Comparison of dimensionality reduction schemes for derivative-free global optimization algorithms

Vladislav Sovrasov

Lobachevsky State University of Nizhni Novgorod, Nizhni Novgorod, Russia
sovrasov.vlad@gmail.com

Abstract
A common approach to solving global optimization problems is to use univariate optimization algorithms in combination with dimensional reduction schemes. The paper considers five types of Peano-like space-filling curves (evolvents), which are used to reduce the dimension in the derivative-free algorithm of global optimization. The algorithm is univariate and developed within the framework of the information-statistical approach. This work is the first one, where convergence rates and implementations details of these five evolvents are considered together and directly compared.

Keywords: Global optimization, Dimension reduction, Derivative-free algorithms, Global search algorithms

1 Introduction

In the present paper, the algorithms for solving the multiextremal optimization problems are considered. In the multiextremal problems, the opportunity of a reliable estimate of the global optimum is based principally on the availability of some information on the function known *a* priori allowing relating the probable values of the optimized function to the known values at the points of performed trials. Very often, such an information on the problem being solved is represented in the form of suggestion that the objective function $\varphi(y)$ satisfies Lipschitz condition with the constant L not known a priori (see, for example, [1, 2, 3]). At that, the objective function could be represented by a "black-box"-function. Many methods destined to solving the problems of the class specified above reduce the solving of a multidimensional problem to solving the one-dimensional subproblems implicitly (see, for example, the methods of diagonal partitions [4, 5] or simplicial partitions [6, 7]). In the present work, we will use the approach developed in Lobachevsky State University of Nizhni Novgorod based on the idea of the dimensionality reduction with the use of Peano space-filling curves $y(x)$ mapping the interval $[0, 1]$ of the real axis onto an n-dimensional cube continuously and unambiguously.

In recent years several methods of constructing Peano-type space-filling curves (*evolvents*) have been proposed [8, 9, 10, 11]. Some of them were successful used to build large-scale

parallel algorithms [12]. Presented paper numerically compares different evolvents in terms of convergence rates and computational efficiency. The comparison is done by solving sets consisting of hundreds of test problems and collecting different characteristics of convergence. As a result, practical recommendations about the usage of each evolvent type are given. For example, some evolvents are suitable for parallel computations, while others are quite useful in case of sequential solving of problems with time-consuming objective functions.

2 Statement of Multidimensional Global Optimization Problem

In this paper, the core class of optimization problems, which can be solved using univariate algorithm of global search [8], is formulated. This class involves the multidimensional global optimization problems without constraints, which can be defined in the following way:

$$\begin{gathered}\varphi(y^*) = \min\{\varphi(y) : y \in D\}, \\ D = \{y \in \mathbb{R}^N : a_i \le y_i \le b_i, 1 \le i \le N\}\end{gathered} \tag{1}$$

with the given boundary vectors a and b. It is supposed, that the objective function $\varphi(y)$ satisfies the Lipschitz condition

$$|\varphi(y_1) - \varphi(y_2)| \le L\|y_1 - y_2\|, y_1, y_2 \in D, \tag{2}$$

where $L > 0$ is the Lipschitz constant, and $||\cdot||$ denotes the norm in $\mathbb{R}^N$ space.

Usually, the objective function $\varphi(y)$ is defined as a computational procedure, according to which the value $\varphi(y)$ can be calculated for any vector $y \in D$ (let us further call such a calculation *a trial*). It is supposed that this procedure is time-consuming.

3 Methods of Dimension Reduction

3.1 Single evolvent

Within the framework of the information-statistical global optimization theory, the Peano space-filling curves (or evolvents) $y(x)$ mapping the interval $[0, 1]$ onto an N-dimensional hypercube D unambiguously are used for the dimensionality reduction [13], [8], [12], [14].

As a result of the reduction, the initial multidimensional global optimization problem (1) is reduced to the following one-dimensional problem:

$$\varphi(y(x^*)) = \min\{\varphi(y(x)) : x \in [0, 1]\}. \tag{3}$$

It is important to note that this dimensionality reduction scheme transforms the Lipschitzian function from (1) to the corresponding one-dimensional function $\varphi(y(x))$, which satisfies the uniform Hölder condition, i. e.

$$|\varphi(y(x_1)) - \varphi(y(x_2))| \le H|x_1 - x_2|^{\frac{1}{N}}, x_1, x_2 \in [0, 1], \tag{4}$$

where the constant H is defined by the relation $H = 2L\sqrt{N+3}$, L is the Lipschitz constant from (2), and N is the dimensionality of the optimization problem (1).

