International Congress on Evolutionary Methods for Design,
Optimization and Control with Applications to Industrial Problems
EUROGEN 2003
G. Bugeda, J. A- Désidéri, J. Periaux, M. Schoenauer and G. Winter (Eds)

MULTI-OBJECTIVE ROBUST OPTIMIZATION USING IOSO TECHNOLOGY PART I: MAIN FEATURES

Igor N. Egorov-Yegorov*, Gennany V. Kretinin*,
Igor A. Leshchenko*, Sergey V. Kuptcov*

*IOSO Technology Center
Vekovaya 21, office 203, Moscow, 109544, Russia
e-mail: {egorov, kretinin, leschenko, kuptcov}@iosotech.com

Key words: evolutionary algorithm, robust design optimization, multi-objective optimization, multi-disciplinary optimization

Abstract: *This paper presents the main capabilities of IOSO (Indirect Optimization based on Self-Organization) technology algorithms, tools and software, which can be used for the optimization of complex systems and objects, including air engine and aircraft. IOSO implements a novel evolutionary response surface strategy. This strategy differs significantly from both the traditional approaches of nonlinear programming and the traditional response surface methodology. That is why, IOSO algorithms have higher efficiency, provide a wider range of capabilities, and are practically insensitive with respect to the types of objective function and constraints. They could be smooth, non-differentiable, and stochastic, with multiple extrema, with the portions of the design space where objective function and constraints could not be evaluated at all, with the objective function and constraints dependent on mixed variables, etc. The capabilities of IOSO software are demonstrated using well-known test problems of solving complex single-objective and multi-objective problems.*

1 INTRODUCTION

The creation of any complex technical object or system assumes the search for combinations of parameters for that system that would provide maximum or minimum values of one or more efficiency indicators. To accomplish this one should, at the designing stage, conduct complex research to evaluate the influence of design parameters on the objects efficiency. A distinctive feature of solving such tasks is that the object can be investigated at different levels of complexity. With aircraft engines, for example, both internal efficiency indicators (such as specific fuel consumption or specific weight) and higher-level indicators (such as lifecycle cost or direct maintenance charges for aircraft equipped with the engine being investigated) can be used. Obviously, the most reliable results can be obtained by using high-level criteria, which take into consideration the efficiency of the whole system and not only one of its particular elements. Hence, to optimize parameters of an object that is part of a higher-level system, a multidisciplinary approach should be used. This allows one to consider multiple disciplines of analysis when designing the system.

This paper presents the basic capabilities and features of algorithms, tools, and software that utilize various IOSO (Indirect Optimization based on Self-Organization) Technology algorithms [1]. This software product is designed to solve a wide range of real-life problems in various fields of science and technology [1]. This paper presents some samples of IOSO Technology for different optimization test problems.

2 SUMMARY OF IOSO ALGORITHM

This *multi*-objective optimization problem maximizes a vector of n objective functions

$$\max\ F_i(\bar{x}) \qquad \text{for } i = 1, ..., n \tag{1}$$

subject to a vector of inequality constraints

$$g_j(\bar{x}) \le 0 \qquad \text{for } j = 1, ..., m \tag{2}$$

and a vector of equality constraints

$$h_q(\bar{x}) = 0 \qquad \text{for } q = 1, ..., k \tag{3}$$

Conditions (2) and (3) confirm set D which lies in the search of one or several vectors $x^* \in D$ that ensures the best efficiency.

Our approach is based on the widespread application of the response surface technique, which depends upon the original approximation concept, within the frameworks which adaptively use global and middle-range multi-point approximation. One of the advantages of the proposed approach is the possibility of ensuring good approximating capabilities using the minimum amount of available information. This possibility is based on self-organization and evolutionary modeling concepts[1]. During the approximation, the approximation function structure is being evolutionarily changed, so that it allows for the successful approximation of the optimized functions and constraints having sufficiently complicated topology. The obtained approximation functions can be used by multi-level procedures with the adaptive change of simulation levels within both single and multiple disciplines of object analysis, and also for the solution of their interaction problems.

