

# Globalizer – A Parallel Software System for Solving Global Optimization Problems

Alexander Sysoyev<sup>(✉)</sup>, Konstantin Barkalov, Vladislav Sovrasov,  
Ilya Lebedev, and Victor Gergel

Lobachevsky State University of Nizhni Novgorod, Nizhny Novgorod, Russia  
alexander.sysoyev@itmm.unn.ru

**Abstract.** In this paper, we describe the Globalizer software system for solving global optimization problems. The system implements an approach to solving the global optimization problems using the block multistage scheme of the dimension reduction, which combines the use of Peano curve type evolvents and the multistage reduction scheme. The scheme allows an efficient parallelization of the computations and increasing the number of processors employed in the parallel solving of the global optimization problems many times.

**Keywords:** Multidimensional multiextremal optimization · Global search algorithms · Parallel computations · Dimension reduction · Block multistage dimension reduction scheme

## 1 Introduction

The development of optimization methods that use high-performance computing systems to solve time-consuming global optimization problems is an area receiving extensive attention. The theoretical results obtained provide efficient solutions to many applied global optimization problems in various fields of scientific and technological applications. At the same time, the practical software implementation of these algorithms for multiextremal optimization is quite limited. Among the software for the global optimization, one can select the following systems: LGO (Lipschitz Global Optimization) [14], GlobSol [11], LINDO [12], IOSO (Indirect Optimization on the basis of Self-Organization) [3], MATLAB Global Optimization Toolkit [23], TOMLAB system [10], BARON (Branch-And-Reduce Optimization Navigator) [15], GAMS (General Algebraic Modeling System) [2], Global Optimization Library in R [13].

In this paper, a novel Globalizer software system is considered. The development of the system was conducted based on the information-statistical theory of multiextremal optimization aimed at developing efficient parallel algorithms for global search – see, for example, [21, 22]. The Globalizer advantage is that the system is designed to solve time-consuming multiextremal optimization problems. In order to obtain global optimized solutions within a reasonable time and cost, the system efficiently uses modern high-performance computer systems.

## 2 Statement of Multidimensional Global Optimization Problem

In this paper, the core class of optimization problems which can be solved by using Globalizer is examined. This involves multidimensional global optimization problems without constraints, which can be defined in the following way:

$$\varphi(y) \rightarrow \inf, y \in D \subset \mathbb{R}^N, \quad (1)$$

$$D = \{y \in \mathbb{R}^N: a_i \leq y_i \leq b_i, 1 \leq i \leq N\}. \quad (2)$$

If  $y^*$  is an exact solution of problem (1)–(2), the numerical solution of the problem is reduced to building an estimate  $y^0$  of the exact solution matching to some notion of nearness to a point (for example,  $\|y^* - y^0\| \leq \varepsilon$  where  $\varepsilon > 0$  is a predefined accuracy) based on a finite number  $k$  of computations of the optimized function values.

Regarding to the class of problems considered, the fulfillment of the following important conditions is supposed:

1. The optimized function  $\varphi(y)$  can be defined by some algorithm for the computation of its values at the points of the domain  $D$ .
2. The computation of the function value at every point is a computation-costly operation.
3. Function  $\varphi(y)$  satisfy the Lipschitz condition.

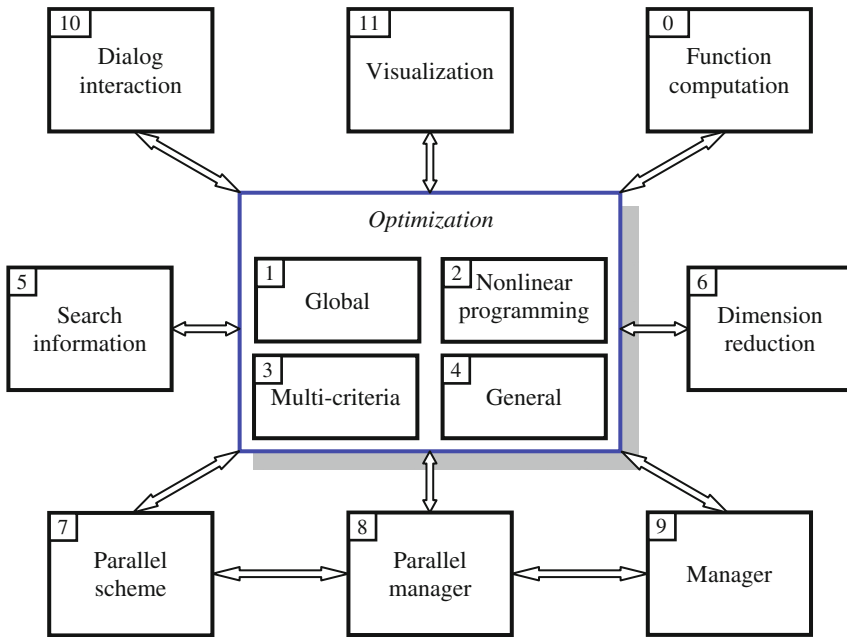
## 3 Globalizer Architecture

The Globalizer expands the family of global optimization software systems successively developed by the authors during the past several years [1, 5].

