Proc. Estonian Acad. Sci. Eng., 2004, **10**, 4, 236–250

A comparative analysis of global search procedures

Alexander Janushevskis[a], Teodor Akinfiev[b], Janis Auzins[a] and Alexander Boyko[a]

[a] Research Laboratory of Machine and Mechanism Dynamics, Riga Technical University, Ezermalas iela 6, LV-1006 Riga, Latvia; janush@latnet.lv

[b] IAI CSIC, Carretera de Campo Real, km. 0.200 La Poveda, Arganda del Rey, Madrid 28500, Spain; teodor@iai.csic.es

Received 1 July 2004

Abstract. This paper presents a brief overview and a comparison of global optimization methods and their software. By solving a set of recognized test problems, the reliability of obtaining the global extremums is compared. It is shown that the Edaopt algorithm, developed formerly by the authors, is more efficient than modern genetic algorithms. In most cases the Edaopt algorithm takes also noticeably less computation time.

Key words: optimization, stochastic global search, genetic algorithms.

1. INTRODUCTION

The development of competitive engineering systems is unimaginable without their optimization. In the development stages of various machines, constructions and complex mechatronic systems, computer modelling with the aim of detailed investigation and improvement of their properties is becoming more and more widespread. In the field of mechanical engineering, the so-called virtual prototyping tools, that is software [1,2] that allows to create automatically mathematical models of mechanical systems to estimate their dynamic and strength properties and carry out their parametrical optimization, are widely used. By that, one is frequently faced with the global optimization task.

From the mathematical point of view, if the optimization criterion does not satisfy the Lipschitz conditions and the search region is not limited, then factually it is incorrectly formulated in the sense that the global optimum cannot be located with a given accuracy after a limited number of calculations of the criterion function. In the case of the general “black box” model, the global optimization is carried out without *a priori* knowledge about the surface of the criterion function, which is defined by the criterion and the constraints. The domain of attraction is defined as a region of this surface where a local minimum exists and the constraints are satisfied. The method of global optimization must have a mechanism that allows to leave local minimums, while local optimization methods have not such a mechanism and therefore attraction regions “catch” the local search methods. For this reason, global search algorithms employ heuristic methods to search for new attraction regions.

The minimums of limited regions are often located using gradient descent, Newton, quasi-Newton and other methods. It must be noted that in many practical tasks finding of the global optimum is incommensurably costly and sub-optimal solutions must suffice. Therefore various possibilities of obtaining the global optimum are sought for. One of the approaches is building and optimization of the so-called metamodels.

2. THE RESPONSE SURFACE METHOD OF SYSTEM OPTIMIZATION

Optimal design is based on a mathematical model of the object. The level of complexity of practical systems is frequently very high and their models are complicated non-linear high-order systems of equations (differential, integral, algebraic and others), the parameters of which are not precisely known. Their parametric and structural identification and solution demands very large computing resources. In such cases, to carry out optimization, the response surface method (RSM) [3–5] or the neural network approach [6] is usually used. The development of metamodels (surrogate models) on the basis of a small number of very time-consuming calculations, mathematical or natural experiments, is often applied to find the global optimum.

In the construction of metamodels, polynomial functions, stochastic kriging models [7], radial basis functions [8] or adaptive regression splines [9] are most often employed. Polynomial functions stand out with the simplicity of their construction and calculation speed that is very important for carrying out global optimization. Using RSM, the acceptable number of criterion and constraint calculations may be significant and reach hundreds of thousands and even millions of tries, since its calculation requires a significantly smaller amount of time than the criterion calculation of the initial model.

3. A SHORT REVIEW OF THE GLOBAL OPTIMIZATION METHODS

In the solution of engineering problems, one is frequently faced with mixed non-linear programming problems with constraints, where optimization parameters are both discrete and continuous variables. In many cases, it is possible to interpret discrete variables as the continuous ones. The constraints are most often taken into account by the transformation of the original problem into a problem without constraints, using penalty functions, barrier methods and Lagrange multipliers. Here we will discuss in greater detail the methods used especially in the case of continuous variables. The global search methods [10] can be divided into deterministic and stochastic ones. Deterministic methods employ such a heuristic as modification of search trajectories in the trace-based methods, as well as introduction of penalties to avoid regions where no optimal solutions exist. Covering methods [11] isolate a region that does not contain a global optimum and discard it, not searching there any further. Thus the search region is reduced. Obtaining the solution requires a very thorough search of the space, that is, these methods are very time-consuming if the size of the problem is large. Branch-and-bound and interval methods recursively divide the search region into smaller sub-regions and exclude the regions that do not contain the optimal solution. They are covering methods that estimate the criterion function's lower boundaries in the search sub-regions, allowing to estimate the quality of the local minimum. Combining this with numerically verifiable optimality conditions, they allow to confirm the global optimality of the best obtained solution. However, in order to guarantee the quality of the solution, the problem must satisfy the Lipschitz conditions. In the worst case, they demand exponentially increasing computational resources and therefore are very time-consuming. In general, this branch-and-bound principle may be successfully employed in other heuristical methods. However, if the search region is large, these methods are not efficient.

