GPU-based Parallel Computations in Multicriterial Optimization

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 the present paper, an efficient approach for solving the time-consuming multicriterial optimization problems, in which the optimality criteria could be the multiextremal ones and computing the criteria values could require a large amount of computations is proposed. The proposed approach is based on the reduction of the multicriterial problems to the scalar optimization ones with the use of the minimax convolution of the partial criteria, on the dimensionality reduction with the use of the Peano space-filling curves, and on the application of the efficient information-statistical global optimization methods. An additional application of the block multistep scheme provides the opportunity of the large-scale parallel computations with the use of the graphics processing units (GPUs) with thousands of computational cores. The results of the numerical experiments have demonstrated such an approach to allow improving the computational efficiency of solving the multicriterial optimization problems considerably – hundreds and thousands.

Keywords: Decision making · Multicriterial optimization · Global optimization · High performance computations · Dimensionality reduction · Criteria convolution · Global search algorithm · Computational costs

1   Introduction

The multicriterial optimization (MCO) problems are classified as the most general statements of the decision-making problems – the statement of MCO problems covers many classes of optimization problems, including unconstrained optimization, nonlinear programming, global optimization, etc. The opportunity to set several criteria is very useful in the formulating of the complex decision-making problems and is used in the applications widely. Such practical importance has caused a high activity of research in the field of the MCO problems. As a result of the performed investigations, a large number of the efficient methods of solving the MCO problems have been proposed and a great number of the applied problems have been solved – see, for example, the monographs [1–4] and the reviews of the scientific and practical results in the field [5,6].

2   V. Gergel et al.

The present paper is devoted to the solving of the MCO problems, which are used for formulating the decision-making problems in the design of complex technical objects and systems. In such applications, the partial criteria could take a complex multiextremal form, and computing the values of the criteria could require a large amount of computations. In such conditions, finding even a single efficient decision requires a significant amount of computations whereas finding several decisions (or the complete set of these ones) becomes a problem of high computation costs.

Among the directions used for solving the MCO problems widely, the scalarization approach utilizing some methods of the partial criteria convolution to a single scalar criterion is applied – see, for example, [2, 4, 7]. Among such approaches, there are the methods of finding the decisions, which are the closest to the ideal one or to the compromised ones, or to the prototypes existing actually, etc. Among such algorithms, there exists the method of successive concessions, in which some tolerances to possible values of criteria are introduced. The scalarization of a vector criterion allows reducing the solving of a MCO problem to solving a series of the multiextremal optimization problems and, therefore, utilizing all existing highly efficient global search algorithms for the multicriterial optimization.

One of the promising approaches to solving the time-consuming global optimization problems consists in utilizing the graphics processing units (GPUs). At present, a GPU is a high-performance flexible programmable massive parallel processor, which can provide solving many complex computational problems [15]. However, the use of the GPU computational potential in the field of global optimization is quite limited. As a rule, GPUs are used for the parallelization of the algorithms, which are based on the random search concept in any way (see [16–18]). A review of this direction is given in [19].

Further structure of the paper is as follows. In Section 2, the multicriterial optimization problem statement is given and the basics of the developed approach are considered, namely the reduction of the multicriterial problems to the scalar optimization ones using the minimax convolution of the partial criteria and the dimensionality reduction using the Peano space-filling curves. In Section 3, the parallel global search algorithm for solving the reduced scalar optimization problems is described and the block multistep scheme of the dimensionality reduction, which provides the opportunity to use the GPUs with thousands of computational cores is presented. Section 4 includes the results of numerical experiments confirming the proposed approach to be a promising one. In Conclusion, the obtained results are discussed and main possible directions of further investigations are outlined.

2   Problem Statement

The multicriterial optimization (MCO) can be defined as follows:

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

where $y = (y_1, y_2, \ldots, y_N)$ is the vector of the varied parameters, N is the dimensionality of the multicriterial optimization problem being solved, $f(y)$ is the vector efficiency criterion, and D is the search domain representing an N-dimensional hyperparallelepiped

$$D = y \in R^N : a_i \leq y_i \leq b_i, 1 \leq i \leq N \tag{2}$$

at given boundary vectors a and b.

