

ON THE USE OF COORDINATED CALCULATIONS IN THE SOLUTION OF EXTREMAL PROBLEMS

Gintautas DZEMYDA and Vytautas TIEŠIS

Institute of Mathematics and Informatics,
Lithuanian Academy of Sciences,
232600 Vilnius, Akademijos St.4, Lithuania

Abstract. The paper deals with the minimization algorithms which enable us to economize the computing time during the coordinated calculation of the values of an objective function on the nodes of a rectangular lattice by storing and using quantities that are common for several nodes. The algorithm of a uniform search with clustering, the variable metric algorithm and the polytope algorithm are modified.

Key words: optimization, vector optimization, system analysis, clustering.

1. Introduction. Extremal problems that arise in the design of technical systems can be often transformed into the form

$$\min_{X \in [A, B]} f(X), \quad (1)$$

where the function $f(X) : R^n \rightarrow R$ is continuous and multiextremal in a general case, $A = (a_1, \dots, a_n)$, $B = (b_1, \dots, b_n)$, $X = (x_1, \dots, x_n)$, $[A, B] = \{X : a_i \leq x_i \leq b_i, i = \overline{1, n}\}$. The calculation of the value of an objective function in optimization problems, occurring in practice, often requires much

expenditure of computing time. Sometimes the expenditure is so great that it is impossible to solve the problem by classical methods. In such cases it is reasonable to base optimization methods not only on the functional characteristic of the objective function (linearity, convexity, etc.) but also on the structure of a calculation process of the function value. Thus, it is reasonable to combine the algorithms of optimization and calculation of an objective function.

The paper deals with the optimization problems where the coordinated calculation of an objective function on some nodes of a rectangular lattice requires much less computing time than the individual calculation of all those values. Let us denote the values of $f(\cdot)$ on nodes of the rectangular lattice by $f(\bar{x}_1 + s_1^{j_1}, \dots, \bar{x}_n + s_n^{j_n})$, where $s_i^{j_i} \in S_i \subset R$, $\exists s_i^{j_i} = 0$, $j_i = \overline{1, L_i}$, $i = \overline{1, n}$, L_i is the number of discrete levels of the i -th coordinate of the lattice, $\bar{X} = (\bar{x}_1, \dots, \bar{x}_n)$ is the source point (node) of the lattice. The total number of nodes is $L = \prod_{i=1}^n L_i$.

2. Models of the objective function. The structure of some functions in practice enables us to economize the computing time during the coordinated calculation of function values on the nodes of the rectangular lattice by storing and using the quantities that are common for several nodes. Let us define such an approach more in detail.

Every function may be represented in such a form:

$$f(X) = F(g_1(y_1), \dots, g_m(y_m)), \quad (2)$$

where $y_i \subseteq \{x_1, \dots, x_n\}$.

It is obvious that one of the reasons of economy is that it suffices to calculate the values of $g_k(y_k)$, $x_i \notin y_k$, but once when it is necessary to calculate L_i values $f(\bar{x}_1 + s_1^{j_1}, \dots, \bar{x}_i + s_i^{j_i}, \dots, \bar{x}_n + s_n^{j_n})$, $j_i = \overline{1, L_i}$. The process of calculation of $f(\cdot)$ values on all the nodes of the rectangular lattice in a general

case must be organized taking into account the computing time of all the functions composing $f(\cdot)$.

It is possible to construct a variety of models derived from (2) and to organize the correspondent process of calculating the values of the objective function. Let us analyze such examples:

$$f(X) = F_1(g_{i_1}(x_{i_1}), \quad i_1 = \overline{1, n}, \quad \tilde{g}_1(X)), \quad (3)$$

$$f(X) = F_2(g_{i_1 i_2}(x_{i_1}, x_{i_2}), \quad i_1 \leq i_2 = \overline{1, n}, \quad \tilde{g}_2(X)), \quad (4)$$

⋮

$$f(X) = F_k(g_{i_1 i_2 \dots i_k}(x_{i_1}, x_{i_2}, \dots, x_{i_k}), \quad i_1 \leq i_2 \leq \dots \\ \dots \leq i_k = \overline{1, n}, \quad \tilde{g}_k(X)). \quad (5)$$

The functions of the type g and \tilde{g} in (3)–(5) may be vector functions in a general case, i.e., some different functions, depending on the same combination of variables, may occur in the expression of $f(\cdot)$.

The functions

$$f(X) = \overset{n}{\mathbf{H}} g_i(x_i), \quad (6)$$

$$f(X) = \overset{n}{\mathbf{M}} \overset{n}{\mathbf{W}} g_{ij}(x_i, x_j) \quad (7)$$

are the special cases of (3) and (4), respectively, where $\mathbf{H}, \mathbf{M}, \mathbf{W} \in \{\Sigma, \Pi\}$.

With the help of these examples we shall analyze the economy of computing time.

The computational economy (E) is characterized by the relation of the computing time (T_1) of function values on all nodes of the rectangular lattice, without taking into account that the trial points are nodes of the lattice, and of the corresponding computational expenditure (T_2) obtained making use of the coordinated calculations of these values:

$$E = T_1/T_2.$$

Another criterion (N), derived from E , is useful comparing the efficiency of optimization algorithms. Its meaning is the following: the number of values of $f(\cdot)$ calculated during time T_2 in an uncoordinated way:

$$N = L/E.$$

We will use noninteger N to ensure the greater accuracy of further estimations.

