European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS 2000
Barcelona, 11-14 September 2000

TWO APPROACHES TO MULTIDISCIPLINARY OPTIMIZATION PROBLEMS

Igor N. Egorov*, Gennadiy V. Kretinin*, and Igor A. Leshchenko†

* Keldysh Institute of Applied Mathematics
127322, Milashenkova st., 10-201, Moscow, Russia
E-mail: optim@orc.ru, web page: www.orc.ru/~pulsar

† Air Force Engineering Academy
103190, Planetnaya st., 3, Moscow, Russia

Key words: Multidisciplinary optimization, approximation, parallel computations, multilevel optimization.

Abstract. *This paper discusses two approaches to solve multidisciplinary optimization problems regarding complex engineering systems. These approaches have been developed in order to reduce computing time expenditures required for solution of such problems.*

The first approach is based on utilization of parallel computations not only for object function computing, but also for an "internal" operation of the algorithm that is parallelized. This allows for a significantly higher acceleration of the problem solution process then a trivial usage of parallel CPUs for optimization criteria calculation. The results of numerical testing for the new algorithm are presented.

The second approach consists of using multiple fidelity (multilevel) analysis algorithms. The results of a real-life stochastic multiobjective optimization problem solution are presented. The usage of multilevel approach for such optimization problems results in a significant reduction of computing time.

1 INTRODUCTION

Within the frameworks of methodology development for multidisciplinary optimization (MDO) of complicated systems, there are several widely used techniques for approximation of multiparameter non-linear functions. The main reason for using the approximation technologies is that they offer a reduction of the number of calls to algorithms performing laborious mathematical analysis of the object while searching for the optimal solution. One of the promising approaches to achieving this objective consists of building response surface approximations by utilizing various evolutionary algorithms. We are analyzing two approximation based methods of reducing the overall computing time for MDO: computations parallelization and multilevel optimization.

We have developed[1] the highly efficient algorithms for approximation of multiparameter functions which form the foundation of Indirect Optimization based on Self-Organization (IOSO). IOSO is based upon the use of self-organization and evolutionary simulation principles to iteratively correct the response surface approximation structure and the parameters during the search of an extremum. The distinctive advantage of this approach is that it requires an extremely low number of trial points in the experiment plan to initialize the algorithm (typically 30...50 data points for the optimization problems with nearly 100 design variables).

The main feature of this approach is the decomposition of the approximation function under construction into the set of simple approximation functions. The final response function is a multilevel graph. The total power of the final polynomial can be high enough and it is determined in the process of evolutionary synthesis of the response function structure.

The obtained response surface functions are naturally used in the procedures of multilevel optimization. The simulation level is adaptively changed for both single discipline and multiple discipline analysis of the object. At each IOSO iteration the response function is being optimized within the current search area. This step is followed by a direct call to the mathematical model performing the analysis of the object for the obtained point. During IOSO operation the information about the object function behavior is stored in the neighborhood of the extremum so that response function becomes more accurate for this search area. While proceeding from one iteration to another, the following steps are carried out: the modification of the experiment plan; the adaptive selection of the current extremum search area; the choice of the response function type (global or middle-range); the transformation of the response function; the modification of both parameters and structure of the optimization algorithms; and, if necessary, the selection of the new promising points within the researched area.

Flexible structure of IOSO's main algorithm provides wide opportunities concerning development of new approaches aimed at the reduction of the computing time for complex real-life MDO problems. In the present paper we analyze two possible approaches. One of them utilizes parallelization of computations during optimization problem solution. The other is based upon the use of mathematical models with different accuracy levels. We will carry out the analysis of the efficiency of these approaches regarding the multiobjective optimization problems where it is required to obtain a set of Pareto-optimal solutions.

For this purpose a multiobjective version[2] of IOSO is used. This version allows us to find numerically the Pareto-optimal set of solutions. The obtained Pareto-optimal solutions are uniformly distributed within the design space according to the optimization criteria. IOSO can solve the problems of large dimensionality (tens and hundreds of variables and up to 10 simultaneous objectives). The number of solutions (degree of Pareto set discretization) is defined by the researcher and can be purposefully varied during the search.

2 PARALLELIZATION OF COMPUTATIONS

One of the prospective trends concerning MDO process efficiency improvement is the use of computers with multiple parallel processors. In this case the reduction of elapsed (clock) computing time can be achieved through mathematical model solution time reduction by means of parallel computations "inside" the model, as well as by adaptive organization of the optimization process for parallel computations. The first approach supposes the use (or development) of mathematical analysis models suitable for computations using parallel processors. The latter makes it necessary to develop or to modify the corresponding optimization methods.

In this paper we analyze the efficiency of an optimization algorithm for the complex systems which uses parallel computations. The presented algorithm is a modified version of the IOSO.

2.1 Basic scheme of IOSO parallel algorithm

The scheme of the developed optimization algorithm is shown in Fig. 1. In this case a "master-slave" model is used for parallelizing the computations.

Operation of the main IOSO unit is carried out at the master processor. This unit is called data analysis and moving strategy unit. Within the frameworks of the given unit, analysis of stored information is performed about the variable parameters, constrained parameters, and optimization criteria. The current neighborhood of Pareto-optimum solutions set is selected, determination the promising areas for further search is made, and formulation of the sequence of the next operation is performed. Three variants of actions at the current iteration of the optimization process have been used. They are:

I) Optimization process termination. This variant is performed by the stop-criterion activation when working in automatic operation mode, or if a researcher terminates the process when working in an interactive operation mode.

II) Experiment design generation. This variant involves generation of a set of points in the initial search area (at the initial stage of optimization), or in a promising sub-region of the search area. Then, for this set of points, both the optimization criteria calculations and parallel slave processor constrained parameter calculations are made. The obtained information is transmitted back to the unit of data analysis and moving strategy and the transition to the next iteration is performed.