Every iteration of IOSO consists of two steps. The first step is the creation of an analytical approximation of the objective function(s). Each iteration represents a decomposition of the initial approximation function into a set of simple approximation functions. The final response function is a multi-level graph. The second step is the optimization of this approximation function. This approach allows for corrective updates of the structure and the parameters of the response surface approximation. The distinctive feature of this approach is an extremely low number of trial points to initialize the algorithm. The obtained response functions are used in the multi-level optimization while adaptively utilizing various single and multiple discipline analysis tools that differ in their level of sophistication. The optimization of the response function is performed only within the current search area during each iteration of IOSO.

This step is followed by a direct call to the mathematical analysis model or an actual experimental evaluation for the obtained point. During the IOSO operation, the information concerning the behavior of the objective function in the vicinity of the extremum is stored, and the response function is made more accurate only for this search area. While proceeding from one iteration to the next, the following steps are carried out: modification of the experiment plan; adaptive selection of the current extremum search area; choice of the response function type (global or middle-range); transformation of the response function; modification of both parameters and structure of the optimization algorithms; and, if necessary, selection of new promising points within the researched area. Thus, a series of approximation functions for a particular objective of optimization is built during each iteration. These functions differ from each other according to both structure and definition range. The subsequent optimization of these approximation functions allows us to determine a set of vectors of optimized variables.
When solving robust design optimization we have another statement:

$$\max\ F_i(\bar{x}, e) \qquad \text{for } i = 1, ..., n \tag{4}$$

subject to a vector of inequality constraints

$$g_j(\bar{x}, e) \leq 0 \qquad \text{for } j = 1, ..., m \tag{5}$$

and a vector of equality constraints

$$h_q(\bar{x}, e) = 0 \qquad \text{for } q = 1, ..., k \tag{6}$$

The attempt to include uncertainties while design problem formalization results in the necessity to consider relations: $x = x(\bar{x},\xi_x)$; $e = e(\bar{e},\xi_e)$; $f(x,e) = \Psi(\bar{f}(x,e),\xi_f(x,e))$, where $\bar{x},\bar{e},\bar{f}$ are ideal vectors of variable parameters, environmental conditions and the ideal mathematical model; $\xi = (\xi_x,\xi_e,\xi_f)$ is the vector of random values including uncertainties in implementation of variable parameters, environment conditions and the mathematical model accuracy. Generally, to solve a robust design optimization (RDO) problem one must be able to determine the system efficiency values $y = f(x,e)$ for given values of $\bar{x},\bar{e}$, and hence to know the laws of distribution of ξ vector components and functional dependence of $\Psi(\bar{f},\xi_f)$. It means that for RDO we must define some probability criteria (objectives and constraints) for each iteration.

For example, when solving robust design optimization the efficiency values $y = f(x,e)$ are random ones. In this case it is necessary to use probabilistic optimization criteria $\tilde{y}(\bar{x})$. Let us consider some of the probabilistic criteria used when the efficiency index $y(x,e)$ is to be minimized.

1. $\tilde{y}(\bar{x}) = M\{ f(x,e)\}$ - mean value of efficiency;
2. $\tilde{y}(\bar{x}) = \sigma\{ f(x,e)\}$ - magnitude of efficiency value deviation;
3. $\tilde{y}(\bar{x}) = P_c\{ f(x,e) \le y_p \}$ - probability that efficiency value is no worse than the one given;
4. $P_c\{ f(x,e) \le \tilde{y}(\bar{x})\} \ge P_p$ - efficiency value ensured with probability is no less than the one given.

Each of these criteria reflects different robust properties of the project. Fig.1 shows a sample graph of function $y = (x-1)^2 \cdot ((x-5)^4 + 1)$ and probabilistic criteria No. 1, 2 for the case of $x = \bar{x}(1+\xi)$, where ξ is a normally distributed random value with the average distribution equaling 0, and the dispersion equaling 0.3.

