A Highly Parallel Approach for Solving Computationally Expensive Multicriteria Optimization Problems [*]

Victor Gergel[0000−0002−4013−2329] and Evgeny Kozinov[0000−0001−6776−0096]

Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia
gergel@unn.ru, evgeny.kozinov@itmm.unn.ru

Abstract. In this paper, a highly parallel approach for solving multicriteria optimization problems is proposed. The considered approach is based on the reduction of the multicriterial problems to the global optimization ones using the minimax convolution of the partial criteria, the dimensionality reduction with the use of the Peano space-filling curves, and the application of the efficient parallel information-statistical global optimization methods. The required computations can be time-consuming since functions representing individual criteria can be multiextremal and computationally expensive. The proposed approach comprises two different schemes for efficient parallel computations on high performance systems with shared and distributed memory and with a large number of computational units. The computational efficiency is achieved by storing all the computed criteria values and their intensive reuse for finding new solutions. The results of numerical experiments have demonstrated that this approach allows to reduce the computational costs of solving multicriteria optimization problems by a factor between 10 and 100.

Keywords: Multicriteria optimization · Parallel computing · Dimensionality reduction · Criteria convolution · Global optimization algorithm · Computational complexity

1 Introduction

The multicriteria optimization (MCO) problems arise in a general formulation of a decision making problem. Reviews of the state of the art in multicriteria optimization can be found in [1–3, 12] and reviews of relevant techniques and applications in [4–7, 13, 24].

At the same time, the MCO problems are some of the most complex ones. To solve an MCO problem compromise decisions need to be found and having to find a representative set of those increases the computational complexity of

[*] 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 superior-performance supercomputers."

solving the MCO problems. Bearing in mind that finding even a single efficient solution may require a large amount of computations, obtaining the whole Pareto set or even a limited subset of efficient decisions becomes a problem of high computational complexity. In order to solve such problems huge computational capabilities of high-performance systems are required.

The structure of this paper is as follows. Section 2 presents a general statement of a multicriteria optimization problem. A scheme of parallel computations to simultaneously solve a set of multicriteria global optimization problems is proposed in Section 3. A parallel algorithm of multicriteria global search is presented in Section 4. To demonstrate the proposed techniques, Section 5 presents results of numerical experiments.

2 Statement of a Multicriteria Optimization Problem

The problem of multicriteria (or multi-objective) optimization (MCO) can be defined as follows:

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

where $y = (y_1, y_2, \dots, y_N)$ is a vector of varied parameters describing a system being designed, N is the dimensionality of the optimization problem, and D is the search domain being an N-dimensional hyperinterval

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

with given boundary vectors a and b. Without a loss of generality, the efficiency criteria values in the problem (1) are assumed to be non-negative, and the decrease of these ones corresponds to the increase of the efficiency of the considered solutions $y \in D$. In addition it is assumed the efficiency criteria $f_i(y)$, $1 \le i \le s$ can be multiextremal, and the obtaining of the criteria values at the points of the search domain $y \in D$ can require a large amount of computations. It is assumed that the efficiency criteria $f_i(y)$, $1 \le i \le s$ satisfy the Lipschitz condition

$$|f_i(y') - f_i(y'')| \le L_i ||y' - y''||, y', y'' \in D, 1 \le i \le s \tag{2}$$

where L_i, $1 \le i \le s$ are the Lipschitz constants for the corresponding efficiency criteria $f_i(y)$, $1 \le i \le s$, and $||.||$ denotes the norm in the R^N space.

As *a solution* of the MCO problem, any efficient decision is considered. In a general case, it is required to find the whole set of Pareto-optimal solutions $PD(f, D)$, i.e. *the complete solution of an MCO problem.*

A general approach to solving the MCO problem used in this paper consists in the replacement of solving the original MCO problems by solving a series of simpler optimization problems:

$$\begin{aligned} \min\,(\phi(x) = F(\lambda, y(x)), y(x) \in D, x \in [0, 1]), \\ F(\lambda, y(x)) = \max\,(\lambda_i f_i(y(x)), 1 \le i \le s), \\ \lambda = (\lambda_1, \lambda_2, \dots, \lambda_s) \in \Lambda \subset R^s : \sum_{i=1}^{s} \lambda_i = 1, \lambda_i \ge 0, 1 \le i \le s \end{aligned} \tag{3}$$

where $F(\lambda, y(x))$ is the minimax convolution of the efficiency criteria of the MCO problem [2,6,19] with the use of the vector of the convolution coefficients $\lambda \in \Lambda$ and $y(x)$ is a continuous and unambiguous mapping of the interval [0,1] onto an N-dimensional search domain D – see, for example, [7,8].