III) The main variant of actions is in the following:
a) Synthesis of the set of approximation functions used for particular optimization criteria and constrained parameters. The functions differ according to both structure and search area. The process is performed with the help of parallel slave processors.
b) Optimization of the obtained approximation functions using parallel slave processors. The result of this step is a set of points which are supposed to be solutions of the initial optimization problem.
c) Computation of true values of optimization criteria and constrained parameters using parallel slave CPUs. The obtained information is transmitted back to the data analysis and moving strategy unit, and the transition to next iteration is performed.

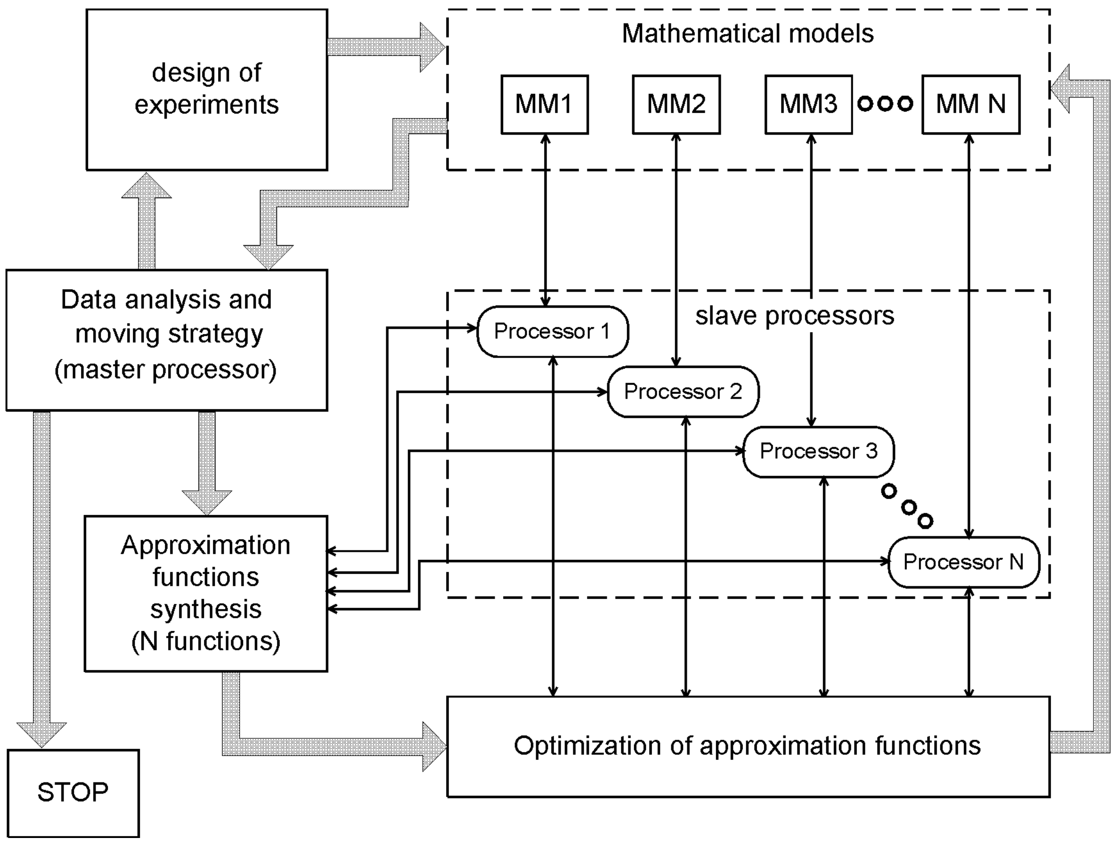

Figure 1: IOSO parallel algorithm scheme

The main difference between the developed parallel optimization algorithm and the basic IOSO algorithm is that at each iteration of the optimization process the information is received by the data analysis and moving strategy unit. This information is not about a single point, but about a whole set of points, the number of which is equal to the number of slave processors. This circumstance can affect the algorithm work efficiency. To evaluate this effect, a testing of the developed algorithm has been carried out.

2.2 The peculiarities of algorithm testing for parallel multicriteria optimization.

While carrying out the testing of the developed algorithm, a number of double-criteria optimization problems have been solved utilizing different numbers of slave processors (CPUs): (N_{CPU}= *1, 5, 10, 20, 30, 40*). Problems of different dimensionalities (N_x= *20, 40, 60, 80*) have been solved for two test functions.

I) "Simple" test.
In this case the following simple functions have been considered as particular optimization criteria.

$$y_1 = \frac{1}{N_x}\sum_{i=1}^{N_x} x_i^2 \rightarrow min; \qquad (1)$$

$$y_2 = \frac{1}{N_x}\sum_{i=1}^{N_x}(x_i - 1)^2 \rightarrow min;$$

$$-2 \le x_i \le 3, \quad i = \overline{1,N_x}.$$

The given problem has the exact solution (the Pareto set), which could be presented in a criterion (objective function) space as the following dependence.

$$y_1 = 1 - y_2, \quad 0 \le y_2 \le 1. \qquad (2)$$

II) "Complex" test.
The two functions considered as the particular optimization criteria are:

$$y_1 = \frac{2}{N_x}\sum_{i=1}^{N_x/2}[(e^{(x_{2i-1})} - x_{2i})^4 + 100\cdot(x_{2i} - 1)^6 + x_{2i-1}^2] \rightarrow min; \qquad (3)$$

$$y_2 = \frac{2}{N_x}\sum_{i=1}^{N_x/2}[(e^{(x_{2i-1}-0.5)} - x_{2i} + 0.5)^4 + 100\cdot(x_{2i} - 1.5)^6 + (x_{2i-1} - 0.5)^2] \rightarrow min;$$

$$-0.5 \le x_i \le 2, \quad i = \overline{1,N_x}.$$

The analytical determination of the Pareto set for the given test is extremely difficult. The "exact" solution has been obtained by means of the detailed scanning of variable parameters space (the scanning interval had the value of 10^{-4}) for $N_x=2$. It is easy to realize that the increase of the optimization problem dimensionality ($N_x=4,6,8,$) would not lead to the "exact" solution change.

