To appear in *Optimization Methods & Software*
Vol. 00, No. 00, Month 20XX, 1–7

GLOBALIZER: A NOVEL SUPERCOMPUTER SOFTWARE SYSTEM FOR SOLVING TIME-CONSUMING GLOBAL OPTIMIZATION PROBLEMS

V. P. Gergel[a*] and K. A. Barkalov[b] and A. V. Sysoev

[a]*Taylor & Francis, 4 Park Square, Milton Park, Abingdon, UK;* [b]*Institut für Informatik, Albert-Ludwigs-Universität, Freiburg, Germany*

(*v5.0 released July 2015*)

In this paper, we describe the Globalizer software system for solving global optimization problems. The system is designed to maximize the use of computational potential by modern high performance computer systems in order to solve the most time-consuming optimization problems. Highly parallel computations are facilitated using various distinctive computational schemes: processing several optimization iterations simultaneously, reducing multidimensional optimization problems using multiple Peano curve evolvents, multi-stage computing based on nested block reduction schemes. This novelty leverages supercomputer system capabilities with shared and distributed memory and with large numbers of processors to efficiently solve global optimization problems.

Keywords: global optimization, information-statistical theory, parallel computing, high-performance computer systems, supercomputer technologies

1. Introduction

Global (or multiextremal) optimization problems are among the most complex problems in both theory and practice for optimal decision making. In these kinds of problems, the criterion to be optimized has several local optima within the search domain, which have different values. The existence of several local optima essentially makes finding the global optimum difficult, since it requires examining the whole feasible search domain. The volume of computations for solving global optimization problems can increase exponentially with increasing number of varied parameters.

In addition, global optimization problem statements are used, as a rule, in the most complex decision making situations, for example, when a computer-aided design of complex technologies, products, and systems is conducted. In such problems, the efficiency criteria are nonlinear, the search domains may be non-contiguous, and, most importantly, the computational complexity of the functions which the optimized criteria and constraints are based on can be quite essential.

These global optimization problem features impose special requirements on the quality of the optimization methods and on the software to implement them. The global optimization methods should be highly efficient, and the software systems should be developed on a good professional basis. In general, global optimization problems can

*Corresponding author. Email: latex.helpdesk@tandf.co.uk

be solved using modern supercomputer systems at a reasonable time and cost by only employing parallel global optimization algorithms.

The general state of the art in the field of global optimization has been presented in a number of key monographs (Trn, A., ilinskas, A., 1989), (Horst, R., Tuy, H., 1990), (Zhigljavsky, A. A., 1991), (Pintr, J. D., 1996), (Strongin, R. G., Sergeyev, Y. D., 2000), (Locatelli, M., Schoen, F., 2013), (Floudas, C. A., Pardalos, M. P., 2016), etc. The development of optimization methods that use high-performance computer systems to solve time-consuming global optimization problems is an area receiving extensive attention see, for example, (Censor, Y., Zenios, S. A., 1998), (Strongin, R. G., Sergeyev, Y. D., 2000), (Ciegis, R., Henty, D., Kgstrm, B., Zilinskas, J., 2009), (Luque, G., Alba, E., 2011), (Strongin, R. G., K. A., Gergel, V. P., Grishagin, V. A., Barkalov, K. A., 2013).

The theoretical results obtained provide efficient solutions to many applied global optimization problems in various fields of scientific and technological applications see, for example, (Floudas, C. A., Pardalos, M. P., 1996), (Pintr, J. D., 1996), (Luque, G., Alba, E., 2011), (Fasano, G., Pintr, J. D., (2013), (Locatelli, M., Schoen, F., 2013), (Pardalos, M. P., Zhigljavsky, A. A., Zilinskas, J., 2016), etc.

At the same time, the practical implementation of these algorithms for multiextremal optimization within the framework of industrial software systems is quite limited. In many cases, software implementations of global optimization algorithms are experimental in nature and are just used by the developers themselves to obtain results from computational experiments required for scientific publication. To a large extent, this situation originates from high development costs for professional software systems that can be used by numerous users. In addition, global optimization problems could rarely be solved in an automatic mode because of their complexity. As a rule, the user should actively control the global search process which implies an adequate level of qualification in the field of optimization (particularly, the user should know and understand global optimization methods well).