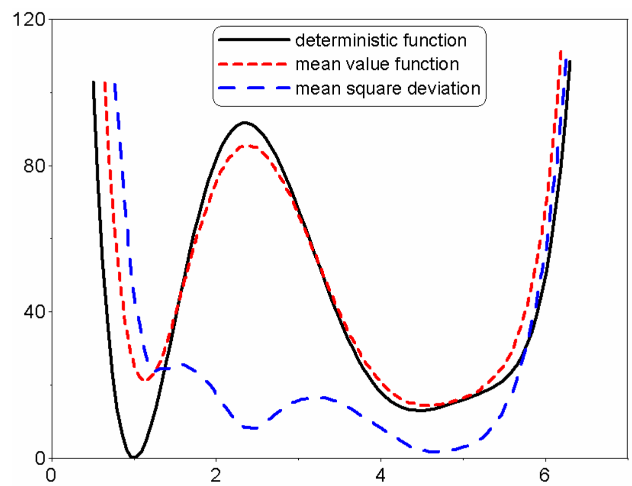


Figure 1. Examples of probability criteria.

Suppose this equation represents some mathematical model of a system where efficiency index y is to be minimized. It is obvious that the minimum of given function (deterministic solution) is reached at $x = 1$. Suppose that the design parameter x is a random value depending on, say, technological spread, than the minimum of the average value of efficiency (probabilistic criterion No.1) is reached as early as $\bar{x} \approx 4.6$. The location of the minimum by probabilistic criterion No.2 also differs from the deterministic solution. The considered example leads to an important conclusion that an RDO problem should not be reduced to the problem of “correction” of the solution obtained through the deterministic approach since extremums of probabilistic criteria may substantially differ, by design parameters, from extremum of deterministic criterion. This, in fact, means that in the vicinity of the deterministic solution there might be a solution improving robust properties of the project. To achieve maximal robust properties the problem must be solved by probabilistic criteria.

It should be noticed that in the case of robust design optimization all of the constrained parameters of the system $g_j(x,e) \le 0,\ j = \overline{1,w}$ are also random values, and it is necessary to evaluate their probabilistic criteria. To simplify procedures of optimization, problem solution, and analysis of the obtained results some integral value can be introduced. This value is the probability of observing of all given constraints - $P_o(\bar{x}) = P(x \in D)$. Thus, from a mathematical point of view the problem of robust design optimization can be formulated as follows: for given laws of distribution of components of vector ξ a vector $\bar{x}^*$ is to be found that will ensure $\tilde{y}(\bar{x}^*) = \underset{P_o(\bar{x}) \ge P_{0p}}{extr}\{\tilde{y}(\bar{x})\}$, where P_{0p} is the given probability of constraints observed.

Value $P_0(\bar{x})$ may be regarded as both additional criterion of stochastic optimization and can be used to calculate other criteria, for example:

5. $\tilde{y}(\bar{x}) = P_\Sigma\{f(x,e) \le y_p\}$ - integrated probability that the value of efficiency is not worse than a given one ($P_\Sigma = P_c \cdot P_0$).

6. $P_\Sigma\{f(x,e) \le \tilde{y}(\bar{x})\} \ge P_p$ - value of efficiency ensured with integrated probability no less than a given one ($P_\Sigma = P_c \cdot P_0$).

The specific features of the optimization task determine the selection of probabilistic criterion. At the same time it is necessary to take into account that when applying criteria No.3…No.6, extra research is to be done to set correct values of P_p and y_p, since in this case a situation with no solution to the set problem may occur. It is quite obvious that the formulated probabilistic criteria can contradict one another. Thus, it will be sufficient to simply ensure high robust properties for a project (high values for P_c, P_0, P_Σ) by artificially decreasing requirements for efficiency value y_p. Hence, robust design optimization problems (even for a single chosen efficiency) are in essence multicriteria ones and appropriate techniques to solve them should be used.

The main problem occurring while solving robust design optimization problems is determining probabilistic criteria values. There are various approaches to solving this problem [2,3]. It is well known that defining probability criteria needs large CPU time. Note, any optimization algorithm uses the iterative procedure with a large number of objective functions and constraints evaluations. The total time of solution for any optimization problem, an RDO problem in particular, can be defined as time of calculation of criteria for one value of variable parameters multiplied by the necessary number of such calculations $T_\Sigma = t_{cr} \cdot N_{calc}$. This simple formula indicates the great importance of choosing an appropriate optimization technique.

