INFORMATICA, 1995, Vol. 6, No. 3, 249–263

MULTIEXTREMAL PROBLEM OF COMPUTER-AIDED DESIGN

Gintautas DZEMYDA

Institute of Mathematics and Informatics
Akademijos St. 4, 2600 Vilnius, Lithuania

Abstract. A multiextremal problem on the synthesis of external circuit of a tunable subnanosecond pulse TRAPATT-generator was investigated using algorithms of local optimization and cluster analysis.

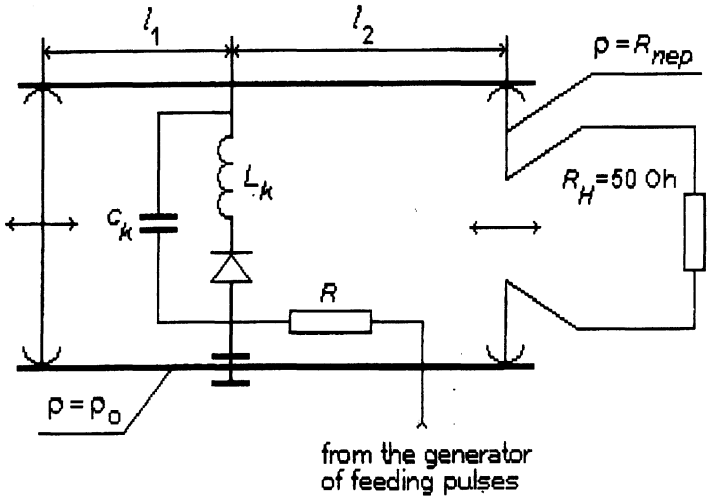
Key words: global optimization, local search, cluster analysis, computer-aided design.

1. Statement of the problem. The goal of this paper is to investigate the efficiency of optimization and data analysis methods in the analysis of the real technical problem: the synthesis of external circuit of a tunable subnanosecond pulse TRAPATT-generator. Global optimization algorithms are often investigated on various sets of test problems (e.g. [1]). Another goal of this paper is to find a multiextremal function such that actually arises, has known properties, is simple for computer realization, and is suitable for comparison of optimization algorithms.

The TRAPATT-diodes are used to generate high-capacity and high-frequency pulses. The description of the problem can be found in [2–6]. The scheme of external circuit of the generator is shown in Fig. 1a, and the generated pulses are presented in Fig. 1b. The external circuit involves two segments of the delay line. The resistance of segments is equal to ρ_0 ohms, and the lengths are equal to l_1 and l_2 centimeters, respectively. The inductance of the frame of diode is equal to L_k nanohenries, and its capacitance is equal to c_k nanofarads. The equivalent load capacity is equal to c_H nanofarads. The resistance of connection is equal to R_{nep} ohms.

Real Re_i, and imaginary Im_i, $i = \overline{1,m}$, parts of harmonics of external circuit impedance are the functions of seven parameters $x_1, \ldots, x_7$. Our

a)



b)

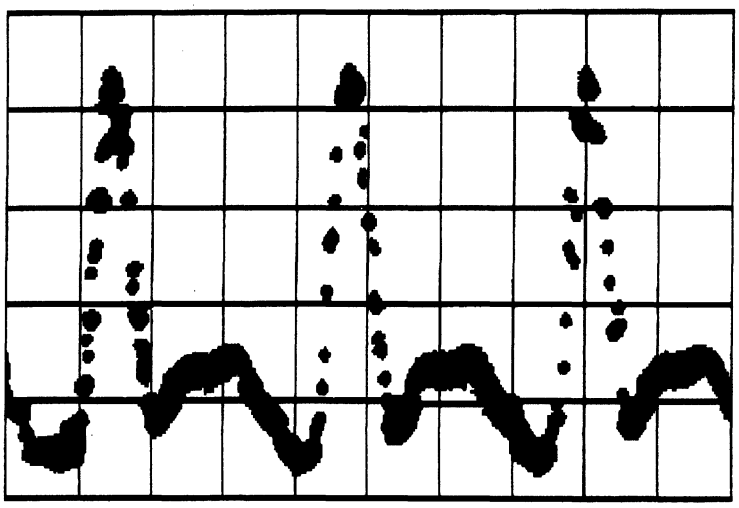


Fig. 1. The scheme of external circuit of the generator (a), and the generated pulses (b): horizontally 20 volts per division, vertically 0.2 nanoseconds per division.