The necessity and sufficiency of this approach for solving MCO problems is a key property of the minimax convolution. The result of the minimization of $F(\lambda, y)$ leads to obtaining an efficient variant[1] of the MCO problem and vice versa, any efficient variant of an MCO problem can be obtained as a result of the minimization of $F(\lambda, y)$ at corresponding values of the convolution coefficients λ_i, $1 \le i \le s$. In order to obtain several efficient decisions (or in order to evaluate the whole Pareto domain) the problem (3) should be solved for the corresponding set of values of the vector $\lambda \in \Lambda$.

The dimensionality reduction applied in (3) allows to replace solving multidimensional MCO problems by the optimization of one-dimensional functions $F(\lambda, y(x))$ which satisfy the uniform Hölder condition i.e.

$$|F(\lambda, y(x')) - F(\lambda, y(x''))| \le H|x' - x''|^{1/N}, x', x'' \in [0,1], \tag{4}$$

where the constant H is defined as $H = 2L\sqrt{N+3}$, L is the Lipschitz constant of the function $F(\lambda, y)$ and N is the dimensionality of the optimization problem (1).

3 Parallel Computations for Solving the Multicriteria Global Optimization Problems

The scalarization of the vector criterion allows reducing the solving of the MCO problem (1) to solving a series of the multiextremal problems (3). Therefore, the problem of development of the methods for solving the MCO problems is resolved by the possibility of an extensive use of efficient parallel global optimization algorithms.

Multiextremal optimization is a research area being actively developed – its state of the art is presented, for example, in [7,9,10]. The information-statistical theory of global optimization is considered to be one of the promising approaches [7]. HPC systems are used widely for solving time-consuming global search problems [7,11,14,15].

The proposed approach to the use of parallel computations for solving time-consuming global optimization problems is based on the following main statements:

- The parallelism of the performed computations is provided by means of simultaneous computing the values of the efficiency criteria $f_i(y)$, $1 \le i \le s$ at several points of the search domain D. Such an approach provides the

[1] More precisely, the minimization of $F(\lambda, y)$ can lead to the obtaining of the weakly-efficient variants (the set of the weakly-efficient decisions includes the Pareto domain).

parallelization of the most computation-costly part of global optimization and is a general one – it can be applied with many global optimization methods for various global optimization problems.
- The parallel computations are provided by means of simultaneous solving of several global optimization problems (3) for varying values of the coefficients λ_i, $1 \leq i \leq s$. For solving the problems of the family (3), a set of computational nodes of the high performance systems with distributed memory can be applied.
- The optimization data obtained in the course of parallel computations is exchanged between all employed processors because of the information compatibility of the global optimization problems of the family (3).

All these statements will be considered in more details below.

3.1 General Scheme of Parallel Computations

As mentioned above, when solving a multicriteria optimization problem (1), solving a series of scalar problems (3) with varying coefficients of the minimax convolution of the efficiency criteria may be required in order to find several different efficient decisions:

$$\Phi(y) = \{\phi_1(y), \dots, \phi_q(y)\}, \phi_i(y) = F(\lambda_i, y), 1 \leq l \leq q. \tag{5}$$

Such problems can be solved sequentially by various global optimization methods. Alternatively, these problems can be solved simultaneously using several processors. It is important to note that the obtained family of the one-dimensional problems $\Phi(y)$ is an information-dependent one because the values of the optimized functions computed for any problem $\phi_l(y)$, $1 \leq l \leq q$ can be transformed to the values of all the remaining problems of the family without time-consuming recalculations of the efficiency criteria values $f_i(y)$, $1 \leq i \leq s$ (see Section 3.2).