Generalized descent methods [12] continue the search trajectory after a local minimum is found. In the first approach, the trajectory methods modify the differential equation that describes the local descent trajectory in such a way that the local minimum can be left. Their weakness is a large number of calculations that must be carried out in regions that are not promising. In the second approach, the criterion function is modified by imposing a penalty so that the algorithm would not return to an already found local minimum. The weakness consists in the fact that the more local minimums are found, the more difficult it becomes to minimize the modified criterion function.

Thus deterministic methods may be divided into point-based methods that calculate the function at discrete points (for example, the generalized descent methods) and region-based methods that calculate the function constraints in compact sets (for example, the covering methods). The point-based methods are unreliable but require less calculation.

The stochastic global optimization methods rely on probabilities by making decisions in searching for extremums. Random search methods include pure random search with single or multiple starts, random search along a line, adaptive random search, partitioning into subsets, substitution of the worst point, evolutionary algorithms and simulated annealing [10]. The simplest way of getting out of a local minimum is to restart. The cluster or grouping methods [13] employ cluster analysis to avoid the already found local minimums. There are two strategies for grouping points around local minimums: 1) retaining only points with relatively small function values, and 2) transferring every point to the local minimum, making only few local search steps. They work badly if the surface of the function is very rough or if the search is captured in a deep ravine surface of local optimums. Methods that are based on stochastic models employ random variables to simulate the unknown values of the criterion function. The Bayesian method [14] is based on a random function and minimizes the expected deviations from the global minimum estimation. Its efficiency is not high.

Simulated annealing [15] uses an analogy with physical phenomena that occur by heating and then slowly cooling metallic objects; a more homogenous crystalline state is obtained, in which the free energy of the base substance has a global minimum. The role of temperature is important, since it allows the system to reach its lowest energy state with a probability according to Boltzmann’s exponential law. In such a way it is possible to step over the energy sub-barriers, which otherwise would have forced the system to remain at the local minimum. Similarly to physical annealing, convergence in simulated annealing may be slow. Therefore many improvements are used to speed up the process.

Genetic algorithms (GA) [16–18] use an analogy to biological evolution, allowing mutations and crossovers between good local optimum candidates in the hope that ever better optimums will be found. In each search stage a configuration of all populations is maintained. Mutations are carried out locally, while crossover operators ensure the possibility to leave the region of the local minimum. The crossover laws have a large probability of creating an offspring of similar or better fitness. The efficiency of GAs depends on correct conditions of selection and crossovers. Interchanging of the coordinates is sufficiently good if the coordinates have a nearly independent influence on the fitness, but if the influence is strongly correlated (as it is with functions with deep narrow ravine surfaces that are not parallel to coordinate axes) then GAs have great difficulties. A successful configuration of a GA demands a thorough investigation of the actual problem.

Taboo search [19] introduces a taboo list that contains information on the search history. In each iteration a local improvement is made. However, thanks to the taboo list, movement towards already located solutions is forbidden, that is, a taboo has been placed. The taboo list protects from returning to the local optimum which the search has recently left. Taboo search gives good results in the solution of large discrete optimization problems.

Stochastic methods are classified as unreliable. However, these methods are often the only ones that allow the solution of large-scale problems with an acceptable computer time. Currently, in engineering practice namely the stochastic methods are most frequently applied. Therefore the Edaopt optimization algorithm [1,20,21] is compared only with these methods.

4. TEST FUNCTIONS

We try to preserve the names of the test functions as they are indicated in initially sources [16,20].

1. Goldstein Price function

$$f_1(x) = [1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)]$$
$$+ [30 + (2x_1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)] \qquad (1)$$

has 4 local minimums and the global minimum $f_1(0, -1) = 3$ in the domain $-2 \le x_i \le 2,\ \ i = 1, 2.$

2. Griewank2 function

$$f_2(x) = \frac{x_1^2 + x_2^2}{200} - \cos x_1 \cos \frac{x_2}{\sqrt{2}} + 1 \qquad (2)$$

has approximately 500 local minimums and the global minimum $f_2(0, 0) = 0$ in the domain $-100 \le x_i \le 100,\ \ i = 1, 2.$

3. Griewank10 function

$$f_3(x) = \sum_{i=1}^{10} \frac{x_i^2}{4000} - \prod_{i=1}^{10} \cos \frac{x_i}{\sqrt{i}} + 1 \qquad (3)$$

has approximately several thousand local minimums and the global minimum $f_3(0, 0, 0, 0, 0, 0, 0, 0, 0, 0) = 0$ in the domain $-600 \le x_i \le 600,\ \ i = 1, \ldots, 10.$