To carry out the numerical testing of the multicriteria optimization algorithm it is necessary to introduce some quality index which is being obtained as a result of the Pareto set. It is common to use the solution mean error, defined as

$$\varepsilon = \frac{1}{N_P}\sum_{k=1}^{N_P}\varepsilon_{Pi} + \frac{1}{2}(\varepsilon_{e1} + \varepsilon_{e2}), \qquad (4)$$

where: N_P - number of obtained Pareto-optimal solutions;
ε_{Pi} - error of *i-th* Pareto-optimal point (see Fig. 2);
ε_{e1}, ε_{e2} - error of extrema determination for particular optimization criteria.

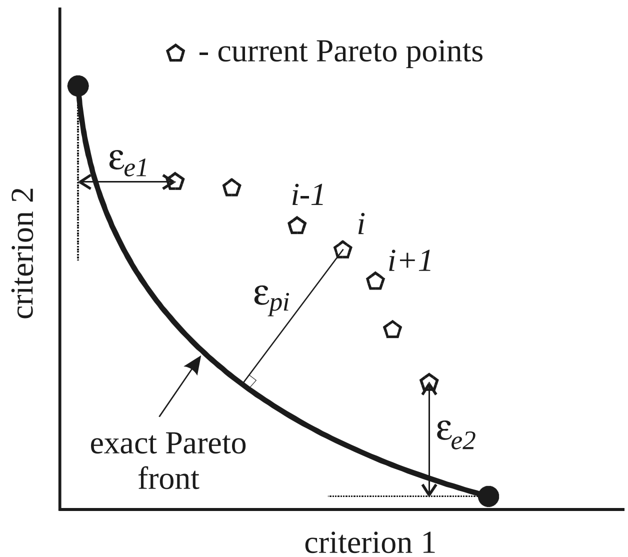

Figure 2: Pareto set accuracy evaluation

The optimization efficiency assessment for the use of parallel processors was carried out using the following indices:

I) Relative processor time defined according to the following dependence

$$RCT = \frac{total\ optimization\ time}{single\ MM\ execution\ time}. \tag{5}$$

This index characterizes the total time spent for the optimization problem solution. On condition that CPU time for mathematical model operation is much more increased in comparison with time of optimization algorithm internal work, the relative CPU time is approximately equal to the total amount of mathematical model runs at every CPU. It is the very case when we deem it advantageous to use computations in parallel.

II) Parallel optimization speed-up at the expense of computations parallelizing

$$S_N = \frac{optimization\ time\ for\ a\ single\ processor}{optimization\ time\ using\ N\ processors}. \tag{6}$$

It must be noted that when using traditional approach to the analysis of parallelization computation efficiency[3] the ideal value of this index is equal to the number of the applied processors ($S_N=N$). However, for the parallel optimization algorithm, the change of the number of processors being used corrects the information volume which is being analyzed when going from one iteration to another. Thus, in our case the effectiveness of computations parallelization (S_N/N) can be both less and more than 1.

2.3 The testing results

When carrying out the given test studies we formulated two tasks:

a) Verification of the algorithm operational capacity.
b) Assessment of the algorithm efficiency.

The solution of a number of test optimization problems confirmed the good reliability of the developed algorithm. It enables to obtain the Pareto-optimal solution set for problems of various dimensionalities. In Fig. 3 an example is shown of alteration of a Pareto-optimal solutions set during the process of the problem solution. One can see that while the relative CPU time increases, the solution becomes more precise. This is evidenced in the sense that the obtained Pareto-optimal points approach the exact solution (ε_P) and by determination accuracy of the particular objective (ε_{e1} ε_{e2}) extrema.

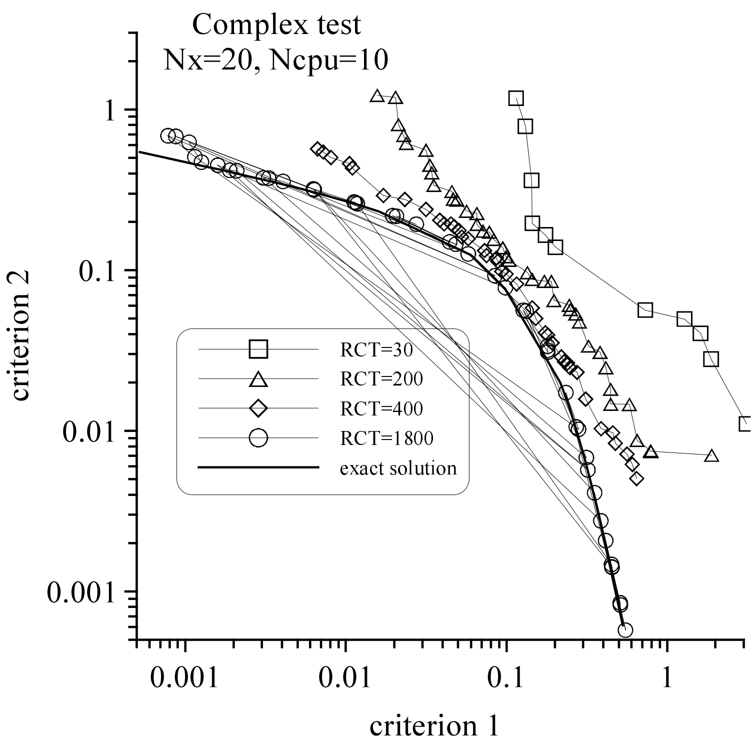

Figure 3. Dynamics of Pareto set search

Figure 4 demonstrates the examples of the obtained Pareto-optimal solution sets for the test functions under consideration while using a single processor and 40 processors. The given sets have been obtained using equal total numbers of objective function evaluations. One can see that the obtained solutions are practically the same and that they are close to the exact problem solution.