Without loss of generality, the values of partial criteria in the problem (1) are supposed to be non-negative, and the decreasing of these ones corresponds to increasing efficiency of the decisions $y \in D$.

In the present work the problem (1) will be considered in application to the most complex decision-making problems, in which the partial criteria $f_i(y)$, $1 \leq i \leq s$ could be multiextremal, and obtaining the criteria values at the points of the search domain $y \in D$ could require a considerable amount of computations. Let us suppose also the partial criteria $f_i(y)$ to satisfy the Lipschitz condition

$$|f_i(y') - f_i(y'')| \leq L_i \|y' - y''\|, y', y'' \in D, 1 \leq i \leq s. \tag{3}$$

where L_i is the Lipschitz constant for the functions $f_i(y)$, $1 \leq i \leq s$ and $\| * \|$ denotes the Euclidean norm in R^N.

The general approach to solving the MCO problem applied in the present work consists in the reduction of solving the MCO problems to the solving of a series of one-dimensional optimization problems:

$$\min \phi(x) = F(\lambda, y(x)), x \in [0, 1], \tag{4}$$

where

$$F(\lambda, y(x)) = \max\left((\lambda_i f_i(y(x)), 1 \leq i \leq s\right) \tag{5}$$

is the minimax convolution of the partial criteria of the MCO problem with the use of the vector of the convolution coefficients

$$\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_s) \in \Lambda \subset R^s : \sum_{i=1}^{s} \lambda_i = 1, \lambda_i \geq 0, 1 \leq i \leq s \tag{6}$$

and $y(x)$ is a continuous and unambiguous mapping of the interval $[0, 1]$ onto the N-dimensional search domain D – see, for example, [8–10].

3   GPU-based Parallel Computations for Solving the Multicriterial Optimization Problems

The convolution of the partial criteria applied within the framework of the developed approach and the dimensionality reduction allow reducing the solving of the MCO problem (1) to solving a series of the reduced multiextremal problems (4). And, therefore, the problem of the development of the methods for

4   V. Gergel et al.

solving the MCO problems is resolved by the opportunity of a wide use of the global search algorithms. The state of the art in the field of global optimization is presented comprehensively enough, for example, in [8, 20, 21].

In the present work, the global search algorithm developed within the framework of the information-statistical theory of the multiextremal optimization is proposed to use for solving the reduced problems (4). This theory served as a basis for the development of a large number of algorithms, which have been substantiated mathematically, have demonstrated high efficiency, and allowed solving many complex optimization problems in various fields of application [8,9,22–25].

The approach proposed for the organization of the parallel computations when solving the time-consuming multiextremal optimization problems is based on the simultaneous computing of the partial criteria values in the MCO problem (1) at several different points of the search domain D. Such an approach provides the parallelization of the most time-consuming part of the global search and is a general one – it can be applied for many global search methods in various global optimization problems.

3.1   Parallel Algorithm of Global Search for Finding the Efficient Decisions in the Multicriterial Optimization Problems

Within the framework of this approach, the multidimensional generalized parallel algorithm of global search (PAGS) for finding the efficient decisions of the multicriterial optimization problems constitutes the basis for the developed optimization methods. The general computational scheme of the algorithm can be described as follows [8–10].

Let p is 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 $x^0 = 0$, $x^1 = 1$. Besides these boundary points, the algorithm should perform additional iterations at the points x^i, $1 < i \le p$, which can be defined *a priori* or computed by any auxiliary computational procedure. Then, let k, $k > p$ global search iterations have been completed, at each of which the computing of the value of the minimized function $\phi(x)$ from (4) (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 trials points of the completed search iterations by the lower indices in the order of increasing coordinate values

$$0 = x_0 < x_1 < \cdots < x_i < \cdots < x_k = 1. \tag{7}$$

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

$$m = \begin{cases} rM, & M > 0 \\ 1, & M = 0 \end{cases}, M = \max_{1 \le i \le k} \frac{|z_i - z_{i-1}|}{\varrho_i} \tag{8}$$

where $z_i = \phi(x_i)$, $\varrho_i = \sqrt[N]{x_i - x_{i-1}}$, $1 \leq i \leq k$. 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), $1 \leq i \leq k$ according to the expression

$$R(i) = \varrho_i + \frac{(z_i - z_{i-1})^2}{m^2 \varrho_i} - 2\frac{(z_i + z_{i-1})}{m}, 1 \leq i \leq k, \qquad (9)$$