The information compatibility of the problems from the family (5) allows to propose the following unified scheme of parallel computations:

1. The family of the information-linked problems $\Phi(y)$ in (5) is distributed among the processors of the computing system.
2. For solving the optimization problems (5), a global optimization method implemented on each processor should be updated following the following rules:
   (a) Upon completing the iteration for any problem $\phi_l(y)$, $1 \leq l \leq q$ at any point $y' \in D$, the point y' with the particular criteria values $f_i(y)$, $1 \leq i \leq s$ computed at this point should be transferred to all employed processors.
   (b) Prior to the start of the next global search iteration, the method should check the queue of the received messages; if there are any data in the queue, the received information should be included into the search information.

The possibility of the asynchronous data transfer is a key feature of such a scheme of the parallel computations. It is important to note that this scheme does not depend on any of the single control nodes, and the number of computational nodes can vary in the course of global optimization.

3.2 Accumulation and Reuse of Calculated Optimization Data

For solving the stated optimization problems, the values of the efficiency criteria $f^i = f(y^i)$ at the points y^i, $1 \leq i \leq k$ of the search domain D are computed. The search information obtained as a result of these computations can be represented in the form of the *Search Information Set* (SIS):

$$\Omega_k = \{(y^i, f^i = f(y^i))^T : 1 \leq i \leq k\}. \tag{6}$$

As a result of the dimensionality reduction, the search information Ω_k from (6) can be transformed into the *Matrix of the Search State* (MSS)

$$A_k = \{(x_i, z_i, l_i)^T : 1 \leq i \leq k\} \tag{7}$$

where x_i, $1 \leq i \leq k$ are the reduced points of the executed global search iterations[2], z_i, $1 \leq i \leq k$ are the values of the scalar criterion of the current optimization problem (3), l_i, $1 \leq i \leq k$ are the indices of the optimization iterations where the points x_i, $1 \leq i \leq k$ were computed.

The matrix of the search state can be used by the optimization algorithms in order to improve the efficiency of the global search – selecting the points for the scheduled iterations can be performed taking into account the results of all previously executed computations.

3.3 Parallel Computations on Multiprocessors with Shared Memory

Within the proposed approach, for solving the reduced one-dimensional multiextremal optimization subproblems (3) it is proposed to use the well-known Parallel Algorithm of Global Search (PAGS) developed within the framework of the information-statistical theory of the multiextremal optimization [7]. This method has a good theoretical background and has demonstrated its efficiency as compared to other global search algorithms (see also the results of numerical experiments in Section 4).

For completeness, let us briefly discuss the general computational scheme of PAGS, which consists in the following.

Let p be the number of employed parallel computers (processors or cores) of a computational system with shared memory. The initial two iterations of the algorithm are performed at the boundaries of the interval $x^0 = 0$, $x^1 = 1$. Apart from these boundary points, the algorithm should perform additional iterations at the points x^i, $1 < i \leq p$ which can be defined *a priori* or computed by any

[2] The lower indices denote the increasing order of the coordinate values of the points x_i, $1 \leq i \leq k$.

auxiliary computational procedure. Then, let $(k > p)$ global search iterations be completed, at each of which the computation of the value of the minimized function $\phi(x)$ from (3) (hereafter called *a trial*) has been performed. The choice of the points for trials performed within the next iteration in parallel is determined by the following rules.

Rule 1. Renumber the trial points of the completed search iterations with the lower indices in the order of increasing coordinate values

$$0 = x_0 < x_1 < \cdots < x_i < \cdots < x_k = 1. \tag{8}$$

Rule 2. Compute the current estimate of the Hölder constant of the function $\phi(x)$:

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

where $z_i = \phi(x_i)$, $\varrho_i = \sqrt[N]{x_i - x_{i-1}}$, $1 \le i \le k$. The constant r, $r > 1$, is the *reliability parameter* of the algorithm.