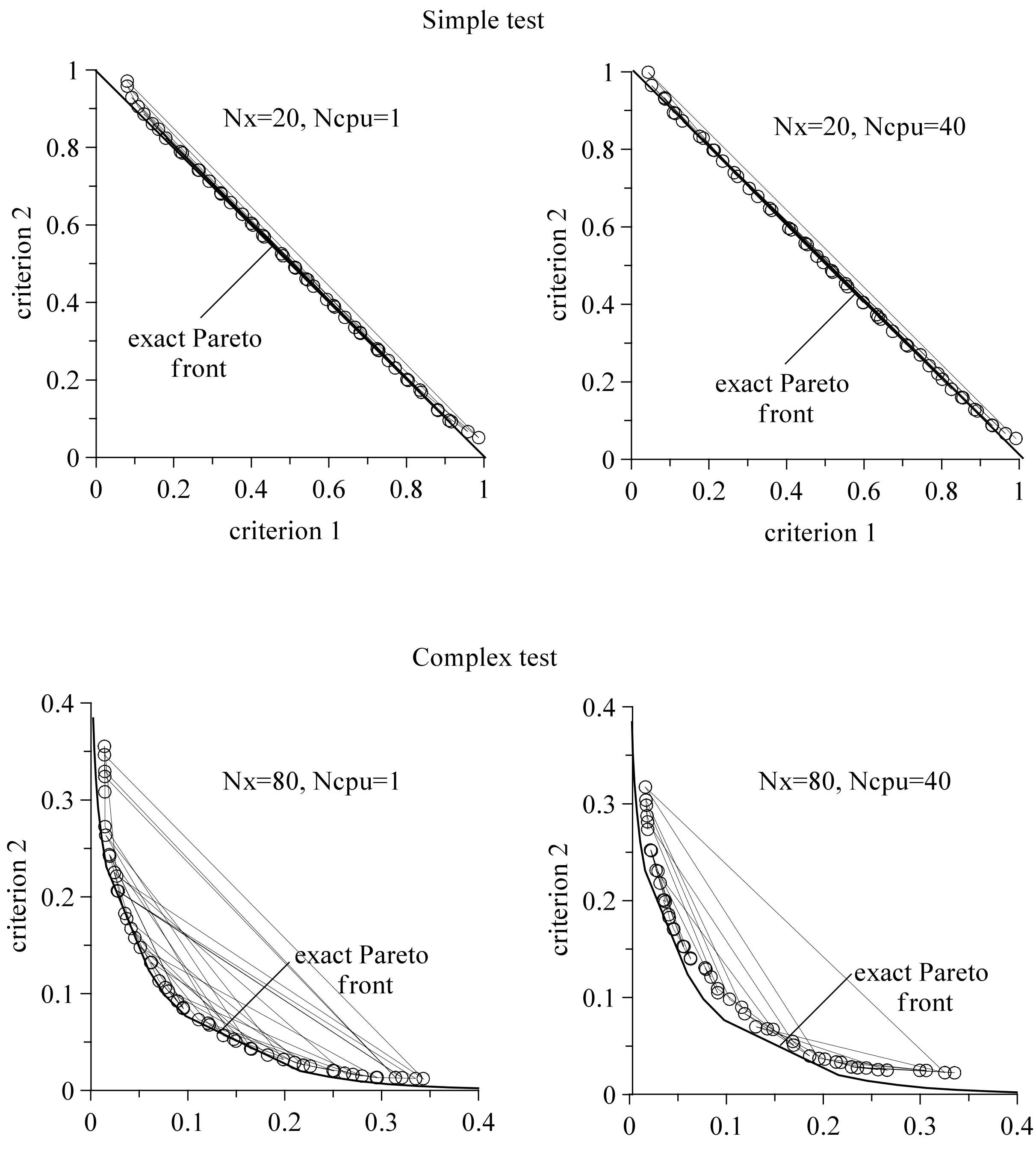

Figure 4. Examples of Pareto sets found using single and multiple processors

Figure 5 shows examples of solution mean error (ε) alteration versus the relative CPU time of the problem solution for different number of processors. One can see that the usage of computation parallelization results in significant reduction of the time for optimization problem solution. In particular, when using 40 processors at the initial stage of optimization, the time required to obtain the same solution accuracy could be reduced by two orders of magnitude in comparison with a single CPU time requirement. While decreasing the mean error, the efficiency of computations parallelization is being reduced somewhat although it still remains relatively high.