4. Hartman3 function

$$f_4(x) = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{3} \alpha_{ij}(x_j - p_{ij})^2} \qquad (4)$$

with coefficients

i	α_{ij}	c_i	p_{ij}
1	(3.0, 10.0, 30.0)	1.0	(0.36890, 0.11700, 0.26730)
2	(0.1, 10.0, 35.0)	1.2	(0.46990, 0.43870, 0.74700)
3	(3.0, 10.0, 35.0)	3.0	(0.10910, 0.87320, 0.55470)
4	(0.1, 10.0, 35.0)	3.2	(0.03815, 0.57430, 0.88280)

is analysed in the domain $0 \le x_i \le 1,\ \ i = 1, \ldots, 3.$ The global minimum is $f_4(0.114614, 0.555649, 0.852547) = -3.86278.$

5. Hartman6 function

$$f_5(x) = -\sum_{i=1}^{4} c_i e^{-\sum_{j=1}^{6} \alpha_{ij}(x_j - p_{ij})^2} \tag{5}$$

with coefficients

i	α_{ij}	c_i	p_{ij}
1	(10, 3, 17, 3.5, 1.7, 8)	1.0	(0.1312, 0.1696, 0.5569, 0.0124, 0.8283, 0.5886)
2	(0.05, 10, 17, 0.1, 8, 14)	1.2	(0.2329, 0.4135, 0.8307, 0.3736, 0.1004, 0.9991)
3	(3, 3.5, 1.7, 10, 17, 8)	3.0	(0.2348, 0.1451, 0.3522, 0.2883, 0.3047, 0.6650)
4	(17, 8, 0.05, 10, 0.1, 14)	3.2	(0.4047, 0.8828, 0.8732, 0.5743, 0.1091, 0.0381)

has an unknown number of local minimums in the domain $0 \le x_i \le 1,\ i = 1, \ldots, 6$. The global minimum is $f_5(0.201690, 0.150011, 0.476874, 0.275332, 0.311652, 0.657300) = -3.32237$.

6. Branin function

$$f_6(x) = \left(x_2 - \frac{5.1}{4\pi^2} x_1^2 + \frac{5}{\pi} x_1 - 6 \right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10 \tag{6}$$

has 3 global minimums $f_6(-3.14159, 12.27500) = f_6(3.14159, 2.27500) = f_6(9.42478, 2.47500) = 0.397887$ in the domain $-5 \le x_1 \le 10;\ 0 \le x_2 \le 15$.

7. Camel Back function

$$f_7(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4 \tag{7}$$

has 4 local minimums and 2 global minimums $f_7(0.0898400, -0.712656) = f_7(-0.0898400, 0.712656) = -1.0316285$ in the domain $-5 \le x_i \le 5,\ i = 1, 2$.

8. F8 function

$$f_8(x) = 2(x - 0.75)^2 + \sin(5\pi x - 0.4\pi) - 0.125 \tag{8}$$

has global minimum $f_8(0.0898400, -0.712656) = -1.12323$ in the domain $0 \le x \le 1$.

9. F9 function

$$f_9(x) = -\sum_{j=1}^{5} \{ j \sin[(j+1)x + j] \} \tag{9}$$

has global minimum $f_9(0.0898400, -0.712656) = -12.03125$ in the domain $-10 \le x \le 10$.

10. Hosc45 function

$$f_{10}(x) = 2 - \prod_{i=1}^{10} \frac{x_i}{n!} \tag{10}$$

has global minimum $f_{10}(1,2,3,4,5,6,7,8,9,10) = 1$ in the domain $0 \le x_i \le i$, $i = 1,\ldots,10$.

11. Pshubert1 function

$$f_{11}(x) = \left\{\sum_{i=1}^{5} i\cos[(i+1)x_1 + i]\right\}\left\{\sum_{i=1}^{5} i\cos[(i+1)x_2 + i]\right\} + 0.5\,[(x_1 + 1.42513)^2 + (x_2 + 0.80032)^2] \tag{11}$$

has global minimum $f_{11}(-1.4251, -0.8003) = -186.73091$ in the domain $-10 \le x_i \le 10$, $i = 1, 2$. By changing in (11) the coefficient from 0.5 to 1.0, this function is named Pshubert2 function.

12. Quartic function

$$f_{12}(x) = \frac{x_1^4}{4} - \frac{x_1^2}{2} + \frac{x_1}{10} + \frac{x_2^2}{2} \tag{12}$$

has global minimum $f_{12}(-1.04668, 0) = -0.35239$ in the domain $-10 \le x_i \le 10$, $i = 1, 2$.

13. Leonard Rastrigin [22] *function*

$$f_{13}(x) = x_1^2 + x_2^2 - \cos 18x_1 - \cos 18x_2 \tag{13}$$

has about 50 local minimums and the global minimum $f_{13}(0,0) = -2$ in the domain $-1 \le x_i \le 1$, $i = 1, 2$.