The Globalizer architecture is presented in Fig. 1. The structural components of the systems are:

- Block 0 is an external block. It consists of the procedures for computing the function values (criteria and constraints) for the optimization problem being solved.
- Blocks 1–4 form the optimization subsystem for solving the global optimization problems (Block 1), nonlinear programming (Block 2), multicriterial optimization (Block 3), and general decision making problems (Block 4).
- Block 5 is a subsystem for accumulating and processing the search information.
- Block 6 contains the dimensional reduction procedures based on the Peano evolvents; this block also provides interaction between the optimization blocks and the initial multidimensional optimization problem.
- Block 7 organizes the choice of parallel computation schemes in the Globalizer system subject to the computing system architecture employed (the numbers of cores in the processors, the availability of shared and distributed memory, the availability of accelerators for computations, etc.) and the global optimization methods applied.

- Block 8 is responsible for managing the parallel processes when performing the global search (determining the optimal configuration of parallel processes, distributing the processes between computing elements, etc.).
- Block 9 is a management subsystem, which controls the computational process when solving global optimization problems.
- Block 10 is responsible for organizing the dialog interaction with users for stating the optimization problem, adjusting system parameters (if necessary), and visualizing and presenting the global search results.
- Block 11 is a set of tools for visualizing and presenting the global search results; the availability of tools for visually presenting the computational results enables the user to provide efficient control over the global optimization process.



**Fig. 1.** Program architecture of Globalizer system (Blocks 1–2, 5–7 have been implemented; Blocks 3–4 and 8–11 are under development)

## 4 Globalizer Approach for Solving the Global Optimization Problems

### 4.1 Methods of Dimension Reduction

The Globalizer implements a block multistage scheme of dimension reduction [1], which reduces the solving of initial multidimensional optimization problem (1)–(2) to the solving of a sequence of «nested» problems of less dimensionality.

Thus, initial vector  $y$  is represented as a vector of the «aggregated» macro-variables

$$y = (y_1, y_2, \dots, y_N) = (u_1, u_2, \dots, u_M) \tag{3}$$

where the  $i$ -th macro-variable  $u_i$  is a vector of the dimensionality  $N_i$  from the components of the vector  $y$  taken sequentially i.e.

$$\begin{aligned} u_1 &= (y_1, y_2, \dots, y_{N_1}), \\ u_2 &= (y_{N_1+1}, y_{N_1+2}, \dots, y_{N_1+N_2}), \dots \\ u_i &= (y_{p+1}, \dots, y_{p+N_i}), \dots \end{aligned} \tag{4}$$

where  $p = \sum_{k=1}^{i-1} N_k$  and  $\sum_{k=1}^M N_k = N$ .

Using the macro-variables, the main relation of the well-known multistage scheme can be rewritten in the form

$$\min_{y \in D} \varphi(y) = \min_{u_1 \in D_1} \min_{u_2 \in D_2} \dots \min_{u_M \in D_M} \varphi(y), \tag{5}$$

where the subdomains  $D_i, 1 \leq i \leq M$ , are the projections of the initial search domain  $D$  onto the subspaces corresponding to the macro-variables  $u_i, 1 \leq i \leq M$ .

It should be pointed that the nested subproblems

$$\varphi_i(u_1, \dots, u_i) = \min_{u_{i+1} \in D_{i+1}} \varphi_{i+1}(u_1, \dots, u_i, u_{i+1}), \quad 1 \leq i \leq M, \tag{6}$$

are the multidimensional ones in the block multistage scheme and this is the key difference from the initial scheme. Thus, this approach can be combined with the reduction of the domain  $D$  (for example, with the evolvent based on Peano curve) for the possibility to use the efficient methods of solving the one-dimensional problems of the multiextremal optimization [20].

The Peano curve  $y(x)$  lets map the interval of the real axis  $[0, 1]$  onto the domain  $D$  uniquely:

$$\{y \in D \subset \mathbb{R}^N\} = \{y(x): 0 \leq x \leq 1\}. \tag{7}$$

The evolvent is the approximation to the Peano curve with the accuracy of the order  $2^{-m}$  where  $m$  is the density of the evolvent.