E must be maximized and N minimized.

The following estimation of E is valid for any function $f(\cdot)$: $E \leq L/L_{\min}$, where $L_{\min} = \min_i L_i$. The estimation can be more accurate depending on the degree of concretization of (2).

The greatest economy E_3 for model (3) is achieved when the computing time of \tilde{g} and F_1 is negligible compared to that of all g_1, \dots, g_n . In this case $L/L_{\max} \leq E_3 \leq L/L_{\min}$, where $L_{\max} = \max_i L_i$ and $L_{\min} = \min_i L_i$. $E_3 = L/L_{\min}$ and $N_3 = L_{\min}$ as $L_i = L_{\min}$, $i = \overline{1, n}$. These estimates of economy are the same for the function (6) when the total computing time of all g_i , $i = \overline{1, n}$, is considerably more than that of H . In a general case the computational expenditure of all the parts of $f(\cdot)$ is weighty and the economy decreases significantly. For instance, let the computing time of all the functions in (3) be the same, $L_i = L_{\min}$, $i = \overline{1, n}$. In this case the economy is as follows: $E'_3 = (2L + nL)/(2L + nL_{\min})$.

Let us also consider special cases of (4) and (5), where there is the strict inequality " $<$ " instead of " \leq ". Let us denote these cases by (4a) and (5a) without writing the corresponding formulae. The values of E for these cases are not greater and the values of N are not smaller than those for (4) or (5), but the formulae of the estimations of E and N are simpler.

The greatest economy E_{4a} for model (4a) is achieved when the computing time of \tilde{g}_2 and F_2 is negligible com-

pared to that of all the other functions from (4a). In this case $L/L_{\max}^2 \leq E_{4a} \leq L/L_{\min}^2$, $E_{4a} = L_{\min}^{n-2}$ and $N_{4a} = L_{\min}^2$ as $L_i = L_{\min}$, $i = \overline{1, n}$. These estimates of economy are the same for function (7), when the total computing time of all g_{ij} , $i \neq j = \overline{1, n}$, is considerably more than that of M , W and g_{ii} , $i = \overline{1, n}$. The estimates are the following:

$$E_4 = L_{\min}^n n / [L_{\min}^2(n-1) + L_{\min}],$$

$$N_4 = [L_{\min}^2(n-1) + L_{\min}] / n,$$

when

- 1) the total computing time of all g_{ij} , $i, j = \overline{1, n}$, from (7) is considerably greater than that of M and W ;
- 2) the computing times of all g_{ij} from (7) are the same.

Analogously, the greatest economy for model (5a) is achieved when the computing time of \tilde{g}_k and F_k is negligible compared to that of all the other functions from (5a). In this case $L/L_{\max}^k \leq E_{5a} \leq L/L_{\min}^k$, $E_{5a} = L/L_{\min}^k$ and $N_{5a} = L_{\min}^k$ as $L_i = L_{\min}$, $i = \overline{1, n}$.

Let n and L_i , $i = \overline{1, n}$, be fixed. From the results above we observe a tendency that the value of E will be greater if the greater part of computing time of a single value of $f(\cdot)$ will be taken for the calculation of g_j , depending on a small number of variables.

The values of E'_3, E_3 (the greatest possible economy), E_4, E_{4a}, N_3, N_4 and N_{4a} are presented in Table 1 for the case $L_i = L_{\min}$, $i = \overline{1, n}$.

The investigation showed that a proper organization of calculations of the objective function values on the nodes of the rectangular lattice allows to obtain considerable economy of computation time in comparison with the calculation of those values without taking into account the specific character of the location of trial points. It is expedient to use this result in the construction of optimization algorithms.

Table 1. Investigation of computation economy

n	L_{\min}	L	E'_3	E_3	N_3	E_4	N_4	E_{4a}	N_{4a}
3	2	8	1.82	4	2	2.4	3.3	2	4
3	3	27	2.14	9	3	3.9	7.0	3	9
4	3	81	2.79	27	3	10.8	7.5	9	9
6	3	729	3.95	243	3	91.1	8.0	81	9
6	5	15625	4.00	3125	5	721.2	21.7	625	25

3. Method of solution. The function $f(X)$ is multiextremal in a general case. Thus, both global and local search are necessary in solving problem (1).

One of the ways to effectively solve the problem mentioned above is to use the well known methods of global and local optimization, where the possibility of calculation of objective function values on nodes of the rectangular lattice is in conformity with the main scheme of these methods.

3.1. Global Search. In search for an approximate value of global extremum, the values of an objective function are often calculated on nodes of some lattice covering the feasible set $[A, B]$ uniformly in a sense. This is done in this paper as well. The ideas of such a global search with clustering (Törn, 1986) are used.

The algorithm of global search is:

1. Construction of several lattices, totally of m points distributed uniformly in a sense in the set $[A, B]$, on the basis of m_2 source points, and calculation of the values of the function at those points.

2. Selection of m_1 least calculated values of the function and the corresponding points $X_i = (x_1^i, \dots, x_n^i)$, $i = \overline{1, m_1}$.

3. Clustering of the obtained m_1 points into p group

and selection of the point, corresponding to the least value of the function, for each group.

We assume these selected p points to be the approximate values of local minimizers.

