Efficient Methods of Multicriterial Optimization Based on the Intensive Use of Search Information

Victor Gergel and Evgeny Kozinov

Abstract In this paper, an efficient approach for solving complex multicriterial optimization problems is proposed. For the problems being solved, the optimality criteria may be multiextremal ones, and calculating the criteria values may require a large amount of computations. The proposed approach is based on reducing multicriterial problems to nonlinear programming problems via the minimax convolution of the partial criteria, reducing dimensionality by using Peano evolvents, and applying efficient information-statistical global optimization methods. The new contribution is that all the search information obtained in the course of optimization is used to find each current Pareto-optimal solution. The results of the computational experiments show that the proposed approach essentially reduces the computational costs of solving multicriterial optimization problems (by tens and hundreds of times).

1 Introduction

Multicriterial optimization (MCO) problems are the subject of intense research and are widely used in applications. The practical demand stimulates extensive research in the field of MCO problems – see, for example, monographs [3,5,20,22,24] and reviews of the scientific and practical results [6,8,17,21,23,28,34]. As a result, a great number of efficient methods for solving MCO problems have been proposed, and the solutions to many practical problems have been reported.

Possible contradictions between the partial efficiency criteria are meaningful issues with multicriterial optimization problems. As a result, finding the optimal (best)

Victor Gergel
Nizhni Novgorod State University, 23, Gagarin av., Nizhni Novgorod, Russia, e-mail: gergel@unn.ru

Evgeny Kozinov
Nizhni Novgorod State University, 23, Gagarin av., Nizhni Novgorod, Russia, e-mail: evgeny.kozinov@itmm.unn.ru

values for all partial criteria simultaneously is impossible. As a rule, improving the efficiency with respect to some partial criteria results in the reduced quality of chosen solutions with respect to other criteria. In these situations, solving a MCO problem consists in finding some compromise solutions, for which the obtained values are coordinated with respect to partial criteria. It is worth noting that the concept of an expedient compromise may change during the course of computations, which may require finding several different compromise solutions.

This work is dedicated to solving MCO problems which are used to describe decision making problems for designing complex engineered devices and systems. In such applications, the partial criteria may take a complex multiextremal form, and the computations of the criteria values are computationally expensive procedures as a rule. In these conditions, finding even one compromise solution requires a significant number of computations, whereas determining several Pareto-efficient solutions (or the entire set of them) becomes a challenging problem. In order to overcome this problem, maximizing the use of search information obtained during the course of computations is proposed. Within this approach, finding every next compromise solution requires fewer and fewer computations down to executing just a few iterations to find the next efficient solution.

This article is organized as follows. In Section 2, the multicriterial optimization problem statement is given. In Section 3, the basics of the approach are presented, namely reducing multicriterial problems to nonlinear programming problems using the minimax convolution of partial criteria, and reducing dimensionality using Peano evolvents. In Section 4, the multidimensional generalized global search algorithm is described for solving the reduced scalar nonlinear programming problems is described, and issues are discussed regarding the reuse of search information obtained during the course of computations. Section 5 includes results from computational experiments. In the Conclusion, the obtained results are discussed, and the main areas for further investigation are presented.

2 Problem Statement

A multicriterial optimization (MCO) problem can be defined as follows:

$$f(y) = (f_1(y), f_2(y), \dots, f_s(y)) \to \min, y \in D, \tag{1}$$

where:

- $y = (y_1, y_2, \dots, y_N)$ is the vector of varied parameters,
- N is the dimensionality of the multicriterial optimization problem being solved,
- D is the search domain being an N-dimensional hyperparallelepiped

$$D = \{y \in R^N : a_i \le y_i \le b_i, 1 \le i \le N\}$$

at a given boundary of vectors a and b. Without any loss of generality, the partial-criteria values in problem (1) are supposed to be non-negative, and reducing these corresponds to an increase in the efficiency of the considered solutions $y \in D$.

The partial criteria in MCO problem (1) are usually contradictory, and there is no solution $y \in D$ that provides optimal values for all criteria simultaneously. In these cases, such solutions $y^* \in D$ are considered to be the solutions of the MCO problem, for which the values of any particular criteria cannot be improved without reducing the efficiency with respect to other criteria. Such un-improvable solutions are called *efficient* or *Pareto-optimal*. Any efficient solution can be considered a *partial solution*, whereas the set of all un-improvable solutions is the *complete solution* of the MCO problem.

As mentioned above, in this work, problem (1) will be applied to the most complex decision making problems, for which the partial criteria $f_i(y)$, $1 \leq i \leq s$, can be multiextremal, and obtaining criteria values at the points of the search domain $y \in D$ may require a considerable number of computations. Let us also assume that the partial criteria $f_i(y)$, $1 \leq i \leq s$, satisfy the Lipschitz condition

$$|f_i(y') - f_i(y'')| \leq L_i \|y' - y''\|, y', y'' \in D, 1 \leq i \leq s. \quad (2)$$

where L_i is the Lipschitz constant for the criterion $f_i(y)$, $1 \leq i \leq s$.