problem was to find values of the parameters such that the obtained meanings of Re_i and Im_i, $i = \overline{1,m}$, differed from the desirable ones ($\overline{\mathrm{Re}}_i$ and $\overline{\mathrm{Im}}_i$) no more than by e_i and d_i percent, respectively. The function to be minimized was as follows:

$$f(X) = \sum_{i=1}^{m} \left\{ \left[\frac{\mathrm{Re}_i - \overline{\mathrm{Re}}_i}{\overline{\mathrm{Re}}_i \alpha_i} \right]^2 + \left[\frac{\mathrm{Im}_i - \overline{\mathrm{Im}}_i}{\overline{\mathrm{Im}}_i \delta_i} \right]^2 \right\},$$

where $X = (x_1, \ldots, x_7)$, $\alpha_i = e_i/100$ and $\delta_i = d_i/100$ are multipliers,

$$\mathrm{Re}_i = \frac{R_{nep}\rho_0^2[1 + \mathrm{tg}^2(\beta_i l_1)]}{A_i^2 + B_i^2},$$

$$\mathrm{Im}_i = \frac{[(x_{nep}^i + \rho_0 \mathrm{tg}(\beta_i l_1))A_i - R_{nep}B_i]\rho_0}{A_i^2 + B_i^2} + \omega_i L_k,$$

$$A_i = \rho_0 - x_{nep}^i \mathrm{tg}(\beta_i l_1) - [x_{nep}^i + \rho_0 \mathrm{tg}(\beta_i l_1)][\omega_i c_k \rho_0 - \mathrm{ctg}(\beta_i l_2)],$$

$$B_i = R_{nep}[\mathrm{tg}(\beta_i l_1) + \omega_i c_k \rho_0 - \mathrm{ctg}(\beta_i l_2)],$$

$$\beta_i = \omega_i / C,\ C = 30,$$

$$x_{nep}^i = -1/(\omega_i c_H),\ \omega_i = i2\pi F_1,\ F_1 = 1.064 \text{ gigahertz},$$

$$R_{nep} = a_1 + (b_1 - a_1)x_1,$$

$$c_H = (a_2 + \varepsilon) + (b_2 - a_2 - \varepsilon)x_2,\ \rho_0 = a_3 + (b_3 - a_3)x_3,$$

$$c_k = a_4 + (b_4 - a_4)x_4,\ L_k = a_5 + (b_5 - a_5)x_5,$$

$$l_1 = [a_6 + (b_6 - a_6)x_6]/F_1,\ l_2 = [a_7 + (b_7 - \varepsilon - a_7)x_7]/F_1,$$

$$\varepsilon = 10^{-10},\ x_j \in [0,1],\ j = \overline{1,7},\ m = 9.$$

From the technical point of view a solution is good if

$$\left| \frac{\mathrm{Re}_i - \overline{\mathrm{Re}}_i}{\overline{\mathrm{Re}}_i} \right| \leqslant \alpha_i, \quad \left| \frac{\mathrm{Im}_i - \overline{\mathrm{Im}}_i}{\overline{\mathrm{Im}}_i} \right| \leqslant \delta_i, \quad i = \overline{1,m}. \tag{1}$$

The values of e_i, d_i, $\overline{\mathrm{Re}}_i$ and $\overline{\mathrm{Im}}_i$ are presented in Table 1. The values of a_j and b_j, $j = \overline{1,7}$, are presented in Table 2.

The problem above may be formulated as a multiple criteria optimization problem with $2m$ criteria and additive utility function [32], too. The utility function is $f(X)$, and $1/\alpha_i^2$, $1/\delta_i^2$, $i = \overline{1,m}$, are weights of criteria in this case.