Attempts to decrease the number of calculations for values of probabilistic criteria N_{calc} leads to the necessity to use “fast” gradient methods of optimizations. However, the efficiency of gradient methods substantially decreases when there is a “noisy” object function under investigation. Hence their usage requires high accuracy of assessment of probabilistic criteria, which, in its turn, leads to a substantial increase in t_{cr}. In addition, the use of gradient techniques places substantial restrictions on the topology of object functions, hence limiting their applicability when solving practical problems. The use of direct optimization techniques could be much more advantageous. They have higher noise immunity and allow for a substantial decrease in the time t_{cr} under minor increase in N_{calc}.

When solving real-life RDO problems we use basic algorithms of IOSO technology with some modifications. The effective noise-proof feature of these algorithms enables us to solve RDO problems by means of the Monte-Carlo technique with an extremely small amount of statistical tests at each search iteration. Figure 2 shows an example of the efficiency of IOSO algorithms when solving optimization problems for “noisy” object functions. It is evident that even under intensive noise IOSO algorithms reliably provide an effective extremum region. With the use of such an approach, a more accurate assessment of probabilistic criteria is done for the solution of optimization problems at the next stage of analysis.

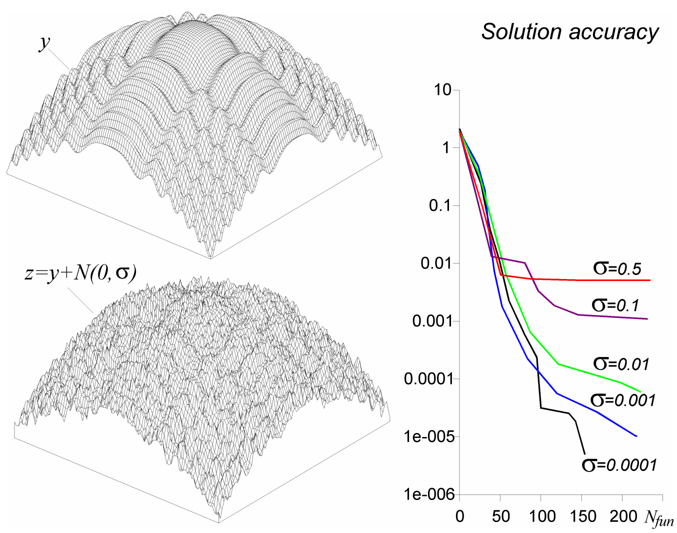


Figure 2. Example of the IOSO algorithms noise-immunity.

We developed the family of IOSO algorithms (Fig.3), which can be used by many scientific fields to solve real-life optimization problems of complex technical systems and objects [3,4,7,10].

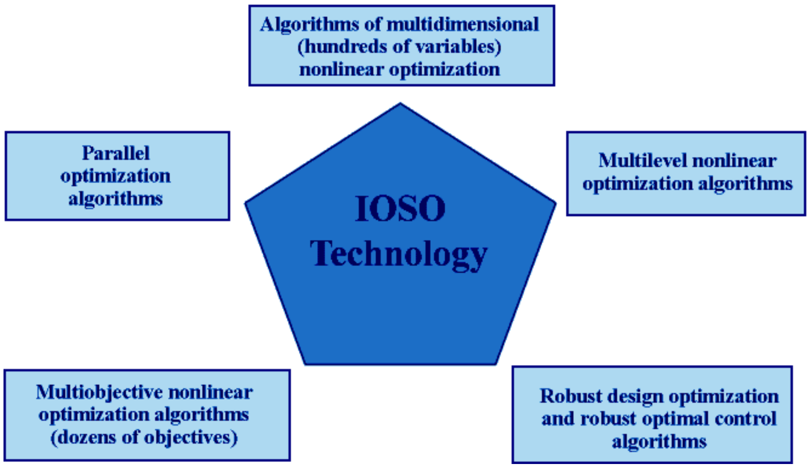


Figure 3. A family of IOSO algorithms.