It is important to note that the feasibility of the Lipschitz condition fits practical applications well – at small variations in the parameter $y \in D$, the corresponding changes of the partial criteria values are limited as a rule.

3 The Basics of the Approach

3.1 Methods of Solving the Multicriterial Optimization Problems

Multicriterial optimization is a field of intensive scientific investigations. Among the approaches developed for solving MCO problems, one can select a *lexicographical optimization* method where the criteria are arranged in a certain way according to their importance, and the optimization of partial criteria is performed step by step as the level of importance decreases – see, for example, [3]. *Interactive methods* [2,21] represent another approach where the researcher (decision maker, DM) is involved in the process of choosing solutions. Another extensively developed area is the development of *evolutionary algorithms* based on the imitation of certain natural phenomena for applying them to solving MCO problems [2,4,31,33].

The *scalarization approach*, in which some convolution methods of a set of partial criteria $f_i(y)$, $1 \leq i \leq s$, are applied to an integrated scalar functional $F(\lambda, y)$, is an extensively developed area for solving MCO problems – see, for example [5,6]. Such an approach reduces the solution of problem (1) to solving a nonlinear programming problem

$$\min F(\lambda,y), y \in D \tag{3}$$

where $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_s)$ is a vector of coefficients used to construct integrated scalar criterion. As part of this approach, a wide set of scalarization methods have been proposed for partial criteria. From among them, one can select various kinds of convolution, including the additive, minimax and multiplicative schemes.

Various methods for defining the preferred solutions, which should be obtained as a result of solving MCO problems, can also lead to scalar criterion. Among such approaches are methods for seeking solutions which are closest to the ideal or to compromise solutions or to existing prototypes, etc. More detailed consideration of this approach in given, for example, in [5,21,24].

The coefficients λ from (3), used in the approaches listed above, are often the requirements for an expedient compromise combination of the partial criteria. Thus, for example, the scalar criterion in the minimax convolution is defined as

$$F(\lambda,y) = \max_{1 \le i \le s} \lambda_i f_i(y) \tag{4}$$

where the coefficients λ_i, $1 \le i \le s$ should be non-negative, and their sum should be balanced to the unit value:

$$\sum_{i=1}^{s} \lambda_i = 1, \lambda_i \ge 0, 1 \le i \le s.$$

The necessity and sufficiency of this approach to solving MCO problems is one of the main properties of the minimax convolution scheme: the results of minimizing $F(\lambda,y)$ lead to obtaining efficient solutions[1] to MCO problems, and, vice versa, any efficient solution of the MCO problem can be obtained as a result of minimizing $F(\lambda,y)$ at the corresponding values of the convolution coefficients λ_i, $1 \le i \le s$ – see, for example, [21].

The coefficients λ_i, $1 \le i \le s$ in (4) can be considered indicators of the importance of the partial criteria – the greater the value of coefficient λ_i for some partial criterion, the greater the contribution of this partial criterion to the integrated scalar criterion $F(\lambda,y)$. Therefore, a method for solving MCO problems where the compromise solution sought is determined during the course of several stages performed sequentially. At every stage, the researcher (decision maker) specifies the desired values of the importance coefficients λ_i, $1 \le i \le s$ then problem (4) formulated in this way is solved. After that, the researcher analyzes the efficient solutions obtained and corrects the specified coefficients λ_i, $1 \le i \le s$ if necessary. Such a multistep method corresponds to the practice of decision making for compromise solutions in complex optimization problems. The possibility of determining several efficient solutions (or an entire set) at a reasonable computation cost becomes a key issue in solving complex multicriterial optimization problems.

[1] More precisely, the minimization of $F(\lambda,y)$ can lead to obtaining weakly efficient solutions (the set of weakly efficient solutions includes the Pareto domain). The situation can be corrected by adding an additional correcting element into (4) – see, for example, [21].

3.2 Dimensionality Reduction

In the general case, finding numerical estimates for globally optimized solutions implies generating coverage of the search domain D – see, for example, [7,9,18-19,25,32,35-36]. As a result, the computational costs of solving global optimization problems are very high, even with a relatively low number of varied parameters (the dimensionality of the problem). A considerable decrease in the number of computations can be achieved if the computing grids obtained when covering the search domain are non-uniform, when the optimization points are only denser in close proximity to the globally optimized solutions. The constricting of such economic non-uniform coverages considerably complicates the computational schemes of global optimization methods. One possible way to reduce this complexity involves using various dimensionality reduction methods [25,27,29-30].

Within the framework of the information-statistical theory of global optimization, Peano *curves* or *evolvents* $y(x)$ mapping the interval $[0,1]$ onto the N-dimensional hypercube D unambiguously are used for dimensionality reduction – see, for example [27,29-30]. As a result of this reduction, the initial multidimensional global optimization problem (4) is reduced to a one-dimensional problem:

$$F(\lambda, y(x^*)) = \min\{F(\lambda, y(x)) : x \in [0,1]\}. \tag{5}$$

The dimensionality reduction scheme considered associates a multidimensional problem with a Lipschitzian minimized function with a series of one-dimensional problems, for which the corresponding objective functions satisfy the uniform Hölder condition (see [29-30]) i. e.

$$|F(\lambda, y(x')) - F(\lambda, y(x''))| \leq H|x' - x''|^{1/N}, x', x'' \in [0,1], \tag{6}$$

where the Hölder constant H is defined by the relationship $H = 4L\sqrt{N}$, where L is the Lipschitz constant of the function $F(\lambda, y)$, and N is the dimensionality of the optimization problem (4).

