**WCCM V**
Fifth World Congress on
Computational Mechanics
July 7-12, 2002, Vienna, Austria
Eds.: H.A. Mang, F.G. Rammerstorfer,
J. Eberhardsteiner

# Optimization Algorithms as Tools for the Solution of Inverse Problems

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

IOSO Technology Center, Moscow, Russia
e-mail: egorov@iosotech.com

**Key words:** optimization, response surface, multilevel optimization, parallel optimization

## Abstract

This paper presents the main capabilities of IOSO (Indirect Optimization based on Self-Organization) technology algorithms, which can be used for the optimization of objective functions resulting from certain formulations of inverse problems. 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. 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, 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. The capabilities of IOSO software are demonstrated using examples of solving complex multidimensional (up to 200 design variables) problems. The examples employ technology of multilevel, multiobjective, and parallel optimization.

# 1 Introduction

When formulating some inverse problems there can be a necessary in non-linear optimization [1]. For their efficient solution one needs in up-to-date optimization techniques. One of them is the optimization based on response surface technology.
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 optimization: computations parallelizing and multilevel optimization. Usage of multilevel and parallel optimization strategies in combination with approximation technology allows us to obtain additional efficiency increment when solving complex optimization problems.

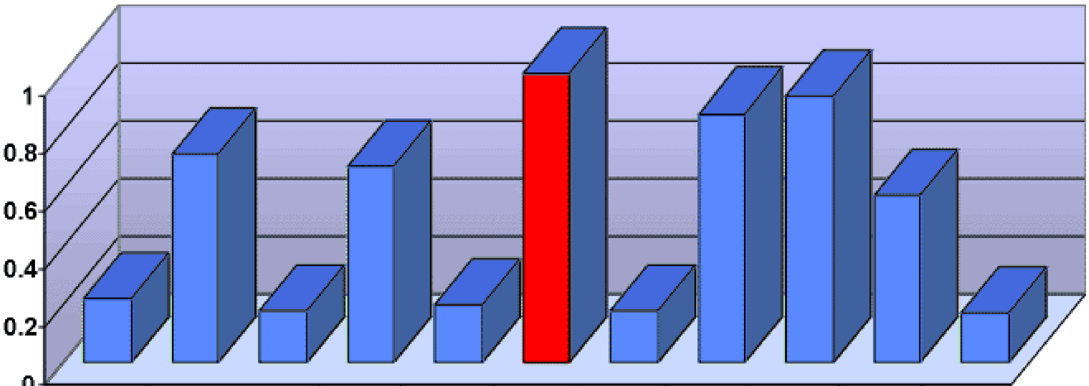
# 2 Main features of algorithms comprising IOSO Technology

## 2.1 Main algorithm

The pre-starting procedure of IOSO main algorithm consists of development the initial plan of experiment $X^W$. For each vector of the variables the value of objective is calculated by direct call to mathematical simulator of the system under study: $Y(x^j),\ \ j=\overline{1,m},\ x^j \in X^W$.
Each iteration during the extremum search consists of two stages: the construction of the simple type function approximating the objective function and the search the extremum of this statistical analogue. We have developed the highly efficient algorithms for approximation of multiparameter functions [6]. These algorithms form the foundation of Indirect Optimization methods based on Self-Organization (IOSO). IOSO implements self-organization and evolutionary simulation principles to iteratively correct the response surface approximation structure and parameters during the search for 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).
Then the approximation function extremum is searched. This does not require much CPU time expenditures when the numerical procedure is used, based upon the random search with adaptation of both search region and direction. The result of these two phases is the point $\tilde{x}_w$ likely as extremum.
The direct call to the complex mathematical simulator of the system under study is reasonable only for the point $\tilde{x}_w$ to correct the value of the optimization criterion $\tilde{y} = f(\tilde{x}_W)$. The obtained value of objective is used within the new experiment plan.
The above stated presents only one iteration of extremum search. This process is in progress till the extremum is found with required accuracy. During this process the plan of experiment is changed adaptively. The effectiveness of this method is in the possibility to shift from one iteration to another by calculating only one value of the goal function for the successful step.
During the problem solving the effectiveness of the search becomes better due to the adaptive changing of the search region, putting into the experiment plan only the points with good prospects. This is due to the collecting of information about the optimized function topology and improvement of the approximation function (especially nearby the extremum point).