The algorithms for the numerical construction of the Peano curve approximations are given in [14].

The computational scheme obtained as a result of the dimensionality reduction consists of the following:

- The optimization algorithm performs the minimization of the reduced one-dimensional function $\varphi(y(x))$ from (3),
- After determining the next trial point x, a multidimensional image y is calculated by using the mapping $y(x)$,
- The value of the initial multidimensional function $\varphi(y)$ is calculated at the point $y \in D$,
- The calculated value $z = \varphi(y)$ is used further as the value of the reduced one-dimensional function $\varphi(y(x))$ at the point x.

3.2 Shifted evolvents

One of the possible ways to overcome the negative effects of using a numerical approximation of evolvent (it destroys the information about the neighbour points in $\mathbb{R}^N$ space, see [9]) consists in using the multiple mappings

$$Y_L(x) = \left\{y^0(x),\ y^1(x), ...,\ y^L(x)\right\} \tag{5}$$

instead of single Peano curve $y(x)$ (see [9, 14, 15]).

Such set of evolvents can be produced by shifting the source evolvent $y^0(x)$ by $2^{-l}, 0 \le l \le L$ on each coordinate. Each evolvent has it's own corresponding hypercude $D_l = \left\{y \in R^N : -2^{-1} \le y_i + 2^{-l} \le 3 \cdot 2^{-1},\ 1 \le i \le N\right\},\ 0 \le l \le L$.

In Fig. 1a the image of the interval $[0, 1]$ obtained by the curve $y^0(x),\ x \in [0, 1]$, is shown as the dashed line. Since the hypercube D from (1) is included in the common part of the family of hypercubes D_l, having introduced an additional constraint function

$$g_0(y) = \max\left\{|y_i| - 2^{-1} :\ 1 \le i \le N\right\}, \tag{6}$$

one can present the initial hypercube D in the form

$$D = \left\{y^l(x) :\ x \in [0, 1],\ g_0(y^l(x)) \le 0\right\},\ 0 \le l \le L,$$

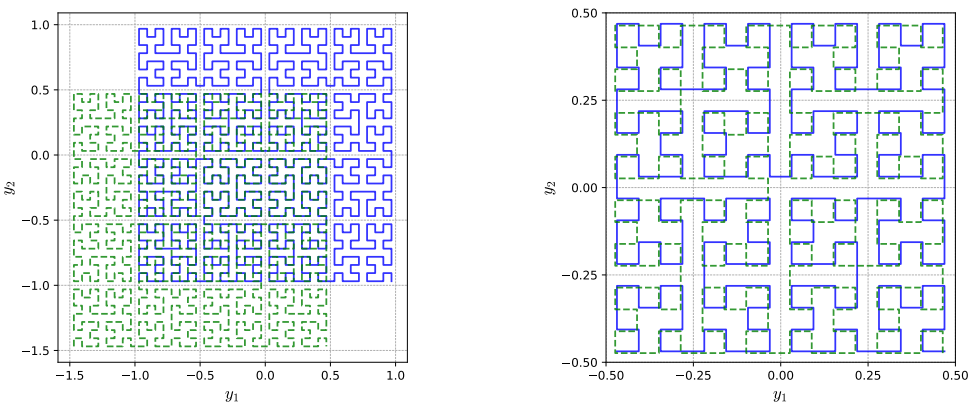
i.e., $g_0(y) \le 0$ if $y \in D$ and $g_0(y) > 0$ otherwise. Consequently, any point $y \in D$ has its own preimage $x^l \in [0, 1]$ for each mapping $y^l(x),\ 0 \le l \le L$.

Thus, each evolvent $y^l(x),\ 0 \le l \le L$, generates its own problem of the type (1) featured by its own extended (in comparison with D) search domain D_l and the additional constraint with the left hand part from (6)

$$\min\left\{\varphi(y^l(x)) : x \in [0, 1],\ g_j(y^l(x)) \le 0,\ 0 \le j \le m\right\},\ 0 \le l \le L. \tag{7}$$

3.3 Rotated evolvents

The application of the scheme for building the multiple evolvents (hereinafter called the shifted evolvents or S-evolvents) described in Subsection 3.2 allows to preserve the information on the nearness of the points in the multidimensional space and, therefore, to provide more precise (as



(a) Two shifted evolvents on the hypercubes D_0 and D_1 (b) Two rotated evolvents on the same plane

Figure 1: Multiple evolvents built with low density

compared to a single evolvent) estimate of Lipschitz constant in the search process. However, this approach has serious restrictions, which narrow the applicability of the parallel algorithms, designed on the base of the S-evolvents (see the end of the section 5.1).