It can be noted that a nested optimization scheme can also be applied for dimensionality reduction – see [1,11-12,30].

4 An Efficient Method for Solving the Multicriterial Optimization Problems Based on Reusing Search Information

The basics of the approach presented in Section 3 allow the solution of the MCO problem (1) to be reduced to solving a series of reduced multiextremal problems (5). And, therefore, the problem of developing methods for solving the MCO problem is resolved by the potentially broad application of global search algorithms.

4.1 Method for Solving Global Optimization Problems

It should be pointed that multiextremal optimization is an area of extensive research – the general state of the art is presented, for example, in [9,18-19,25,30,32,35], etc. The main results from applying dimensionality reduction using Peano evolvents have been obtained through the information-statistical theory of global search developed in [29-30]. This theory has served as the basis for developing a large number of efficient methods for multiextremal optimization – see, for example [1,11-15,26], etc.

Within the framework of this approach, the Generalized Multidimensional Algorithm of Global Search (GMAGS) [13,29-30] forms the basis for the optimization methods being developed. The general computational scheme of this algorithm can be presented as follows.

Let us introduce a simpler notation for reduced one-dimensional problems (5) as

$$\phi(x) = F(\lambda, y(x)) : x \in [0,1]. \tag{7}$$

The initial iteration of the algorithm is performed at an arbitrary point $x^1 \in (0,1)$. Then, let us assume k, $k > 1$ global search iterations to be completed. The choice of the optimization point of the next $(k+1)^{th}$ iteration is determined by the following rules.

Rule 1. Renumber the optimization points by the lower indices in the order of increasing coordinate value

$$0 = x_0 < x_1 < \cdots < x_i < \cdots < x_k < x_{k+1} = 1, \tag{8}$$

the points x_0, x_{k+1} have been introduced additionally for the convenience of further explanation, the values of the minimized function z_0, z_{k+1} at these points are undefined.

Rule 2. Compute the current estimate of the Hölder constant H from (6)

$$m = \begin{cases} rM, & M > 0 \\ 1, & M = 0 \end{cases}, M = \max_{1 \le i \le k} \frac{|z_i - z_{i-1}|}{\rho_i} \tag{9}$$

as the relative difference in the values of the minimized functions $\phi(x)$ from (7) on the set of the points of the executed iterations x_i,$1 \le i \le k$ from (8). Here and hereafter $\rho_i = (x_i - x_{i-1})^{1/N}$, $1 \le i \le k+1$. The constant r, $r > 1$ is the *parameter* for the algorithm.

Rule 3. For each interval (x_{i-1}, x_i), $1 \le i \le k+1$ compute the *characteristic* $R(i)$ where

$$R(i) = \rho_i + \tfrac{(z_i - z_{i-1})^2}{m^2 \rho_i} - 2\tfrac{(z_i + z_{i-1})}{m}, 1 < i \le k,$$

$$R(i) = 2\rho_i - 4\tfrac{z_i}{m}, i = 1, \tag{10}$$

$$R(i) = 2\rho_i - 4\tfrac{z_{i-1}}{m}, i = k+1$$

Rule 4. Determine the interval with the maximum characteristic

$$R(t) = \max_{1 \leq i \leq k+1} R(i) \tag{11}$$

Rule 5. Execute a new trial (computing the value of the minimized function $\phi(x)$) at the point x^{k+1} placed in the interval with the maximum characteristic from (11)

$$\begin{gathered} x^{k+1} = \frac{x_t + x_{t-1}}{2} - sign(z_t - z_{t-1}) \frac{1}{2r} [\frac{|z_t - z_{t-1}|}{m}]^N, 1 < i \leq k \\ x^{k+1} = \frac{x_t + x_{t-1}}{2}, t = 1, t = k+1. \end{gathered} \tag{12}$$

The termination condition, by which the trials are terminated, is defined by the condition

$$\rho_t < \varepsilon, \tag{13}$$

for the interval t with the maximum characteristic $R(t)$ form (11) and $\varepsilon > 0$ is the given accuracy of the solution. If the termination condition is not fulfilled, the iteration number k is incremented by unity, and a new iteration of the global search is performed.

To clarify the presented algorithm, let us note the following. The concrete form of the characteristics $R(i)$, $1 \leq t \leq k+1$, calculated in (10) have been developed with the framework of the information-statistical theory of global optimization, and can be interpreted as some measure of the importance of the intervals with respect to containing the global minimum point within them. As it can be seen, the characteristics stimulate the selection of new iteration points within the longest intervals and with the smallest values of the function to be minimized. Then, the scheme for choosing the interval to execute the next trial described in (11)-(12) becomes clear – the point of every next iteration of a global search is chosen in the interval in which the occurrence of the global minimum point is the most probable.