### 2.2 Comparative analysis of efficiency

In the present work there has been carried out the comparing of IOSO Technology algorithms efficiency with that of up-to-date nonlinear optimization methods. As the tests there were chosen well known test functions [7], which were complex enough nonlinear problems of conditional and unconditional optimization. When comparing optimization methods there was considered one of possible 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 keeping, the number of function evaluations required for obtaining the solution etc. At Fig. 1 there are shown the main results. One can see that IOSO basic algorithm can compete successfully with well-known optimization methods.




| CONMIN | DONLP | EP | GRG | Hooke | IOSO | MMFD | MOST | NLPQL | SAM | SLP |
|---|---|---|---|---|---|---|---|---|---|---|
| 0.22 | 0.72 | 0.18 | 0.68 | 0.2 | 1 | 0.18 | 0.86 | 0.92 | 0.58 | 0.17 |

Figure 1: 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 the scientific problems by the adaptive building of the algorithm for the particular real optimization problem during the extremum search by the computer itself.

## 3 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.

### 3.1 Basic scheme of IOSO parallel algorithm

The scheme of the developed optimization algorithm is shown in Fig. 2. 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.

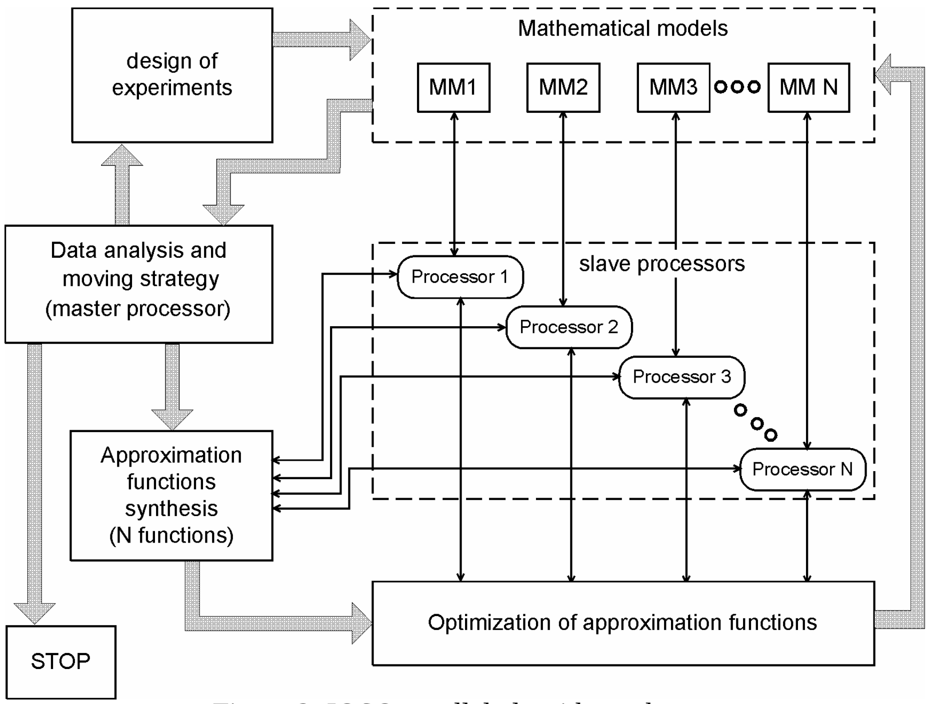



Figure 2: IOSO parallel algorithm scheme.

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.

### 3.2 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 (Nx= 20, 40, 60, 80) have been solved for two test functions. We used the multiobjective version of IOSO [3].

I) "Simple" test.

In this case the following simple functions have been considered as particular optimization criteria.

$$y_1 = \frac{1}{N_x}\cdot\sum_{i=1}^{N_x} x_i^2 \rightarrow min;\ y_2 = \frac{1}{N_x}\cdot\sum_{i=1}^{N_x}(x_i - 1)^2 \rightarrow min;\ -2 \le x_i \le 3,\ i = \overline{1,N_x}\ . \quad (1)$$

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

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

II) "Complex" test.

The two functions considered as the particular optimization criteria are:

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

$$y_2 = \frac{2}{N_x}\cdot\sum_{i=1}^{N_x/2}\left[\left(e^{(x_{2i-1}-0.5)} - x_{2i} + 0.5\right)^4 + 100\cdot\left(x_{2i} - 1.5\right)^6 + (x_{2i-1} - 0.5)^2\right] \rightarrow min; \quad (3)$$

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

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 Nx=2. It is easy to realize that the increase of the optimization problem dimensionality (Nx=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

$$\varepsilon = \frac{1}{N_P}\sum_{k=1}^{N_P}\varepsilon_{Pi} + \frac{1}{2}\cdot\left(\varepsilon_{\hat{a}1} + \varepsilon_{\hat{a}2}\right), \quad (4)$$

where: $N_P$ - number of obtained Pareto-optimal solutions;

$\varepsilon_{Pi}$ - error of i-th Pareto-optimal point (see Fig. 3);

$\varepsilon_{e1}$, $\varepsilon_{e2}$ - error of extrema determination for particular optimization criteria.