3 RESULTS OF TESTING IOSO ALGORITHMS

3.1 Comparative analysis of efficiency

In the present work a comparison has been made of IOSO Technology algorithms for comparing the efficiency with that of up-to-date nonlinear optimization methods. For comparison, we chose well-known test functions [4,11], which were complex, nonlinear problems of conditional and unconditional optimization. When comparing optimization methods, we considered one with complex criteria. This criterion evaluates the efficiency of optimization strategy taking into account the dimensionality of the problem, the number and type of constraints (equality or inequality), the accuracy of solution determination and constraints as well as the number of function evaluations required for obtaining the solution. In Fig. 4 the main results are shown. One can see that the IOSO basic algorithm can compete successfully with well-known optimization methods.

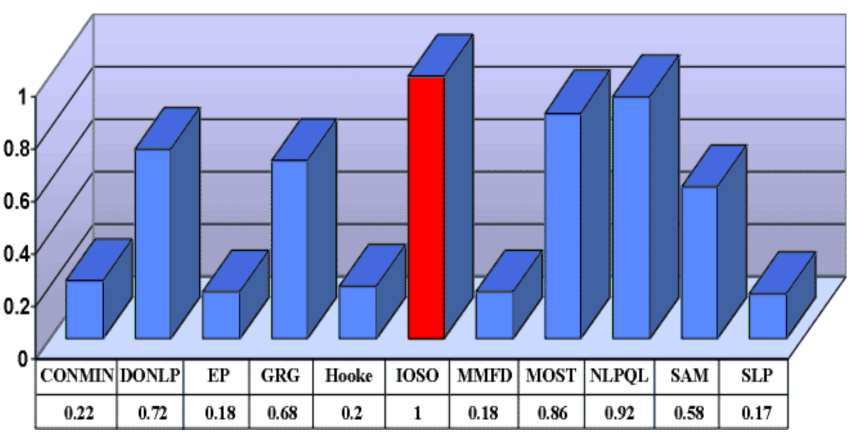


Figure 4. Comparison of IOSO basic algorithm efficiency with other optimization methods.

The experience of applying this method shows that it is possible to solve a wide range of scientific problems by the adaptive building of the algorithm for the particular real optimization problem during the extremum search by the computer itself.

3.2 The main features of IOSO algorithm

IOSO Technology implements the new evolutionary response surface methodology. This methodology differs significantly from both the traditional approaches of nonlinear programming and the traditional response surface approach. Because of these differences IOSO Technology algorithms have higher efficiency, provide a wider range of capabilities, and are practically insensitive with respect to the types of objective function and constraints: smooth, non-differentiable, stochastic, with multiple optima, with the portions of the design space where objective function and constraints could not be evaluated at all, with the objective function and constraints dependent on mixed variables, etc. (see Fig. 5).

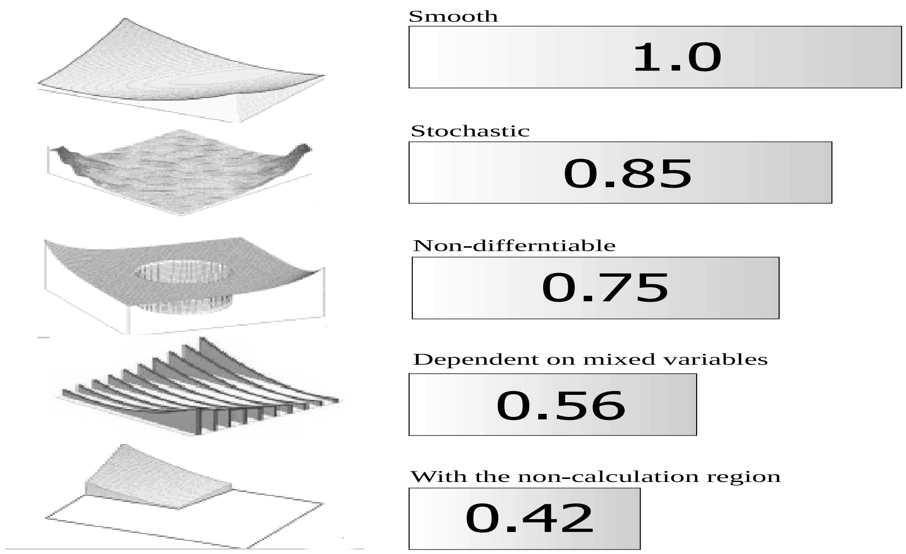


Figure 5. IOSO algorithms efficiency for different objective functions.