14. Shekel5 function

$$f_{14}(x) = -\sum_{i=1}^{5} \frac{1}{(x - \alpha_i)(x - \alpha_i)^{\mathrm{T}} + c_i} \tag{14}$$

with coefficients

i	α_i	c_i
1	(4.0, 4.0, 4.0, 4.0)	0.1
2	(1.0, 1.0, 1.0, 1.0)	0.2
3	(8.0, 8.0, 8.0, 8.0)	0.2
4	(6.0, 6.0, 6.0, 6.0)	0.4
5	(3.0, 7.0, 3.0, 7.0)	0.4

has 5 local minimums at $f_{14}(\alpha_i) \approx -1/c_i$ and the global minimum $f_{14}(4.00004, 4.00013, 4.00004, 4.00013) = -10.1532$ in the domain $0 \le x_i \le 10$, $i = 1, \ldots, 4$.

15. Shekel7 function

$$f_{15}(x) = -\sum_{i=1}^{7} \frac{1}{(x-\alpha_i)(x-\alpha_i)^{\mathrm{T}} + c_i} \tag{15}$$

with coefficients

i	α_i	c_i
1	(4.0, 4.0, 4.0, 4.0)	0.1
2	(1.0, 1.0, 1.0, 1.0)	0.2
3	(8.0, 8.0, 8.0, 8.0)	0.2
4	(6.0, 6.0, 6.0, 6.0)	0.4
5	(3.0, 7.0, 3.0, 7.0)	0.4
6	(2.0, 9.0, 2.0, 9.0)	0.6
7	(5.0, 5.0, 3.0, 3.0)	0.3

has 7 local minimums at $f_{15}(\alpha_i) \approx -1/c_i$ and the global minimum $f_{15}(4.00057, 4.00069, 3.99949, 3.99961) = -10.4029$ in the domain $0 \le x_i \le 10$, $i = 1, \ldots, 4$.

16. Shekel10 function

$$f_{16}(x) = -\sum_{i=1}^{10} \frac{1}{(x-\alpha_i)(x-\alpha_i)^{\mathrm{T}} + c_i} \tag{16}$$

with coefficients

i	α_i	c_i
1	(4.0, 4.0, 4.0, 4.0)	0.1
2	(1.0, 1.0, 1.0, 1.0)	0.2
3	(8.0, 8.0, 8.0, 8.0)	0.2
4	(6.0, 6.0, 6.0, 6.0)	0.4
5	(3.0, 7.0, 3.0, 7.0)	0.4
6	(2.0, 9.0, 2.0, 9.0)	0.6
7	(5.0, 5.0, 3.0, 3.0)	0.3
8	(8.0, 1.0, 8.0, 1.0)	0.7
9	(6.0, 2.0, 6.0, 2.0)	0.5
10	(7.0, 3.6, 7.0, 3.6)	0.5

has 10 local minimums at $f_{16}(\alpha_i) \approx -1/c_i$ and the global minimum $f_{16}(4.00075, 4.00059, 3.99966, 3.99951) = -10.5364$ in the domain $0 \le x_i \le 10$, $i = 1, \ldots, 4$.

17. Function used by Vilnis Eglajs [20]

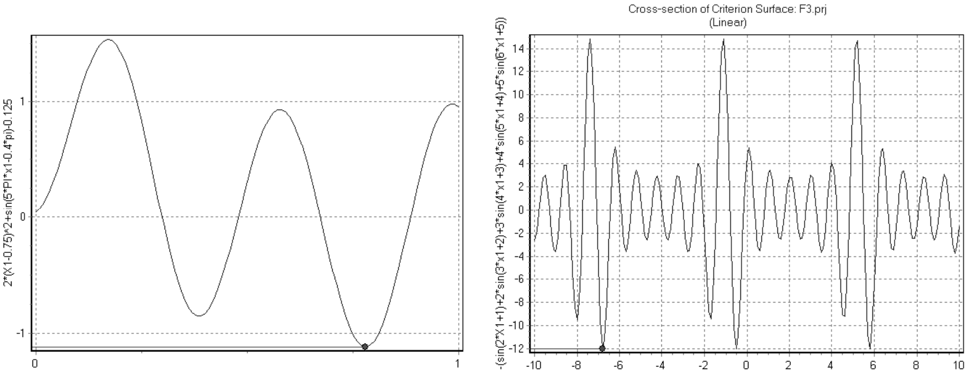
$$f_{17}(x) = \sum_{i=1}^{4} (x_i^2 + ix_i) + 40 \prod_{j=1}^{4} \sin x_j . \tag{17}$$

There is a great number of local minimums in the domain $-50 \le x_i \le 50$, $i = 1, \ldots, 4$. The global minimum is $f_{17}(1.471979, -1.543479, -1.567407, -1.591337) = -42.92577$.