Table 1. The values of e_i d_i, $\overline{\mathrm{Re}_i}$ and $\overline{\mathrm{Im}_i}$

i	e_i	d_i	$\overline{\mathrm{Re}_i}$	$\overline{\mathrm{Im}_i}$
1	15	30	1.2	−2.4
2	16.875	33.75	1.5	−3.9
3	18.75	37.5	2.1	−7.1
4	20.625	41.25	3.9	−15.0
5	22.5	45	12.2	−40.0
6	24.375	48.75	261.0	−113.0
7	26.25	52.5	19.3	105.0
8	28.125	56.25	5.6	70.2
9	30	60	4.8	57.8

Table 2. The values of a_j and b_j

j	a_j	b_j
1	1	20
2	0	0.1
3	20	80
4	0	0.002
5	0	2
6	10	15
7	10	15

2. The strategy of investigation. Local descents were executed starting from 20 common random points and the obtained results were investigated using various clustering algorithms. The following question arises: maybe it would be better to use one optimization algorithm instead of 11, and to perform 220 descents? The experiments proved the correctness of our approach: descents from the same point gave from 3 to 8 different solutions due to various algorithms used. This is because the hyper surface of $f(X)$ is very difficult. Assumptions on $f(X)$ in the optimization algorithms also influenced the results. The results of local search starting from the same points allow us to compare the efficiency of the used methods, too. So, the investigation also showed the best algorithm for solving problems of this type.

The problem was investigated using 11 algorithms of local optimization:

1. POLC31.2 – the steepest descent using a projected gradient (Shah, Buehler and Kempthorne [9]);
2. POLC32.1 – a conjugate projected gradient (Fletcher and Reeves [10], Hestenes and Stiefel [11]);
3. POLC36.1 – a variable metric (Davidon [12], Fletcher and Powell [13], Goldfarb [14]);
4. POLC36.2 – a variable metric (Goldfarb [14, 17], Broydenn [15], Fletcher [16], Shanno [18]);
5. POLC39.4 – a variable metric (Oren and Spedicato [19]);
6. POLC41.2 – the steepest descent using a reduced gradient (Shah, Buehler and Kempthorne [9]);
7. POLC42.4 – a conjugate reduced gradient (Hestenes and Stiefel [11]);
8. POLC47.2 – a variable metric (Gill and Murray [20]);
9. POLC51.5 – quasi-Newton (Lukšan [21]);
10. WMCWD – a variable metric (Schittkowski [22]);
11. MIVAR4 – a variable metric (Tiešis [23]).

The first nine algorithms are taken from the system SPONA-82 [28, 29]. The algorithm MIVAR4 is from [30]. These algorithms differ in their complexity: from the gradient-type to a variable metric. The search for the best values of parameters for the programs realizing the algorithms above has not been performed. These values were used as suggested in the examples prepared by authors. Only the maximum number of function calculations was fixed to be the same in all the programs: 5000. In SPONA-82 this number is used as a default value. The other parameter rather frequently used is the number of iterations. The optimization process was never terminated due to this parameter during our experiments.

The results of optimization were analyzed using four clustering algorithms:

1. The centroid clustering algorithm (CCA) [25].
2. Ward's minimum variance clustering algorithm (WCA) [25].
3. The average linkage hierarchical clustering algorithm (ALCA) [25].
4. The algorithm proposed in [7] and theoretically based in [8] (DCA).

A lot of investigations in the field of cluster analysis indicates that WCA and ALCA are the best among the hierarchical clustering algorithms (e.g., see [26]). The programs realizing CCA, WCA and ALCA are taken from SAS

[25]. Some details on DCA may be found in [31].

3. The results

3.1. Local descents. The following results are presented in Table 3:

1. S is the averaged number of calculations of $f(X)$ for a single local descent.
2. N is the number of results of local descents satisfying conditions (1).
3. $S1$ is the averaged number of calculations of $f(X)$ used to obtain the results satisfying conditions (1).
4. $S2$ is the averaged number of calculations of $f(X)$ used to obtain the results not satisfying conditions (1).
5. $S1/S2$ is the ratio of $S1$ and $S2$.

