A two-phase approach in a global optimization algorithm using multiple estimates of hölder constants

Cite as: AIP Conference Proceedings **2070**, 020033 (2019); https://doi.org/10.1063/1.5090000
Published Online: 12 February 2019

Daniela Lera, and Yaroslav D. Sergeyev

A Two-Phase Approach in a Global Optimization Algorithm Using Multiple Estimates of Hölder Constants

Daniela Lera[1,a)] and Yaroslav D. Sergeyev[2,3]

[1]*Dipartimento di Matematica e Informatica, Università di Cagliari, Cagliari, Italy*
[2]*Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria, Italy*
[3]*Department of Software and Supercomputing Tecnologies, Lobachevsky University of Nizhni Novgorod, Russia*

[a)]Corresponding author: lera@unica.it

Abstract. In this paper, the global minimization problem of a multi-dimensional black-box Lipschitzian function is considered. In order to pass from the original Lipschitz multi-dimensional problem to a univariate one, an approach using space-filling curves to reduce the dimension is applied. The method does not use derivatives and, at each iteration, works with a set of estimates of the Hölder constant of the reduced one-dimensional problem. A two-phase technique is applied to accelerate the search of the global minimum. Numerical experiments carried out on several hundreds of test functions show a promising performance of the discussed algorithm in comparison with its direct competitors.

INTRODUCTION

We consider the following global optimization problem:

$$\min\{F(y) : \ y \in [a,b]\}, \qquad |F(y') - F(y'')| \le L\|y' - y''\|, \qquad y', y'' \in [a,b], \tag{1}$$

where $[a,b]$ is a hyperinterval in R^N and F is a multiextremal non-differentiable function that satisfies the Lipschitz condition with a constant L, $0 < L < \infty$, generally unknown. In the literature, one can find several algorithms that deal with the Lipschitzian global optimization problem (see e.g. [1–21]).

In this paper, in order to solve the problem (1) we use space-filling curves introduced by Peano in 1890 and independently by Hilbert in 1891. The curves under consideration emerge as the limit objects generated by an iterative process: they are fractals constructed using the principle of self-similarity. It is possible to prove that the curves fill in the hyperinterval $[a,b] \subset R^N$, i.e., they pass through every point of $[a,b]$. It is known that it is possible to reduce the dimension of the global optimization problem (1) by using the curves and to move from a multivariate problem to a univariate one (see [22–26]). More precisely, Strongin has proved (see [24, 27]) that finding the global minimum of the Lipschitz function $F(y), y \in R^N$, over a hyperinterval is equivalent to determining the global minimum of the function $f(x)$:

$$f(x) = F(p(x)), \qquad x \in [0,1], \tag{2}$$

where $p(x)$ is the Peano curve. Moreover, it follows that

$$|f(x') - f(x'')| \le H|x' - x''|^{1/N}, \qquad x', x'' \in [0,1], \tag{3}$$

i.e., $f(x)$ is a Hölderian function and $H = 2L\sqrt{N+3}$, where L is the Lipschitz constant of the N-dimensional function $F(y)$. In this way one can solve the problem (1) by using algorithms proposed for minimizing Hölderian functions in one dimension. In Fig. 1-left an example of a reduced function in 1 dimension is shown.

In this paper, we discuss and test the GOSH method proposed in [28] in which, by using Peano curves, we worked with the Hölder information in one dimension in order to obtain, at each iteration, several estimates of the Hölder constant using the DIRECT methodology from [5]. In [28], we have proposed a new two-phase procedure intended to accelerate the search and, in particular, the importance of the local improvement phase for an acceleration of the search has been discussed in the context of GOSH method introduced in [28].

Proceedings LeGO – 14th International Global Optimization Workshop
AIP Conf. Proc. 2070, 020033-1–020033-4; https://doi.org/10.1063/1.5090000
Published by AIP Publishing. 978-0-7354-1798-4/$30.00

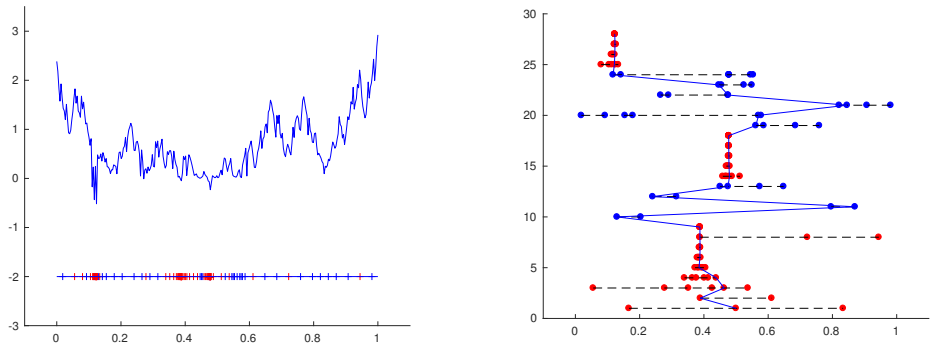

FIGURE 1. *An example of a reduced one-dimensional function and the respective trial points produced by the GOSH at the interval* $[0, 1]$ *(left); the dynamics of 28 iterations executed by the GOSH (right). Red points represents trials calculated in local phase.*