The condition where the presented algorithm converges has been considered, for example, in [30]. Thus, at a proper estimate of the Hölder constant ($m > 2^{2-1/N}H$, m is from (9)) the algorithm converges to all existing global minimum points.

It is worth noting that the results obtained for GMAGS in this work are applicable to the majority of multiextremal optimization methods that can be formulated in accordance with the general characteristic scheme [15]. Moreover, the proposed approach can be extended efficiently to parallel computations – see, for instance [13,30].

4.2 *Reusing Search Information to Efficiently Solve Multicriterial Problems*

Solving multicriterial optimization problems may require a large number of computations. The main problem for computation costs consists of the fact that, in general, multiextremal problems (5) must be solved several times. The full usage of all search information obtained during the course of computations could overcome this problem.

The numerical solution of multicriterial optimization problems usually consists of successive computations of the partial criteria values $f^i = f(y^i)$ at the points y^i, $1 \leq i \leq k$, of the search domain D (see, for example, the rules of the GMAGS algorithm). The search information obtained as a result of the computations can be represented as a set (*set of the search information*, SSI):

$$\Omega_k = \{(y^i, f^i = f(y^i))^T : 1 \leq i \leq k\}. \tag{14}$$

It is important to note that SSI contains all of the available information on the optimization problem being solved, and any possible increasing in the efficiency of the global search can be organized based on the information stored in SSI. Given that, the size of the search information when solving complex multidimensional problems may appear to be large enough. However, as a rule, any reduction in the stored data volume results in executing excess global search iterations.

As a result of scalarizing vector criterion (3) or (4), dimensionality reduction (5), and the need for the ordered placement of the trial points (see Rule 1 of GMAGS) SSI is transformed into a *set of optimization data* (SOD)

$$A_k = \{(x_i, z_i, l_i)^T : 0 \leq i \leq k+1\}, \tag{15}$$

where

- x_i, $0 \leq i \leq k+1$ are the reduced points of the executed global search iterations, in which the criteria values have been computed; the arranged placement of the points according to Rule 1 of GMAGS is reflected by the use of the lower index i. e.
$$0 = x_0 < x_1 < \dots < x_{k+1} = 1;$$
- z_i, $0 \leq i \leq k+1$ are the scalar criterion values for the current optimization problem (5) being solved at the points x_i, $0 \leq i \leq k+1$, i. e.
$$z_i = \phi(x_i) = F(\lambda, y(x_i)), 1 \leq i \leq k$$
(the values z_0, z_{k+1}, are undefined and are not used in the computations);
- l_i, $1 \leq i \leq k$ are the global search iterations indices, for which the points x_i, $1 \leq i \leq k$ have been computed; these indices are used to store the correspondence between the reduced points of the executed iterations and the multidimensional ones, i.e.
$$y^j = y(x_i), j = l_i, 1 \leq i \leq k.$$

In contrast to SSI, the set of optimization data contains the search information reduced to the current scalar reduced problem (5) being solved. In addition, the search information in SOD is represented in a form allowing efficient performing of the global search algorithms. Thus, for example, SOD supports the ordered placement of iteration points necessary for GMAGS – see Rule 1.

The definition of SSI and SOD has a very crucial impact on the essential reduction of the computational cost for solving multicriterial optimization problems. The optimization methods may use SOD to adaptively perform the scheduled search iterations (taking into account the results of previous computations). And, as the main contribution of the proposed approach, the availability of SSI allows the previous computations to be recalculated in SOD for the values of the current optimization problem (5) to be solved without any costly computation of the values for the partial criteria $f_i(y)$, $1 \leq i \leq s$ from (1). And, therefore, all search information can be employed in the ongoing computations – when searching for the next Pareto-optimal solution, GMAGS can start with optimization data from SOD instead of starting the computations from scratch. Within this approach, Rule 1 of the GMAGS algorithm can be formulated as follows:

Rule 1 (updated). Take the iteration points x_i, $0 \leq i \leq k+1$, and the scalar criterion values z_i, $0 \leq i \leq k+1$, from SOD (15).

(as can be seen, the iteration points in SOD are placed in order of increasing coordinate values).

In general, reusing search information requires fewer and fewer computations for solving each successive optimization problem down to executing just a few iterations to find the next efficient solutions.

5 Results of Computational Experiments

In this section, the results of computational experiments are presented. In the beginning, the ability to apply other optimization methods to solving multicriterial optimization problems was estimated. It is worth noting that the comparison of the methods needed to estimate this possibility should be performed on the optimization problems, the solution upon which the considered approach is oriented (the criteria can be multiextremal and computationally expensive – see Section 2). As a result, within the framework of the comparison performed, the multicriterial optimization methods, which are oriented to simpler MCO problem statements (for example, with the linear optimality criteria) may not be considered. It should also be stressed that in this paper, the approach based on scalarizing the vector criterion (see Section 3) is applied. Within this approach, each scalar problem (3) is a multiextremal optimization. Thus, the problem of comparing multicriterial optimization methods can be reduced to comparing the global search algorithms. Such comparisons have been reported in a sufficiently large number of publications – see, for example [1,11-15,26].

