



Numerical methods using two different approximations of space-filling curves for black-box global optimization

Yaroslav D. Sergeyev^{1,2} · Maria Chiara Nasso¹ · Daniela Lera³

Received: 11 January 2022 / Accepted: 20 July 2022 / Published online: 8 August 2022
© The Author(s) 2022

Abstract

In this paper, multi-dimensional global optimization problems are considered, where the objective function is supposed to be Lipschitz continuous, multiextremal, and without a known analytic expression. Two different approximations of Peano-Hilbert curve applied to reduce the problem to a univariate one satisfying the Hölder condition are discussed. The first of them, piecewise-linear approximation, is broadly used in global optimization and not only whereas the second one, non-univalent approximation, is less known. Multi-dimensional geometric algorithms employing these Peano curve approximations are introduced and their convergence conditions are established. Numerical experiments executed on 800 randomly generated test functions taken from the literature show a promising performance of algorithms employing Peano curve approximations w.r.t. their direct competitors.

Keywords Deterministic global optimization · Lipschitz and Hölder conditions · Space-filling curves · Black-box functions

1 Introduction

A wide variety of real-life applications requires a deep study of global optimization problems with many local minima and maxima (see, e.g. [1–10]). In particular, applications in engineering, machine learning, electronics, optimal control, etc. (see, e.g., [11–22]) are interested

✉ Yaroslav D. Sergeyev
yaro@dimes.unical.it

Maria Chiara Nasso
mc.nasso@dimes.unical.it

Daniela Lera
lera@unica.it

¹ Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria, Rende, Italy

² Institute of Information Technology, Mathematics and Mechanics, Lobachevsky State University, Nizhni Novgorod, Russia

³ Dipartimento di Matematica ed Informatica, Università di Cagliari, Cagliari, Italy

in finding the global (called also *absolute*) solution to the problem, since local solutions can often be unsatisfactory.

In this work, we consider the following global optimization problem

$$F^* = F(y^*) = \min F(y), \quad y \in D, \quad (1)$$

where $D = [a, b]^N$ is a hyperinterval in \mathbb{R}^N ,

$$D = \{y \in \mathbb{R}^N : a \leq y_j \leq b, \quad 1 \leq j \leq N\},$$

and $F(y)$ is the objective black-box function that satisfies the Lipschitz condition with an unknown Lipschitz constant L , $0 < L < \infty$, i.e.,

$$|F(y') - F(y'')| \leq L \|y' - y''\|, \quad y', y'' \in D, \quad (2)$$

where $\|\cdot\|$ denotes the Euclidean norm. Problems of this kind attract a great attention of researchers since they can be faced frequently in applications (see, e.g. [23–33]).

It has been shown (see, e.g., [33]) that solving the problem (1), (2) is equivalent to solve the following one-dimensional problem:

$$f(x^*) = F(y(x^*)) = \min F(y(x)), \quad x \in [0, 1], \quad (3)$$

where $y(x)$ is a Peano curve mapping the interval $[0, 1]$ in $[a, b]^N$. This kind of curve, first introduced by Peano in [34] passes through every point of $[a, b]^N$ (in the present paper, Hilbert's version of a Peano curve proposed in [35] will be used).

In addition, it can be also proved (see [33]) that the function $f(x)$ from (3) satisfies the Hölder condition

$$|f(x') - f(x'')| \leq H |x' - x''|^{1/N}, \quad x', x'' \in [0, 1], \quad (4)$$

with the constant

$$H = 2L\sqrt{N+3}.$$

Thus, we can use one-dimensional algorithms proposed in [31, 33] to solve (3), (4). Obviously, computable approximations to the Peano curve should be used in numerical algorithms. In several works (see, e.g., [31, 33, 36–38]) a *piecewise-linear approximation* hereinafter indicated as $l_M(x)$ was applied, where $M \in \mathbb{N}$ is the level of approximation, in the sense that it is constructed dividing the hypercube D into 2^{MN} subcubes with edge length equal to $2^{-M}(b-a)$.