Table 3. Results of local descents

Algorithm	S	N	$S1$	$S2$	$S1/S2$
POLC31.2	3968	8	5000	3279	1.52
POLC32.1	3977	10	5000	2954	1.69
POLC36.1	934	3	1553	824	1.88
POLC36.2	692	3	748	682	1.10
POLC39.4	689	5	1025	577	1.78
POLC41.2	4431	5	5000	4242	1.18
POLC42.4	4044	3	5000	3875	1.29
POLC47.2	826	1	957	819	1.17
POLC51.5	834	3	953	813	1.17
WMCWD	454	6	425	467	0.91
MIVAR4	852	7	1056	742	1.42

From Table 3 we observe a tendency of using a greater number of function calculations for obtaining solutions satisfying (1) in comparison with the solutions not satisfying (1). The best optimization results are obtained by POLC32.1 and POLC31.2, which are the simplest among the algorithms. A bit worse are MIVAR4 and WMCWD. But they are significantly better from the view point of the used number of function calculations.

3.2. Application of cluster analysis. The results of 220 local descents were analyzed using cluster analysis. The goal of such an analysis was to determine groups of points located around different local minima.

The analysis was made in two stages:

1. Analysis of the results satisfying (1) (54 points).
2. Analysis of all the results (220 points).

Analysis of the results satisfying special conditions. The first question to be answered was as follows: do the points form one cluster or more? If more, then how many?

Fig. 2 shows the densities of appearance of the values of different variables and the function $f(X)$. Intervals of change, i.e., the intervals between the least and the greatest value obtained in 54 descents, were divided into 7 parts. The values of x_3, x_5 and x_6 make two obvious clusters. The densities for other variables are more fuzzy. Thus, it is necessary to use additional information for determining the number of clusters.

We introduce a term of an "unreliable" point. These are points where the optimization stops due to the restriction on the number of function calculations. These points may be far enough from the local minima. Formal criteria devoted to determine the number of clusters may also be used.

The analysis was made using the algorithm DCA [7, 8, 31]. The values of variables were normalized seeking the same influence in the analysis. The normalization was made as follows: the values of any variable were divided by the length of the interval of change in this variable.

The partition into two clusters gave the following result: 28 points were attached to the first cluster, and 26 points were attached to the second one. This partition coincides with the optimal visual partition of x_3, x_5 and x_6 into two clusters. The partition into three clusters gave the following result: 28, 20, 6. The third cluster contains "unreliable" points only. The partition into four clusters gave the following result: 28, 18, 6, 2. Two last clusters here contain "unreliable" points only. Thus, the most reliable optimal partition is that of two clusters. Note that the clustering results have a hierarchical structure while DCA is not a hierarchical procedure.

The number of clusters was examined using a gamma-criterion. It is proposed in [24]. In [27] it gave very good results in the search for the optimal number of clusters. The value of the gamma-criterion was calculated for partitions into 2, 3, and 4 clusters. The maximum of the criterion was obtained in the case of clustering into two groups. This is the second motive that the number of clusters is equal to 2.