5. ANALYSIS OF OBTAINED RESULTS

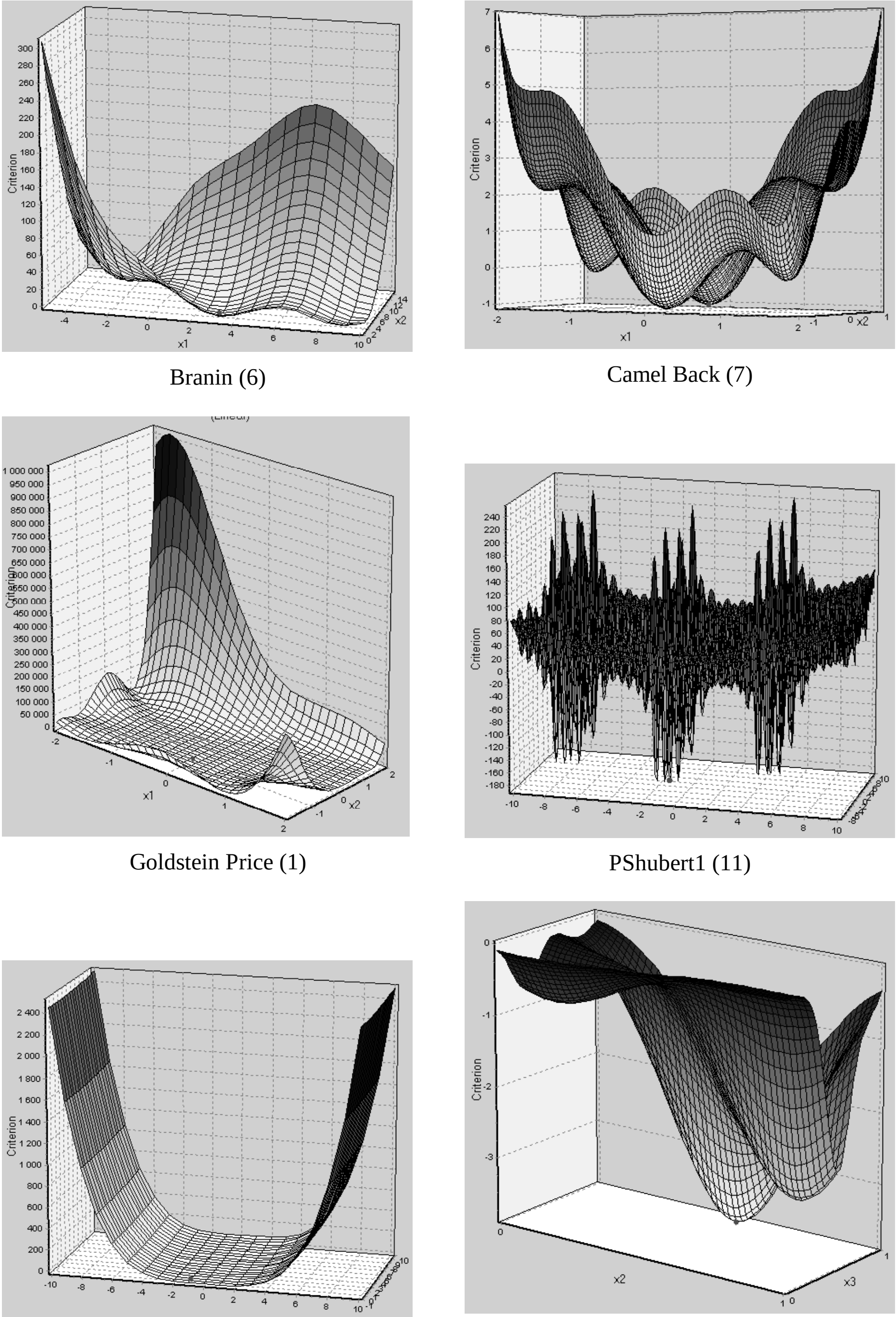
To evaluate the efficiency of the random search two-phase multistart optimization algorithm Edaopt [1,21], it was tested by solving a set of test problems. Figures 1–3 show the test functions or their characteristic sections with the global minimums found with Edaopt. They were found with a practically 100% success in all cases. In solving some of the problems (for example, Griewank10, which contains several thousands of local minimums in a 10 parameter space), the algorithm occasionally converged to extremes close to the global optimum rather than to the global optimum itself. In this way in all cases the most promising optimum region was found. This has a great significance, since in practical problems it is very important not to miss these regions.

Table 1 shows results obtained with Edaopt, standard (GA) and improved (GA+) genetic algorithms. To make objective comparison possible, it was necessary to adhere to the calculation conditions given in literature [16]. Since stochastic methods were evaluated, the minimum of each function was found 100 times. The search was considered successful if the global minimum was found with the given accuracy. Table 1 shows that, standard GA guarantees a 100%



F8 (8) F9 (9)

Fig. 1. Minimums of the test functions, found with Edaopt.