Rule 4. Arrange the characteristics of the intervals (x_{i-1}, x_i), $1 \leq i \leq k$ obtained according to (9) in the decreasing order

$$R(t_1) \geq R(t_2) \geq \cdots \geq R(t_{k-1}) \geq R(t_k) \qquad (10)$$

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

Rule 5. Perform new trials at the points x^{k+j}, $1 \leq j \leq p$ placed in the intervals with the maximum characteristics from (10) according to the expressions

$$x^{k+j} = \frac{x_{t_j} + x_{t_j-1}}{2} - sign(z_{t_j} - z_{t_j-1})\frac{[\frac{|z_{t_j} - z_{t_j-1}|}{m}]^N}{2r}, 1 \leq j \leq p. \qquad (11)$$

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

$$\varrho_t \leq \varepsilon, 1 \leq t_j \leq p. \qquad (12)$$

Various modifications of this algorithm and the corresponding theory of convergence are presented in [8, 9].

3.2   Multilevel Decomposition of the Parallel Computations

Further development of the methods of the parallel computations in the multicriterial global optimization problems and, therefore, the expansion of the possible quantity of the employed processors/cores can be ensured by the use of one more dimensionality reduction method in the decomposition scheme of the MCO problems (1) – the multistep scheme of decomposition of the optimization problems [8, 9, 25, 26]. According to this scheme, the solution of a multidimensional optimization problem can be obtained by solving a series of nested one-dimensional problems:

$$\min\{\phi(y) : y \in D\} = \min_{a_1 \leq y_1 \leq b_1} \min_{a_2 \leq y_2 \leq b_2} \cdots \min_{a_N \leq y_N \leq b_N} \phi(y). \qquad (13)$$

The original multistep reduction scheme (13) can be generalized for the use in combination with the dimensionality reduction scheme based on the Peano

6   V. Gergel et al.

curves [27]. According to the generalized block multistep scheme, the vector of variables $y \in D$ of the global optimization problem (1) is considered as a set of the block variables

$$y = (y_1, y_2, \ldots, y_N) = (u_1, u_2, \ldots, u_M), \qquad (14)$$

where the ith block variable u_i is a vector with the dimensionality N_i of the elements of the vector y taken sequentially i.e.

$$u_i = (y_{n_i+1}, y_{n_i+2}, \ldots, y_{n_i+N_i}),$$
$$n_0 = N_0 = 0, n_i = n_{i-1} + N_{i-1}, 1 \leq i \leq M \qquad (15)$$

and $N_1 + N_2 + \cdots + N_M = N$.

Using the new variables, main equation of the multistep reduction scheme (13) can be rewritten in the form

$$\min \{\phi(y) : y \in D\} = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \ldots \min_{u_M \in D_M} \phi(y). \qquad (16)$$

where the subdomains D_i, $1 \leq i \leq M$ are the projections of the initial search domain D onto the subspace corresponding the variables u_i, $1 \leq i \leq M$. As a result, in the generalized block multistep reduction scheme, the nested subproblems

$$\phi_i(u_1 \ldots u_i) = \min_{u_{i+1} \in D_{i+1}} \phi_{i+1}(u_1 \ldots u_i, u_{i+1}), 1 \leq i \leq M - 1 \qquad (17)$$

are the multidimensional ones, and the dimensionality reduction method based on the Peano curves can be applied to solve these ones.