In this paper, we propose to use another type of M -approximation of the Peano curve $y(x)$, called *Peano non-univalent approximation* and indicated by $n_M(x)$ (see [33]). Fig. 1 compares $l_1(x)$ (left picture) with $n_1(x)$ (right picture) and Fig. 2 shows $l_2(x)$ and $l_3(x)$ with $D = [-1, 1]^2$ whereas Fig. 3 presents $n_2(x)$ and $n_3(x)$. In [33], $n_M(x)$ was applied in so-called information methods. In contrast, in this paper, the non-univalent approximation is used in the framework of geometric global optimization algorithms that so far used the piecewise-linear approximation $l_M(x)$ only (see [31, 33, 36–38]).

The Peano non-univalent approximation is constructed using the same partition of D as $l_M(x)$ does but it reflects the following property (see [33]) of the Peano curve $y(x)$: a point in D can have several inverse images in $[0, 1]$ (but not more than 2^N). In order to describe this approximation, let us denote with $p(M, N)$ the uniform grid of the interval $[0, 1]$ composed by

$$p(M, N) = \left\{ \frac{j}{2^{(M+1)N} - 2^{MN}} : 0 \leq j \leq 2^{(M+1)N} - 2^{MN} \right\}$$

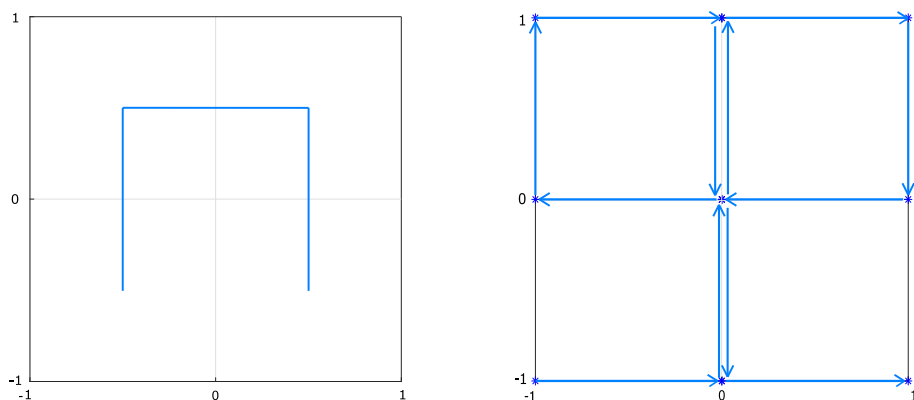


Fig. 1 Peano piecewise-linear approximation (left) and Peano non-univalent approximation (right) for $N = 2$, $M = 1$

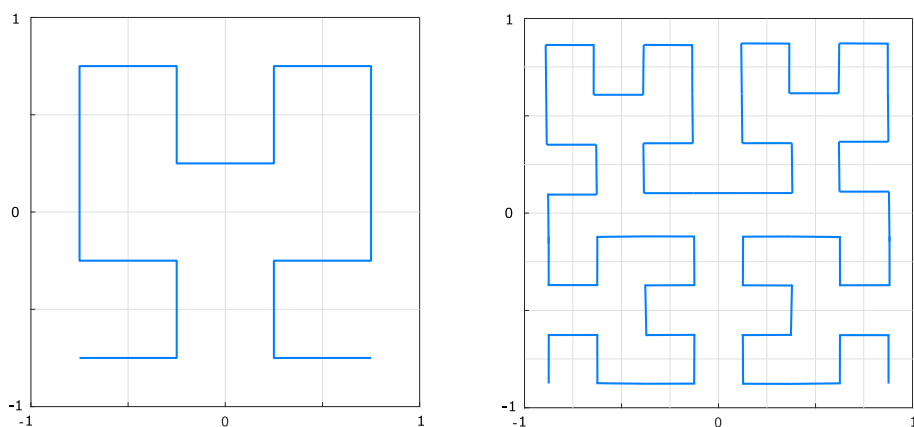


Fig. 2 Peano piecewise-linear approximation for $N = 2$ and $M = 2, 3$

and with $P(M, N)$ the grid of the M -partition of D which has the mesh width equal to $2^{-M}(b-a)$ (composed by the vertices of the subcubes of the M -partition of D) and satisfying another useful condition:

$$P(M, N) \subset P(M + 1, N).$$

Thus, the evolvent $n_M(x)$ maps the uniform grid $p(M, N)$ onto the grid $P(M, N)$. In this way, the result obtained by computing the value $F(n_M(\bar{x}))$ for a point \bar{x} of the uniform grid in $[0, 1]$ allows one to know the values $F(n_S(\bar{x}_i))$, where $S \geq M$ and the s , $1 \leq s \leq 2^N$, points $\bar{x}_i \in [0, 1]$, $1 \leq i \leq s$, are the *inverse images* of $n_M(\bar{x})$ with respect to the correspondence $n_M(x)$.

In Fig. 3.a), integers around the nodes of $P(2, 2)$ indicate the corresponding inverse images of $[0, 1]$. For example, the points $x_1 = \frac{8}{48} = \frac{1}{6}$, $x_2 = \frac{24}{48} = \frac{1}{2}$, $x_3 = \frac{40}{48} = \frac{5}{6} \in [0, 1]$ are all mapped through $n_2(x)$ in the point $(0, 0)$. In this way, computing a single evaluation, for example, at the point x_1 we know that the values of the function $F(n_M(x))$ at the points x_2 and x_3 will be the same for $M \geq 2$.

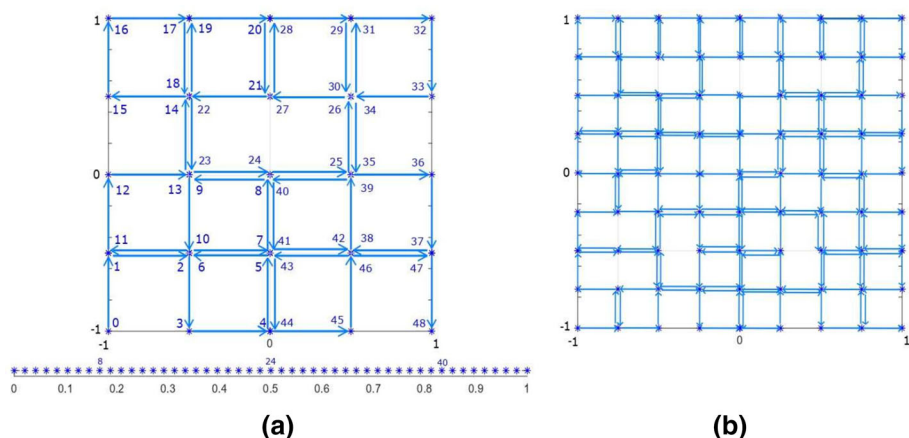


Fig. 3 Peano non-univalent approximation for $N = 2$ and $M = 2, 3$

The rest of the paper is structured as follows. In the next section, theorems discussing solution to the problem (1), (2) using Peano non-univalent approximations are given. In Sect. 3, two new geometric optimization algorithms are introduced and their convergence conditions are established. In Sect. 4, we discuss results of some numerical experiments and Sect. 5 is dedicated to conclusions.

2 Establishing a lower bound for the multi-dimensional objective function

Let us prove the following theorem which allows us to establish a lower bound for the function $F(y)$ over the entire multi-dimensional region D using a lower bound for the function $F(y)$ evaluated only along the approximation $n_M(x)$, $x \in p(M, N)$. It should be noticed that a similar result has been proven in [36] for the function $F(l_M(x))$ employing the piecewise-linear approximation $l_M(x)$. Even though the two approximations are very different (see illustrations in Figs. 1–3) we show that it is possible to link estimates of global minimizers of $F(y)$ over D and over the non-univalent approximation $n_M(x)$, as well.