Software and tools of IOSO Technology consist of several independent algorithms intended for solving the following nonlinear optimization problems:

- Single-objective
- Multi-objective
- Parallel single and multi-objective
- Multilevel with adaptive change of the model fidelity (low-, middle-, high fidelity models)
- Robust design optimization and robust optimal control.

All IOSO technology algorithms were developed according to the single concept of formulating optimization problems, providing initial data, data exchange with the user’s program, and analysis of the obtained results. An important feature of IOSO Technology software is its capability to solve a wide range of analysis of the obtained results.

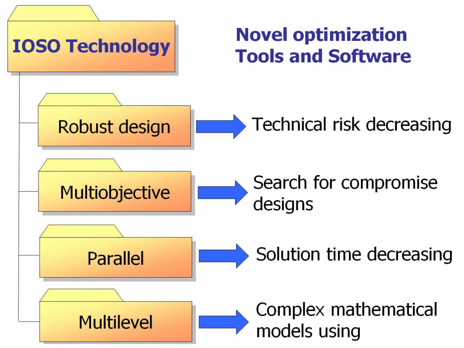


Figure 6. Tools and Software of IOSO technology.

An important feature of IOSO Technology software is its capability to solve a wide range of optimization problems having different types of objective functions.

IOSO Technology Tools implement highly efficient evolutionary self-organizing algorithms. The efficiency is guaranteed by internal adaptive choice of the algorithm suitable for each particular problem. This feature results in solving complex optimization problems with a minimal number of evaluations of the system mathematical model [3,4,7,10]

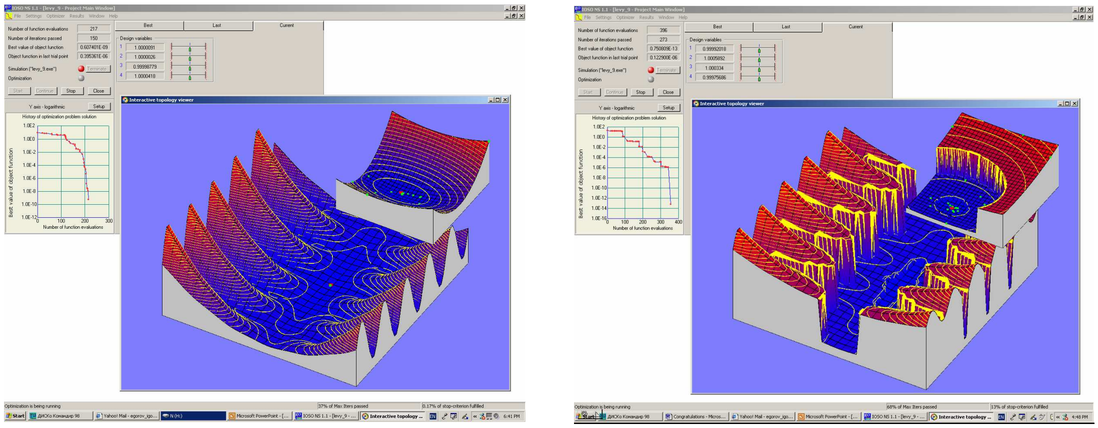
This optimization procedure is universal. It is uniquely powerful according to the relationship between the required number of calls to the analysis module and response topography complexity. On smooth object function it works as well as gradient methods. However, for complex (more probable to be faced by a designer in practice) object functions, having incomputability areas, discontinuities, multiple extremums and noise, the number of function calls required to find the global extremum is being increased considerably, while gradient methods are inapplicable for such task solutions.

For example, Figure 7a shows the results of optimization of well know Levy #8 test problem with 4 design variables[6]. This optimization problem is a multi-extremum optimization function with more then 626 local minima.