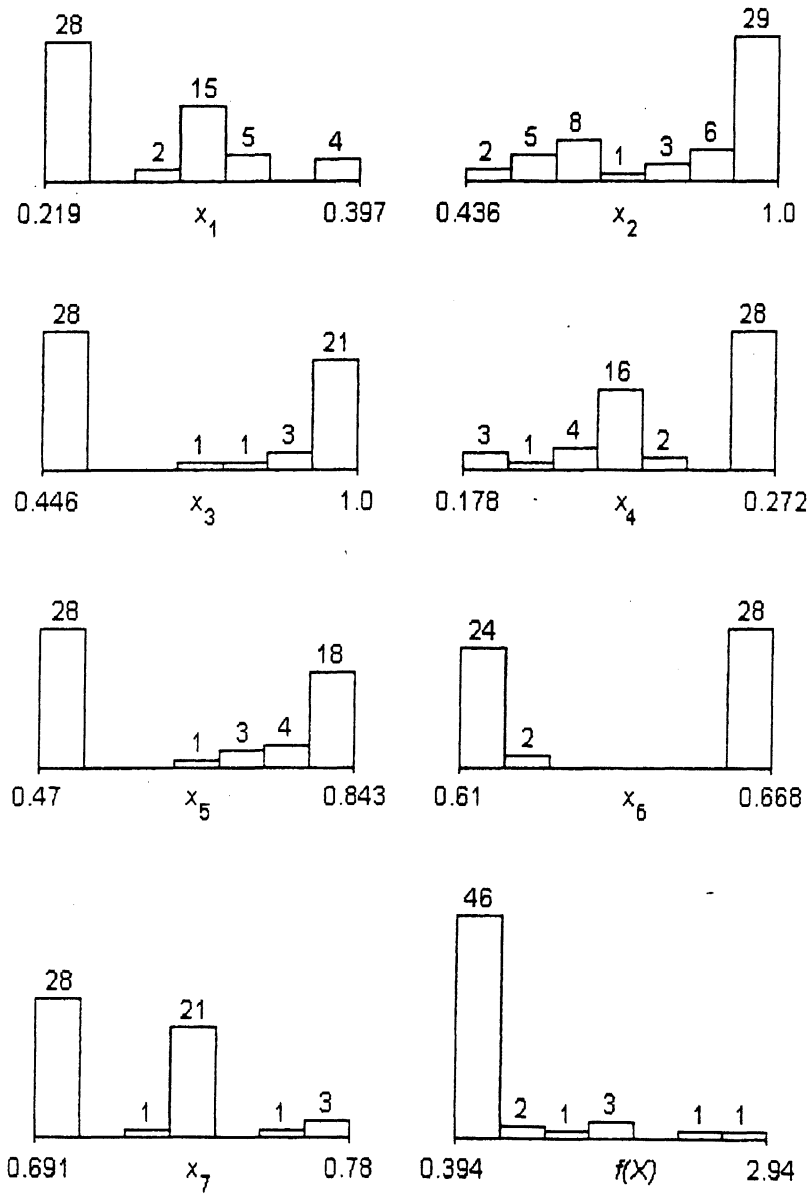


Fig. 2. The densities of appearing values of different variables and the function $f(X)$.

So, the function $f(X)$ is multiextremal and has two minima near to zero:

1. $f(X_1) = 0.393896$, $X_1 = (0.31861, 1., 0.95557, 0.22026, 0.80798, 0.61446, 0.73354)$.
2. $f(X_2) = 0.447964$, $X_2 = (0.22120, 1., 0.48568, 0.26192, 0.49754, 0.66697, 0.69334)$.

The surface of $f(X)$ is shown in Fig. 3a. The values of $f(X)$ vary in a wide range, so the values of $\lg[f(X)]$ were used in Fig. 3a. The values of $x_1, \ldots, x_5$ were fixed at the middle point between X_1 and X_2, and the values of x_6 and x_7 were varied in [0,1]. Fig. 3b details a part of Fig. 3a. We can observe here the existence of two local minima.

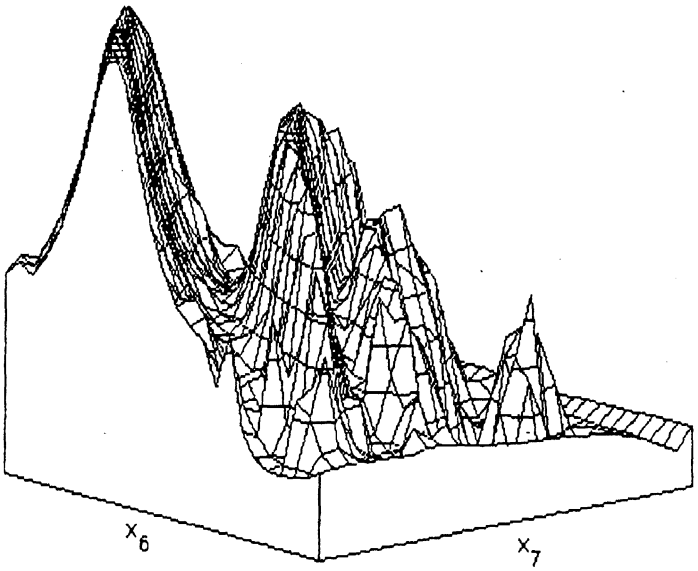
Complete analysis. The results of 220 local descents were analyzed using three hierarchical clustering algorithms: the centroid clustering algorithm (CCA), Ward's minimum variance clustering algorithm (WCA), the average linkage hierarchical clustering algorithm (ALCA) [25]. The data were not normalized because the obtained values of all variables vary in the similar ranges: $x_1, \ldots, x_6 \in [0, 1]$, $x_7 \in [0, 0.909]$.