Branin (6)

Camel Back (7)

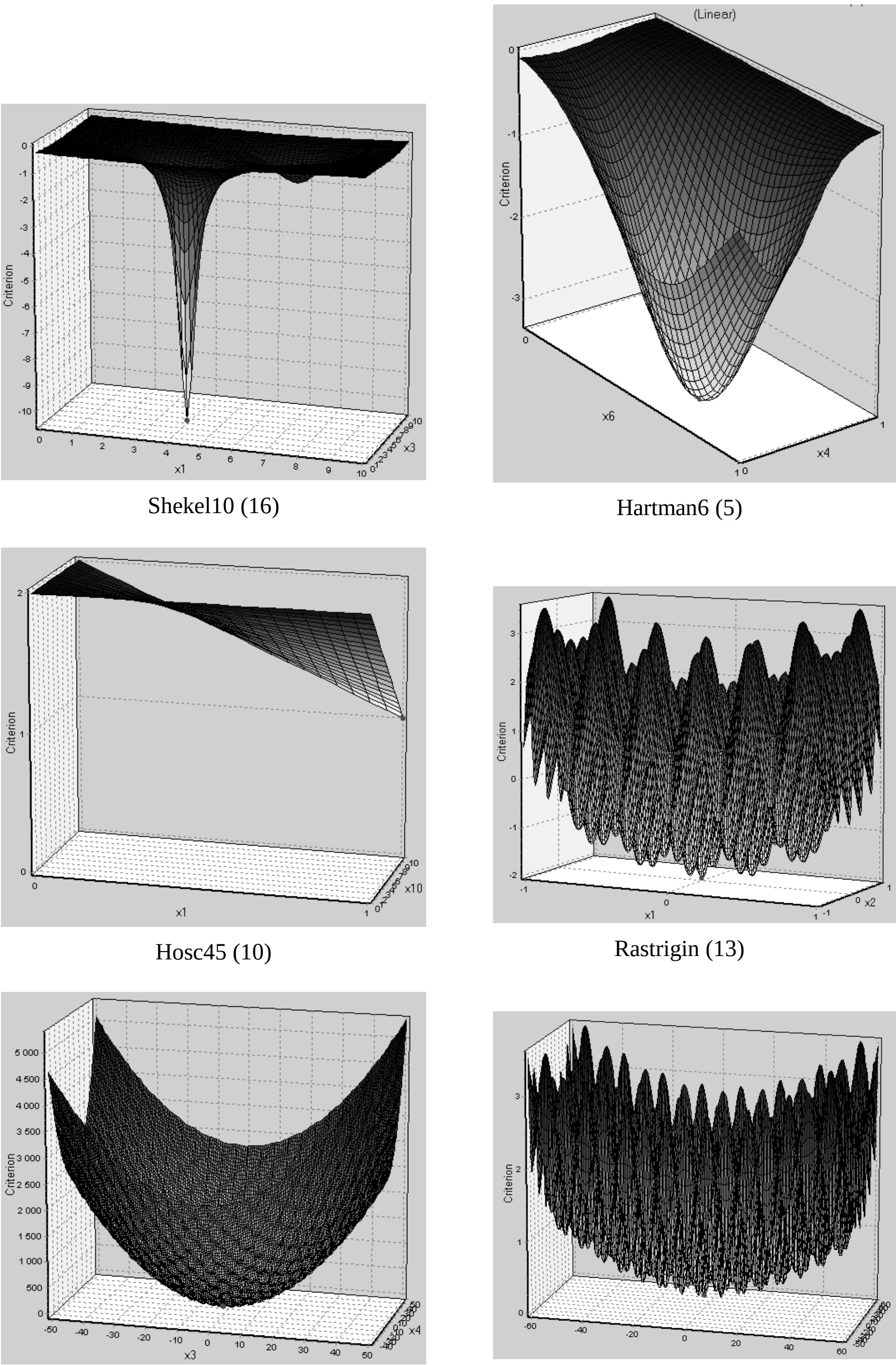
Goldstein Price (1)

PShubert1 (11)

Quartic (12)

Hartman3 (4)

Fig. 2. Minimums of test functions, found with Edaopt.



Shekel10 (16)

Hartman6 (5)

Hosc45 (10)

Rastrigin (13)

Eglajs (17)

Griewank10 (3)

Fig. 3. Minimums of test functions, found with Edaopt.