To provide the parallel computations in the block multistep reduction scheme, at each decomposition level one can generate several optimization problems simultaneously for the parallel solving of these ones [26, 27] (see Figure 1). The resulting set of problems to be solved in parallel can be controlled by means of the predefined parallelization vector

$$\pi = (\pi_1, \pi_2, \ldots, \pi_M), \qquad (18)$$

where π_i, $1 \leq i < M$ is the number of subproblems being solved in parallel at the $(i+1)^{th}$ level of decomposition arising as a result of performing the parallel iterations at the ith level. For the Mth level, the quantity π_M means the number of parallel trials in the course of minimization of the function

$$\phi_M(u_1, \ldots, u_M) = \phi(y_1, \ldots, y_N) \qquad (19)$$

with respect to the variable u_M at fixed values $u_1, \ldots, u_{M-1}$, i.e. the number of values of the objective function $\phi(y)$ computed in parallel. Then, the total number of the employed processors/cores will be

$$\prod = 1 + \sum_{i=1}^{M-1} \prod_{j=1}^{i} \pi_j \qquad (20)$$

The resulting multilevel scheme of parallel computations allows ensuring the efficient employment of all processors/nodes available in the high-performance systems with a large number of the computational nodes (including the ones with the distributed memory). The generation of a large number of optimization problems solved in parallel initiates a promising direction on a wide employment of GPUs with a large number of computational cores. This direction has been tested in solving the time-consuming global optimization problems [28–30]. In the present work, the possibility of utilizing the GPUs for solving the multicriterial optimization problems has been evaluated.
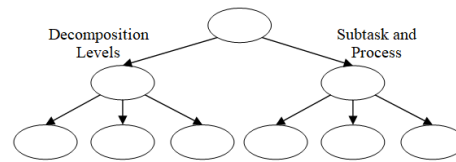

Fig. 1. General scheme of generating the parallel problems using the block multistep dimensionality reduction scheme

The PAGS algorithm combined with the block multistep scheme of dimensionality reduction will be called hereafter Generalized Parallel Algorithm of Global Search (GPAGS).

4   Results of Numerical Experiments

The numerical experiments have been carried out on the Lobachevsky supercomputer at State University of Nizhni Novgorod (operating system – CentOS 6.4, management system – SLURM). Each supercomputer node included 2 Intel Sandy Bridge E5-2660 2.2 GHz, 64 Gb RAM processors. The central processor units were the 8-core one (i. e. total 16 CPU cores per a node were available). At each node, three NVIDIA Kepler K20 GPUs were installed. To provide parallel computations MPI and CUDA technologies are applied.

The evaluation of the efficiency of the developed approach for solving the MCO problems without using the computational accelerators has already been performed earlier [11–14]. Let us consider, for example, the results of experiments from [13]. For comparison, a bicriterial test problem proposed in [31] was used:

$$f_1(y) = (y_1 - 1)y_2^2 + 1, f_2(y) = y_2, 0 \leq y_1, y_2 \leq 1. \tag{21}$$

As a solution of this MCO problem, the construction of a numerical approximation of the Pareto domain (PDA) was considered. For evaluating the quality of approximation, the completeness and the uniformity of the coverage of the Pareto domain were compared with the use of the following two indicators [13, 31]:

8   V. Gergel et al.

- The *hypervolume index* (HV) defined as the volume of the subdomain of the values of the vector criterion $f(y)$ dominated by the points of the Pareto domain approximation. This indicator characterizes the completeness of the Pareto domain approximation (a higher value corresponds to more complete coverage of the Pareto domain).
- The *distribution uniformity* index (DU) of the points from the Pareto domain approximation. This indicator characterizes the uniformity of coverage of the Pareto domain (a lower value corresponds to more uniform coverage of the Pareto domain).

Within the described experiment, five multicriterial optimization algorithms were compared: the Monte-Carlo (MC) method, the genetic algorithm SEMO from the PISA library [5, 32], the Non-uniform coverage (NUC) method [5], the bi-objective Lipschitz optimization (BLO) method proposed in [32], and the serial variant of the GPAGS algorithm proposed in the present paper.

Total for the GPAGS algorithm 50 subproblems were solved at various values of the convolution coefficients λ distributed in Λ uniformly. The results of experiments from [13] are presented in Table 1.

Table 1. Results of numerical experiments from [13] for the test problem (21)

Method	Iterations	PDA points	HV	DU
MC	500	67	0.300	1.277
SEMO	500	104	0.312	1.116
NUC	515	29	0.306	0.210
BLO	498	68	0.308	0.175
GPAGS	370	100	0.316	0.101

The results of performed experiments have demonstrated the GPAGS algorithm to have a considerable advantage as compared to considered multicriterial optimization methods even in solving relatively simple MCO problems.