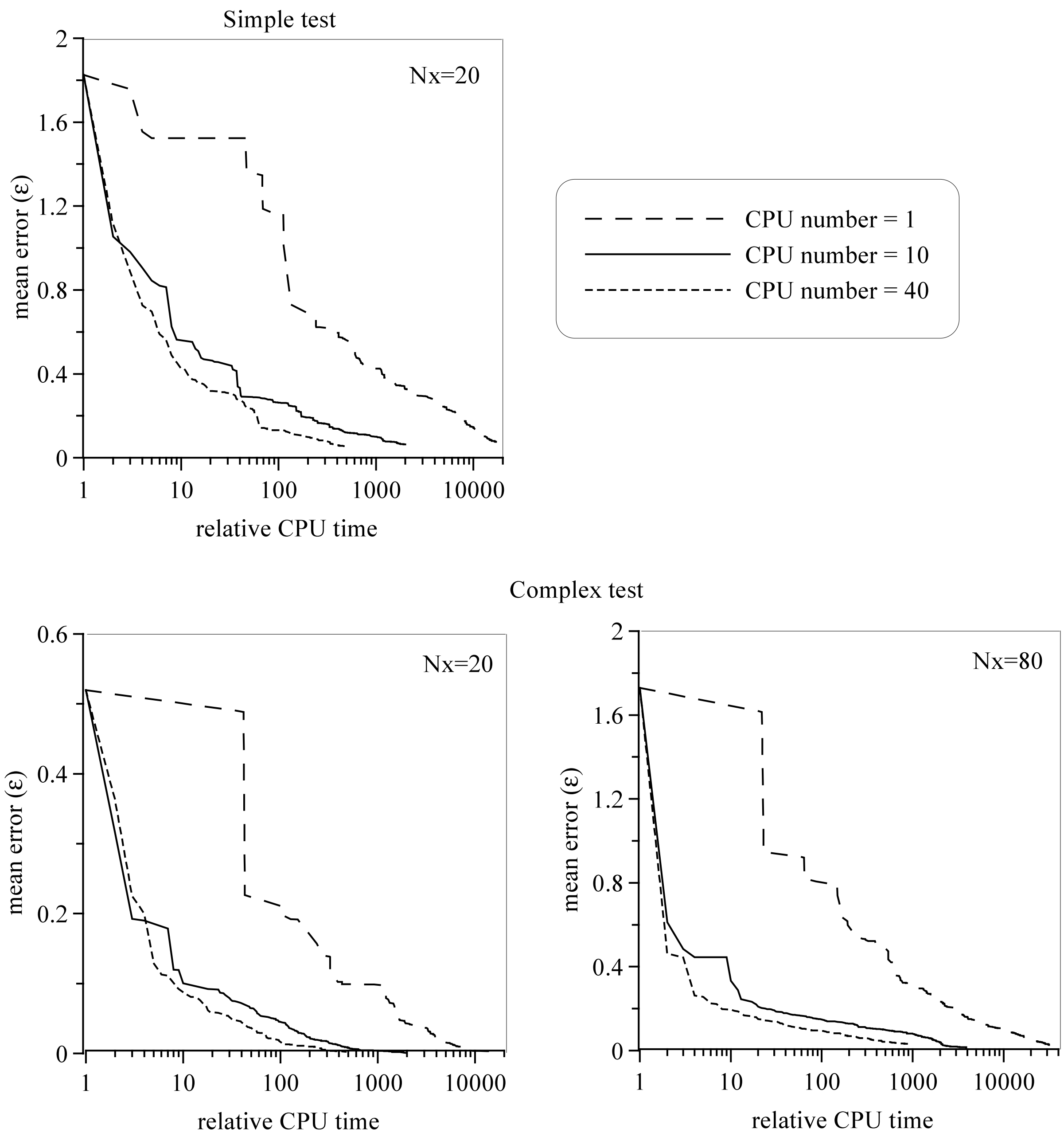

Figure 5. Mean error evolution for single and multiple processor optimization

Figure 6 shows an example of S_N parameter variation demonstrating the high parallelized computations efficiency. One can see that for the considered test conditions the S_N parameter value exceeds the total number of operational CPUs. It proves that this parallel optimization algorithm has the higher "internal" efficiency than one of IOSO basic (single-CPU) version. It also proves that the given optimization procedure has been developed according to the parallel optimization process, and that it does not represent a trivial usage of several CPUs for the solution of optimization problems.

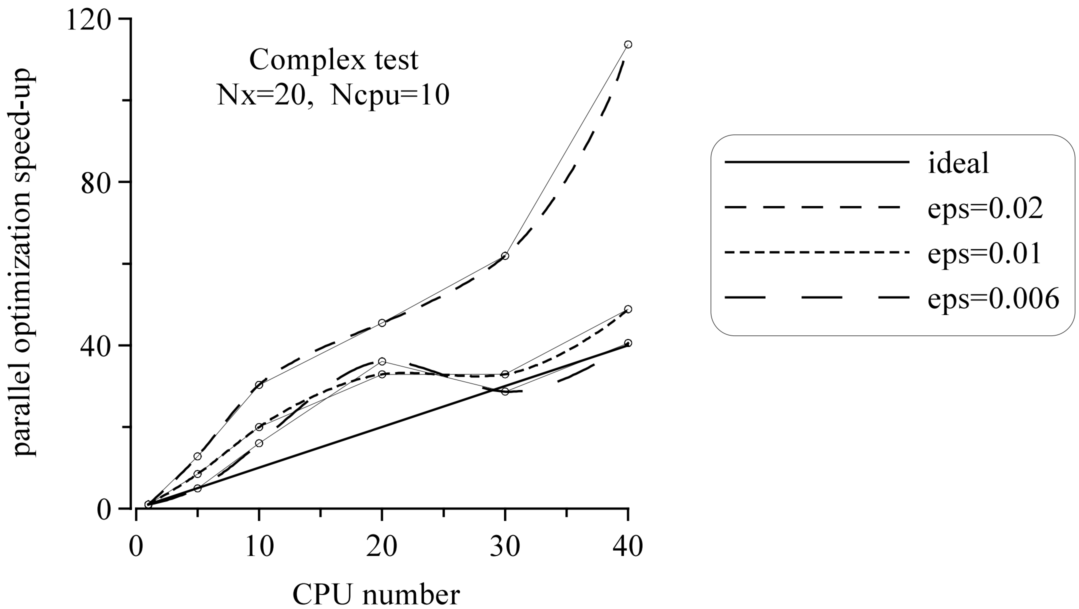

Figure 6. Parallel optimization efficiency

It must be noted that the given results have been obtained using a fixed set of the algorithm internal parameters. Some of these parameters are the number of points in the initial design of experiment; the desired number of Pareto-optimal solutions that are uniformly distributed; the probability of generation stage initialization in the basic algorithm, etc. Variation of these parameters can affect the resulting performance indices of the parallel optimization algorithm. Preliminary results show that the general dependence trend (Fig. 6) is still being preserved.