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 [2] 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.

### 3.3 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. 4 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.

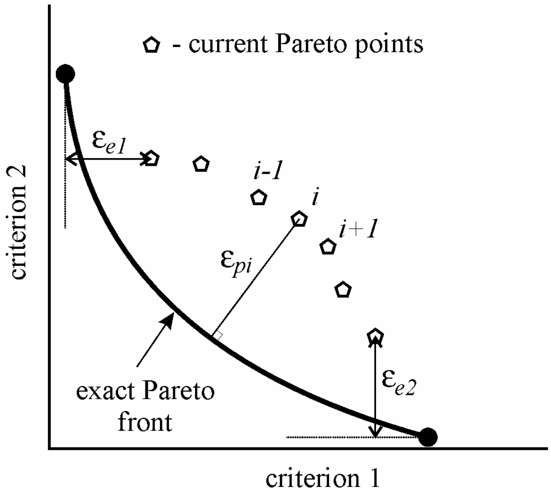



Figure 3: Pareto set accuracy evaluation.

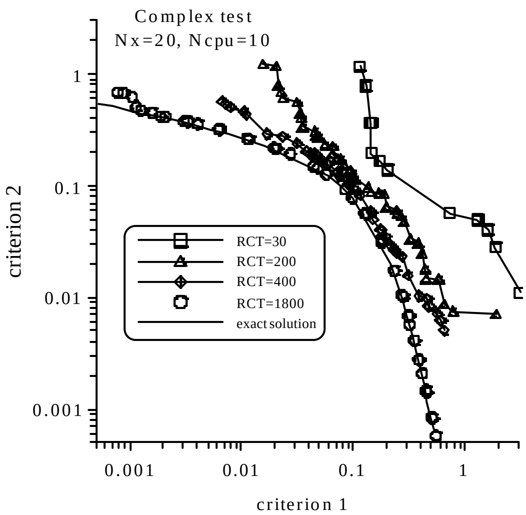



Figure 4: Dynamics of Pareto set search.

Figure 5 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.

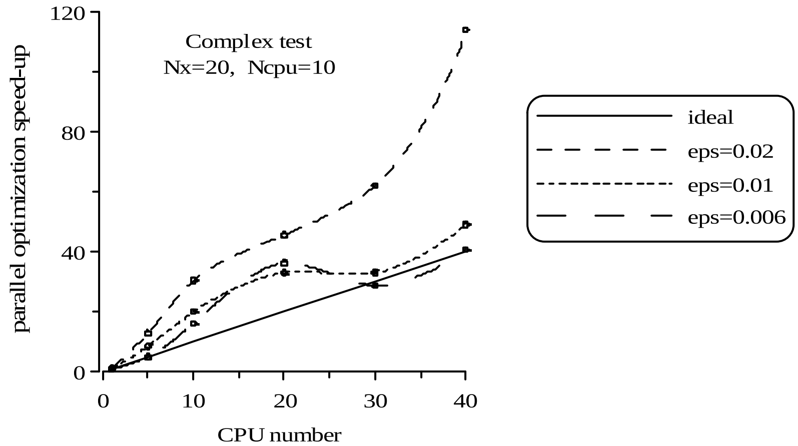



Figure 5. Parallel optimization efficiency.

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 uniformly distributed Pareto-optimal solutions; 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.5) is still being preserved.

# 4 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.

## 4.1 The basic scheme of multilevel optimization

The simplified scheme of work for the multilevel optimization procedure [4, 5] can be represented as follows (Fig.6).

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 discretization 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.