Rule 3. Compute *the characteristic* $R(i)$ for each interval (x_{i-1}, x_i), $1 \le i \le k$ according to the expression

$$R(i) = \varrho_i + \frac{(z_i - z_{i-1})^2}{m^2 \varrho_i} - 2\frac{(z_i + z_{i-1})}{m}, 1 \le i \le k. \tag{10}$$

Rule 4. Arrange the characteristics of the intervals (x_{i-1}, x_i), $1 \le i \le k$ obtained according to (10) in the decreasing order

$$R(t_1) \ge R(t_2) \ge \cdots \ge R(t_{k-1}) \ge R(t_k) \tag{11}$$

and select p intervals with the indices t_j, $1 \le j \le p$ having the maximum values of the characteristics.

Rule 5. Perform new trials at the points x^{k+j}, $1 \le j \le p$ placed in the intervals with the maximum characteristics from (11) according to the expressions

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - sign(z_{t_j} - z_{t_j-1}) \frac{[\frac{|z_{t_j} - z_{t_j-1}|}{m}]^N}{2r}, 1 \le j \le p. \tag{12}$$

Termination condition for the algorithm, according to which the execution of the algorithm is terminated, consists of checking the lengths of the intervals in which the scheduled trials are performed with respect to a required *accuracy* of the problem solution, i. e.

$$\varrho_{t_j} \le \varepsilon, 1 \le t_j \le p. \tag{13}$$

As the current estimate of the optimization problem solution, the minimum computed value of the optimization function is accepted i. e.

$$z_k^* = \min\{z_i : 1 \le i \le k\}. \tag{14}$$

Within the framework of the proposed approach, PAGS is applied to solving the MCO problem (1) according to the scheme (3). The availability of SIS from (6) allows transforming the results of the previous computations to the values of a next optimization problem (3) without any time-consuming computations of the efficiency criteria values for new values of the convolution coefficients, i.e.

$$z'_i = \max\,(\lambda'_j f^j_i, 1 \le j \le s), 1 \le i \le k. \tag{15}$$

As a result, all the search information accumulated so far can be employed for continuing the computations. In general, the reuse of the search information will provide a smaller and smaller amount of computations for solving every successive optimization problem (see the results of numerical experiments in Section 4).

The method obtained as a result of such extension is called hereafter Parallel Multicriteria Global Algorithm (PMGA) for multiprocessors with shared memory. The PMGA algorithm in combination with a general scheme of the parallel computations presented in Section 3.1 will be referred to as the Multilevel Parallel Multicriteria Global Algorithm (MPMGA) for high-performance computational systems with shared and distributed memory.

Theoretical properties of MPMGA can be established by a theoretical analysis of the sequential one-dimensional variant of the method, i.e. the MGA algorithm. Here is a summary (without a proof) of the theoretical results from [16].

Theorem 1. When switching from solving the optimization subproblem (3) with the convolution coefficients $\lambda' \in \Lambda$ to solving the next subproblem (3) with the coefficients $\lambda'' \in \Lambda$, the value of the function $F(\lambda'', y(x))$ at the point of the global minimum differs from the estimated minimum value z^*_k from (14) obtained by minimizing the function $F(\lambda', y(x))$, by no more than the bounded value

$$|z^*_k - F(\lambda'', y(x^*))| \le \frac{\alpha H \varepsilon}{2} + \delta, \alpha > 1, \tag{16}$$

where δ is the maximum possible change of the function $\phi(x)$ (being minimized) from one subproblem (3) to the next:

$$\begin{aligned} \delta = \Delta\phi_{max} &= \max\,\{\Delta\lambda_{max} f^{max}_i, 1 \le i \le s\}, \\ \Delta\lambda_{max} &= \max\,\{|\lambda'_i - \lambda''_i|, 1 \le i \le s\}, \\ f^{max}_i &= \max\,\{f_i(y), y \in D\}, 1 \le i \le s. \end{aligned} \tag{17}$$

and for the value m from (9) the following inequality is satisfied

$$m \ge H\left(1 + \sqrt{\frac{\alpha + 4\beta(1+\beta)}{\alpha - 1}}\right), \beta = \frac{2r\delta}{H\varepsilon(r-1)}$$

with z^*_k from (14), $\phi(x^*)$ from (3), ε from (13), r from (9) and H from (4).