Application the mappings of this kind allows reducing multidimensional problem (1)–(2) to a one-dimensional one

$$\varphi(y^*) = \varphi(y(x^*)) = \min\{\varphi(y(x)): x \in [0, 1]\}. \tag{8}$$

#### 4.2 Method for Solving the Reduced Global Optimization Problems

The information-statistical theory of global search formulated in [22] has served as a basis for the development of a large number of efficient multiextremal optimization methods – see, for example, [6–9, 16–19], etc. Within the framework of information-statistical theory, a general approach to parallelization computations when solving global optimization problems has been proposed – the parallelism of computations is

provided by means of simultaneously computing the values of the minimized function  $\varphi(y)$  at several different points within the search domain.

The general computation scheme of Parallel Multidimensional Algorithm of Global Search (PMAGS) that is implemented in Globalizer is fully described in [1].

### 4.3 Implementation of Parallel Algorithm of Global Optimization

Let us consider a parallel implementation of the block multistage dimension reduction scheme described in Subsect. 4.1.

For the description of the parallelism in the multistage scheme, let us introduce a vector of parallelization degrees

$$\pi = (\pi_1, \pi_2, \dots, \pi_M), \quad (9)$$

where  $\pi_i$ ,  $1 \leq i \leq M$ , is the number of the subproblems of the  $(i + 1)$ -th nesting level being solved in parallel, arising as a result of execution of the parallel iterations at the  $i$ -th level. For the macro-variable  $u_i$ , the number  $\pi_i$  means the number of parallel trials in the course of minimization of the function

$$\varphi_M(u_1, \dots, u_M) = \varphi(y_1, \dots, y_N)$$

with respect to  $u_i$  at fixed values of  $u_1, u_2, \dots, u_{i-1}$ , i.e. the number of the values of the objective function  $\varphi(y)$  computed in parallel.

In the general case, the quantities  $\pi_i$ ,  $1 \leq i \leq M$  can depend on various parameters and can vary in the course of optimization, but we will limit ourselves to the case when all components of the vector  $\pi$  are constant.

At every nested multistage dimension reduction level PMAGS is used. Let us remind that the parallelization is implemented by selection not a single point for the next trial (as in the serial version) but  $p$  points, which are placed into  $p$  intervals with the highest characteristics. Therefore, if  $p$  processors are available,  $p$  trials can be executed in these points in parallel. As a result, the solving of the problem at the  $i$ -th level generates  $p$  subproblems for the  $(i + 1)$ -th level.

## 5 Numerical Results

The computational experiments were conducted using the Lobachevsky supercomputer at the State University of Nizhny Novgorod (<http://hpc-education.unn.ru/en/resources>). The problems generated by the GKLS-generator [4] were selected for the test problems.

The results of the numerical experiments with Globalizer on an Intel Xeon Phi are provided in Table 1. The computations were performed using the Simple and Hard function classes with the dimensions equal to 4 and 5.

**Table 1.** Average number of iterations

		$p$	N = 4		N = 5	
			Simple	Hard	Simple	Hard
I	<b>Serial computations</b> Average number of iterations	1	11953	25263	15920	>148342(4)
II	<b>Parallel computations on CPU</b> Speedup	2	2.51	2.26	1.19	1.36
		4	5.04	4.23	3.06	2.86
		8	8.58	8.79	4.22	6.56
III	<b>Parallel computations on Xeon Phi</b> Speedup	60	8.13	7.32	9.87	6.55
		120	16.33	15.82	15.15	17.31
		240	33.07	27.79	38.80	59.31

In the first series of experiments, serial computations using MAGS were executed. The average number of iterations performed by the method for solving a series of problems for each of these classes is shown in row I. The symbol “>” reflects the situation where not all problems of a given class were solved by a given method. It means that the algorithm was stopped once the maximum allowable number of iterations  $K_{\max}$  was achieved. In this case, the  $K_{\max}$  value was used for calculating the average number of iterations corresponding to the lower estimate of this average value. The number of unsolved problems is specified in brackets.

In the second series of experiments, parallel computations were executed on a CPU. The relative “speedup” in iterations achieved is shown in row II; the speedup of parallel computations was measured in relation to the serial computations ( $p = 1$ ).

The final series of experiments was executed using a Xeon Phi. The results of these computations are shown in row III; in this case, the speedup factor is calculated in relation to the PMAGS results on a CPU using eight cores ( $p = 8$ ).

## 6 Conclusion

In this paper, the Globalizer global optimization software system was presented for implementing a general scheme for the parallel solution of globally optimized decision making. The work is devoted to the investigation of the possibility to speedup the process of searching the global optimum when solving the multidimensional multiextremal optimization problems using the approach based on the application of the parallel block multistage scheme of the dimension reduction.