The number of nodes of the rectangular lattice obtained from one source point is chosen equal to 3^n , i.e., $L_i = 3$, $i = \overline{1, n}$. That is limited by insufficiency of computer resources. $m_2 = 2^n$ source points are used.

Two modes are proposed for construction of the lattice in $[A, B]$. We shall detail these modes.

Mode 1. The rectangular lattice with a uniform distribution of nodes on individual variables (Sobolj, 1979) is constructed in the first mode. The source points for the lattice are the following:

$$(a_1 + \varepsilon_1 i_1, \dots, a_n + \varepsilon_n i_n), \quad i_1, \dots, i_n \in \{1; 4\},$$

$$\varepsilon_k = \frac{b_k - a_k}{5}, \quad k = \overline{1, n}.$$

The sets S_1, \dots, S_n (see, their description in Introduction) are the same for any source point $\bar{X} = (\bar{x}_1, \dots, \bar{x}_n)$:

$$S_k = \{-\varepsilon_k; 0; \varepsilon_k\}.$$

However, such a lattice is not effective when the objective function depends weakly on some variables. That was analyzed by Valevičienė and co-workers (1988). For instance, let the values of $f(\cdot)$ be dependent in fact only on one variable. In this case we shall have only 6 values of $f(\cdot)$ in order to evaluate the point of its minimum. This mode is also not effective for separable functions like a sum of functions depending on a single variable.

Mode 2. This mode allows to avoid the mentioned above imperfections of mode 1. It requires to spread out 3^n points

for calculating the objective function in each of 2^n subsets of the same volume, covering the feasible set $[A, B]$ completely. The length of each edge of any subset is equal to half of the corresponding edge of $[A, B]$. The coordinates of the source point $\bar{X} = (\bar{x}_1, \dots, \bar{x}_n)$ are chosen in each subset at random with constraint on the distance between this point and the subset boundary:

$$c = \min_{k=1, n} \left[\min \left(2 \cdot \frac{\bar{x}_k - \bar{a}_k}{b_k - a_k}, 2 \cdot \frac{\bar{b}_k - \bar{x}_k}{b_k - a_k} \right) \right] > \Delta,$$

where Δ is a certain constant, \bar{a}_k and \bar{b}_k are the lower and the upper boundaries, respectively, of the k -th edge of the investigated subset, $\bar{b}_k - \bar{a}_k = (b_k - a_k)/2$.

In order to construct the rectangular lattice, on the basis of the chosen source point in the subset, the following sets

$$S_k = \{ -c \cdot (\bar{b}_k - \bar{a}_k); 0; c \cdot (\bar{b}_k - \bar{a}_k) \}, \quad k = \overline{1, n},$$

are proposed to be used.

The examples of location of trial points by modes 1 and 2 are presented in Fig.1 for $n = 2$, $A = (0; 0)$, $B = (1; 1)$. The source points are denoted by ■. The dotted line bounds the region of random generation of the source points.

Clustering. The problem is to find the partition of m_1 points $X_i = (x_1^i, \dots, x_n^i)$, $i = \overline{1, m}$, from $[A, B]$ into p nonintersecting and homogeneous in a sense groups. The cluster-procedure based on the extremal parameter grouping is used. It is proposed by Dzemyda and Valevičienė (1988). This nonhierarchical cluster-procedure enables us to determine the clusters which can be unconvex, i.e., it is possible that the clusters obtained should not be separated by hyperplanes. The square matrix $R = \{r_{ij}, i, j = \overline{1, m}\}$ is analyzed.

$$r_{ij} = \exp \{ -\alpha \rho^2(X_i, X_j) \}, \quad \alpha > 0,$$

$$\rho^2(X_i, X_j) = \sum_{k=1}^n (x_k^i - x_k^j)^2 / (b_k - a_k)^2.$$

The procedure maximizes some functional of partition quality. It allows to select the value of p automatically. The results of Dzemyda (1990) are also related with this cluster-procedure.

Any other cluster-procedure can be used.

Efficiency. Let us denote the versions of global search algorithms with different modes for construction of the lattice in $[A, B]$ (mode 1 and mode 2) by $G1$ and $G2$, respectively.

The property of $G1$ and $G2$ is the following: the number of calculations of objective function values depends on n and is equal to $V = 6^n$. This number grows significantly for greater n . The problem arises to economize the computing time of these values taking into account the fact of location of trial points on the nodes of 2^n rectangular lattices.

Let us consider functions (6) and (7). Let the assumptions on the calculation durations of their constituent parts be made as in Chapter 2. In this case the coordinated calculation of V values of function (6) requires the same time as $P_6 = 2^n N_3$ values calculated in an uncoordinated way. The coordinated calculations of V values of function (7) require the same time as $P_7 = 2^n N_4$ or $P_{7a} = 2^n N_{4a}$ values, calculated in an uncoordinated way. The dependences V, P_6, P_7 and P_{7a} on n are presented in Table 2 for $L_i = 3, i = \overline{1, n}$.