On the market for global optimization software, one can select from the following systems:

- LGO (Lipschitz Global Optimization) is designed to solve global optimization problems for which the criteria and constraints satisfy the Lipschitz condition (see, for example, (Pintr, J. D., 1996)). The system is a commercial product based on diagonal extensions of one-dimensional multiextremal optimization algorithms.
- GlobSol (Kearfott, R. B., 2009) is oriented towards solving global optimization problems as well as systems of nonlinear equations. The system includes interval methods based on the branch and bound method. There are some extensions of the system for parallel computations, and it is available to use for free.
- LINDO (Lin, Y., Schrage.L., 2009) is features by a wide spectrum of problem solving mehtods which it can be used for these include linear, integer, stochastic, nonlinear, and global optimization problems. The ability to interact with the Microsoft Excel software environment is a key feature of the system. The system is widely used in practical applications and is available to use for free.
- IOSO (Indirect Optimization on the basis of Self-Organization) is oriented toward solving of a wide class of the extremal problems including global optimization problems (see, for example, (Egorov, I. N., Kretinin, G. V., Leshchenko, I. A., Kuptzov, S. V., 2002)). The system is widely used to solve applied problems in various fields. There are versions of the system for parallel computational systems. The system is a commercial product, but is available for trial use.
- MATLAB Global Optimization Toolkit (see, for example, (Venkataraman, P., 2009))

includes a wide spectrum of methods for solving the global optimization problems, including multistart methods, global pattern search, simulated annealing methods, etc. The library is compatible to the TOMLAB system (see, for example, (Holmstrm, K., Edvall, M. M., 2004)), which is an additional extension the widely-used MATLAB. It is also worth noting that similar libraries for solving global optimization problems are available for MathCAD, Mathematica, and Maple systems as well.
- BARON (Branch-And-Reduce Optimization Navigator) is designed to solve continuous integer programming and global optimization problems using the branch and bound method (Sahinidis, N. V., 1996). BARON is included in the GAMS (General Algebraic Modeling System) system used extensively see (Bussieck, M. R., Meeraus, A., 2004).
- Global Optimization Library in R is a large collection of optimization methods implemented in the R language (see, for example, (Mullen, K. M., 2014)). Among these methods, are stochastic and deterministic global optimization algorithms, the branch and bound method, etc.

The list provided above is certainly not exhaustive additional information on software systems for a wider spectrum of optimization problems can be obtained, for example, in (Rios, L. M., Sahinidis, N. V., 2013), (Mongeau, M., Karsenty, H., Rouz, V., Hiriart-Urruty, J. B., 2000), (Pintr, J. D., 2009), etc. Nevertheless, even from such a short list the following conclusions can be drawn (see also Liberti, L., 2006):

- the collection of available global optimization software systems for practical use is insufficient,
- the availability of numerous methods through these systems allows complex optimization problems to be solved in a number of cases, however, it requires a rather high level of user knowledge and understanding in the field of global optimization,
- the use of the parallel computing to increase the efficiency in solving complex time-consuming problems is limited, therefore, the computational potential of modern supercomputer systems is very poorly utilized.

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, (Strongin, R. G., 1978), (Strongin, R. G., Sergeyev, Y. D., 2000), (Strongin, R. G., K. A., Gergel, V. P., Grishagin, V. A., Barkalov, K. A., 2013). The advantage of the Globalizer 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.

This paper is further structured as follows. In Section 2, a statement of the multidimensional global optimization problem is presented, and an approach to reducing these to one-dimensional optimization problems is described. In Section 3, parallel computation schemes are presented, and parallel optimization methods are described. In Section 4, the Globalizer architecture is examined. In Section 5, the results are presented from computational experiments that confirm the systems high level of efficiency. Finally, Section 6 presents the conclusions and some ideas for future research.

2. Multidimensional Global Optimization Problems and Dimension Reduction

In this paper, the core class[1] of optimization problems which can be solved using the Globalizer is examined. This involves multidimensional global optimization problems without constraints, which can be defined in the following way:

$$\varphi(y^*) = \min\{\varphi(y) : y \in D\} \\ D = \{y \in \mathbf{R}^N : a_i \leqslant x_i \leqslant b_i, 1 \leqslant i \leqslant N\} \tag{1}$$

where the objective function (y) satisfies the Lipschitz condition

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

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

Let us further assume that the minimized function $\varphi(y)$ is defined as a computational procedure, according to which the value $\varphi(y)$ can be calculated for any value of vector $y \in D$ (let us further call the process of obtaining a value for the minimized function a trial). As a rule, this procedure is computational-costly i.e. the overall costs of solving optimization problem (1) are determined, first of all, by the number of trials executed. It is also worth noting the essence of the assumption on satisfying the Lipschitz condition, since one can construct an estimate of the global minimum based on a finite number of computed values from the optimized function in this case only.