This statement means that if the deviation $\Delta\phi_{max}$ from (17) of the estimates of the minimum values of $F(\lambda'', y(x))$ is acceptable, then solving the subproblem

(3) with the new convolution coefficients $\lambda'' \in \Lambda$ does not require any additional optimization iterations because the estimates of the minimum value of the function $F(\lambda'', x)$ can be obtained according to (16) using the values z_i, $1 \leq i \leq k$, located within the search information A_k from (7).

As a result, two different schemes for selecting various convolution coefficients $\lambda \in \Lambda$ can be proposed, with whose use the required conditions of the Theorem 1 will be satisfied. In the first scheme, initially a sparse grid can be built on the set Λ, so that when setting new values of λ, the values that are close to the ones already established on the grid built in Λ earlier could always be found. In the second scheme, new values of λ could be chosen by small perturbations of the values used earlier (this method is often used in solving practical MCO problems when the decision maker wants to refine the estimates of the efficient solutions obtained earlier). It is worth noting that the reuse of the search information could be beneficial even with significant differences in the chosen values of the coefficients λ from the previously established ones, see the results of numerical experiments in Section 4.

4 Results of Numerical Experiments

The numerical experiments have been carried out on the *Lobachevsky* supercomputer at State University of Nizhni Novgorod. Each supercomputer node has 2 Intel Sandy Bridge E5-2660 2.2 GHz octa-core processors, 64 Gb RAM.

To evaluate the efficiency the proposed approach, the MGA[3] algorithm was compared with a number of other popular multicriteria optimization methods. In [20] MGA was tested against four multicriteria optimization methods: the Monte-Carlo (MC) method where the trial points are selected within the search domain D randomly and uniformly, the genetic algorithm SEMO from the PISA library [21], the Non-Uniform Coverage (NUC) method [17] and the Bi-objective Lipschitz Optimization (BLO) method [18].

In [16] a series of numerical experiments were executed to compare MGA with four other MCO methods considered in [22], namely the Linear Combination Method (LCM), the Multi-Objective Genetic Algorithm (MOGA), the Global Criterion Method (GCM), the ε-Constraint Method (ECM).

The results of the numerical experiments have demonstrated that MGA has a considerable advantage compared to the considered multicriteria optimization methods even when solving the relatively simple MCO problems.

In this paper, we present the results of computational experiments to evaluate the efficiency of the MPMGA algorithm. The computations have been carried out for 100 of four-dimensional MKO problems with ten criteria, i.e. $N = 4$, $s = 10$. The criteria in these problems were obtained by the GKLS generator [23] that generates multiextremal optimization problems with a priori known properties: the number of local minima, the size of their attraction domains, the point of the global minimum, etc.

[3] MGA is the sequential implementation of the parallel MPMGA method

In these experiments 100 multicriteria problems has been solved. To obtain the Pareto domain approximation, each problem has been solved for 50 coefficients λ uniformly distributed in Λ from (3). The obtained results were averaged over the number of solved MCO problems. The accuracy ε from (13) was 0.025 and the reliability parameter r from (9) was 5.6.

The following approach was used to evaluate the accuracy of the Pareto domain approximation calculated for each of the 100 solved MCO problems. First of all, for the domain D from (1) a uniform grid D_δ was constructed with the step $\delta = 0.01$ for each coordinate. On this grid D_δ, the minimum values were found for all 50 subproblems $F(\lambda, y(x))$ from (3) generated in accordance with the chosen set of coefficients $\lambda \in \Lambda$ – the resulting set $y \in PD_\delta(f, D) \subset D_\delta$ was taken as the δ-approximation of the Pareto domain of the MCO problem being solved.