To overcome complexity of the S-evolvent and to preserve the information on the nearness of the points in the N-dimensional space, one more scheme of building of the multiple mappings was proposed. The building of a set of Peano curves not by the shift along the main diagonal of the hypercube but by rotation of the evolvents around the coordinate origin is a distinctive feature of the proposed scheme [11]. In Fig. 1b two evolvents being the approximations to Peano curves for the case $N = 2$ are presented as an illustration. Taking into account the initial mapping, one can conclude that current implementation of the method allows to build up to $N(N-1)+1$ evolvents for mapping the N-dimensional domain onto the corresponding one-dimensional intervals. Moreover, the additional constraint $g_0(y) \leq 0$ with $g_0(y)$ from (6), which arises in shifted evolvents, is absent. This method for building a set of mappings can be "scaled" easily to obtain more evolvents (up to 2^N) if necessary.

3.4 Non-Univalent evolvent

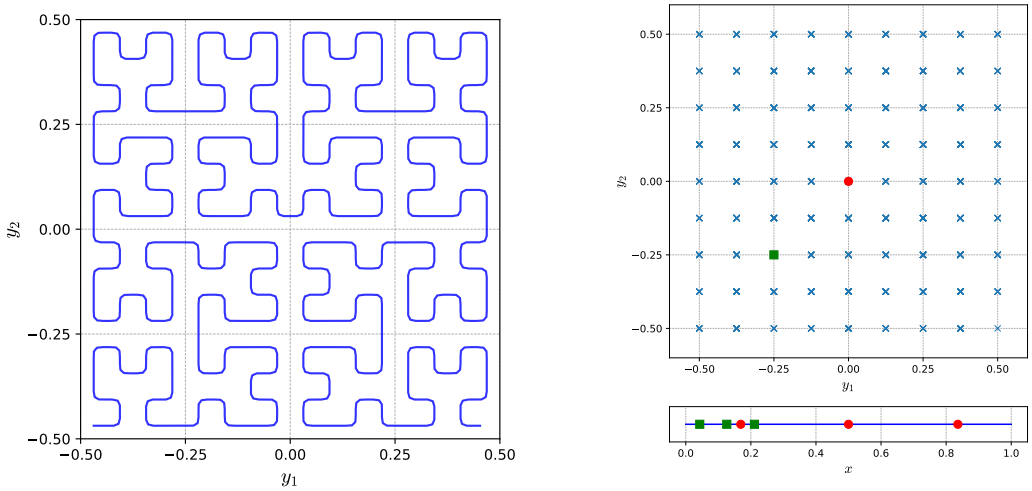
As it has been already mentioned above (Sec. 3.2), the loss of information on the proximity of the points in the multidimensional space could be compensated in part by the use of multiple mappings $Y_L(x) = \{y^1(x), ..., y^L(x)\}$. However, the Peano-type curve preserves a part of this information itself: it is not an injective mapping. Therefore, if a single image $y(x) \in \mathbb{R}^N$ is available, on can obtain several different preimages $t_j \in [0,1], t_j \neq x$, which could be added into the search information of the method later.

The Peano-type curve used in (3) for the dimensionality reduction is defined via the transition to the limit. Therefore, it cannot be computed directly. In the numerical optimization, some approximation of this curve is used, and it is an injective piecewise-linear curve. In [8] a non-univalent mapping of a uniform grid in the interval $[0,1]$ onto a uniform grid in a hypercube D has been proposed. Each multidimensional node can have up to 2^N one-dimensional preimages. In Fig. 2b, the grid in the $\mathbb{R}^2$ space is marked by the crosses, for two nodes of which the corresponding one-dimensional preimages from $[0,1]$ are pointed (marked by the squares and circles). Each node mentioned above has 3 preimages.

A potentially large number of preimages (up to 2^N) and the inability to use the parallel scheme for the multiple mappings are the disadvantages of the non-univalent evolvent.