The experiments were organized as follows. First of all, a comparison of the GMAGS algorithm with the DIRECT method [10], which is a widely used global optimization method, was performed on a large set of multiextremal optimization problems. Experimental results have shown that GMAGS is more efficient. Based on these results and on the results of comparing various multiextremal optimization methods [1,11-15,26] in other experiments, the efficiency evaluation has only been

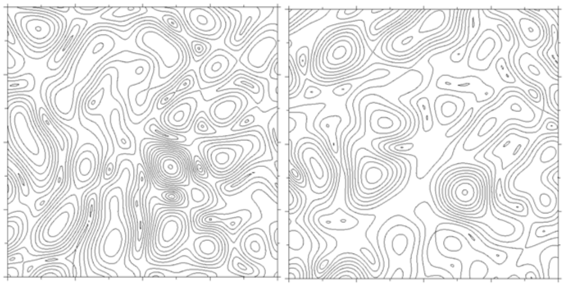

Fig. 1 Contour plots of two multiextremal functions from the test optimization problem family

performed for GMAGS. First, solving bi-criteria univariate MCO problems was performed. Then, the experiments for bi-criteria two-dimensional optimization problems were performed. Finally, experiments for a multicriterial optimization problem with 10 criteria were conducted.

In the first series of experiments, GMAGS is compared with the DIRECT method [10], which is a well-known global optimization method. The set of the test optimization functions consists of the multiextremal functions defined by relationship [14]:

$$\begin{aligned} AB &= \left(\textstyle\sum_{i=1}^{7} \sum_{j=1}^{7} [A_{ij}a_{ij}(y_1,y_2) + B_{ij}b_{ij}(y_1,y_2)] \right)^2 \\ CD &= \left(\textstyle\sum_{i=1}^{7} \sum_{j=1}^{7} [C_{ij}a_{ij}(y_1,y_2) - D_{ij}b_{ij}(y_1,y_2)] \right)^2 \\ \phi(y_1,y_2) &= -\{AB+CD\}^{1/2} \end{aligned} \tag{16}$$

where

$$\begin{aligned} a_{ij}(y_1,y_2) &= \sin(\pi i y_1)\sin(\pi j y_2), \\ b_{ij}(y_1,y_2) &= \cos(\pi i y_1)\cos(\pi j y_2) \end{aligned}$$

are defined in the domain $0 \le y_1, y_2 \le 1$, and the parameters $-1 \le A_{ij}, B_{ij}, C_{ij}, D_{ij} \le 1$ are independent random numbers distributed uniformly. The minimization of such functions arises, for example, in the problem for estimating the maximum strain on a thin plate (determining its strength) at the transversal loading. The contour plots of two functions from this family are shown in Fig. 1 – one can see that these types of function are essentially multiextremal.

In order to draw more substantiated conclusions on the efficiency of the compared methods, a set of 100 multiextremal problems has been solved.

For GMAGS the reliability parameter $r = 3$ and the search accuracy $\varepsilon = 0.01$ were used. All 100 problems have been solved with the required accuracy and the average number of executed optimization iterations is 512. For DIRECT, a search accuracy $\varepsilon = 10^{-6}$ has been used. In this case 91 problems have been solved and the average number of executed optimization iteration is 688. With a lower accuracy $\varepsilon = 10^{-2}$ the number of problems solved was 37.

Additional results from numerical comparisons of the GMAGS and DIRECT methods were also presented in [26].

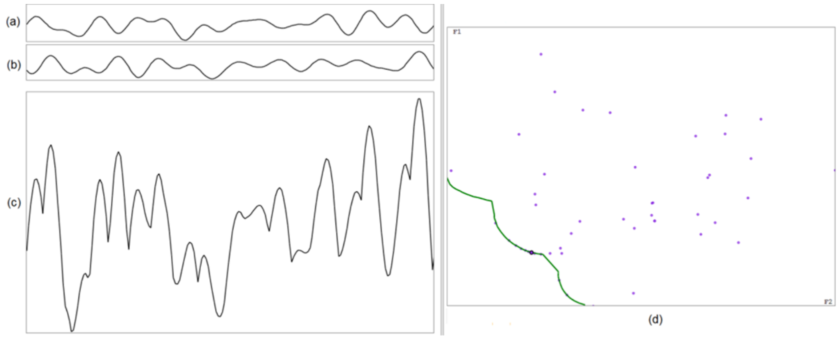

Fig. 2 An example of the test MCO problem: the criteria (a,b), the minimax convolution for $\lambda_1 = \lambda_2 = 0.5$ (), the Pareto domain and the criteria values at the executed optimization iteration points (d)

In the next series of experiments, bi-criteria univariate MCO problems, i.e. for $N = 1$, $s = 2$ were solved. Like the criteria for the problems, the multiextremal functions are defined by the relationships [16]:

$$\phi(x) = A_0 + \sum_{i=1}^{14} (A_i \sin(2i\pi x) + B_i \cos(2i\pi x)), 0 \leq x \leq 1$$

where the coefficients A_0, A_i, B_i, $1 \leq i \leq 14$ were generated by a random number generator within the interval $[-1, 1]$. A graph of the function from such MCO problems is presented in Fig. 2: the criteria and the minimax convolution for $\lambda_1 = \lambda_2 = 0.5$ are shown in the left panel, and a view of the Pareto domain and the criteria values at the executed optimization iteration points are shown in the right panel (the ordinate axis corresponds to the values of the first criterion, the abscissa axis – to the values of the second criterion).