Table 1. Characteristics of optimization algorithms (GA and GA+ results taken from [16]; success of Edaopt in all cases is 100%)

Function		Number of calculations			Absolute error			Success %	
No	Name	GA	GA+	Edaopt	GA	GA+	Edaopt	GA	GA+
1	Goldstein Price	8 185	4 632	816	0.229	0.013	0.0127	59	100
4	Hartman3	1 993	1 680	1 150	0.025	0.020	0.0197	94	100
5	Hartman6	19 452	53 792	650 475	0.144	0.033	0.0118	23	92
6	Branin	8 125	2 040	593	0.003	0.002	0.0018	81	100
7	Camel Back	1 316	1 316	346	0.005	0.005	0.0048	98	100
8	F8	5 566	784	156	0.000	0.000	0.0000	100	100
9	F9	5 347	744	131	0.001	0.014	0.0002	100	100
10	Hosc45	11 140	126 139	14 020	1.000	0.392	0.0000	0	2
11	PShubert1	7 192	8 853	32 849	4.563	0.983	0.4467	63	100
11	PShubert2	7 303	4 116	1 430	4.772	0.986	0.8593	59	100
12	Quartic	8 181	3 168	1 134	0.003	0.002	0.0018	83	100
14	Shekel5	7 495	36 388	500 187	6.067	0.072	0.0521	1	97
15	Shekel7	8 452	36 774	390 185	4.856	0.165	0.0939	0	98
16	Shekel10	8 521	36 772	390 175	5.126	0.074	0.0950	0	100

success (global minimum found in all 100 attempts) only with one-dimensional functions *F8* and *F9*, while with other functions the success is modest and in some cases GA is entirely unable to find the global minimum.

Significantly better results are obtained with GA+, the "heuristic coefficients" of which have been improved. From practice it is known that for specific test problems these coefficients may be fitted in such a way that global extremums for these functions can be found with only a few iterations, while for the optimization of other functions the algorithm becomes practically useless. Nevertheless, the Edaopt algorithm is exactly compared with GA+. The table shows data only for the functions with a number of parameters up to 10, for which paper [16] gives data for GA+. By optimizing *F8*, *F9*, Branin, Camel Back, Goldstein Price, PShubert2 and Quartic functions, the accuracy of finding the global minimum is higher and simultaneously the number of function calculations is 3 to 5.5 times lower with Edaopt than with GA+. Thus the efficiency of Edaopt is definitely higher. This is especially obvious by optimization of the function Hosc45, where the location of the global minimum with Edaopt requires 9 times less points (function calculations), and the percentage of success is 100% as compared to 2% for GA+. That shows high reliability of Edaopt, at the same time signifying rather unsuccessful fit of coefficients with the GA+ method.

The obtained results do not, however, show that GA is not suitable for global search procedures. On the contrary, GA, simulated annealing and taboo search are among the most effective methods, since finding of a practical solution is never limited with a few search series. It is always connected with a thorough and detailed investigation, namely, building of sensitivity curves and evaluation of functioning stability in optimality regions, etc.

It should be noted that it is hard to achieve a 100% success rate with stochastic search methods, since there is always a probability of carrying out an inefficient search with a limited number of points. For example, when attempting to find global minimums for the Hartman6 and Shekel functions with 5 and 7 local minimums in a 4 parameter space with 100% success, it turned out that the number of points necessary using Edaopt is about ten times greater than using the GA+ method with a corresponding 92%, 97% and 98% success. This fact does not indicate the superiority of one or another method, but it shows that in order to achieve reliability close to 100%, the minimal number of points must be suitable. To achieve a more or less objective comparison, it would be necessary to ensure a precise coincidence of absolute errors and percentage of successes. Since the aim was to obtain not a formal numerical evaluation, but a qualitative evaluation of the algorithms, such a comparison was not carried out. Moreover, with the Edaopt standard interface the search is not terminated on a given precision, but on the computer precision (10-byte float point calculation), and the only parameter necessary to provide is the maximal number of iterations (no other parameters like “heuristic coefficients” are required). It must be noted that with some test problems we had to carry out the comparison with high relative error level 1%, when the deviation of the parameters from their optimal values may be significant. Manipulation with precision may bring a great amount of subjectivity into the evaluation.

Finally, it should be noted that the global minimums for the Griewank2, Rastrigin and Eglajs functions were located with Edaopt without difficulty. Searching for minimums for the Branin function, each of the 3 global minimums was found with a 1/3 probability, while for the Camel Back function both global minimums were found with a 1/2 probability.

5. CONCLUSIONS

The solution of a wide scope of test problems has shown that the Edaopt algorithm gives a significantly higher reliability in searching for global optimums in comparison with traditional standard stochastic search algorithms. In most cases high reliability is obtained with a noticeably smaller amount of computation. However, that is less important when the metamodel approach is applied. In cases when the RSM is used, the reliability of the optimum finding is the most important characteristic of the optimization algorithm. Besides, the Edaopt software allows the user to orientate himself visually in the seemingly endless optimization jungle.

ACKNOWLEDGEMENT

This work was partly supported by the European Community Competitive and Sustainable Growth Programme project AURORA and 6th Framework Programme research project COCOMAT (http://www.cocomat.de).