The clustering was proceeded until the moment of appearance of X_1 and X_2 in different clusters. The clustering results are presented in Table 4. The minimal values of function $f(X)$ and the corresponding points for different clusters are presented in the rows 1 – 23. The symbol "$*$" indicates "unreliable" points. The symbol "+" indicates that this minimum has been picked out by corresponding clustering algorithm. Symbol "–" indicate that this minimum has not been picked out by the corresponding clustering algorithm. The clustering results indicate that the greatest number of clusters has been chosen by CCA.

4. Conclusions. The local descents from random starting points formed a set of solutions which may be considered as local minima of the TRAPATT-generator design problem. Data analysis methods were applied to the optimization results. There are no general methods suitable to determine the number of clusters - minima. The human decision is the final one in this paper. The investigation showed that the problem has many local minima. However, only two minima satisfy technical restrictions (1). These two minima are near to zero. The values of other local minima are considerably greater.

The problem of computer-aided synthesis of the external circuit of the tunable subnanosecond pulse TRAPATT-generator may be used as a test for comparison of the efficiency of optimization methods.

a)



b)

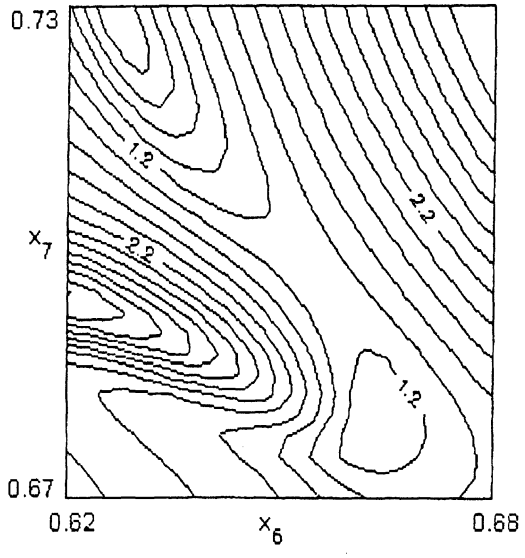


Fig. 3. The surface of $f(X)$.

Table 4. The appearance of local minima during cluster analysis

No	Clustering algorithm			$f(X)$	x_1	x_2	x_3	x_4	x_5	x_6	x_7
	CCA	WCA	ALCA								
1	+	+	+	0.39390	0.31861	1.	0.95557	0.22026	0.80798	0.61446	0.73354
2	+	+	+	0.44796	0.22120	1.	0.48568	0.26192	0.49754	0.66697	0.69334
3	+	−	+	10.760*	0.17811	1.	0.17743	0.10982	0.19104	0.71817	0.75039
4	−	−	+	11.781*	0.23025	0.68393	0.19796	0.45920	0.27033	0.66085	0.68431
5	+	+	+	21.077*	1.	1.	1.	0.17940	0.48921	0.58138	0.86487
6	+	+	+	46.642	0.17270	0.91101	0.20983	0.71157	0.32420	0.67186	0.56522
7	+	−	+	63.308	0.04937	0.65308	0.	0.	0.	0.94765	0.44999
8	+	−	+	66.481*	0.79100	0.40447	0.70460	0.13916	0.19970	0.82659	0.89488
9	+	−	−	68.838	0.12854	0.55221	0.01321	0.61665	0.	0.92699	0.69371
10	+	−	+	68.976	0.03056	1.	0.44369	0.	0.04028	0.95621	0.36175
11	+	−	+	72.827	0.67927	1.	0.	0.60539	0.	1.	0.86814
12	+	+	+	73.476	1.	1.	0.	0.60962	0.03257	0.	0.84975

to be continued

Table 4. The appearance of local minima during cluster analysis (continuation)