In the present paper, the numerical experiments were conducted in order to evaluate the efficiency of the developed approach in solving the MCO problems with the use of the GPUs. In the conducted series of experiments, the solving of the bicriterial six-dimensional MCO problems (i.e. $N = 6$, $s = 2$) has been performed. As the test problem criteria, the multiextremal functions obtained with the use of the GKLS generator [33] were used.

In the course of experiments, 50 multicriterial problems of this class have been solved. In each problem, the search of the Pareto-optimal decisions was performed for 10 convolution coefficients λ from (4) distributed in Λ uniformly (i.e. 500 global optimization subproblems have been solved). In solving the problems, two levels of the block dimensionality reduction scheme were used. At the first level of the reduction scheme, the optimization with respect to the first two variables was performed (i.e. $u_1 = (y_1, y_2)$, $N_1 = 2$). The optimization with respect to the rest variables was performed at the second decomposition level

(i.e. $u_2 = (y_3, y_4, y_5, y_6)$, $N_2 = 4$). Trials at the second decomposition level are executed on GPU. Computations are implemented in accordance with the master/slave scheme which is not required any data communication between GPUs. Trial points are sent by CPU just before every iteration and are stored in the GPU global memory.

As the parameters the accuracy of the method $\varepsilon = 0.025$ and the reliability of the method $r = 6.5$ for the first level of the block reduction scheme and $r = 4.5$ for the second level of decomposition were used. The results of the numerical experiments are presented in Table 2.

Table 2. Comparison of the times of solving of the six-dimensional bicriterial MCO problems

Nodes	P	Th	P*Th	Core type	Time, s	Speedup
1	1	16	16	CPU	7 186.4	1.0
16	16	16	256	CPU	957.3	7.5
16	32	4 032	129 024	GPU	529.9	13.6
16	64	2 016	129 024	GPU	291.8	24.6
16	128	1 008	129 024	GPU	272.4	26.4
32	128	2 016	258 048	GPU	214.9	33.4
32	256	1 008	258 048	GPU	253.2	28.4

In Table 2, the column "Nodes" shows the number of the supercomputer nodes employed, the column P shows the number of the parallel subproblems generated at the first level of the block reduction scheme (the parameter π_1 from (18)), the column Th shows the number of generated points of trials performed in parallel at the second decomposition level (the parameter π_2 from (18)).

The results of experiments presented in the first row show the averaged time of solving of the MCO problems with the use a single computational node of the supercomputer ($\pi_1 = 1$, $\pi_2 = 16$). In the second row of the table, the averaged time of solving of the MCO problems with the use of sixteen computational nodes of the supercomputer ($\pi_1 = 16$, $\pi_2 = 16$) is given. As follows from the presented results, the resulting speedup of the computations was 7.5 times. In the rows 3-7, the averaged times of solving the MCO problems with the use of 16 and 32 supercomputer nodes are given. At each node, 3 GPUs were employed. The number of parallel subproblems generated at the first level of the block reduction scheme (the parameter π_1 from (18)) ranged from 32 to 256 whereas the number of generated points of trials performed in parallel at the second decomposition level (the parameter π_2 from (18)) ranged from 1008 to 4032. Total number of employed GPU cores was 129024 with using 16 nodes, and 258048 with using 32 nodes. The maximal speedup of computations achieved was 33.4 times.

It is worth noting that the speedup of computations in the time depends on the time of computing the values of the criteria of the MCO problem being solved. This time is relatively small in the test optimization problems, but it can be essential when solving the applied problems in various scientific and

10   V. Gergel et al.

technical applications. As a result, along with the evaluation of the achieved speedup of computations in the time, it is reasonable to evaluate the speedup of computations with respect to the reduction of the number of iterations performed in the course of computations. The results of experiments performed to evaluate such speedup are presented in Table 3.