Table 2. The dependence V, P_6, P_7 and P_{7a} on n

n	V	P_6	P_7	P_{7a}
2	36	12	24	36
3	216	24	56	72
4	1296	48	120	144
5	7776	96	249.6	288
6	46656	192	512	576

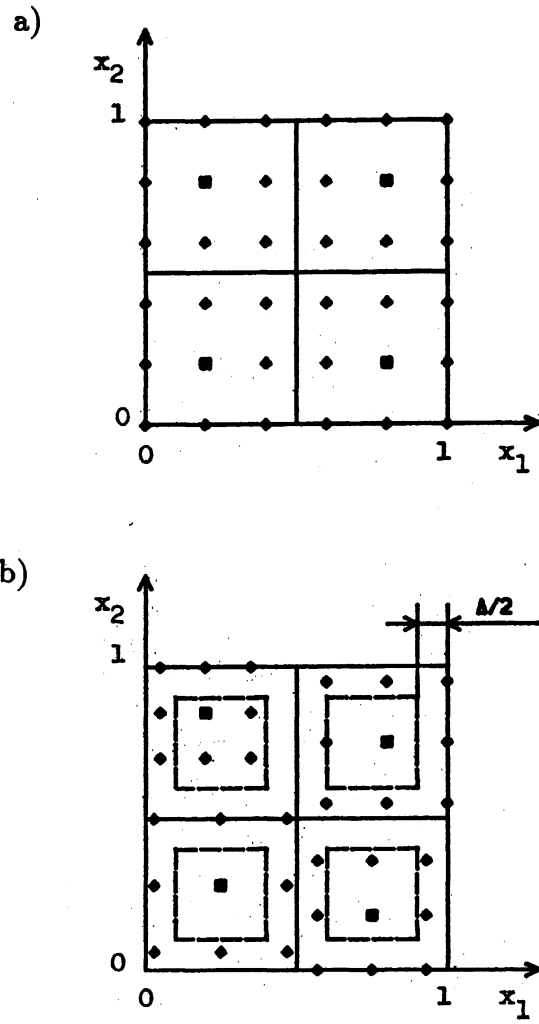


Fig.1. The examples of location of trial points:

a) mode 1,

b) mode 2.

The results of Table 2 enable us to forecast that the efficiency of $G1$ and $G2$ for most of the functions of such a type will be significantly better in comparison with that of any algorithm, which uses uncoordinated calculations of values of $f(\cdot)$, when the fixed calculating time for values of $f(\cdot)$ is $2^n T_2$. Analogous results can be forecasted for any function of type (3), (4) and (4a), when the total computing time of all g_1, \dots, g_n in (3), all g_{ij} in (4) and (4a) is considerably greater than that of F_1 and \tilde{g}_1 in (3), F_2 and \tilde{g}_2 in (4) and (4a). In a general case it is necessary to analyze the structure of the function for better organization of coordinated calculations of its values.

3.2. Local search. Variable metric algorithms are very effective in the regions of convexity of objective functions. A polytope algorithm is robust for nonsmooth and unconvex functions. These algorithms make the basis for our modifications.

The variable metric algorithm. The variable metric algorithm, proposed by Tiešis (1975) and presented in the description of the package MINIMUM edited by Dzemyda (1985), was modified. This algorithm combines the algorithms of Biggs (1971) and Goldfarb (1969). The nodes of the rectangular lattice are formed consecutively for numerical differentiation by either a forward or a second order difference formula. The differentiation step is selected in depending on the function values calculated earlier. Any realization of model (2) can be used in the calculation of values of the objective function on nodes of the lattice. The main scheme of the variable metric algorithm remains the same and the trajectory of descent remains the same as in the algorithm proposed by Tiešis. Theoretically, with the help of such an approach in the best case it is possible to accelerate the calculation of the gradient by n times in comparison with uncoordinated calculations of objective function values on the lattice nodes. The acceleration of calculation of the gradient is $n/2$ or n/k times when the

objective function corresponds to the data needed to obtain E_{4a} or E_{5a} , respectively.

The polytope algorithm. The ideas of the polytope algorithm are taken from the algorithm of Nelder and Mead (1963). We use the rectangular lattice of 3 levels instead of the simplex: $L_i = 3$, $s_i^3 = -s_i^1 = d_i$, $s_i^2 = 0$, $i = \overline{1, n}$. We construct a new lattice in each iteration. This lattice can be shrunk or expanded. The source point $\overline{X}^k = (\hat{x}_1^k, \dots, \hat{x}_n^k)$ of the new k -th lattice can be transferred or remains the same.

The shrinkage is performed when $Z^{k-1} = \overline{X}^{k-1}$ or $Z^{k-2} = Z^{k-1}$, where $Z^k = (z_1^k, \dots, z_n^k)$ is the point from the set of nodes of the k -th lattice with a minimal value of the objective function. Then, $\overline{X}^k = Z^{k-1}$ and $d_i^k = d_i^{k-1} \beta^2$ if the shrinkage was performed in the $(k-1)$ -th iteration, or $d_i^k = d_i^{k-1} \beta$ otherwise. Here $0 < \beta < 1$.

In other cases the lattice moves towards the minimum of the objective function. Then $\hat{x}_i^k = \hat{x}_i^{k-1}$ and $d_i^k = d_i^{k-1} \beta$ if $z_i^{k-1} = \hat{x}_i^{k-1}$. If $z_i^{k-1} = \hat{x}_i^{k-1} + d_i^{k-1}$, then $\hat{x}_i^k = z_i^{k-1} + d_i^k$. Here $d_i^k = d_i^{k-1} \alpha$ ($\alpha > 1$) if $f(\hat{x}_i^{k-1} - d_i^{k-1}) > f(\hat{x}_i^{k-1}) > f(z_i^{k-1})$, and $d_i^k = d_i^{k-1}$ otherwise. $f(y_i)$ denotes the value of the function $f(\cdot)$ at such a point: $Y = (\hat{x}_i^k, \dots, \hat{x}_{i-1}^{k-1}, y_i, \hat{x}_{i+1}^{k-1}, \dots, \hat{x}_n^{k-1})$. The definition of \hat{x}_i^k for the case $z_i^{k-1} = \hat{x}_i^{k-1} - d_i^{k-1}$ is analogous as above.