3 MULTILEVEL OPTIMIZATION

The typical situation while solving a problem of optimization of complicated engineering systems is that the user has several tools of various degree of fidelity to perform the analysis. These tools differ according to their levels of complexity of modeling the actual physical phenomena and their different levels of numerical accuracy. The high-fidelity tools could be represented by detailed non-linear mathematical models of the researched systems or even by the experimental samples of such systems. However, the use of such tools in optimization is associated with significant time expenditures. The low-fidelity models also allow carrying out optimization search, but the reliability of the obtained results can be rather low. Therefore, within the framework of the development of MDO methodology for complicated systems, the methods based on a combination of various fidelity analysis tools are widely practiced. The objective here is to offer a procedure of multiobjective optimization of complicated systems based upon the adaptive use of analysis tools of various levels of complexity. The intention is to minimize the use of complicated time-consuming tools for the analysis. This approach ensures the possibilities to search Pareto-optimal set of solutions, and also ensures the improvements of the lower level mathematical model.

3.1 The basic scheme of multilevel optimization

The simplified scheme of work for the multilevel optimization procedure can be represented as follows (Fig.7).

I) Solving the multiobjective optimization problem based upon a simplified mathematical model for the analysis. For this purpose, the method of indirect optimization based on the self-organization (IOSO) is used. This method allows finding the Pareto-optimal set of solutions numerically. The found Pareto-optimal set is uniformly distributed in the design space with respect to optimization objectives. IOSO can solve the problems of large dimensionality (tens and hundreds of variables and up to 10 objectives). The number of solutions (degree of discritization of the Pareto set) is specified by the user and can be purposefully varied during search.

II) For the obtained Pareto set the indicators of effectiveness are updated using the high-fidelity analysis tools.

III) The identification of the simplified mathematical model is performed. Depending upon the peculiarities of the applied mathematical simulation, the identification procedure can performed using various approaches. One such approach involves non-linear corrective dependencies construction that includes evaluation of the results deviation approximation functions obtained with different fidelity analysis tools. The other possible approach is application of internal parameters nonlinear estimation.

IV) Replacement of the simplified mathematical model by the identified one and the return to step I).

The particular features of the problem define the number of iterations for such a multilevel procedure. The number of applications of high fidelity analysis tools is limited to the product of the number of iterations and the number of Pareto-optimal solutions.

While solving the problem of optimization of complicated engineering systems the accuracy difference between high fidelity and low fidelity tools can be rather large. For such cases the series of intermediate middle-fidelity tools can be applied. The scheduling of their application during the problem solution process must be adaptively changed to minimize the temporal expenditures. This is true for MDO in multiobjective cases as well, where the available analysis tools for various disciplines can be substantially different at the accuracy and execution speed levels. In this case, when formulating the problem one should take into account these features to assure optimal distribution of the analysis tools.

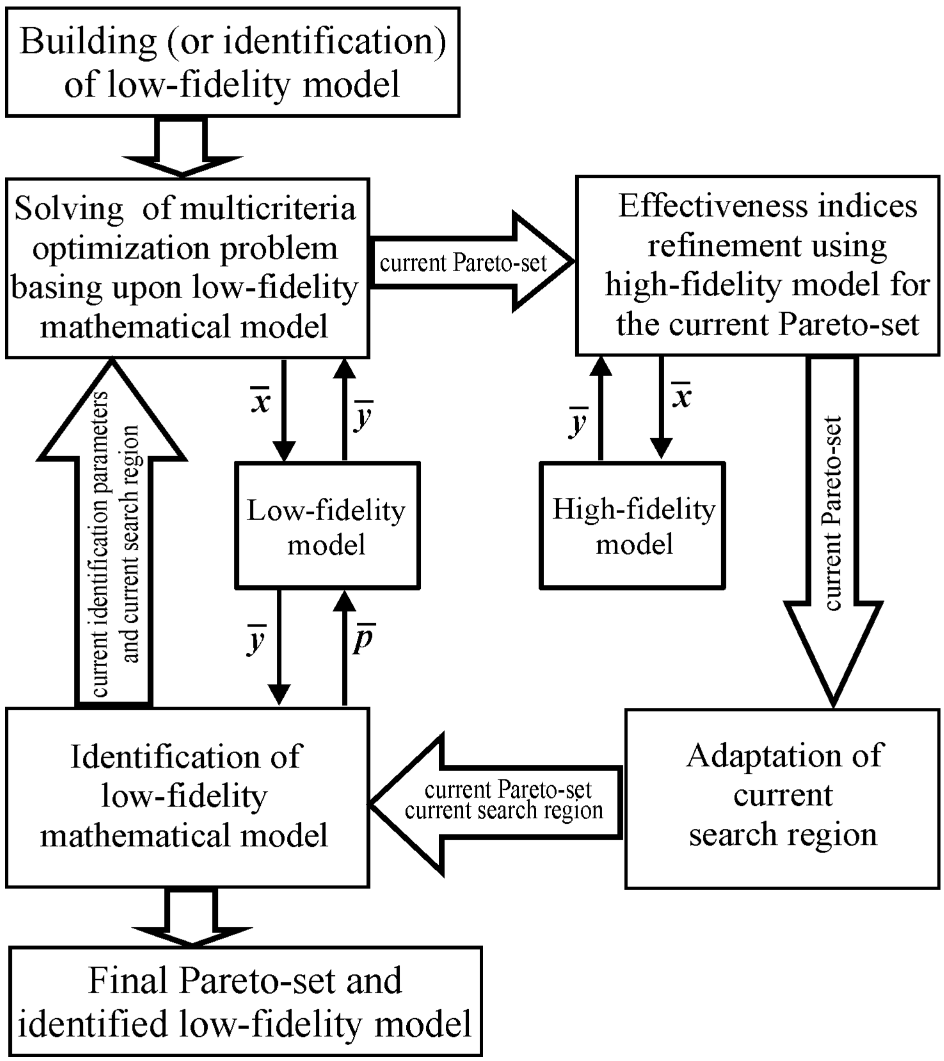

Figure 7. The scheme of multilevel optimization procedure

The information stored during the search is used to improve the simplified models. After the given analysis procedure termination, one can construct the researched object response functions. However, both identification and approximation are correct not for the entire initial search area but only for certain neighborhood of the obtained Pareto set. This ensures purposeful improvement of approximating properties only in the area of optimal solutions that noticeably reduce the computing effort to construct these functions.