Table 3. Comparison of the number of iterations when solving the six-dimensional bicriterial MCO problems

Nodes	P	Th	P*Th	Core type	Iterations	Speedup
1	1	16	16	CPU	12 279 179.8	1.0
16	16	16	256	CPU	808 858.8	15.2
16	32	4 032	129 024	GPU	3 086.5	3 978.4
16	64	2 016	129 024	GPU	2 426.9	5 059.6
16	128	1 008	129 024	GPU	2 910.2	4 219.4
32	128	2 016	258 048	GPU	1 581.5	7 764.4
32	256	1 008	258 048	GPU	2307.5	5 321.4

The results of experiments presented above demonstrate the speedup of computations with respect to the reduction of the number of performed global search iterations to be considerable. Thus, when employing 16 computer nodes, the speedup can be more than 5000 times whereas when employing 32 computer nodes – more than 7700 times.

5   Conclusion

In the present paper, an efficient approach for solving the complex multicriterial optimization problems, in which the criteria of optimality can be the multiextremal ones, and the computing of the criteria values can require a large amount of computations has been proposed. The proposed approach is based on the reduction of the multicriterial problems to the scalar optimization ones by means of the minimax convolution of the partial criteria, the dimensionality reduction with the use of Peano space-filling curves, and the application of the efficient information-statistical methods of global optimization.

The opportunity of the large-scale parallel computations is provided by application of the block multistep dimensionality reduction scheme and by the use of the GPUs with many thousands computational cores. The results of numerical experiments have shown the developed approach to allow reducing the computational costs of solving the multicriterial optimization problems considerably – by hundreds and thousands times.

The results of the performed experiments have demonstrated the developed approach to be a promising one and to require further investigations. First of all, it is necessary to continue carrying out the numerical experiments on solving the multicriterial optimization problems at larger number of the partial criteria

GPU-based Parallel Computations in Multicriterial Optimization   11

of efficiency and for larger dimensionality of the optimization problems being solved.

Acknowledgements

References

1. Marler, R.T., Arora, J.S.: Multi-Objective Optimization: Concepts and Methods for Engineering, VDM Verlag (2009)
2. Ehrgott, M.: Multicriteria Optimization, Springer, 2005. (2nd ed., 2010)
3. Collette, Y., Siarry, P.: Multiobjective Optimization: Principles and Case Studies (Decision Engineering), Springer (2011)
4. Pardalos, P.M., Žilinskas, A., Žilinskas, J.: Non-Convex Multi-Objective Optimization, Springer (2017)
5. Hillermeier, C., Jahn, J.: Multiobjective optimization: survey of methods and industrial applications, Surv. Math. Ind. **11**, pp. 1–42 (2005)
6. Cho, J.-H., Wang, Ya., Chen, I.-R., Chan, K.S., Swami, A.: A Survey on Modeling and Optimizing Multi-Objective Systems, IEEE Communications Surveys & Tutorials **19**(3), pp. 1867–1901 (2017)
7. Eichfelder, G.: Scalarizations for adaptively solving multi-objective optimization problems, Comput. Optim. Appl. **44**, pp. 249–273 (2009)
8. Strongin, R., Sergeyev, Ya.: Global optimization with non-convex constraints. Sequential and parallel algorithms, Kluwer Academic Publishers, Dordrecht, 2000 (2nd ed. 2013, 3rd ed. 2014).
9. Strongin, R., Gergel, V., Grishagin, V., Barkalov, K.: Parallel computations for global optimization problems, Moscow State University Press (2013) (in Russian)
10. Sergeyev, Y.D., Strongin, R.G., Lera, D.: Introduction to global optimization exploiting space-filling curves, Springer (2013)
11. Gergel V.P., Kozinov, E.A.: Accelerating Parallel Multicriterial Optimization Methods Based on Intensive Using of Search Information, Procedia Computer Science, **108**, pp. 1463-1472 (2017)
12. Gergel, V., Kozinov, E.: Parallel computing for time-consuming multicriterial optimization problems, Lecture Notes in Computer Science, **10421**, pp. 446–458 (2017). DOI: 10.1007/978-3-319-62932-2_43
13. Gergel, V., Kozinov, E.: Efficient methods of multicriterial optimization based on the intensive use of search information, Springer Proceedings in Mathematics and Statistics, **197**, pp. 27–45 (2017). DOI: 10.1007/978-3-319-56829-4_3
14. Gergel, V., Kozinov, E.: An approach for parallel solving the multicriterial optimization problems with non-convex constraints, Communications in Computer and Information Science, **793**, pp. 121-135 (2017). DOI: 10.1007/978-3-319-71255-0_10
15. Cai, Y., See, S.: GPU Computing and Applications, Springer (2015)
16. Ferreiro, A.M., Garcia, J.A., Lopez-Salas, J.G., Vazquez, C.: An efficient implementation of parallel simulated annealing algorithm in GPUs, J. Glob. Optim., **57**(3), pp. 863–890 (2013)