Theorem 2.1 *Let U_M^* be a lower bound along $n_M(x)$ for a multidimensional function $F(y)$ satisfying Lipschitz condition with the constant L , i.e.,*

$$U_M^* \leq F(n_M(x)) \quad x \in p(M, N),$$

then the value

$$U = U_M^* - 2^{-(M+1)} L \sqrt{N} (b - a), \quad (5)$$

is a lower bound for $F(y)$ over the entire region D , i.e.,

$$U \leq F(y), \quad y \in D.$$

Proof Due to the construction of the non-univalent approximation $n_M(x)$, every point y in D can be approximated by s , $1 \leq s \leq 2^N$, different points $\alpha_i(y)$, hereinafter called *images* such that

$$\alpha_i(y) = \operatorname{argmin}\{\|y - \bar{y}\| : \bar{y} = n_M(x), x \in p(M, N)\},$$

with $1 \leq i \leq s$. This is due to the fact that a hypercube belonging to $P(M, N)$ and containing y has 2^N vertices, therefore, the closest to y vertices are less than 2^N . Let us consider now a point y and one of its images $\alpha(y)$. Since the function $F(y)$ satisfies the Lipschitz condition, we have

$$\begin{aligned} |F(y) - F(\alpha(y))| &\leq L\|y - \alpha(y)\|, \\ F(y) &\geq F(\alpha(y)) - L\|y - \alpha(y)\|. \end{aligned}$$

By hypothesis $F(\alpha(y)) \geq U_M^*$, therefore

$$F(y) \geq U_M^* - Ld_M, \quad (6)$$

where

$$d_M = \max_{y \in D} \|y - \alpha(y)\|.$$

It is easy to understand that d_M is equal to the distance between the center of a sub-cube of $P(M, N)$ and one of its vertex. In other words, this distance is half the length of the diagonal of a sub-cube of $P(M, N)$ which is

$$d_M = 2^{-(M+1)}\sqrt{N}(b-a). \quad (7)$$

Thus, from (6) and (7) we obtain

$$F(y) \geq U_M^* - L2^{-(M+1)}\sqrt{N}(b-a)$$

that concludes the proof. \square

Corollary 2.2 *The following condition holds*

$$\min\{F(n_M(x)) : x \in p(M, N)\} - F^* \leq 2^{-(M+1)}L\sqrt{N}(b-a), \quad (8)$$

where F^* from (1) is the global minimum of $F(y)$ over the search region D .

Proof Let us consider the value

$$U_M^* = \min\{F(n_M(x)) : x \in p(M, N)\}$$

that is the minimum of $F(y)$ along $n_M(x)$. It then follows from (5) that

$$F(y) \geq U_M^* - 2^{-(M+1)}L\sqrt{N}(b-a), \quad y \in D,$$

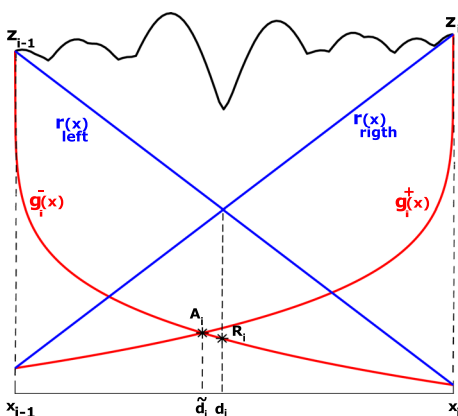
in particular,

$$\min\{F(y) : y \in D\} \geq U_M^* - 2^{-(M+1)}L\sqrt{N}(b-a),$$

which is equivalent to (8). \square

The theorem and its corollary mean, in particular, that, given the required accuracy to solve the problem (1), (2), it is possible to choose an appropriate level of approximation M for $n_M(x)$ in order to minimize $F(y)$ with the prescribed accuracy.