3.5 Smooth evolvent

The methods of constructing the evolvents considered in the previous paragraphs produce the curve $y(x)$, which in not a smooth one (see Fig. 1a). The absence of smoothness may affect the properties of the reduced one-dimensional function $\varphi(y(x))$ adversely since a smooth curve reflect the information on the growth/decay of the initial function better. On the basis of initial algorithm of constructing the non-smooth evolvent, a generalized algorithm allowing constructing a smooth space-filling curve has been proposed [10]. As an illustration, a smooth evolvent for the two-dimensional case is presented in Fig. 2a. An increased computational complexity (several times as compared to the piecewise-linear curves) is a disadvantage of the smooth evolvent. This caused by computing of the nonlinear smooth functions.



(a) Smooth evolvent

(b) Non-univalent evolvent

Figure 2: Different evolvents built with low density

4 Core Multidimensional Algorithm of Global Search

The optimization methods applied in Globalizer to solve the reduced problem (3) are based on the MAGS method, which can be presented as follows — see [8], [14].

The initial iteration of the algorithm is performed at an arbitrary point $x^1 \in (0, 1)$. Then, let us suppose that k, $k \geq 1$, optimization iterations have been completed already. The selection of the trial point x^{k+1} for the next iteration is performed according to the following rules.

Rule 1. Renumber the points of the preceding trials by the lower indices in order of increasing value of coordinates $0 = x_0 < x_1 < ... < x_{k+1} = 1$.

Rule 2. Compute the characteristics $R(i)$ for each interval $(x_{i-1}, x_i), 1 \leq i \leq k + 1$.

Rule 4. Determine the interval with the maximum characteristic $R(t) = \max_{1 \leq i \leq k+1} R(i)$.

Rule 5. Execute a new trial at the point x^{k+1} located within the interval with the maximum characteristic from the previous step $x^{k+1} = d(x_t)$.

The stopping condition, which terminated the trials, is defined by the inequality $\rho_t < \varepsilon$ for the interval with the maximum characteristic from Step 4 and $\varepsilon > 0$ is the predefined accuracy of the optimization problem solution. If the stopping condition is not satisfied, the index k is incremented by 1, and the new global optimization iteration is executed.

The convergence conditions and exact formulas for decision rules $R(i)$ and $d(x)$ of the described algorithm are given, for example, in [14].

5 Results of Numerical Experiments

The computational experiments have been carried out on the Lobachevsky supercomputer at State University of Nizhni Novgorod. A computational node included 2 Intel Sandy Bridge E5-2660 2.2 GHz processors, 64 GB RAM. The CPUs had 8 cores (i. e. total 16 cores were available per a node). All considered algorithms and evolvents were implemented using C++ within the Globalizer software system [16].

The comparison of the global optimization algorithms was performed by the evaluation of the quality of solving a set of problems from some test class. In the present paper, the test class generated by GKLS generator [17] was considered. This generator allows constructing the complex multiextremal problems of various dimensions. In the present work, the series of 100 problems from the classes of the dimensions of 2, 3, 4, and 5 were considered. Each class had two degrees of complexity — *Simple* and *Hard*. The parameters of the generator for the considered classes were given in Ref. [17].

In order to evaluate the efficiency of an algorithm on a given set of 100 problems, we will use the operating characteristics [18], which are defined as a curve, showing the dependency of number of solved problems vs the number of iterations.

5.1 Comparison of the sequential evolvents

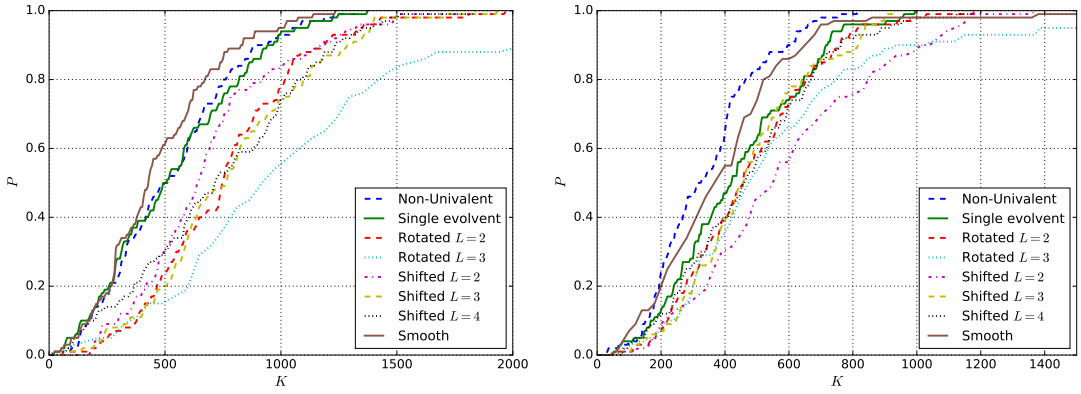
In order to understand whether any type of evolvents listed above has an essential advantage as compared to other ones, the operating characteristics of the index method with different types of evolvents have been obtained for the classes GKLS 2d Simple and GKLS 3d Simple. The global minimum was considered to be found if the algorithm generates a trial point y^k in the δ-vicinity of the global minimizer, i.e. $\left\|y^k - y^*\right\|_\infty \leq \delta$. The size of the vicinity was selected as $\delta = 0.01 \left\|b - a\right\|_\infty$. In case of GKLS $\delta = 0.01$.