Figure 7b illustrates the results for the same problem which has the following modification:

$$y_{non\ diff} = \begin{cases} y + 10^{6-i}, if\ \ y\ > 10^{5-i}, i = \overline{1,17} \\ y \end{cases} \tag{7}$$

This means that this test function is discontinuous. It has 17 levels of decreasing shock patterns.



a) b)

Figure 7. Optimization of Levy #8 test problem.

The IOSO tolls and software are designed to solve complex problems of constrained and unconstrained optimization with various classes of objective function: smooth, non-differentiable, stochastic, with multiple optima, with the portions of the design space where objective function and constraints could not be evaluated at all, with the objective function and constraints dependent on mixed variables, etc. Algorithms of IOSO have good invariant features, a high level of stability of calculation while optimizing complex objects. They also ensure a solution for extremum with the presence of incomputability areas. These features of the algorithms make it possible to substantially expand classes of problems solved, facilitating the use of this software for complex practical problems.

IOSO tools and software work with only executable modules written to represent mathematical models. This significantly facilitates the customizing of the interaction of user's model and the optimization procedure since it does not require either shared PC memory spaces for data exchange or specific programming language to write the analysis code. Data exchange is provided by means of text files on a disk drive, making it easy to integrate the analysis codes into IOSO tools and software package. IOSO software has user friendly GUI and is simple to use. The software provides all necessary information to the user interactively. The parameters of IOSO technology are pre-programmed and are adaptively changing during the search for extremum without the user's intervention. Most of the algorithm's tunings are done internally, that is, they are hidden from the user who is not required to have any knowledge of nonlinear programming or optimization procedures. The only important thing for the user to understand is the physics of the problem and to have a mathematical model of the system. Creating an interface between IOSO and mathematical model typically takes several minutes.

The optimization process is visually represented in real time (displayed in current values of the design variables and their bounds) representing the objective function history. The user is able to control the optimization process. Users can interrupt the optimization process to tune up parameters with the ability to restart from the specified point, thus, cleaning up a "hanged" or crashed user's mathematical model. IOSO algorithm can be used in different computer systems such as Windows, Linux, and UNIX.

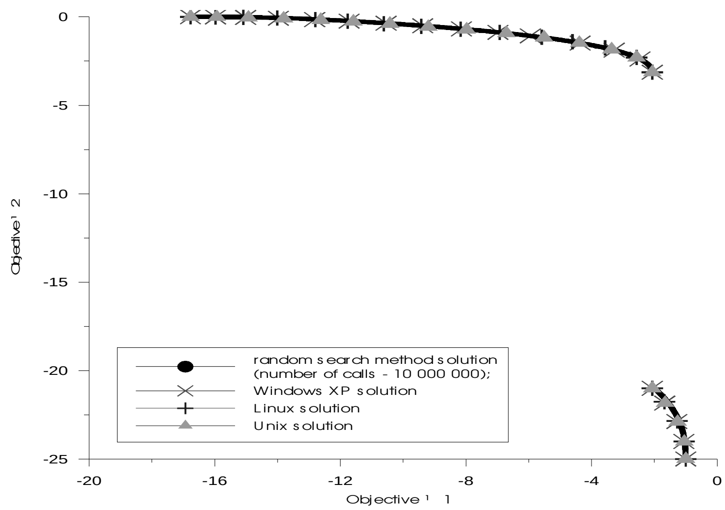
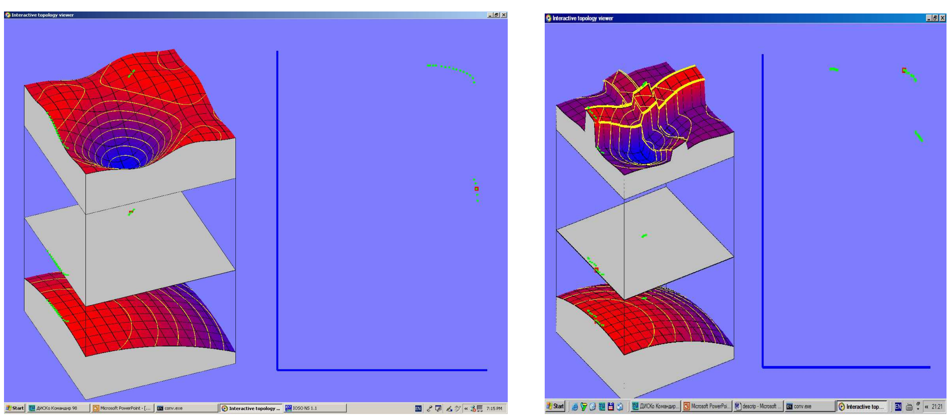


Figure 8. Solution obtained using IOSO NM.