No	Clustering algorithm			$f(X)$	x_1	x_2	x_3	x_4	x_5	x_6	x_7
	CCA	WCA	ALCA								
13	+	−	+	87.101	0.07767	0.29702	0.	0.93343	0.	1.	0.44907
14	+	−	+	90.493	0.02380	1.	1.	0.	0.	0.98469	0.07379
15	+	−	+	100.00	1.	0.02023	0.	0.53049	0.01652	0.	0.84187
16	+	−	+	106.22	0.55520	0.73895	0.00490	0.01351	0.	0.24011	0.90204
17	+	−	+	108.49	0.	0.76743	1.	1.	0.	0.99549	0.25523
18	+	−	−	124.07	0.02149	1.	0.	0.	0.	0.93044	0.
19	+	−	−	141.80	0.09631	0.71147	0.74683	1.	0.47238	0.81684	0.16298
20	+	−	+	155.43	0.35778	0.02097	0.	0.78976	0.13742	0.	0.68731
21	+	−	+	330.68	0.	1.	0.	0.	0.36805	0.	0.69813
22	+	−	+	350.75	0.32727	0.00632	0.10260	1.	1.	0.97789	0.29401
23	+	−	+	1110.9	1.	0.	0.	0.	0.	0.	0.

REFERENCES

[1] 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.

[2] Vaitiekūnas, F.K., and coworkers (1981). Analysis of the external circuit of pulse generator based on TRAPATT-diode. *Technika Sredstv Sviazi, Radioizmeriteljnaja Technika,* **3**(35), 11–17 (in Russian).

[3] Dzemyda, G., and co-authors (1984). Solution of the problems of optimal design and selection of model parameters' values using the package of applied programs MINIMUM. In A. Žilinskas (Ed.), *Teorija Optimaljnych Reshenij,* Vol. 10. Inst. Math. Cybern., Vilnius. pp. 77–98 (in Russian).

[4] Dzemyda, G (1987). On the solving of two optimal design problems. In *European Congress on Simulation,* Vol. A. Academia, Prague. pp. 122–126.

[5] Dzemyda, G., J. Valevičienė and L. Lukšan (1990). Investigation of the TRAPATT-generator design problem. In *Teorija Optimaljnych Reshenij,* Vol. 14. Inst. Math. Cybern., Vilnius. pp. 9–21. (in Russian).

[6] Dzemyda, G. (1993). Investigation of the multiextremal problem. In *16th IFIP Conference on System Modelling and Optimization,* Abstracts, Vol. 2. Compiegne (France). pp. 893–895.

[7] Dzemyda, G., and J. Valevičienė (1988). The extremal parameter grouping in cluster analysis. In *Teorija Optimaljnych Reshenij,* Vol. 13. Inst. Math. Cybern., Vilnius. pp. 36–53 (in Russian).

[8] Dzemyda, G. (1990). Grouping on the sphere. In *Teorija Optimaljnych Reshenij,* Vol. 14. Inst. Math. Cybern., Vilnius. pp. 22–40 (in Russian).

[9] Shah, B.V., R.J. Buehler, and O. Kempthorne (1964). Some algorithms for minimizing a function of several variables *SIAM J. Appl. Math.,* **12**, 74–92.

[10] Fletcher, R., and C.M. Reeves (1964). Function minimization by conjugate gradients. *Computer Journal,* **7**, 149–154.

[11] Hestenes, M.R., and C.M. Stiefel (1964). Methods of conjugate gradient for solving linear systems. *J. Res. of the NBS,* **49**, 409–436.

[12] Davidon, W.C. (1959). *Variable Metric Method for Minimization.* Argonne National Lab. Report No. ANL-5990.

[13] Fletcher, R., and M.J.D. Powell (1963). A rapidly convergent descent method for minimization. *Computer Journal,* **6**, 163–168.

[14] Goldfarb, D. (1969). Extension of Davidon's variable metric method to maximization under inequality and equality constraints. *SIAM J. Appl. Math.,* **17**, 739–764.