As has been previously shown by many researchers, finding numerical estimates of globally optimal extrema implies constructing coverage of the search domain D. As a result, the computational costs of solving global optimization problems are already very high even for a small number of varied parameters (the dimensionality of the problem). A notable reduction in the volume of computations can be achieved when the coverages of the search domain being obtained are non-uniform, i.e. the series of trial points is dense only in terms of its nearness to the sought-after globally optimized variants. The generation of such non-uniform coverages could only be provided in an adaptive way when the selection of the next trial points is determined by the search information (the preceding trial points and the values of the minimized function at these points) obtained in the course of computation. This necessary condition considerably complicates the computational schemes for global optimization methods, since it implies a complex analysis of a large amount of multidimensional search information. As a result, many optimization algorithms use, to some extent, various methods of dimensional reduction.

Within the framework of the information-statistical approach, Peano curves (or evolvents) $y(x)$ mapping the interval $[0, 1]$ onto an N-dimensional hypercube D are unambiguously used for dimensional reduction (see, for example, Strongin, R. G., Sergeyev, Y. D., 2000).

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

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

The dimensional reduction scheme considered above superimposes a multidimensional

[1]In general, the Globalizer can be applied for solving multicriterial multiextremal multidimensional optimization problems with nonlinear constraints.

problem with a minimized Lipschitz function to a one-dimensional problem with a corresponding minimized function to satisfy the uniform Hlder condition (see Strongin, R. G., 1978, Strongin, R. G., Sergeyev, Y. D., 2000) i.e.

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

where the constant H is defined by the relationship $H = 4L\sqrt{N}$, 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 presented in (Strongin, R. G., Sergeyev, Y. D., 2000). As an illustration, an approximation of a Peano curve for the third level of density is shown in Figure 1. The curve shown in Figure 1 demonstrates the winding order of a two-dimensional domain; the precision of the Peano curve approximation is determined by the level of density used in the construction.

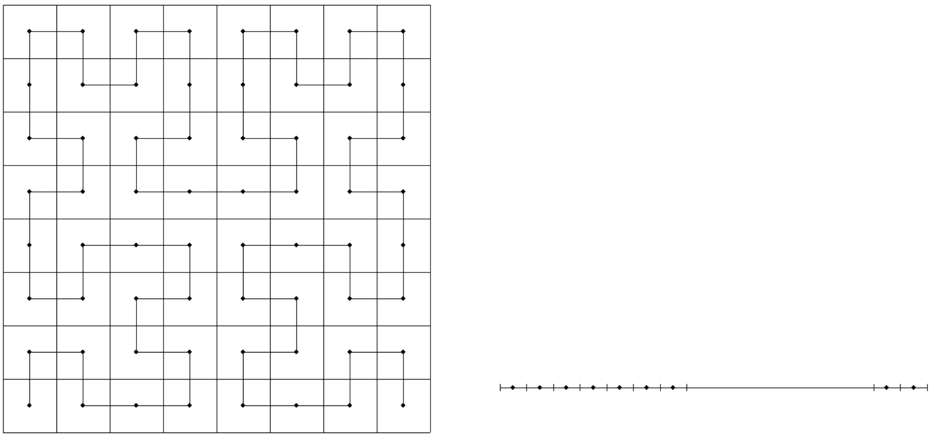

Figure 1. A Peano curve approximation for the third level of

The computational scheme obtained as a result of the dimensional reduction consists of the following (see Figure 2):

- The optimization algorithm performs the minimization of the reduced one-dimensional function $\varphi(y(x)$,
- After determining the next trial point x, a multidimensional image in the mapping $y(x)$ is calculated,
- The value of the initial multidimensional function $\varphi(y)$ is calculated at the multidimensional point $\in D$,
- 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. The Parallelization Approach and Global Optimization Algorithms

Let us consider the parallelization methods used widely in the theoretical and practical application of parallel computing within the context of global optimization problems:

- The distribution of the search domain D among the available computing units (data parallelization scheme). In the case of optimization problems, this approach is insuffi-