The i -th coordinate of a new source point is transferred from the i -th bound $[a_i, b_i]$ by the distance $d_i^k = d_i^{k-1} \beta^2$ when the calculated above \hat{x}_i^k is beyond $[a_i, b_i]$. The coordinates of some nodes of the lattice can be calculated as being beyond $[A, B]$. These nodes are placed on the bound of the set.

More complicated polytope transformations enable us to accelerate the algorithm. That was obtained for the simplex algorithm by Gill, Murray and Wright (1981).

4. Vector optimization. The objective function in (1) is a vector function, i.e., $f(X) = (f^1(X), \dots, f^M(X))$, when the problem of vector optimization is solved. The computing

time economy in this case may be obtained in two ways. First of all, the values of the individual functions $f^1(\cdot), \dots, f^M(\cdot)$ must be calculated on nodes of the rectangular lattice by a coordinated way. On the other hand, it is necessary to look for the constituent parts, common to all the functions $f^1(\cdot), \dots, f^M(\cdot)$. In control problems such a common constituent part is often the solution of differential equations and this takes the greatest part of computing time during the calculation of values of these functions (see, e.g., Didžgalvis and coworkers (1976)).

In the case of vector optimization the Pareto set is usually accepted as a primary solution. We approximate the Pareto set by the set of points chosen with the help of the Pareto criterion from the points generated in $[A, B]$. The Pareto criterion is verified and improper points are rejected after constructing a new lattice.

5. Results of experimental investigation. All the algorithms proposed are realized in FORTRAN and can be used on the IBM/370 or IBM PC/AT type computer.

5.1. Test Problems. The experiments were carried out with nine test functions:

1) Rastrigin's function:

$$f_1(x_1, \dots, x_n) = \sum_{i=1}^n \frac{2}{n} \cdot (x_i^2 - \cos(18x_i)),$$

$$x_i \in [-0.3; 0.6], \quad i = \overline{1, n}.$$

2) Rastrigin's function, $n = 4$, $x_1, x_3 \in [-0.25; 0.5]$, $x_2, x_4 \in [-0.125; 0.625]$.

3) Branin's function, presented by Dixon and Szegö (1978):

$$f_3(x_1, x_2) = a(x_2 - bx_1^2 + cx_1 - d)^2 + e(1 - f) \cos x_1 + e,$$

where $a = 1$, $b = 5.1/(4\pi^2)$, $c = 5/\pi$, $d = 6$, $e = 10$, $f = 1/(8\pi)$, $x_1 \in [-5; 10]$, $x_2 \in [0; 15]$.

4) the sum of Branin's functions:

$$f_4(x_1, \dots, x_4) = f_3(x_1, x_2) + f_3(x_3, x_4),$$

$$x_1, x_3 \in [-5; 10], x_2, x_4 \in [0; 15].$$

5) the first version of Shekel's function, presented by Dixon and Szegö (1978), $n = 4$, $x_i \in [0; 10]$, $i = \overline{1, n}$.

6) the second version of Shekel's function, $n = 4$, $x_i \in [0; 10]$, $i = \overline{1, n}$.

7) Hartman's function, presented by Dixon and Szegö (1978), $n = 3$, $x \in [0; 1]$, $i = \overline{1, n}$.

8) the test function No.26 from the list of test functions, proposed by Himmelblau (1972), $n = 4$, $x_i \in [0; 1]$, $i = \overline{1, n}$.

$$9) f_9(X) = \sum_{i=1}^n \sum_{j=i+1}^n \sin [(i+j)x_i x_j],$$

$$x_i \in [0; 1], i = \overline{1, n}.$$

Let us denote the problems of search for the minima of these functions by $T1-T9$.

5.2. Efficiency of global search.

5.2.1. Search for optimal Δ in $G2$. The problems $T1$ ($n = 4$), $T2$ and $T4$ have been solved 50 times because it is impossible to estimate the efficiency of this algorithm by a single solution, and the optimization results (the least function values obtained) have been averaged for each fixed value of Δ . The results are presented in Table 3.

The investigation showed that the optimal Δ value is in $[0.2; 0.3]$.

5.2.2. Investigation of the algorithms $G1$ and $G2$. The algorithms $G1$ and $G2$ belong to the class of algorithms of passive global search. The comparison of their efficiency with that of the random search with a uniform distribution of trial points in the feasible set $[A, B]$ and with that of LP -search proposed by Sobolj (1979) was performed. Let us denote these algorithms by RS and LP . These algorithms also

Table 3. The search for optimal Δ

Δ	$T1$	$T2$	$T3$
0	-1.64	-1.63	5.99
0.05	-1.68	-1.73	6.03
0.1	-1.73	-1.77	4.51
0.15	-1.77	-1.79	3.94
0.2	-1.76	-1.80	3.18
0.25	-1.77	-1.84	2.96
0.3	-1.75	-1.81	2.76
0.35	-1.66	-1.71	2.97

perform the passive global search. LP is one of the most efficient algorithms of this class. The coordinates of a new trial point depend on its number in LP and are selected randomly in RS . The Bayesian method for active global search, proposed by Mockus (1980) and modified by Tiešis (1980), was also used. Let us denote it by BA .