The developed methodical approaches considerably increase the effectiveness of complicated multiobjective optimization problems solution. The stored information for identification of the simplified analysis tools and for approximation allows expanding hierarchically the thoroughness of the problem review. It also allows for incorporation of new disciplines in the object analysis.

3.2 A real-life example

Some examples of the developed multilevel optimization procedure application can be found in the recent publications[4,5]. As an example, stochastic multiobjective optimization of a multistage axial compressor will be illustrated.

One of the significant problems in implementation of the optimal technical solution using the traditional deterministic approach, is the impossibility of the accurate reproduction of optimal design parameters in real-life manufacturing conditions. The presence of accidental deviations of design parameters from their defined values leads to the necessity to consider additional optimization criteria which are connected with the probability of implementation (manufacturability) of the obtained optimal solutions. This can be accomplished using the approach of the stochastic optimization problems solution[6,7,8].

While carrying out the optimization of engineering systems considering current level of manufacturing technology, it is necessary to determine the numerical values of probabilistic optimization criteria. As a rule, Monte-Carlo method is used for this purpose. This method, when applied to the high level analysis systems, requires excessive amounts of computing time for optimization. In this case, the use of multilevel procedure when the probabilistic criteria are computed according to the lower level identified models, can be rather effective.

In this example, the objective is to increase the efficiency of a compressor as a function of blade rows geometrical parameters variety (42 variables). The compressor design is to satisfy a set of pre-defined constraints: air mass flow rate, pressure ratio increase, and gasdynamic stability margins. There is also the criteria of computability, the failure of which corresponds to the analysis model malfunction for a given vector of variables. When considering the blade rows real-life manufacturing conditions, the components of variable parameters vector are considered as random values, distributed according to normal distribution law with the dispersion defined by the level of manufacturing technology.

Thus, the optimization objective is to maximize the compressor efficiency (the deterministic criterion) subject to a given constraints implementation probability. As a high fidelity analysis model we adopted a quasi-3D multistage flow-field analysis code that take into account viscosity effects. As a low fidelity model we adopted a simple and considerably less computionally intensive 2D single blade row flow-field analysis code. This model was suitable for identification. During each iteration of multilevel optimization procedure the low fidelity model identification was performed according to the compressor efficiency and constrained parameters.

While solving this optimization problem, only 6 iterations of the developed multilevel optimization procedure were needed. During each iteration, 10 Pareto-optimal solutions were determined. Figure 8 shows the alteration values of compressor efficiency in comparison with the initial version. These values were computed using low fidelity and high fidelity flow-field analysis models. One can see that during the process of multilevel optimization the range of deterministic criterion is being changed and the predictive capabilities of inexpensive low fidelity model are being improved.

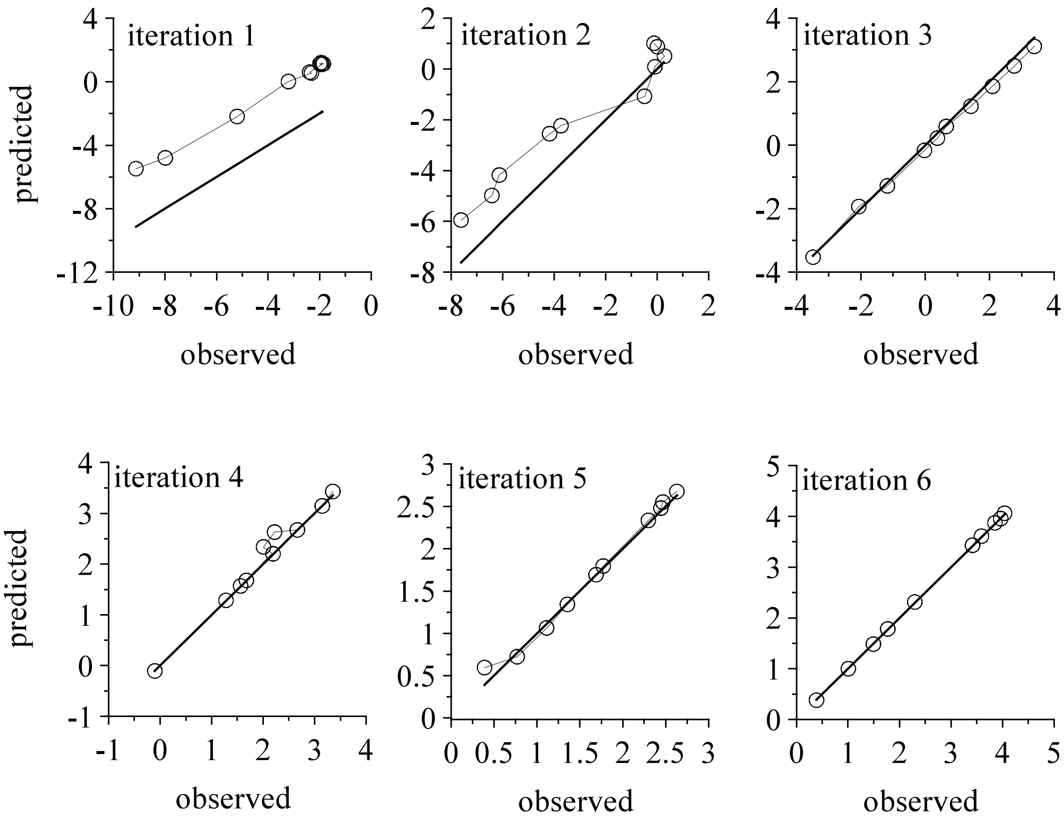

Figure 8. Improvement in the prediction accuracy of the low fidelity model

Figure 9 shows the dynamics of relative computational error changes according to the low-fidelity model, which has been used to evaluate the compressor efficiency and all constrained parameters within the set of Pareto-optimal solutions. One can see the improvement of the low fidelity model accuracy. A slight increase of the low fidelity model error while going from the third iteration to the forth is related to a considerable change of the current search area. At the final Pareto set (6^{th} iteration), the mean error of the low fidelity model computation does not exceed 0.06%, that is quite acceptable for this type of problems.