Fig. 4 The point (\tilde{d}_i, A_i) is approximated by the point (d_i, R_i)



3 Geometric methods and their convergence conditions

In this section, we introduce a Geometric Algorithm using Peano non-univalent approximation, GAP for short, for solving problem (1), (2). In order to proceed, we need to introduce some notations.

We indicate with the term *trial point* a point $x_i \in [0, 1]$ in which the evaluation of the objective function (this operation is called *trial*), denoted by $z_i = f(x_i)$ has been performed.

Let us consider a partition of $[0, 1]$ made by c trial points x_i , such that $x_i < x_{i+1}$, $1 \leq i \leq c-1$, and the following functions

$$\begin{aligned} g_i^-(x) &= z_{i-1} - H(x - x_{i-1})^{1/N}, \quad x \in [x_{i-1}, x_i], \\ g_i^+(x) &= z_i - H(x_i - x)^{1/N}, \quad x \in [x_{i-1}, x_i], \end{aligned}$$

and

$$G^c(x) = \bigcup_{i=2}^c g_i(x) = \bigcup_{i=2}^c \max\{g_i^-(x), g_i^+(x)\}.$$

It has been shown in [39] that $G^c(x)$ is a minorant over $[0, 1]$ for the function $f(x)$ which satisfies (4), (see Fig. 4). In [39], the authors use $G^c(x)$ with a priori given Hölder constant H and then find the “peak” points (\tilde{d}_i, A_i) which are the intersection of the graphs of the two functions $g_i^-(x)$ and $g_i^+(x)$ in order to choose among these points the new trial point (see Fig. 4). However, as it has been shown in [40], solving the nonlinear system of equations determining the peak point (\tilde{d}_i, A_i) can be difficult when N increases and the curves $g_i^-(x)$ and $g_i^+(x)$ tend to be flat. Clearly, another problem consists in the fact that it is difficult to know the constant H a priori.

For this reasons we follow [31, 36, 40] and consider an adaptive global estimate of the Hölder constant H and approximate the points (\tilde{d}_i, A_i) using the method proposed in [40] which does not need to solve a nonlinear system of equations of degree N and requires a smaller computing time. Let us describe this approach briefly.

The adaptive global estimate H_c of the Hölder constant H can be computed as follows

$$H_c = \max\{\xi, M_c\}, \quad (9)$$

where $\xi > 0$ is a small positive number and

$$M_c = \max\{m_i : 2 \leq i \leq c\},$$

with

$$m_i = \frac{|z_i - z_{i-1}|}{|x_i - x_{i-1}|^{1/N}}, \quad 2 \leq i \leq c.$$

Then the point (\tilde{d}_i, A_i) can be approximated with the point (d_i, R_i) , where

$$d_i = 0.5(x_i + x_{i-1}) - \frac{z_i - z_{i-1}}{2r H_c(x_i - x_{i-1})^{\frac{1-N}{N}}}, \quad (10)$$

is found as intersection of the following lines (see an illustration in Fig. 4)

$$\begin{aligned} r_{left}(x) &= -r H_c(x_i - x_{i-1})^{\frac{1-N}{N}} x + r H_c(x_i - x_{i-1})^{\frac{1-N}{N}} x_{i-1} + z_{i-1}, \\ r_{right}(x) &= r H_c(x_i - x_{i-1})^{\frac{1-N}{N}} x - r H_c(x_i - x_{i-1})^{\frac{1-N}{N}} x_i + z_i, \end{aligned}$$

with $r > 1$ being a reliability parameter of the global optimization method, while,

$$R_i = \min\{z_{i-1} - r H_c(d_i - x_{i-1})^{1/N}, z_i - r H_c(x_i - d_i)^{1/N}\}, \quad (11)$$

is found as the minimum value of the auxiliary functions $g_i^-(x)$, $g_i^+(x)$ evaluated at the point d_i . Figure 4 shows the point (\tilde{d}_i, A_i) found as intersection of $g_i^-(x)$, $g_i^+(x)$ and the point (d_i, R_i) found using $r_{left}(x)$, $r_{right}(x)$.