REFERENCES

1. Auzins, J., Janushevskis, A. and Rikards, R. Software tool Edaopt for optimization of complex systems. In *Book of Abstracts of XXXI International Conference "Advanced Problems in Mechanics"*. St. Petersburg, 2003, 24–25.
2. Janushevskis, A., Lavendelis, E. and Onzevs, O. Method for optimisation of random vibrations of nonlinear mechanical system. In *Solid Mechanics and its Applications*. Kluwer, Dordrecht, 2000, **73**, 159–168.
3. Myers, R. H. and Montgomery, D. C. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments*. J. Wiley, New York, 2002.
4. Auzins, J. and Janushevskis, A. New experimental designs for metamodelling and optimization. In *Proc. 5th World Congress on Computational Mechanics* (Mang, H. A., Rammerstorfer, F. G. and Eberhardsteiner, J., eds.). Vienna, 2002, 10 pages.
5. Rikards, R. and Auzins, J. Response surface method in optimum design of lightweight composite structures. In *Computational Analysis of Composite Lightweight Structures in Aerospace Applications* (Rolfes, R. and Teßmer, J., eds.). DLR Institut für Strukturmechanik, Braunschweig, 2003, 111–120.
6. Waszcyszyn, Z. and Ziemianski, L. Neurocomputing in identification problems of structural mechanics. In *Book of Abstracts of the 5th EUROMECH Solid Mechanics Conference ESMC 5* (Aifantis, E. C., ed.). Giapoulis, Thessaloniki, 2003, 299.
7. Simpson, T. W., Peplinski, J. D., Koch, P. N. and Allen, J. K. Metamodels for computer-based engineering design: survey and recommendations. *Eng. Comput.*, 2001, **17**, 129–150.
8. Dyn, N., Levin, D. and Rippa, S. Numerical procedures for surface fitting of scattered data by radial basis functions. *SIAM J. Sci. Stat. Comput.*, 1986, **7**, 639–659.
9. Friedman, J. H. Multivariate Adaptive Regression Splines. *Ann. Stat.*, 1998, **19**, 1–67.
10. Horst, R. and Pardalos, P. M., eds. *Handbook of Global Optimization*. Kluwer, New York, 1995.
11. Horst, R. and Tuy, H. *Global Optimization: Deterministic Approaches*. Springer–Verlag, Berlin, 1993.
12. Vincent, T. L., Goh, B. S. and Teo, K. L. Trajectory – following algorithms for min–max optimization problems. *J. Optim. Theory Appl.*, 1992, **75**, 501–519.
13. Torn, A. and Viitanen, S. Topological global optimization. In *Recent Advances in Global Optimization* (Floudas, C. A. and Pardalos, P. M., eds.). Princeton Univ. Pr., 1992, 385–398.
14. Mockus, J. Application of Bayesian approach to numerical methods of global and stochastic optimization. *J. Global Optim.*, 1994, **4**, 347–365.
15. Aarts, E. and Korst, J. *Simulated Annealing and Boltzmann Machines*. J. Wiley, New York, 1989.
16. Andre, J., Siarry, P. and Dognon, T. An improvement of the standard genetic algorithm fighting premature convergence in continuous optimization. *Adv. Eng. Softw.*, 2001, **32**, 49–60.
17. Koza, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Pr., Cambridge, 1993.
18. Renders, J. M. and Flasse, S. P. Hybrid methods using genetic algorithms for global optimization. *IEEE Trans. Syst. Man Cybern. B: Cybern.*, 1996, **26**, 243–258.
19. Battiti, R. The Reactive Tabu Search. *ORSA J. Comput.*, 1994, **6**, 126–140.
20. Auzins, J., Janushevskis, A. and Onzevs, O. Optimization of multibody vibration response by global search procedure. In *Proc. European Conference on Computational Mechanics-99*. Munich, 1999, 17 pages, CD-ROM.
21. Auzins, J., Janushevskis, A. and Rikards, R. Software tools for experimental design, metamodelling and optimization. In *Book of Abstracts of the 5th EUROMECH Solid Mechanics Conference ESMC 5* (Aifantis, E. C., ed.). Giapoulis, Thessaloniki, 2003, 182.
22. Rastrigin, L. and Ripa, K. *Automation Theory of Random Search*. Zinatne, Riga, 1973 (in Russian).

Globaalsete optimeerimisprotseduuride võrdlev analüüs

Alexander Janushevskis, Teodor Akinfiev, Janis Auzins
ja Alexander Boyko

Esitatakse lühike ülevaade globaalsetest optimeerimismeetoditest ja nende tarkvarast. Erinevaid meetodeid kasutades on lahendatud rida standardseid ülesandeid ning on võrreldud saadud globaalsete ekstreemväärtuste usaldatavust. Tulemused näitavad autorite poolt loodud optimeerimisalgoritmi *Edaopt* kõrget efektiivsust teiste kaasaegsete algoritmidega võrreldes. Seejuures on enamikul vaadeldud juhtudest globaalne optimum saavutatud tunduvalt väiksema arvutiajaga.