The Two-Phase Method

The popular DIRECT algorithm from [5] uses at each iteration several estimates of the Lipschitz constant for selecting a suitable set of subintervals, corresponding to the minimal characteristic, in the central points of which to evaluate the objective function. This selection can be easily done thanks to a smart representation of the intervals in a diagram in two dimensions. Let us see what happens when we work with the Hölder information related to the reduced function $f(x)$ from (2). In this case, it follows from [28] that for the one-dimensional function $f(x)$ satisfying (3) the following minorant (support function) can be constructed:

$$f(x) \geq G^k(x) = g_i(x), \quad x \in [a_i, b_i], \quad 1 \leq i \leq k, \tag{4}$$

$$g_i(x) = \begin{cases} g_i^-(x) = f(m_i) - H(m_i - x)^{1/N}, & x \in [a_i, m_i], \\ g_i^+(x) = f(m_i) - H(x - m_i)^{1/N}, & x \in [m_i, b_i], \end{cases} \tag{5}$$

where $G^k(x)$ is a discontinuous nonlinear minorant for $f(x)$ over each subinterval $d_i = [a_i, b_i]$, $1 \leq i \leq k$, and

$$m_i = (a_i + b_i)/2. \tag{6}$$

The values R_i called *characteristics of intervals* are calculated as follows

$$R_i = R_i(H_1) = \min_{x \in [a_i, b_i]} g_i(x) = f(m_i) - H_1|(b_i - a_i)/2|^{1/N}. \tag{7}$$

They are (see [28]) lower bound for $f(x)$ over each interval $[a_i, b_i]$, if H_1 is an overestimate of the constant H.

Let us show that in the framework of the Hölderian optimization a simple transposition of ideas of DIRECT from Lipschitz to Hölder metric does not work. In Fig. 2-left we represent each interval $d_i = [a_i, b_i]$ by points with coordinates $(h_i, f(m_i))$ following the scheme of DIRECT. If we consider a fixed overestimate H_1 (H_2) of the Hölder constant, we can observe the corresponding nonlinear support functions (5), shown in blue solid lines (green dashed lines), related to the intervals represented by dots A, B, C, D, and E. The characteristic $R_C(H_1)$ of the interval represented by the dot C is obtained as the intersection of the curve (5) constructed at the point C with the vertical coordinate axis. At each iteration intervals having the minimum characteristics are selected w.r.t all possible estimates of the constant H from zero to infinity. In Fig. 2-left we observe a lot of intersections among the curves corresponding to H_1 and the curves corresponding to H_2. As a result, it becomes unclear how to select the set of intervals that should be partitioned at each iteration for all possible estimates of H. In order to solve this difficulty, a new graphical representation of subintervals d_i has been introduced in [29] by using the Hölderian metric instead of the Euclidean one. Each interval is represented by a dot P_i with the coordinates (h_i, w_i), Fig. 2-right, where $h_i = |(b_i - a_i)/2|^{1/N}$, and $w_i = f(m_i)$ with m_i from (6). The introduction of the Hölderian metric allows us to avoid the non-linearity and the intersection of minorants giving as a result the diagram in Fig. 2-right. In Fig. 2-right we can see the representation of the same intervals

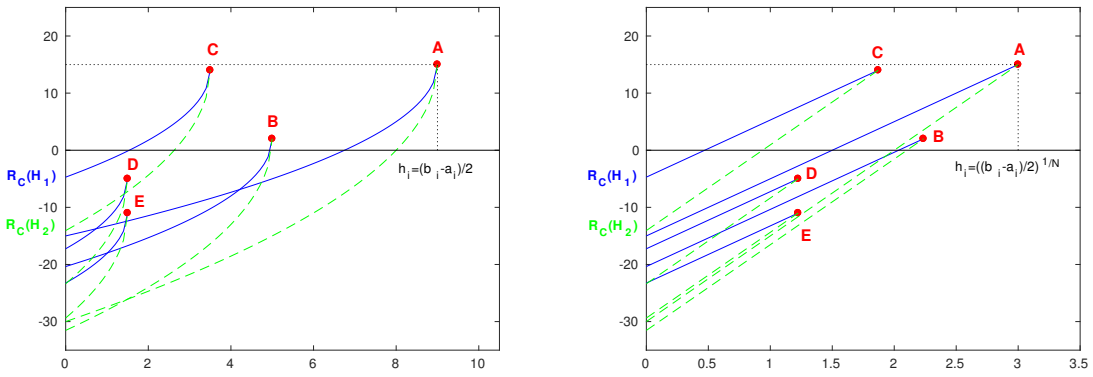

FIGURE 2. *Representation of intervals in Euclidean (left) and Hölder (right) metrics*