To solve each subproblems $F(\lambda, y(x))$ in (3), the termination condition (13) was replaced by the termination condition by the condition

$$F(\lambda, y(x^{k+j})) \leq F(\lambda, y^*) + \varepsilon, 1 \leq j \leq p, F(\lambda, y^*) = \min_{y \in D_\delta} F(\lambda, y).$$

After solving subproblems (3) for 50 values of λ a numerical approximation of the Pareto domain $PD_{\Omega_k}(f, D)$ is constructed using the search information accumulated in Ω_k from (6) as result of the solution of the MCO problem. The resulting estimation of the accuracy of the calculated approximation $PD_{\Omega_k}(f, D)$ is defined as the closeness to the δ-approximation $PD_\delta(f, D)$ of the Pareto domain of the MCO problem

$$\Delta = \frac{1}{M} \sum_{y \in PD_\delta(f,D)} d(y), d(y) = \min_{\overline{y} \in PD_{\Omega_k}(f,D)} \|y - \overline{y}\|, \tag{18}$$

where $\|.\|$ denotes the norm in the R^N space, and M is the number of points used in the approximation $PD_\delta(f, D)$.

Results of the numerical experiments are presented in Table 1. The first two columns in Table 1 show the numbers of the processors (P) and of the parallel computational cores on each processor (Q) employed. The third column (P*Q) contains the total number of cores employed. In the fourth (*Iters*) column the average numbers of iterations spent on solving the problems for the corresponding numbers of the different coefficients λ from (3) are given. In the fifth column (*Nums*) the average numbers of the calculated solutions used to establish the approximation $PD_{\Omega_k}(f, D)$ are shown. In the sixth column (*Approx*) the accuracy Δ of the δ-approximation $PD_\delta(f, D)$ is presented.

The last two columns show the speedup [4] of the parallel computations obtained with the use of the search information ($S1$) and without ($S2$).

The obtained results of experiments demonstrate that even simple reuse of the search information allows reducing the total amount of computations by the

[4] Due to the initial assumption that the computational complexity of multicriteria optimization problems is determined by the complexity of calculations of the criteria values, the speedup is defined as the reduction of the number of executed iterations.

Table 1. Numerical results for solving four-dimensional problems with ten criteria

P	Q	P*Q	Iters	Nums	Approx	S1	S2
Computations without the reuse of the search information							
1	1	1	82,244.11	26,014.27	0.05		1,0
Computations with the reuse of the search information							
1	1	1	23,791.81	8,185.07	0.08	1.00	3.5
5	1	5	4,977.63	8,077.38	0.08	4.78	16.5
1	20	20	1,223.45	8,452.35	0.08	19.45	67.2
25	1	25	1,063.22	8,704.86	0.08	22.38	77.4
1	40	40	616.37	8,416.06	0.07	38.60	133.4
50	1	50	594.46	9,561.09	0.07	40.02	138.4
5	20	100	252.65	8,510.26	0.07	94.17	325.5
5	40	200	131.68	8,914.21	0.07	180.68	624.6
25	20	500	66.54	11,022.32	0.07	357.56	1236.0
25	40	1000	36.81	12,226.82	0.07	646.34	2234.3
50	20	1000	37.94	12,552.88	0.07	627.09	2167.7
50	40	2000	20.98	14,399.94	0.07	1,134.02	3920.1

factor of 3.5 without the use of additional computational resources. When 1000 computational cores were used, the obtained speedup varied from 627 to 646. If 2000 computational cores are used, the speedup with the reuse of the search information reaches 1134. The overall speedup in this case relative to the initial algorithm without the reuse of the search information was 3920.

5 Conclusion

In this paper, a new computationally efficient approach is proposed that allows a parallel solution of time-consuming multicriteria optimization problems where individual criteria can be multiextremal and their evaluation may require a large computing effort. The main feature of this approach is its efficient handling of the computational complexity of solving such multicriteria optimization problems. Improvements of the efficiency resulting in a significant reduction of the required computations have been achieved by an intensive use of the search information obtained in the course of computations. Within the framework of this approach, methods for a re-use of the already available search information for a currently handled scalar nonlinear pro-gramming problem have been proposed. Such search information is then used by the optimization methods for the adaptive planning of the iterative process of global search.

Results of numerical experiments demonstrate that such an approach allows reducing the computation costs of solving multicriteria optimization problems by a factor between tens and hundreds.

In conclusion, it was shown that the developed approach is a promising one that needs a further investigation. It is important to continue numerical experiments of solving multicriteria optimization problems with a larger number of criteria and of a larger dimensionality.