To optimize the criteria convolution (5), GMAGS presented in Section 4 with the reliability parameter $r = 2$ and the search accuracy $\varepsilon = 0.001$ has been used. In each experiment, solving the problem (5) at several values of the convolution coefficients λ (from 1 up to 50 different values) has been performed both with search information and without it. The results are presented in Table 1.

The results presented here demonstrate that using search information in the experiments allowed the number of optimization iterations to be reduced in the solving multicriterial optimization problems by 3.2 to 18.9 times. In addition, as previously mentioned, solving each subsequent scalar problem (5) in order to find the next particular solution for the MCO problems requires performing fewer and fewer optimization iterations. For a clear presentation of this key property of the developed approach, the results of experiments are shown in Table 2 separately for the sequence of groups of solved problems (5); each group includes the scalar problems (5) for 10 values of the convolution coefficients λ (in total, 50 coefficients were selected for solving the MCO test problem).

Table 1 Experimental results for univariate MCO problems both with and without search information

Number of convolution coefficient values	Computations without using search information		Computations using search information		**Reduction in the number of optimization iterations**
	Total	Average per problem	Total	Average per problem	
1	34	34	34	34	**1**
10	497	49,7	153	15,3	**3,2**
20	1087	54,35	171	8,55	**6,4**
30	1578	52,6	174	5,8	**9,1**
40	2466	61,65	176	4,4	**14**
50	3363	67,26	178	3,56	**18,9**

Table 2 The results of experiments for univariate MCO problems shown separately for the groups of 10 problems each

Number of problems included in a group	Computations without using search information		Computations using search information		**Reduction in the number of optimization iterations**
	Total	Average per problem	Total	Average per problem	
1-10	497	49,7	153	15,3	**3,2**
11-20	590	59	18	1,8	**32,8**
21-30	491	49,1	3	0,3	**163,7**
31-40	888	88,8	2	0,2	**444**
41-50	897	89,7	2	0,2	**448,5**

The results of the experiments presented in Table 2 demonstrate a considerable reduction in the number of optimization iterations as the amount of obtained information increases. Thus, when solving a group of 10 scalar problems (5), starting from 21 convolutions of the partial criteria, only 2-3 optimization iterations are required until the termination condition is sufficiently fulfilled. The number of optimization iterations for the group of 41 to 50 problems was reduced by 448.5 times.

In the following series, experiments were performed to solve bi-criteria two-dimensional MCO problems, i.e. for $N = 2$, $s = 2$. The problems were formed using the multiextremal functions from (16) according to the rule

$$f_i(y) = (f_{1i}(y), f_{2i}(y)), 1 \le i \le 100,$$

where i, $1 \le i \le 100$ is the serial number of the multicriterial problem, and

$$f_{1i}(y) = \phi_i(y)$$

$$f_{2i}(y) = \begin{cases} \phi_{i+50}(y), & 1 \le i \le 50 \\ \phi_{i-50}(y), & 51 \le i \le 100 \end{cases}$$

(the lower indices at the functions $\phi_i(y)$ indicate the serial numbers for the multiextremal functions within the family of optimization problems).

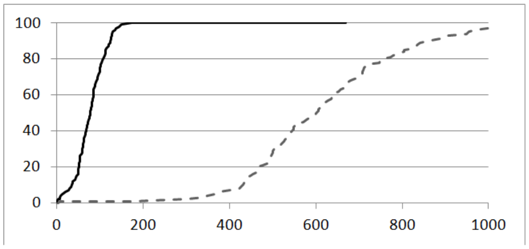

Fig. 3 Operational characteristics of GMAGS when solving multicriterial optimization problems without using the search information (dashed line) and with the search information (solid line)

The evaluation of the set of the efficient solutions (the Pareto set) necessary to conduct computational experiments has been performed numerically for each multicriterial problem being solved by means of scanning all the nodes of a uniform (within the search domain D) grid with a step of 10^{-3} in each coordinate (i.e., the grid had 1 million nodes total).

For a fuller representation of the efficiency of the developed approach, the GMAGS operational characteristics (see [14,30]) have been constructed according to the results of the experiments performed. The operational characteristic (OC) of a method is a curve demonstrating the dependence of the number of solved problems from a certain class (the ordinate axis) on the number of trials (the abscissa axis) and is a set of pairs:

$$OC = \{(k, p(k)) : 1 \leq k \leq K\},$$

where k is the number of the optimization iterations, $p(k)$ is the fraction of the test class problems successfully solved within a given number of iterations, and K is the total number of the executed iterations. These indicators can be calculated based on the results of numerical experiments and are shown graphically in the form of a piecewise line graph. In general, one can examine the operational characteristic to show the probability of finding the global minimum with the required accuracy subject to the number of optimization iterations performed by the method.