The experimental investigation was divided into two stages. The goal of the first one was to estimate the efficiency of $G1$ and $G2$ (in the sense of the obtained values of $f(\cdot)$) in comparison with RS and LP when RS and LP use the same number of values of $f(\cdot)$ as $G1$ and $G2$ (BA was not used in these experiments since it requires much more computing time as the number of trial points is large). The goal of the second stage was to estimate the efficiency of $G1$ as compared to LP and BA when the last two algorithms use the same amount of computing time for values of $f(\cdot)$ as $G1$ ($G2$ and RS were not used in these algorithms because the results of the first stage enable us to compare the efficiency of $G2$ to that of $G1$, and of RS to that of LP). $\Delta = 0.25$ and $p = 1$ were used in $G1$ and $G2$.

The first stage. The procedure, similar to that proposed by Šaltenis (1984), generating the test problem class

$$\min_{X \in D} f^i(X), \quad X = (x_1, \dots, x_n), \quad i = \overline{1, M},$$

$$D = \{0 \leq x_j \leq b_j, \quad j = \overline{1, n}\},$$

on the base of the problem

$$\min_{X \in D} f(X)$$

was used, trying to reach a greater objectivity of experimental results.

The basic problem was transformed by moving the point of its minimum into a random place in D :

$$f^i(X') = f(X), \quad X = (x_1, \dots, x_n), \quad X' = (x'_1, \dots, x'_n)$$

$$x'_j = \begin{cases} x_j + \xi_j^i & \text{as } x_j + \xi_j^i \leq b_j \\ x_j + \xi_j^i - b_j & \text{as } x_j + \xi_j^i > b_j \end{cases}$$

where ξ_j^i is the random value, distributed uniformly in the interval $[0; b_j]$, $j = \overline{1, n}$, $i = \overline{1, M}$.

The investigation results for $M = 50$ are presented in Table 4.

Table 4. The performance of $G1$, $G2$, RS and LP

Class	G1	G2	RS		LP	
			G1	G2	G1	G2
T5	-1.37	-1.79	0.49	1.40	0.42	1.29
T6	-1.64	-2.00	0.50	1.17	0.45	1.27
T7	-3.46	-3.62	0.30	0.53	0.17	0.50
T8	1.49	0.54	0.05	0.25	0.04	0.17

The columns $G1$ and $G2$ of Table 3 indicate the averaged best values (obtained by these algorithms) of the functions from the corresponding classes. The columns $RS/G1$ and $RS/G2$ indicate the ratio of the averaged number of calculations of objective function values by RS and that by $G1$ and $G2$, correspondingly, (it is equal to 6^n) in order to attain the same optimization result. The columns $LP/G1$ and $LP/G2$ have the same meaning for LP -search as $RS/G1$ and $RS/G2$ for the random search, respectively. The experiments witnessed to the possible gain of $G1$ and $G2$ usage as compared to RS and LP (e.g., the case $T5$, $RS/G2$), though the loss is also possible (e.g., the case $T8$, $LP/G1$). The result is very dependent on a concrete function and it is impossible to determine it in advance without some investigation.

The second stage. The problem $T9$ was solved. The results of investigation (obtained minimal values of function f_9) are presented in Table 5 for various n . Function f_9 is a detailed case of (7). Thus, LP and BA used P_{7a} (see, Table 2) values of the objective function during minimization. Here we assume that the summation of g_{ij} is considerably faster than the calculation of all g_{ij} .

The best results are obtained by $G1$. They are considerably better than those obtained by LP and BA .

Table 5. Comparison of $G1$, LP and BA

n	$G1$	LP	BA
3	-1.57	-0.51	-0.81
4	-3.96	-1.51	-3.40
5	-7.19	-3.59	-6.53
6	-11.56	-7.16	-10.38

5.3. Efficiency of local search. The economy of the proposed modification of the variable metric algorithm is achieved by accelerating calculations of the gradient. The main part of computer time during the minimization of functions in practice is used for the calculation of values of the objective function. So we shall estimate the economy (in times) of computing time by the formula:

$$E = N/(N_f + N_g/E_g),$$

where N_f is the number of values of $f(\cdot)$ lying on the descent trajectory, N_g is the number of additional values of $f(\cdot)$ used for calculations of the gradient $N = N_f + N_g$, E_g is the ratio, indicating how much faster (in times) the calculation of a single additional value of $f(\cdot)$ continues in comparison with that of a single value of $f(\cdot)$ on the descent trajectory.

The experiments were carried out with the problems $T1$, $T3$, $T4$ and $T9$. The function f_1 can be considered as a detailed case of (6), f_3 and f_4 that of (3), f_9 that of (7). The main charge of computing time of their values falls on the calculation of g_1, \dots, g_n for f_1 , g_1 for f_3 , g_1 and g_3 for f_4 , and g_{ij} for f_9 . The estimates of E are presented in Table 6. They are obtained without regard to computing times of the other functions composing f_1, f_3, f_4 and f_9 . So, E_g is equal to n for f_1, f_3, f_4 , and E_g is equal to $n/2$ for f_9 .