References

1. Miettinen, K.: Nonlinear Multiobjective Optimization. In: Springer (1999)
2. Ehrgott, M.: Multicriteria Optimization. In: Springer (2nd ed., 2010)
3. Collette, Y., Siarry, P.: Multiobjective Optimization: Principles and Case Studies (Decision Engineering). In: Springer (2011)
4. Marler, R. T., Arora, J. S.: Survey of Multi-Objective Optimization Methods for Engineering. In: Struct. Multidisciplinary Optimization, **26**, pp. 369–395 (2004)
5. Figueira,J., Greco, S., Ehrgott, M. (eds.): Multiple Criteria Decision Analysis: State of the Art Surveys. In: New York (NY), Springer (2005)
6. Eichfelder, G.: Scalarizations for Adaptively Solving Multi-Objective Optimization Problems. In: Comput. Optim. Appl., **44**, pp. 249–273 (2009)
7. Strongin, R., Sergeyev, Ya.: Global Optimization with Non-Convex Constraints. Sequential and Parallel Algorithms. In: Kluwer Academic Publishers, Dordrecht (2000, 2nd ed. 2013, 3rd ed. 2014)
8. Sergeyev, Y.D., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. In: Springer (2013)
9. Floudas, C.A., Pardalos, M.P.: Recent Advances in Global Optimization. In: Princeton University Press (2016)
10. Locatelli, M., Schoen, F.: Global Optimization: Theory, Algorithms, and Applications. In: SIAM (2013)
11. Sergeyev, Y.D., Grishagin, V.A.: Parallel Asynchronous Global Search and the Nested Optimization Scheme. In: J. Comput. Anal. Appl., **3**(2), pp. 123–145 (2001)
12. Marler, R. T., Arora, J. S.: Multi-Objective Optimization: Concepts and Methods for Engineering. In: VDM Verlag (2009)
13. Hillermeier, C., Jahn, J.: Multiobjective Optimization: Survey of Methods and Industrial Applications. In: Surv. Math. Ind., **11**, pp. 1–42 (2005)
14. Gergel, V., Sidorov, S.: A Two-Level Parallel Global Search Algorithm for Solution of Computationally Intensive Multiextremal Optimization Problems. In: LNCS, Springer-Verlag, Berlin Heidelberg, **9251**, pp. 505–515 (2015)
15. Gergel, V.: An Unified Approach to Use of Coprocessors of Various Types for Solving Global Optimization Problems. In: 2nd International Conference on Mathematics and Com-puters in Sciences and in Industry, pp. 13–18 (2015)
16. Gergel, V., Kozinov, E.: Efficient Multicriteria Optimization Based on Intensive Reuse of Search Information. In: Journal of Global Optimization, **71**(1), pp. 73–90 (2018)
17. 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)
18. Žilinskas, A., Žilinskas, 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)
19. Pardalos, P.M., Žilinskas, A., Žilinskas, J.: Non-Convex Multi-Objective Optimization. In: Springer (2017)
20. Gergel, V., Kozinov, E.: Accelerating Parallel Multicriterial Optimization Methods Based on Intensive Using of Search Information. In: Procedia Computer Science, **108**, pp. 1463–1472 (2017)
21. Bleuler, S., Laumanns, M., Thiele, L., Zitzler, E.: PISA-A Platform and Programming Language Independent Interface for Search Algorithms. Evolutionary Multi-Criterion Optimization. In: LNCS, Springer-Verlag, Berlin Heidelberg, **2632**, pp. 494–508 (2003)

22. Chiandussi, G., Codegone, M., Ferrero, S., Varesio, F.E.: Comparison of Multi-Objective Optimization Methodologies for Engineering Applications. In: Comput. Math. Appl., **63**(5), pp. 912–942 (2012)
23. Gaviano, M., Kvasov, D.E., Lera, D., Sergeyev, Ya.D.: Software for Generation of Classes of Test Functions with Known Local and Global Minima for Global Optimization. In: ACM Transactions on Mathematical Software, **29**(4), pp. 469–480 (2003)
24. Borisenko, A., Gorlatch, S.: Parallelizing Metaheuristics for Optimal Design of Multiproduct Batch Plants on GPU. In: LNCS, Springer, Cham, **10421**, pp. 405–417 (2017)