Figure 8 shows an example of a well-known multi-objective test problem proposed by Poloni[8]. It was hard to obtain the exact solution analytically, that is why the results of IOSO NM work were compared with the solution which had been obtained with a random search method (requiring 10 million calls). Figure 9a show topology and solution of this test problem. Figure 9b demonstrates possibilities of solving multi-objective optimization problems with discontinuous objective function.



a) b)

Figure 9. Optimization of Poloni test problem.

IOSO technology algorithms implement a new evolutionary response surface strategy. This strategy differs significantly from both the traditional approaches of nonlinear programming and the traditional response surface methodology. Because of that, IOSO algorithms have higher efficiency, provide wider range of capabilities, and are practically insensitive with respect to the types of objective function and constraints. They could be smooth, non-differentiable and stochastic; with multiple optima, with the portions of the design space where objective function and constraints could not be evaluated at all; with the objective function and constraints dependent on mixed variables etc. [9,10,11,12].

4 CONCLUSIONS

A novel optimization algorithm (IOSO) was shown to be a highly efficient and reliable optimization tool for well-known single-objective and multi-objective test problems. We tried to demonstrate main possibilities of IOSO algorithms. It demonstrates that IOSO algorithms can be used for optimization of real-life complex technical systems and objects.

REFERENCES

1. I. N. Egorov, *“Indirect Optimization Method on the Basis of Self-Organization”*, Curtin University of Technology, Optimization Techniques and Applications (ICOTA’98), Vol.2, pp. 683-691, Perth, Australia (1998).

2. J. Marczyk, “*Stochastic Multidisciplinary Improvement: Beyond Optimization”*, AIAA-2000-4929, Proceedings of 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Long Beach, USA, (2000).

3. I. N. Egorov, “Optimization of a Multistage Axial Compressor. Stochastic Approach”, ASME paper 92-GT-163, (1992).

4. I. N. Egorov, G. V. Kretinin and I.A. Leshchenko, “*How to Execute Robust Design Optimization”*, AIAA-2002-4328, Proceedings of 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, USA, (2002).

5. E. Sandgren, "*The Utility of Nonlinear Programming Algorithms"*, Ph.D. thesis, Purdue University, IN, USA, (1977).

6. J. More, B.S. Gabow, and K.E.Hillstrom, *"Testing Unconstrained Optimization Software,"* ACM Trans. Math. Software, pp.17-41 (1981).

7. I. N. Egorov, G. V. Kretinin and I.A. Leshchenko, *“Optimization Algorithms as Tools for the Solution of Inverse Problems”*. WCCM V, July 7-12, 2002, Vienna, Austria (2002).

8. C. Poloni and V. Pediroda, “*GA coupled with computationally expensive simulations: tools to improve efficiency”*. In Quagliarella, D., Periaux, J., Poloni, C., and Winter, G., editors, *Genetic Algorithms and Evolution Strategies in Engineering and Computer Science. Recent Advances and Industrial Applications*, chapter 13, pages 267-288. John Wiley and Sons, West Sussex, England (1997).

9. *User Guide*, IOSO NS version 1.1, IOSO Technology Center, Moscow, Russia (2002).

10. I. N. Egorov, G. V. Kretinin, I. A. Leshchenko and S.V. Kuptzov, “*IOSO Optimization Toolkit – Novel Software to Create Better Design*”, AIAA paper 2002-4329, Proceedings of 8th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, GA, USA, (2002).

11. *Users Guide*, IOSO NM version 1.0, IOSO Technology Center, Moscow, Russia (2002).

12. *Users Guide*, IOSO NS robust version 1.0, IOSO Technology Center, Moscow (2003).