The termination condition for GMAGS when solving the scalar problem (5) was the falling of the global optimization iteration point into the δ-nearness of any efficient solution from Pareto domain i. e.

$$\|y^k - y^*\| < \delta.$$

When carrying out the computational experiments, the following parameter values were used: the reliability parameter $r = 3$, the search accuracy $\varepsilon = 0.001$, and the size of δ-nearness $\delta = 0.001$. When solving each multicriterial optimization problem, a search has been performed for the efficient solutions to 50 various criteria convolution coefficients distributed uniformly.

The operational characteristics calculated based on the experimental results are presented in Fig. 3. The dashed line in Fig. 3 corresponds to solving the problems without reusing the search information while the solid line corresponds to reusing the search information.

As one can see, the results of this series of computational experiments confirm a significant reduction in the number of optimization iterations for solving multicriterial optimization problems using the search information obtained during the course of computations. Thus, within the execution of about 150 global optimization iterations without reusing search information, no problems were solved. The use of the search information has allowed practically all problems to be solved.

Similar to the data presentation on solving one-dimensional problems (Table 2), the computational results are shown separately for the sequence of the groups of solved problems (5); 10 scalar problems (5) have been included in each group.

Table 3 The results of experiments for groups of 10 problems separately (the data have been averaged over the results of solving 100 multicriterial two-dimensional problems)

Number of problems included in a group	Computations without using search information		Computations using search information		**Reduction in the number of optimization iterations**
	Total	Average per problem	Total	Average per problem	
1-10	5821,2	582,1	1232	123,2	**4,7**
11-20	7244,4	724,4	582,6	58,3	**12,4**
21-30	5258,2	525,8	512,7	51,3	**10,3**
31-40	6736,1	673,6	876,8	87,7	**7,7**
41-50	6112,1	611,2	779,5	78,0	**7,8**

According to the results presented above, the use of the search information in this series of experiments reduced the amount of optimization iterations performed up to 7.8 times.

In the final experiment, the solution of the two-dimensional MCO problem with 10 criteria, i.e. for $N = 2$, $s = 10$, has been determined. The problems were formed using the multiextremal functions from (16). The following parameter values were used: the reliability parameter $r = 3.4$, search accuracy $\varepsilon = 0.005$, and the size of δ-nearness $\delta = 0.01$. As in previous experiments, the MCO problem (5) are solved using 50 various values of the convolution coefficients (49 problems have been solved with the required accuracy). The executed experiments demonstrated that the average number of optimization iterations for solving a single problem (5) without using the accumulated search information is 1813. In the case when GMAGS is taking into account the search information, the average number of optimization iterations is 305 (more than a 5-fold reduction).

6 Conclusions

In this paper, an efficient approach is proposed for solving complex multicriterial optimization problems, where the optimality criteria may be multiextremal, and computing the criteria values may require a large volume of computations. The basis of the proposed approach consists of reducing multicriterial problems to nonlinear programming problems using the minimax convolution of the partial criteria, dimensionality reduction using Peano evolvents, and applying the efficient information-statistical methods of global optimization.

The key aspect of the approach consists in overcoming the high computational complexity of the global search for multiple efficient solutions in solving multicriterial optimization problems. A considerable enhancement in the efficiency and significant reduction in the volume of computations was achieved by maximizing the use of all search information obtained during the course of computations. For this purpose, it was necessary to provide the ability to store a large amount of search information, efficient processing, and using the search data during the course of solving multicriterial optimization problems. Within the framework of the developed approach, methods have been proposed for reducing all available search information to the values of current scalar nonlinear programming problem being solved. The search information reduced to the current state is used by the optimization methods for adaptive planning of the global search iterations to be performed.

According to the results of the computational experiments, this approach significantly reduces the computational costs for solving multicriterial optimization problems – by tens and hundreds times.

In conclusion, it is worth noting that the approach developed is promising and requires further investigation. First of all, it is necessary to continue conducting computational experiments to solve multicriterial optimization problems for a larger number of partial criteria of efficiency and for greater dimensionality of the optimization problems to be solved. Also, it is necessary to estimate the ability to organize parallel computations because of the high computation costs of solving global optimization problems.

Acknowledgements

This work has been supported by Russian Science Foundation, project No 16-11-10150 ”Novel efficient methods and software tools for time-consuming decision making problems using supercomputers of superior performance.”

References

1. Barkalov K.A., Gergel V.P. Multilevel scheme of dimensionality reduction for parallel global search algorithms. In: Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization., pp. 2111–2124 (2014).
2. Branke J., Deb K., Miettinen K., Slowinski R., editors. (2008). Multi-Objective Optimization - Interactive and Evolutionary Approaches. Springer, Berlin.
3. Collette Y., Siarry P. (2011). Multiobjective Optimization: Principles and Case Studies (Decision Engineering). Springer-Verlag, Berlin, Heidelberg.