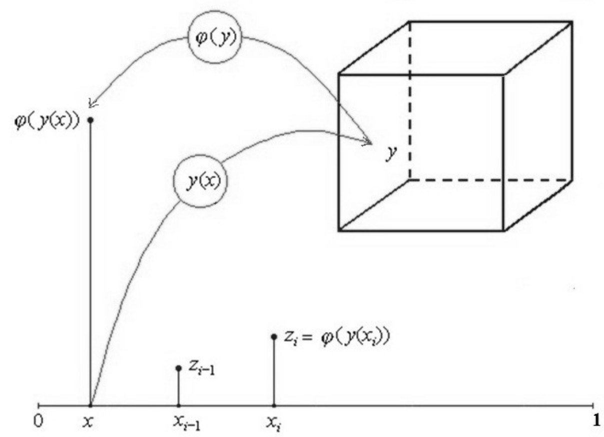

Figure 2. The computational scheme for obtaining the value of the reduced one-dimensional function $\varphi(y(x)$

cient, since in organizing such computations the subdomain, which contain the sought global minimum, will be processed by only one processor, and, therefore, all of the remaining processors would perform the excess computations.

- The parallelization of the computational schemes for optimization algorithms (task parallelization scheme). This approach is also insufficient since the direct computational costs for executing optimization algorithms are relatively low (the majority of computations in the course of a global search are represented by calculations of the optimized function values due to the initial assumption of considerable computational costs for such calculations).
- The parallelization of the computations executed in order to obtain values for the optimized function. This approach can be applied since the most computationally costly part of the global search process will be parallelized. However, this method is not featured by the generality (the development of parallelization methods is to be performed every time from scratch while solving each particular optimization problem).

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 D* see, for example, (Strongin, R. G., Sergeyev, Y. D., 2000), (Strongin, R. G., K. A., Gergel, V. P., Grishagin, V. A., Barkalov, K. A., 2013). This approach provides parallelizaion for the most costly part of computations in the global search process.

The global optimization algorithms implemented in Globalizer will be described step-by-step below. In Subsection 3.1, the core sequential multidimensional algorithm of global search (MAGS) will be presented. In Subsection 3.2, a parallel generalization of the MAGS algorithm for parallel computations on computer systems with shared memory will be described. In Subsection 3.3, the scheme for parallel computations by multiprocessor systems with distributed memory will be provided.

3.1 *Core Multidimensional Generalized Algorithm of Global Search*

The information-statistical theory of global search formulated in (Strongin, R. G., 1978), (Strongin, R. G., Sergeyev, Y. D., 2000) has served as a basis for the development of a large number of efficient multiextremal optimization methods — see, for example, (Gergel, V. P., 1996, 1997), (Gergel, V. P., Strongin, R. G., 2003, 2005), (Grishagin, V. A., Strongin, R. G., 1984), (Sergeyev Y. D., 1995, 1999), (Sergeyev, Y. D., Grishagin V. A., 1994, 2001), (Sergeyev, Y. D., Strongin, R. G., Lera, D., 2013), (Barkalov K. A., Gergel V. P., 2014), etc.

Multidimensional Algorithm of Global Search (MAGS) established the basis for the methods applied in Globalizer. The general computational scheme of MAGS can be presented as follows — see (Strongin, R. G., 1978), (Strongin, R. G., Sergeyev, Y. D., 2000).

Let us introduce a simpler notation for the problem being solved on a computing node

$$f(x) = \varphi(y(x)) : x \in [0;1]\} \tag{5}$$

The initial iteration of the algorithm is performed at an arbitrary point $x^1(0,1)$. Let us assume further k, $k > 1$, iterations of a global search are to be completed. The selection of the trial point $k+1$ for the next iteration is performed according to the following rules.

3.2 *Parallel Computations for Systems with Shared Memory*

3.3 *Parallel Computations for Systems with Distributed Memory*

4. Globalizer System Architecture

5. Results of Numerical Experiments

5.1 *Parallel Computations Using Intel Xeon Phi processors*

5.2 *Parallel Computations Using General Purpose Graphic Processors*

6. Conclusions

In this paper, the Globalizer global optimization software system was examined for implementing a general scheme for the parallel solution of globally-optimized decision making using a wide spectrum of computer systems: with shared and distributed memory, NVIDIA graphic processors and Intel Xeon Phi coprocessors. This approach enables the efficient use of modern supercomputer systems to solve the most computational costly multiextremal optimization problems. The computational experiments have confirmed the prospects for this proposed approach.

Acknowledgements

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