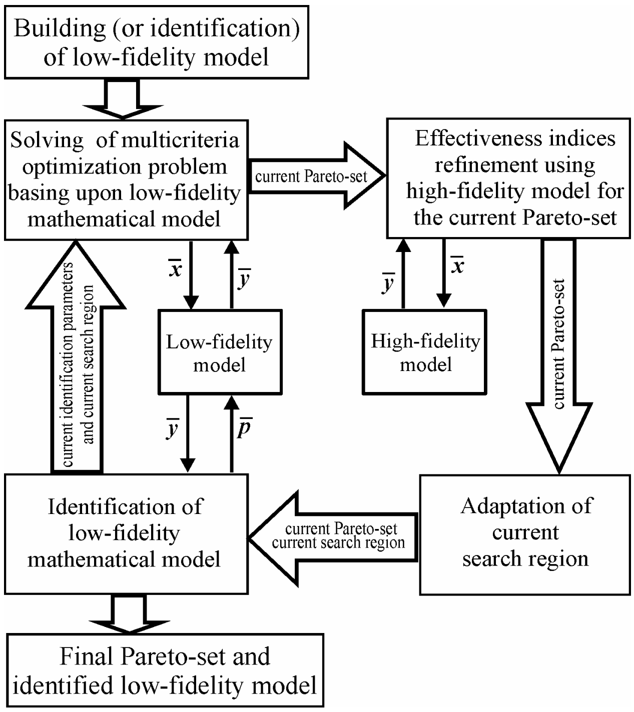



Figure 6. The scheme of multilevel optimization procedure.

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.

### 4.2 A real-life example

This problem was in search of multistage axial compressor optimum design parameters to improve its efficiency at two operating modes, namely, at design-point operating mode, corresponding to maximal thrust of GTE, and at the cruise operating mode, corresponding to minimal specific fuel consumption. Thus, as the optimization criteria we treated the compressor efficiency values for two operating modes. As these criteria could be mutually contradictory, the problem was in determining of the set of Pareto-optimal design parameters vectors. Let us consider the concrete example for the four-stage compressor.
The variable parameters: constructive inlet and exit angles of 7 blade rows in 3 sections by radius.
The criteria: the efficiency at two operating modes.
The constraints: stall margins at two operating modes; the constraints by criteria computability.
High fidelity tool: quasi-3D model with viscosity effect simulation.
Low fidelity model: 2D axis-symmetric model capable to be identified.
In accordance with developed multilevel optimization procedure at the first stage we have found 10 Pareto-optimal vectors of design parameters using low-fidelity model. After testing of this set by high-fidelity model it appeared that 2 of obtained vectors were out of computability region (i.e. the high-fidelity model didn't allow to get the values of criteria and constrained parameters). Other 8 vectors didn't satisfy the stall margin constraints. Thus, first iteration didn't allow to find any admissible solution. Meanwhile, the obtained results were used for the identification of low-fidelity model, and 3 iterations more were conducted. The results of Pareto-optimal solutions set transformation within criteria plane are presented in Fig. 7.

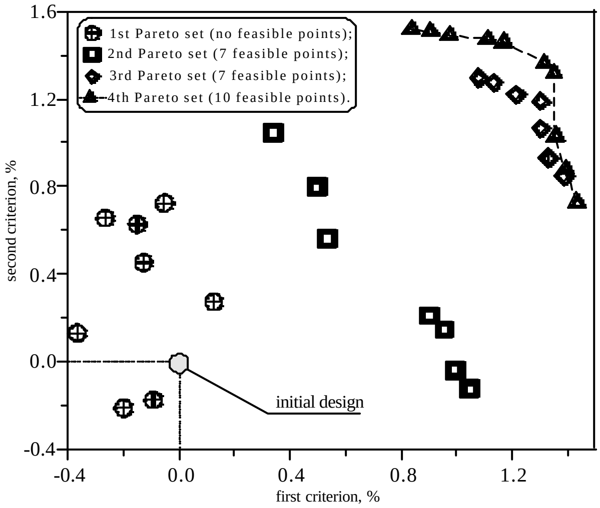



Figure 7. The results of Pareto set transformation.

# 5 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.

**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] B. H. Dennis, Z.-X. Han, I. N. Egorov, G. S. Dulikravich, C. Poloni, *Multi-objective optimization of turbomachinery cascades for minimum loss, maximum loading, and maximum gap-to-chord ratio*, *AIAA Multidisciplinary Analysis and Optimization Conference and Exhibit*, Long Beach, CA, September 5-8 (2000).

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

[3] 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*, Beijing, China, (1996), pp. 112-120.

[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, New York, (1999), pp. 414-417.

[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, (1999), pp. 143-149.

[6] I.N. Egorov, *Indirect optimization method on the basis of self-organization*, Curtin University of Technology, *Optimization Techniques and Applications (ICOTA'98)*, Perth, Australia, (1998), Vol.2, pp. 683-691.

[7] E. Sandgren, *The Utility of Nonlinear Programming Algorithms*, a thesis submitted for the Degree of Doctor of Philosophy,Purdue Univercity, (1977).