considered in Fig. 2-left. Notice that in Fig. 2-right the values in the horizontal axis are calculated in the Hölderian metric, while the vertical axis values coincide with those of Fig. 2-left.

The two-phase approach is related to the intervals selection procedure: at each iteration k the method should select in a suitable way a promising set of subintervals in which it intends to intensify the search and execute new trials (*trial* is evaluation of $f(x)$ at a point x that is called *trial point*). To accelerate the search, a two-phase technique that balances the global and local information collected during the work of the method is introduced. At a generic iteration k of the algorithm the interval $[0, 1]$ is subdivided into subintervals $d_i = [a_i, b_i], i = 1, ..., I(k)$, belonging to the current partition D_k and each interval is represented by a point in a two-dimensional diagram. The two phases are the following: the *global phase* in which large unexplored intervals are investigated in order to find attraction regions of local minimizers that are better than the current best found solution, and the *local phase* in which a local improvement of the current best found solution is performed. The local and global phases can be repeated several times during the work of the method changing one another. During the global phase, the algorithm explores mainly large intervals and identifies a set of promising intervals not among all groups of intervals but only among some sub-groups selected by using a calculated index that represents a separator between the groups of large intervals and small ones. The global phase is performed until a function value improving the current minimal value on at least 1% is obtained. When this happens, the method switches to the local phase in the course of which the obtained new solution is improved locally. In the case when the algorithm is not switched to the local phase during more than a fixed number of iterations (the improvement of the current minimum is still not found by exploring large intervals), it performs one "security" iteration in which determines promising intervals considering all groups of intervals present in the diagram. Once the selection phase (local or global) has been concluded, the chosen intervals are subdivided in order to produce new trial points (for a detailed discussion see [28]).

In order to illustrate the performance of the GOSH, we show in Fig. 1-left a reduced one-dimensional function corresponding to a two-dimensional problem and the respective trial points produced by GOSH at the interval $[0, 1]$ (red points represent trials calculated during the local phase). Fig. 1-right shows the dynamics (from bottom to top) of 28 iterations executed by GOSH, for the same function. It can be seen that each iteration contains more than one trial and the dashed line connects points with the best function value obtained during that iteration. The GOSH method was compared with three algorithms: (i) MGAS proposed in [29] that uses only the global phase; (ii) the original DIRECT [5]; (iii) and its locally biased modification LBDirect from [30]. Fig. 3 shows the behavior of the four methods on two classes of 100 test functions generated by the GKLS-generator (see [31] for a detailed description of its usage) in dimension $N = 3$ (left) and $N = 6$ (right): on the horizontal axis the number of function evaluations is exposed and the vertical coordinate of each curve shows how many problems have been solved by one or another method after executing the number of function evaluations corresponding to the horizontal coordinate.

ACKNOWLEDGMENTS

Research of D. Lera and Y.D. Sergeyev was supported by the INdAM-GNCS 2018 Research Project "Numerical methods in optimization and ODEs" (the authors are members of the INdAM Research group GNCS).

