, V.V. (1979). *The decomposition in the problems of design*, 1. Izv. AN SSSR,

Technicheskaya kibernetika, 2, pp. 7-17. (In Russian)

7. Fiacco, A.V. and McCormick, G.P. (1968). *Nonlinear programming: sequential unconstrained minimization techniques*. Wiley, N.Y.

8. Strongin, R.G. (1985). *Numerical methods for multiextremal nonlinear programming problem with nonconvex constraints*. In: *Lecture Notes in Economics and Mathematical Systems*, 255. Springer Verlag, pp. 278-282.

9. Strongin, R.G., Gergel, V.P. and Markin, D.L. (1991). *Multicriteria multiextremal optimization with nonlinear constraints*. In: *Lewandovski, A. and Volkovich, V. (eds). Lectures Notes in Economics and Mathematical Systems*, 351. Springer Verlag, pp. 120-127.

10. Uosaki, K., Imamura, H., Tasaka, M. and Sugiyama, H. (1970). *A heuristic method for maxima searching in case of multimodal surfaces*. *Technol. Repts., Osaka Univ.*, 20, pp. 337-344.

Received by Publishing Department

on June 15, 1992.

Гергель В.П. и др.

E11-92-252

Система программ для многоэкстремальной оптимизации

Система многоэкстремальной оптимизации (СИМОП) разработана для численного решения многокритериальных многоэкстремальных многомерных задач оптимизации со сложными ограничениями. Система написана на языке фортран-4 и поставлена на вычислительных машинах типа IBM и VAX. Она требует минимально 200 Кб оперативной и 2 Мб внешней памяти ЭВМ.

Работа выполнена в Лаборатории сверхвысоких энергий ОИЯИ.

Gergel V.P. a. o.
A Software System...

E11-92-252.

Сообщение Объединенного инстит

Gergel V.P. et al.
A Software System for
Optimization

The system for
(package) was design
for multiextremal
problems with compli
cations in FORTRAN-4 has
been implemented on
type of computer
with 200 Kbytes of external memo

The investigation was
carried out in the
Physics Laboratory