This research was supported by the Russian Science Foundation, project No 16-11-10150 “Novel efficient methods and software tools for the time consuming decision making problems with 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. Bussieck, M.R., Meeraus, A.: General algebraic modeling system (GAMS). In: Kallrath, J. (ed.) Modeling Languages in Mathematical Optimization, pp. 137–157. Springer, Boston (2004). doi:[10.1007/978-1-4613-0215-5\\_8](https://doi.org/10.1007/978-1-4613-0215-5_8)
3. Egorov, I.N., Kretinin, G.V., Leshchenko, I.A., Kuptzov, S.V.: IOSO optimization toolkit - novel software to create better design. In: 9th AIAA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Atlanta, Georgia (2002). [http://www.iosotech.com/text/2002\\_4329.pdf](http://www.iosotech.com/text/2002_4329.pdf)
4. Gaviano, M., Lera, D., Kvasov, D.E., Sergeyev, Y.D.: Software for generation of classes of test functions with known local and global minima for global optimization. ACM Trans. Math. Software **29**, 469–480 (2003)
5. Gergel, V.P.: A software system for multiextremal optimization. Eur. J. Oper. Res. **65**(3), 305–313 (1993)
6. Gergel, V.P.: A method of using derivatives in the minimization of multiextremum functions. Comput. Math. Math. Phys. **36**(6), 729–742 (1996)
7. Gergel, V.P.: A global optimization algorithm for multivariate function with Lipschitzian first derivatives. J. Glob. Optim. **10**(3), 257–281 (1997)
8. Gergel, V., Lebedev, I.: Heterogeneous parallel computations for solving global optimization problems. Procedia Comput. Sci. **66**, 53–62 (2015)
9. Gergel, V.P., Strongin, R.G.: Parallel computing for globally optimal decision making. In: Malyshekin, V.E. (ed.) PaCT 2003. LNCS, vol. 2763, pp. 76–88. Springer, Heidelberg (2003). doi:[10.1007/978-3-540-45145-7\\_7](https://doi.org/10.1007/978-3-540-45145-7_7)
10. Holmström, K., Edvall, M.M.: The TOMLAB optimization environment. In: Kallrath, J. (ed.) Modeling Languages in Mathematical Optimization. Applied Optimization, vol. 88, pp. 369–376. Springer, Boston (2004). doi:[10.1007/978-1-4613-0215-5\\_19](https://doi.org/10.1007/978-1-4613-0215-5_19)
11. Kearfott, R.B.: GlobSol user guide. Optim. Methods Softw. **24**, 687–708 (2009)
12. Lin, Y., Schrage, L.: The global solver in the LINDO API. Optim. Methods Softw. **24**, 657–668 (2009)
13. Mullen, K.M.: Continuous global optimization in R. J. Stat. Softw. **60**(6) (2014)
14. Pintér, J.D.: Global Optimization in Action: Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications. Springer, New York (1996). doi:[10.1007/978-1-4757-2502-5](https://doi.org/10.1007/978-1-4757-2502-5)
15. Sahinidis, N.V.: BARON: a general purpose global optimization software package. J. Glob. Optim. **8**(2), 201–205 (1996)
16. Sergeyev, Y.D.: An information global optimization algorithm with local tuning. SIAM J. Optim. **5**(4), 858–870 (1995)
17. Sergeyev, Y.D.: Multidimensional global optimization using the first derivatives. Comput. Math. Math. Phys. **39**(5), 743–752 (1999)
18. Sergeyev, Y.D., Grishagin, V.A.: A parallel method for finding the global minimum of univariate functions. J. Optim. Theor. Appl. **80**(3), 513–536 (1994)
19. Sergeyev, Y.D., Grishagin, V.A.: Parallel asynchronous global search and the nested optimization scheme. J. Comput. Anal. Appl. **3**(2), 123–145 (2001)
20. Sergeyev, Y., Strongin, R.G., Lera, D.: Introduction to Global Optimization Exploiting Space-Filling Curves. Springer, New York (2013). doi:[10.1007/978-1-4614-8042-6](https://doi.org/10.1007/978-1-4614-8042-6)

21. Strongin, R.G., Gergel, V.P., Grishagin, V.A., Barkalov, K.A.: Parallel Computations for Global Optimization Problems. Moscow State University, Moscow (2013). (In Russian)
22. Strongin, R.G., Sergeyev, Y.D.: Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms. Springer, New York (2000). doi: [10.1007/978-1-4615-4677-1](https://doi.org/10.1007/978-1-4615-4677-1)
23. Venkataraman, P.: Applied Optimization with MATLAB Programming. Wiley, Chichester (2009)