4. Deb K. (2001). Multi-Objective Optimization using Evolutionary Algorithms. Wiley, Chichester.
5. Ehrgott M. (2005). Multicriteria Optimization. Springer. (2nd ed., 2010).
6. Eichfelder G. Scalarizations for adaptively solving multi-objective optimization problems. In: Comput. Optim. Appl., **44**, pp. 249–273 (2009).
7. Evtushenko Yu.G., Posypkin M.A. A deterministic algorithm for global multi-objective optimization. In: Optimization Methods and Software, **29(5)**, pp. 1005–1019 (2014).
8. Figueira J., Greco S., Ehrgott M., editors. (2005). Multiple criteria decision analysis: State of the art surveys. Springer, New York.
9. Floudas C.A., Pardalos M.P. (2016). Recent Advances in Global Optimization. Princeton University Press.
10. Gablonsky J.M., Kelley C.T. A Locally-Biased Form of the DIRECT Algorithm. In: Journal of Global Optimization, **21(1)**, 27–37 (2001).
11. Gergel V.P., Grishagin V.A., Gergel A.V. Adaptive nested optimization scheme for multidimensional global search. In: Journal of Global Optimization, **66(1)**, 1–17 (2015).
12. Gergel V., Grishagin V., Israfilov R. Local Tuning in Nested Scheme of Global Optimization. In: Procedia Computer Science, **51**, pp. 865–874 (2015).
13. Gergel V.P., Strongin R.G. Parallel Computing for Globally Optimal Decision Making on Cluster Systems. In: Future Generation Computer Systems, **21(5)**, pp. 673–678 (2005).
14. Grishagin V.A. Operating characteristics of some global search algorithms. In: Problems of Stochastic Search, **7**, pp. 198–206 (1978) (In Russian).
15. Grishagin V.A., Sergeyev Y.D., Strongin R.G. Parallel characteristic algorithms for solving problems of global optimization. In: Journal of Global Optimization, **10**, pp. 185–206 (1997).
16. Hill J.D. A Search Technique for Multimodal Surfaces. In: IEE Transactions on Systems and Cybernetics, **5(1)**, pp. 2–8 (1969).
17. Hillermeier C., Jahn J. Multiobjective optimization: survey of methods and industrial applications. In: Surv. Math. Ind., **11**, pp. 1–42 (2005).
18. Horst R., Tuy H. (1990). Global Optimization: Deterministic Approaches. Springer-Verlag, Berlin.
19. Locatelli M., Schoen F. (2013). Global Optimization: Theory, Algorithms, and Applications. SIAM.
20. Koksalan M.M., Wallenius J. and Zionts S. (2011). Multiple criteria decision making: From early history to the 21st century. World Scientific, Singapore.
21. Marler R.T., Arora J.S. Survey of multi-objective optimization methods for engineering. In: Struct. Multidisciplinary Optimization, **26**, pp. 369–395 (2004).
22. Marler R.T., Arora J.S. (2009). Multi-Objective Optimization: Concepts and Methods for Engineering. VDM Verlag.
23. Mardani A., Jusoh A., Nor K., Khalifah Z., Zakwan N., Valipour A. Multiple criteria decision-making techniques and their applications – a review of the literature from 2000 to 2014. In: Economic Research-Ekonomska Istraživanja , **28(1)**, pp. 516–571 (2015). doi: 10.1080/1331677X.2015.1075139
24. Miettinen K. (1999). Nonlinear Multiobjective Optimization. Springer.
25. Pintér J.D. Global Optimization in Action (continuous and Lipschitz optimization: algorithms, implementations and applications). Kluwer Academic Publishers, Dortrecht.
26. Sergeyev Y.D., Kvasov D.E. Global search based on efficient diagonal partitions and a set of Lipschitz constants. In: SIAM J. on Optimization, **16(3)**, 910–937 (2006).
27. Sergeyev Y.D., Strongin R.G., Lera D. (2013). Introduction to global optimization exploiting space filling curves. Springer.
28. Siwale I. Practical multi-objective programming. In: Technical Report RD-14-2013. Apex Research Limited.
29. Strongin R.G. (1978). Numerical methods in multiextremal problems: information-statistical algorithms Nauka, Moscow (in Russian).
30. Strongin R., Sergeyev Ya. (2000). Global optimization with non-convex constraints. Sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht.

31. Tan K.C., Khor E.F., Lee T.H. (2005). Multi-objective Evolutionary Algorithms and Applications. Springer-Verlag, London.
32. Torn A., Zilinskas A. (1989). Global Optimization. In: Lecture Notes in Computer Science 350. Springer-Verlag, Berlin.
33. Yang X.-S. (2008). Nature-inspired metaheuristic algorithms. Luniver Press, Frome.
34. Zavadskas E.K., Turskis Z., Kildiene S. State of art surveys of overviews on MCDM/MADM methods. In: Technological and Economic Development of Economy, **20**, pp. 165–179 (2014).
35. Zhigljavsky A.A. (1991). Theory of Global Random Search. Dordrecht: Kluwer Academic Publishers.
36. Zilinskas A., Zilinskas J. Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems. In: Commun Nonlinear Sci Numer Simulat, **21**, pp. 89–98 (2015).