In all experiments, the evolvent construction density parameter $m = 12$. The minimum value of the reliability parameter r was found for each type of evolvents by scanning over a uniform grid with the step 0.1.

On the GKLS 2d Simple class at the minimum r, the non-univalent evolvent and the smooth one provide a faster convergence (Fig. 3b). The same was observed at $r = 5.0$ as well (Fig. 3a). In the latter case, the shifted evolvent and the rotating one begin to lag behind the rest since the value $r = 5.0$ is too big for them.

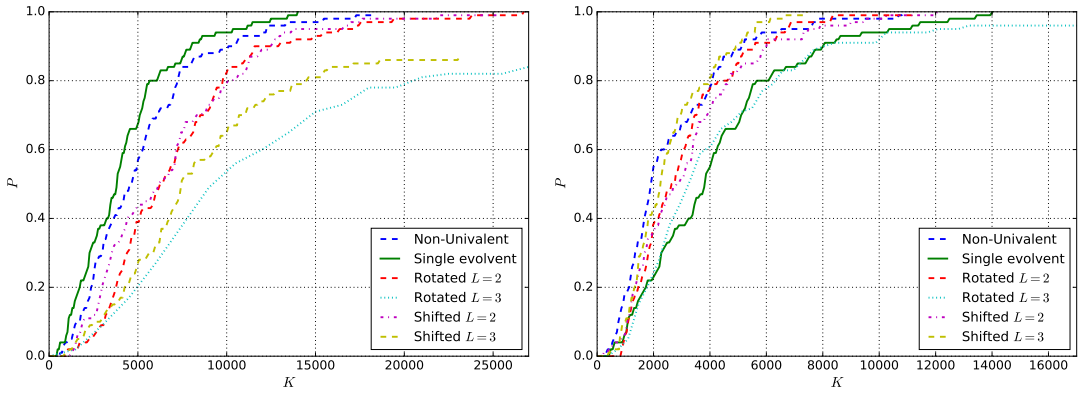
On the GKLS 2d Simple class at the minimum r, the non-univalent evolvents and multiple ones have a considerable advantage over the single evolvent (Fig. 4b). The value $r = 4.5$ is too big for the rotated evolvents and for the shifted one (Fig. 4a).

Overhead costs when using the shifted evolvents. In all experiments presented above, the number of computations of the objective function from the GKLS class was taken into



(a) $r = 5.0$ (b) Minimal r

Figure 3: Operating characteristics on GKLS 2d Simple class



(a) $r = 4.5$ (b) Minimal r

Figure 4: Operating characteristics on GKLS 3d Simple class

account when plotting the operating characteristics. However, in the case of the shifted evolvent, the index method solves the problem with the constraint g_0 from (6). At the points where g_0 is violated, the value of the objective function is not computed. Nevertheless, these points are stored in the search information producing the additional computational costs. In Table 1, the averaged numbers of calls to g_0 and to the objective function are presented. At $L = 3$, the constraint g_0 was computed almost 20 times more than the objective function φ i. e. 95% of the whole search information account for the auxiliary points. Such overhead costs are acceptable when solving the problems of small dimension with the computation costly objective functions. However, when increasing dimensionality and total number of trials other types of evolvents are preferred.