12   V. Gergel et al.

17. Zhu, W.: Massively parallel differential evolution–pattern search optimization with graphics hardware acceleration: an investigation on bound constrained optimization problems, J. Glob. Optim. **50**(3), pp. 417–437 (2011)
18. Garcia-Martinez, J.M., Garzon, E.M., Ortigosa, P.M.: A GPU implementation of a hybrid evolutionary algorithm: GPuEGO, J. Supercomput (2014). DOI:10.1007/s11227-014-1136-7
19. Langdon, W.B.: Graphics processing units and genetic programming: an overview, Soft. Comput. **15**(8), pp. 1657–1669 (2011)
20. Locatelli, M., Schoen, F.: Global Optimization: Theory, Algorithms, and Applications, SIAM (2013)
21. Floudas, C.A., Pardalos, M.P.: Recent Advances in Global Optimization, Princeton University Press (2016)
22. Gergel, V.P., Strongin, R.G.: Parallel Computing for Globally Optimal Decision Making, Lecture Notes in Computer Science, **2763**, pp. 76–88 (2003)
23. Gergel, V.P., Kuzmin, M.I., Solovyov, N.A., Grishagin, V.A.: Recognition of surface defects of cold-rolling sheets based on method of localities, International Review of Automatic Control **8**(1), pp. 51–55 (2015)
24. Modorskii, V.Y., Gaynutdinova, D.F., Gergel, V.P., Barkalov, K.A.: Optimization in design of scientific products for purposes of cavitation problems, AIP Conference Proceedings, **1738**, pp. 400013 (2016). DOI: 10.1063/1.4952201
25. Grishagin, V., Israfilov, R., Sergeyev, Y.: Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes, Applied Mathematics and Computation, **318**, pp. 270–280 (2018). DOI: 10.1016/j.amc.2017.06.036
26. Sergeyev, Y., Grishagin, V.: Parallel asynchronous global search and the nested optimization scheme, J. Comput. Anal. Appl., **3**(2), pp. 123–145 (2001)
27. Barkalov, K.A., Gergel, V.P.: Multilevel scheme of dimensionality reduction for parallel global search algorithms, Proceedings of the 1st International Conference on Engineering and Applied Sciences Optimization, pp. 2111–2124 (2014)
28. Gergel, V., Lebedev, I.: Heterogeneous Parallel Computations for Solving Global Optimization Problems, Procedia Computer Science, **66**, pp. 53–62 (2015)
29. Gergel, V.P., Sidorov, S.V.: A Two-Level Parallel Global Search Algorithm for Solving Computationally Intensive Multi-extremal Optimization Problems, Lecture Notes in Computer Science, **9251**, pp. 505–515 (2015)
30. Gergel, V.: An Unified Approach to Use of Coprocessors of Various Types for Solving Global Optimization Problems, 2nd International Conference on Mathematics and Computers in Sciences and in Industry, MCSI, **7423935**, pp. 13–18 (2016)
31. Evtushenko, Yu.G., Posypkin, M.A.: A deterministic algorithm for global multi-objective optimization, Optimization Methods & Software, **29**(5), pp. 1005–1019 (2014)
32. Žilinskas, A., Žilinskas, J.: Adaptation of a one-step worst-case optimal univariate algorithm of bi-objective Lipschitz optimization to multidimensional problems, Commun Nonlinear Sci Numer Simulat, **21**, pp. 89–98 (2015)
33. 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, ACM Transactions on Mathematical Software **29**(4), pp. 469–480 (2003)