Thus, for each interval $[x_{i-1}, x_i]$, $2 \leq i \leq c$, its *characteristic* R_i is computed and then an interval $[x_{t-1}, x_t]$ for performing the next function evaluation x^{c+1} is chosen as follows

$$R_t = \min\{R_i : 2 \leq i \leq c\}.$$

Let us return now to the function $F(n_M(x))$. As said previously, the function $n_M(x)$ is defined only on the points of the grid $p(M, N)$ and not over the entire interval $[0, 1]$. Thus, when we need to evaluate the function $F(n_M(x))$ at a new trial point x^{c+1} we should proceed as follows: if x^{c+1} belongs to the grid $p(M, N)$, execute the next trial point at the point x^{c+1} i.e., $z^{c+1} = F(n_M(x^{c+1}))$, otherwise detect the point w of $p(M, N)$ which is the nearest to x^{c+1} from the left and compute $z^{c+1} = F(n_M(w))$. With abuse of notation, in both cases we will indicate with x^{c+1} , z^{c+1} the points in which the evaluation was performed and its result, respectively.

As compensation for losing some information in approximating the point x^{c+1} with w , at the cost of a single function evaluation we can obtain more than one trial point (at most 2^N), computing all the inverse images w_1, \dots, w_j , of the point $n_M(x^{c+1})$ with respect $n_M(x)$. In order to avoid redundancy, we will decide which inverse images to include in our trial points. Let us introduce two different strategies which lead to two different optimization algorithms. For this purpose, let us consider the ordered array W_0 composed by x^{c+1} and $\{w_1, \dots, w_j\}$ and define the following selection procedures discussing their meaning.

Selection rule 1:

Input: Array W_0

Output: Array W_1

Reject, if there are any, inverse images which are already included in our current trial points x_1, x_2, \dots, x_c . Namely, W_1 has as elements the points $w \in W_0$ such that $w \neq x_i$, $1 \leq i \leq c$.

Selection rule 2:**Input:** Array W_1 **Output:** Array W_2

Keeping x^{c+1} , discard from W_1 inverse images that are close to each other less than the width of the grid $p(M, N)$. In particular, we consider the following procedure:

$s = \text{width of the grid } p(M, N)$

$W_2(1) = W_1(1)$

for $i = 1 : \text{length}(W_1)$ **do**

if $\text{abs}(W_1(i) - W_2(\text{end})) > s$ **then**

$W_2(\text{end} + 1) = W_1(i)$

end

else if $0 < (x^{c+1} - W_2(\text{end})) < s$ **then**

$W_2(\text{end}) = x^{c+1}$

end

end

Selection rule 3.1:**Input:** Array W_2 **Output:** Array W'_3

Reject from W_2 the points inside $[x_{t-1}, x_t]$ different from x^{c+1} and also those points that are external from $[x_{t-1}, x_t]$ but too close to its endpoints i.e., W'_3 is the ordered array which has as elements the point x^{c+1} and all the points $w \in W_2$ which satisfy

$$(w < x_{t-1} \vee w > x_t) \wedge (|w - x_t| > \varepsilon) \wedge (|w - x_{t-1}| > \varepsilon),$$

with $\varepsilon > 0$ sufficiently small.

Selection rule 3.2:**Input:** Array W_2 **Output:** Array W''_3

If an improvement on at least 1% of the minimal function value

$$z_{\min} = \min\{z_i : 1 \leq i \leq c\}$$

is reached in x^{c+1} i.e.,

$$z^{c+1} \leq z_{\min} - 0.01 \cdot z_{\min},$$

and, in addition, the interval $[x_{t-1}, x_t]$ containing the current best point is not the smallest one, then $W''_3 = W_2$, otherwise W''_3 contains only x^{c+1} .