Table 1: Averaged number of computations of g_0 and of φ when solving the problems from GKLS 3d Simple class using the shifted evolvent

L	$calc(g_0)$	$calc(\varphi)$	$\frac{calc(g_0)}{calc(\varphi)}$ ratio
2	96247.9	6840.14	14.07
3	153131.0	7702.82	19.88

6 Conclusions

In the present work, 5 different Peano curve-type mappings applied to the dimensionality reduction in the global optimization problems were considered. From the preliminary comparison conducted in Sec. 5.1, one can make the following conclusions:

- the smooth evolvent and the non-univalent one demonstrate the best result in the problems of small dimensionality and can be applied successfully in solving the problems with the computational costly objective functions. The properties of these evolvents doesn't allow developing the optimization algorithms scalable onto several cluster nodes based on these ones.

- the shifted evolvents introduce large overhead costs on the operation of the method due to the requirement to adding an auxiliary functional constraint into the problem (1). The experiments have demonstrated that up to 95% of the search information account for the points, in which the auxiliary constraint is computed only. However, if the objective function is computation-costly enough, the use of these evolvents could make sense.

- the rotated evolvents have provided an acceptable speed of convergence in the problems of small dimensionality in the sequential mode. The use of these ones doesn't result in the introduction of the auxiliary constraints that allows constructing an efficient parallel algorithm based on these evolvents.

References

[1] Jones, D.R. The direct global optimization algorithm. In *The Encyclopedia of Optimization*, pages 725–735. Springer, Heidelberg, 2009.

[2] Gablonsky, J.M., Kelley, C.T. A locally-biased form of the direct algorithm. *J. Glob. Optim.*, 21(1):27–37, 2001.

[3] Evtushenko, Y., Posypkin, M. A deterministic approach to global box-constrained optimization. *Optim. Lett.*, 7:819–829, 2013.

[4] Sergeyev, Ya.D., Kvasov, D.E. Global search based on efficient diagonal partitions and a set of lipschitz constants. *SIAM J. Optim.*, 16(3):910–937, 2006.

[5] Sergeyev, Ya.D., Kvasov, D.E. A deterministic global optimization using smooth diagonal auxiliary functions. *Communications in Nonlinear Science and Numerical Simulation*, 21(1-3):99–111, 2015.

[6] Žilinskas, J. Branch and bound with simplicial partitions for global optimization. *Math. Model. Anal.*, 13(1):145–159, 2008.

[7] Paulavičius, R., Žilinskas, J. Simplicial lipschitz optimization without the lipschitz constant. *J. Glob. Optim.*, 59(1):23–40, 2014.

[8] R.G. Strongin. *Numerical Methods in Multiextremal Problems (Information-StatisticalAlgorithms)*. Moscow: Nauka (In Russian), 1978.

[9] R. G. Strongin. Algorithms for multi-extremal mathematical programming problems employing a set of joint space-filling curves. *J. Glob. Optim.*, 2:357–378, 1992.

[10] Goryachih, A. A class of smooth modification of space-filling curves for global optimization problems. In *Models, Algorithms, and Technologies for Network Analysis. NET 2016*, pages 152–159. Springer, Cham, 2017.

[11] Strongin, R.G., Gergel, V.P., Barkalov, K.A. Parallel methods for global optimization problem solving. *Journal of instrument engineering*, 52:25–33, 2009 (In Russian).

[12] Strongin, R.G., Gergel, V.P., Grishagin, V.A., Barkalov, K.A. *Parallel Computations for Global Optimization Problems.* Moscow University Press, Moscow (in Russian), 2013.

[13] Sergeyev, Y.D., Strongin, R.G., Lera, D. *Introduction to Global Optimization Exploiting Space-Filling Curves.* Springer Briefs in Optimization, Springer, New York, 2013.

[14] Strongin R.G., Sergeyev Ya.D. *Global optimization with non-convex constraints. Sequential and parallel algorithms.* Kluwer Academic Publishers, Dordrecht, 2000.

[15] R. G. Strongin. Parallel multi-extremal optimization using a set of evolvents. *Comp. Math. Math. Phys.*, 31(8):37–46, 1991.

[16] Gergel V.P., Barkalov K.A., and Sysoyev A.V. A novel supercomputer software system for solving time-consuming global optimization problems. *Numerical Algebra, Control & Optimization*, 8(1):47–62, 2018.

[17] Gaviano, M., Kvasov, D.E., Lera, D., Sergeev, 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):469–480, 2003.

[18] V.A. Grishagin. Operating characteristics of some global search algorithms. *Problems of Statistical Optimization*, 7:198–206, 1978 (In Russian).