Figure 10 shows the obtained Pareto-optimal set in the dual objective function space. One can see that there are significant compressor efficiency improvement potentials and satisfaction of manufacturing constraints probabilities as compared with the initial design. It is important to note that the obtained set of 10 Pareto-optimal solutions has been determined using only 60 direct calls to the high fidelity analysis model.

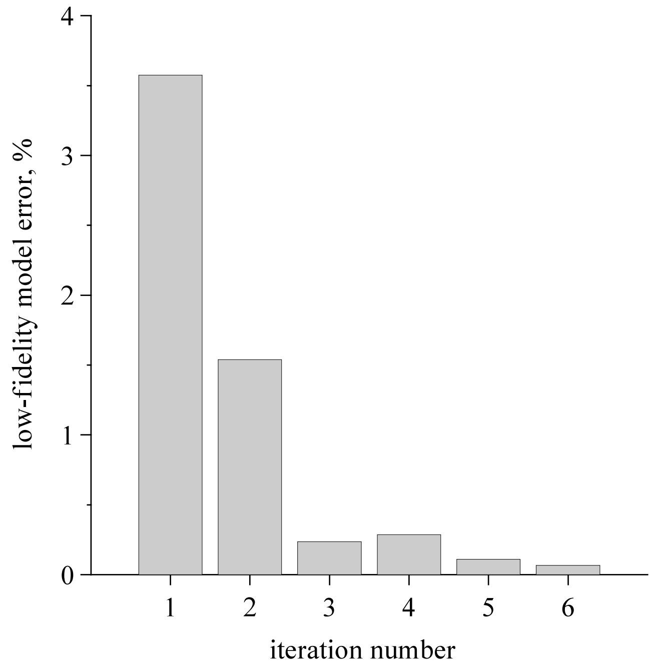

Figure 9. Improvements in low fidelity model accuracy

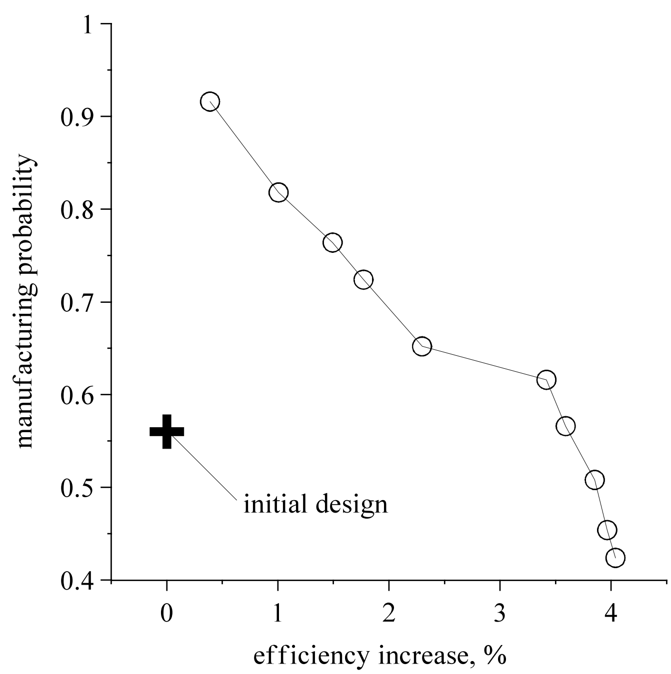

Figure 10. Pareto set found

4 CONCLUSION

One of the promising research directions in the complex engineering systems MDO is the use of decomposition methods combined with multiobjective problem statement. In this case the connections among different disciplines are changed by additional objectives. Then, the multiobjective optimization is carried out within the frameworks of each discipline and the obtained Pareto-optimal solutions sets are coordinated in order to form one or several competitive versions.

Usage of the proposed parallel optimization algorithm and multilevel optimization procedure (separate or joint) offers a significant reduction of the computing time expenditures for the solution of complex real-life problems while maximizing the probability of manufacturing of the optimized object.

5 ACKNOWLEDGMENTS

The authors wish to express their deepest gratitude for Prof. George S. Dulikravich for his council throughout this work. We also wish to express our thanks to Mr. Brian H. Dennis for his assistance in preparing these materials.

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] I.N. Egorov and G.V. Kretinin, “Search for compromise solution of the multistage axial compressor’s stochastic optimization problem”, World Publishing Corporation, *Aerothermodynamics of Internal Flows III*, pp. 112-120, Beijing, China, (1996).

[3] G. Golub and J.M. Ortega, *Scientific computing. an introduction with parallel computing,* Academic Press, ISBN 0-12-289253-4, (1993).

[4] I.N. Egorov, G.V. Kretinin and I.A. Leshchenko, “The multilevel optimization of complex engineering systems”, ISSMO, Short paper, in Proceedings of 3rd WCSMO, pp. 414-417, New York, (1999).

[5] I.N. Egorov, G.V. Kretinin, I.A. Leshchenko and Y.I. Babiy, "Optimization of complex engineering systems using variable-fidelity models", MCB University Press, ISBN: 0-86176-650-4, in Proceedings of the 1st ASMO UK/ISSMO Conference on Engineering Design Optimization, pp. 143-149, (1999).

[6] I.N. Egorov, "Optimization of a multistage axial compressor. stochastic approach", ASME paper 92-GT-163, (1992).

[7] I.N. Egorov and G.V. Kretinin, “Optimization of gas turbine engine elements by probability criteria”, ASME paper 93-GT-191, (1993).

[8] I.N. Egorov, G.V. Kretinin, I.A. Leshchenko and S.S. Kostiuk, “The methodology of stochastic optimization of parameters and control laws for the aircraft gas-turbine Engines flow passage components”, ASME paper 99-GT-227, (1999).