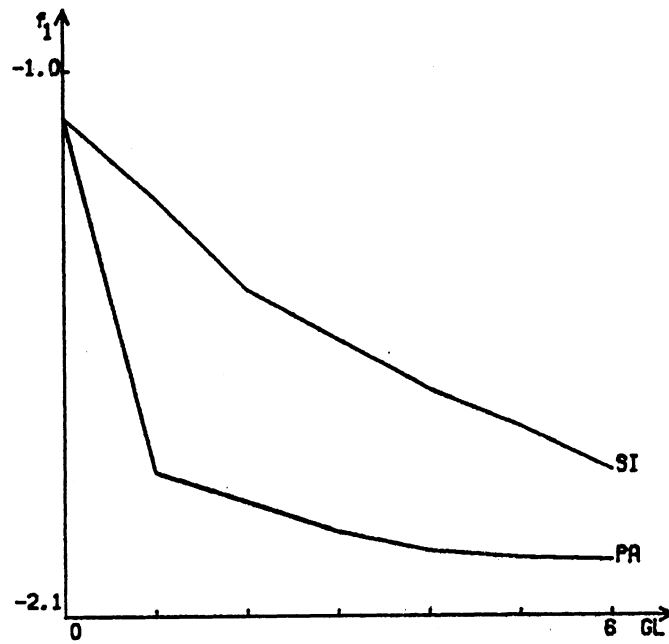
The investigation showed that a greater computational economy can be achieved for larger n .

The efficiency of the polytope algorithm was compared to that of the simplex one, proposed by Himmelblau (1975). Let us denote these algorithms by PA and SI , correspondingly.

The problem $T1$ ($n = 4$) was solved. Function f_1 is a detailed case of (6). 50 descents were performed by PA and SI from random points. The averaged results of minimization are presented in Fig.2.

Table 6. Economy of the variable metric algorithm

The problem .	n	N_g	N_f	E_g	E
$T1$	2	19	27	2	1.26
$T1$	4	47	34	4	1.77
$T1$	6	122	46	6	2.53
$T1$	10	218	44	10	3.98
$T3$	2	11	12	2	1.31
$T4$	4	45	24	4	1.96
$T9$	4	40	22	2	1.48
$T9$	6	77	27	3	1.97

**Fig.2.** The results of PA and SI comparison.

The numbers of generated lattices (GL) by PA are put on the argument axis. One iteration of PA takes the same computing time for 3^n values of f_1 as SI for 3 values of f_1 (see N_3 in Table 1 for $n = 4$ and $L_{\min} = 3$). This correspondence is used in Fig.2. PA was found to be much more effective.

5.4. Illustration of minimization. The performance of the proposed algorithms is illustrated by the test problem $T3$. Branin's function has three minima which are all global. Minimization results are presented in Fig.3.

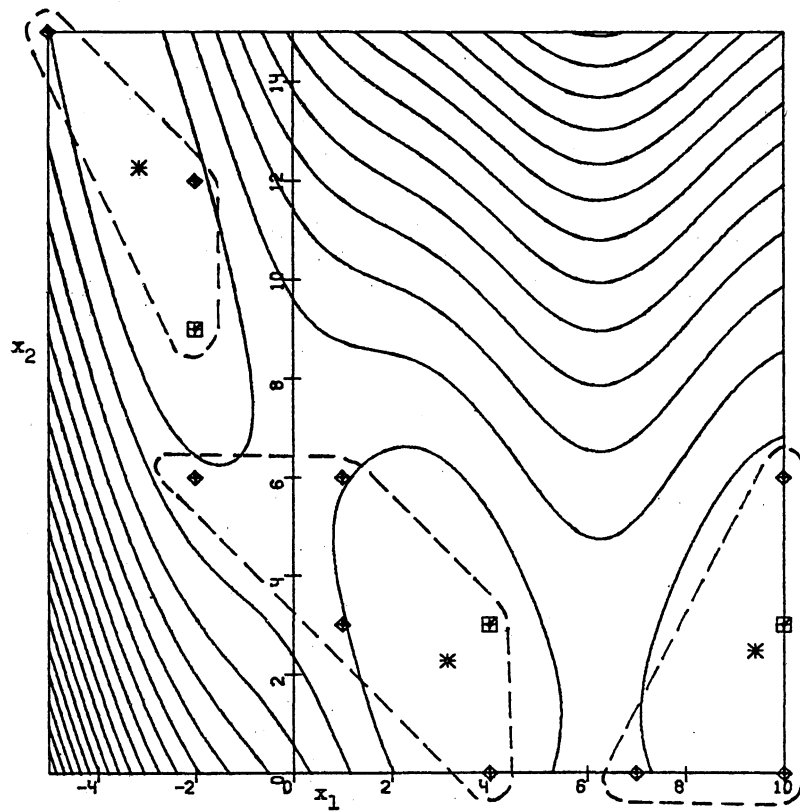


Fig.3. The minimization results.

The values of the function were calculated on the nodes of the lattice constructed by mode 1. 12 points were selected, having the least values of the function, and were clustered. Two symbols (\boxplus and \boxtimes) denote these points. The number of clusters was selected automatically. The dotted lines denote 3 obtained clusters. The symbol \boxplus denotes the point of the cluster where the value of the function is least. There are 3 such points. The symbol $*$ denotes the points of local minima. These points were obtained by the local search from the points denoted by \boxplus . Thus, all the local minima of Brańin's function were found.