[15] Broydenn, C.G. (1970). The convergence of a class of double rank minimization algorithms. Part 1 – General considerations. Part 2 – The new algorithm. *J. Inst. Math. Applics.,* **6**, 79–90 , 222-231.

[16] Fletcher, R. (1970). A new approach to variable metric algorithms. *Computer Journal,* **13**, 317–322.

[17] Goldfarb, D. (1970). A family of variable metric algorithms derived by variational means. *Math. of Computation,* **24**, 23–26.

[18] Shanno, D.P. (1970). Conditioning of quasi-Newton methods for function minimization. *Math. of Computation,* **24**, 647–656.

[19] Oren, S.S., and E. Spedicato (1976). Optimal conditioning of self scaling variable metric algorithms. *Math. Programming,* **10**, 70–90.

[20] Gill, P.E., and W. Murray (1973). *Quasi-Newton Method for linearly Constrained Optimization,* Report NAC 32, National Physical Lab., England.

[21] Lukšan, L. (1982). Quasi-Newton methods without projections for linearly constrained minimization. *Kybernetika,* **18**, 307–319.

[22] Schittkowski, K. (1982). Nonlinear programming methods with linear least squares subproblem. In J.M. Mulvey (Ed.), *Evaluating Mathematical Programming Techniques.* Springer, New-York. pp. 200–213.

[23] Tiešis, V. (1).975 The variable metric method for local minimization of multivariate functions subject to simple bounds. In R. Chomskis (Ed.), *Vyczislitelnaja Technika,* Proceedings of the Conference. Kaunas Polytechnic Institute, Kaunas. pp. 111-114 (in Russian).

[24] Baker, F.B., and L.J. Hubert (1975). Measuring the power of hierarchical cluster analysis. *Journal of the American Statistical Association,* **70**, 31–38.

[25] SAS (1982). *SAS User's Guide: Statistic,* 1982 Edition. SAS Institute Inc., Cary. 584 pp.

[26] Milligan, G.W. (1980). An examination of the effect of six types of error perturbation of fifteen clustering algorithms. *Psychometrika,* **45**(3).

[27] Milligan, G.W., and M.C. Cooper (1985). An examination of procedures for determining the number of clusters in a data set. *Psychometrika,* **50**(2), 159–179.

[28] Lukšan, L. (1987). *Poznamky k používani souboru optimalizačnich programu SPONA.* Vyzkumna zprava č, V-276, SVT ČSAV, Praha, duben 1987. 82 pp.

[29] Lukšan, L., M. Zamazal and S. Kočkova (1987). *SPONA-83. Uživatelsky Popis Souboru Programu pro Optimalizaci a Nelinearni Optimalizaci /*Verse 1982*/.* Vyzkumna zprava V-125, SVT ČSAV, leden 1983, Praha. 301 pp.

[30] 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., Vilnius. 41 pp. (in Russian).

[31] Dzemyda, G. (1996). Clustering of parameters on the basis of correlations via simulated annealing. To appear in the *Special Issue of Control and Cybernetics on Simulated Annealing Applied to Combinatorial Optimization.*

[32] Dzemyda, G., and V. Šaltenis (1994). Multiple criteria decision support system: methods, user's interface and applications. *Informatica,* **5**(1–2), 31–42.

Received September 1995

G. Dzemyda received his Ph.D. degree 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, and an Associate Professor at the Vilnius Pedagogical University. His research interests include interaction of optimization and data analysis.

DAUGIAEKSTREMALINIS AUTOMATIZUOTO PROJEKTAVIMO UŽDAVINYS

Gintautas DZEMYDA

Analizuojamas subnanosekundinių impulsų generatoriaus išorinės grandinės projektavimo uždavinys. Optimizuojama funkcija yra daugiaekstremali. Tyrime daugkartinių lokalinių nusileidimų rezultatai, gauti pasinaudojant 11 lokalinės paieškos algoritmų, analizuoti 4 klasterizavimo algoritmais. Tyrimo tikslas – ištirti funkcijos daugiaekstremališkumo mastą. Siūloma analizės metodika gali būti naudojama ir kitiems optimizavimo uždaviniams tirti.