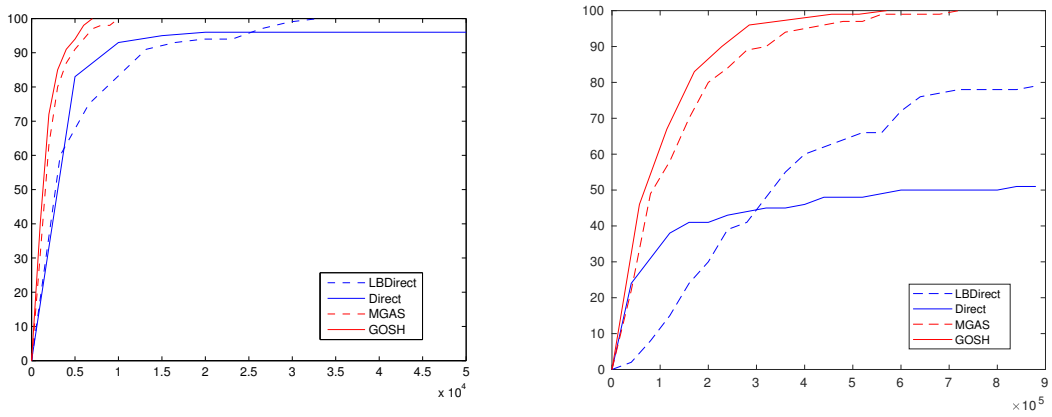

FIGURE 3. *Numerical results in dimension $N = 3$ (left) and $N = 6$ (right)*

REFERENCES

[1] Y. G. Evtushenko, *Numerical Optimization Techniques*, Translations Series in Mathematics and Engineering (Springer–Verlag, Berlin, 1985).
[2] D. Famularo, P. Pugliese, and Y. D. Sergeyev, Automatica **35**, 1605–1611 (1999).
[3] V. P. Gergel, M. I. Kuzmin, N. A. Solovyov, and V. A. Grishagin, Int. Rev. Automat. Contr. **8**, 51–55 (2015).
[4] R. Horst and P. M. Pardalos, eds., *Handbook of Global Optimization*, Vol. 1 (Kluwer Academic Publishers, Dordrecht, 1995).
[5] D. R. Jones, C. D. Perttunen, and B. E. Stuckman, J. of Optimization Theory and Applications **79**, 157–181 (1993).
[6] D. E. Kvasov and Y. D. Sergeyev, Advances in Engineering Software **80**, 58–66 (2015).
[7] R. Paulavičius and J. Žilinskas, *Simplicial Global Optimization*, SpringerBriefs in Optimization (Springer, New York, 2014).
[8] J. D. Pintér, *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications)* (Kluwer Academic Publishers, Dordrecht, 1996).
[9] R. G. Strongin and Y. D. Sergeyev, J. Global Optim. **27**, 25–50 (2003).
[10] A. A. Zhigljavsky and A. Žilinskas, *Stochastic Global Optimization* (Springer, New York, 2008).
[11] Y. D. Sergeyev and D. E. Kvasov, Commun. Nonlinear. Sci. Numer. Simul. **21**, 99–111 (2015).
[12] K. A. Barkalov and R. G. Strongin, J. Global Optim. **71(1)** (2018).
[13] Y. D. Sergeyev and V. Grishagin, J. of Computational Analysis and Applications **3(2)**, 123–145 (2001).
[14] Y. D. Sergeyev, Computational Optimization and Applications **34**, 229–248 (2006).
[15] K. Barkalov, V. Gergel, and I. Lebedev, AIP Conference Proceedings **1738**, p. 400006 (2016).
[16] V. A. Grishagin, R. Israfilov, and Y. D. Sergeyev, Appl. Math. Comput. **318**, 270–280 (2018).
[17] Y. D. Sergeyev, M. S. Mukhametzhanov, and D. E. Kvasov, Scientific Reports **8, 453** (2018).
[18] V. A. Grishagin and R. Israfilov, AIP Conference Proceedings **1738**, p. 400010 (2016).
[19] D. E. Kvasov and Y. D. Sergeyev, Optimization Letters **3**, 303–318 (2009).
[20] D. E. Kvasov and Y. D. Sergeyev, J. of Computational and Applied Mathematics **236**, 4042–4054 (2012).
[21] D. E. Kvasov, C. Pizzuti, and Y. D. Sergeyev, Numer. Math. **94**, 93–106 (2003).
[22] D. Lera and Y. D. Sergeyev, SIAM J. Optimization **23**, 508–529 (2013).
[23] Y. D. Sergeyev, R. G. Strongin, and D. Lera, *Introduction to Global Optimization Exploiting Space-Filling Curves*, SpringerBriefs in Optimization (Springer, New York, 2013).
[24] R. G. Strongin, *Numerical Methods in Multiextremal Problems: Information–Statistical Algorithms* (Nauka, Moscow, 1978) (In Russian).
[25] D. Lera and Y. D. Sergeyev, Appl. Numer. Maths. **60**, 115–129 (2010).
[26] D. Lera and Y. D. Sergeyev, J. Global Optim. **48**, 99–112 (2010).
[27] R. G. Strongin and Y. D. Sergeyev, Parallel Computing **18(11)**, 1259–1273 (1992).
[28] D. Lera and Y. D. Sergeyev, J. Global Optim. **71** (2018).
[29] D. Lera and Y. D. Sergeyev, Commun. Nonlinear. Sci. Numer. Simul. **23**, 328–342 (2015).
[30] M. J. Gablonsky and C. T. Kelley, J. Global Optim. **21**, 27–37 (2001).
[31] M. Gaviano, D. E. Kvasov, D. Lera, and Y. D. Sergeyev, ACM Trans. Math. Software **29**, 469–480 (2003).