6. Conclusions. The proposed minimization algorithms are oriented to use the values of the objective function calculated on nodes of the rectangular lattice. The investigations showed that the coordinated calculations of values of the objective function on nodes of the rectangular lattice allow to economize the computing time in comparison with uncoordinated calculations of these values. The extent of such an economy depends on the structure of the objective function and can vary in a wide range.

The proposed algorithms enable us to solve the problems with the restricted number of variables (in fact $n \leq 7$).

The ideas implied by these algorithms can be applied in the construction of minimization algorithms for parallel computers.

REFERENCES

- Biggs, M.S. (1971). Minimization algorithms making use of non-quadratic properties of the objective function. *Journal of the Institute of Mathematics and Applications*, **8**, 315–327.
- Didžgalvis, R., A.Žilinskas, K.Řagulskis and V.Tiešis (1976). On optimal energy transmission by skew impact. In J.Mockus (Ed.),

- Teorija Optimaljnych Reshenij*, Vol.2. Inst. Phys. Math. Lithuanian Acad. Sci., Vilnius. pp. 9–27 (in Russian).
- Dixon, L.C.W., and G.P.Szegö (1978). The global optimization problem: an introduction. In L.C.W.Dixon and G.P.Szegö (Eds.) *Towards Global Optimization 2*. North-Holland, Amsterdam. pp. 1–15.
- Dzemyda, G. (Ed.) (1985). *The Package of Applied Programs for the Dialogue Solving of Multiextremal Problems MINIMUM: The Description of Using*, The State Fund of Algorithms and Programs (reg. No 50860000112). Inst. Math. Cybern. Lithuanian Acad. Sci., Vilnius. 41pp. (in Russian).
- Dzemyda, G. (1990). Grouping on the sphere. In A.Žilinskas (Ed.), *Teorija Optimaljnych Reshenij*, Vol.14. Inst. Math. Cybern. Lithuanian Acad. Sci., Vilnius. pp. 22–40 (in Russian).
- Dzemyda, G., and J.Valevičienė (1988). The extremal parameter grouping in cluster analysis. In A.Žilinskas (Ed.), *Teorija Optimaljnych Reshenij*, Vol.13. Inst. Math. Cybern. Lithuanian Acad. Sci., Vilnius. pp. 36–53 (in Russian).
- Gill, P.E., W.Murray and M.H.Wright (1981). *Practical Optimization*. Academic Press, London. 402pp. Goldfarb, D. (1969). Extension of Davidon's variable metric method to minimization under linear inequality constraints. *SIAM Journal on Applied Mathematics*, **17**, 739–764.
- Himmelblau, D. (1972). *Applied Nonlinear Programming*. McGraw-Hill.
- Mockus, J. (1980). The simple Bayesian algorithm for the multi-dimensional global optimization. In F.Archetti and M.Cugiani (Eds.), *Numerical Techniques for Stochastic Systems*, North Holland, Amsterdam. pp. 369–377.
- Nelder, J.A., and R.Mead (1963). A simplex method for function minimization. *Computer Journal*, **7**, 308–313.
- Sobolj, I.M. (1979). On systematic search in a hypercube. *SIAM J. Numer. Anal.*, **16**, 790–793.
- Tiesis, V. (1975). The variable metric method for local minimiza-

- tion of multivariate functions subject to simple bounds. In R.Chomskis (Ed.), *Vyčislitel'naja Technika*, Proceedings of the Conference. Kaunas Polytechnic Institute, Kaunas. pp. 111–114 (in Russian).
- Tiešis, V. (1980). On the computation rate and efficiency of some global minimization algorithms. In A. Žilinskas (Ed.), *Teorija Optimaljnych Reshenij*, Vol.6. Inst. Math. Cybern. Lithuanian Acad. Sci., Vilnius. pp. 95–105 (in Russian).
- Törn, A.A. (1986). Clustering methods in global optimization. In N.K.Sinha and L.A.Telksnys (Eds.), *Stochastic Control*, 2nd IFAC Symposium, Vilnius, USSR, 19–23 May, 1986, Preprints, Part 2. IFAC, Moscow. pp. 138–143.
- Valevičienė, J., G.Dzemyda, V.Tiešis and A.Karminskij (1988). The combination of processes of minimization and of objective function calculation. In A.Žilinskas (Ed.), *Teorija Optimaljnych Reshenij*, Vol.13. Inst.Math.Cybern. Lithuanian Acad.Sci., Vilnius. pp. 17–35 (in Russian).
- Šaltenis, V. (1984). A relationship between the efficiency of LP-search and the structure of the problem. In *Internationaler Kongreß über Anwendungen der Mathematik in den Ingenieurwissenschaften*, Heft 4. Hochschule für Architektur und Bauwesen, Weimar. pp. 81–83.

Received January 1991

G.Dzemyda received the degree of Candidate of Technical Sciences from the Kaunas Polytechnic Institute, Kaunas, Lithuania, in 1984. He is a senior researcher at the Department of Optimal Decision Theory of the Institute of Mathematics and Informatics of the Lithuanian Academy of Sciences. His research interests include interaction of optimization and data analysis.

V.Tiešis is a researcher at the Department of Optimal Decision Theory of the Institute of Mathematics and Informatics, Lithuanian Acad.Sci. His research interests include nonlinear and integer optimization algorithms.