By combining these selection rules we obtain the following two algorithms:

GAP1 : the method employing the Geometric Algorithm using Peano non-univalent approximation and selection rules as written in Step 8.1;

GAP2 : the method employing the Geometric Algorithm using Peano non-univalent approximation and selection rules as written in Step 8.2.

As can be seen from the selection procedures described above, GAP1 avoids to generate too small intervals where the search region has already been well studied, whereas GAP2 avoids excessive partition of intervals that are possibly faraway from global minimizer.

Let us now describe the general scheme of the Geometric Algorithm.

Step 0. Initialization. Compute the first two function evaluations at the points $x^1 = 0, x^2 = 1$, i.e., compute the values $z^j = F(n_M(x^j))$, $j = 1, 2$. Set the iteration counter $k = 2$ and the trial points counter $c = 2$. The choice of the $(c+1)$ trial point is done as follows.

Step 1. Renumbering. Renumber the trial points x^1, x^2, \dots, x^c , and the corresponding function values z^1, z^2, \dots, z^c , of the previous c trial points by subscripts so that

$$0 = x_1 < x_2 < \dots < x_c = 1,$$

and each value z_i corresponds to the trial points x_i with $1 \leq i \leq c$. In addition, memorize the current record $z_{\min} = \min\{z^1, z^2, \dots, z^c\}$.

Step 2. Estimates of the Hölder constant. Calculate the current estimates H_c of the Hölder constant using (9).

Step 3. Calculation of characteristics. For each interval $[x_{i-1}, x_i]$, $2 \leq i \leq c$, compute the point d_i from (10) and the characteristic R_i from (11).

Step 4. Subinterval selection. Select an interval $[x_{t-1}, x_t]$ for performing the next function evaluation such that

$$R_t = \min\{R_i : 2 \leq i \leq c\}.$$

Step 5. Stopping criterion. Check whether the following condition holds

$$|d_{t(k)} - d_{t(k-1)}| \leq \delta,$$

where $\delta > 0$ is a given search accuracy. In the affirmative case, **Stop** and return $\tilde{x} = \arg \min\{F(n_M(x_i)), i = 1, \dots, c\}$ as an estimate of the global minimizer. Otherwise go to **Step 6**.

Step 6. New function evaluation. Execute the next trial at the point $x^{c+1} = d_t$.

Step 7. Calculation of inverse images. Compute all inverse images w_1, \dots, w_j , of the point $n_M(x^{c+1})$ with respect to $n_M(x)$.

Step 8. Selection of inverse images. Insert in an ordered array W_0 all points of $x^{c+1} \cup \{w_1, \dots, w_j\}$. To obtain **GAP1** go to **Step 8.1**, to obtain **GAP2** go to **Step 8.2**.

Step 8.1. Perform Selection 1,2,3.1 starting from W_0 (after applying Selection rule 1 to W_0 , the array W_1 is obtained, then the Selection rule 2 is applied to W_1 in order to obtain W_2 to which Selection rule 3.1 is applied). Denote by ν the number of selected inverse images contained in the obtained array W'_3 which will be all inserted in the list of trial points. Go to **Step 8.3**.

Step 8.2. Perform Selection 1,2,3.2 starting from W_0 (after applying Selection rule 1 to W_0 , the array W_1 is obtained, then the Selection rule 2 is applied to W_1 in order to obtain W_2 to which Selection rule 3.2 is applied). Denote by ν the number of selected inverse images contained in the obtained array W''_3 which will be all inserted in the list of trial points. Go to **Step 8.3**.

Step 8.3. Set $k = k + 1$, $c = c + \nu$, and go to **Step 1**.

Let us study convergence conditions of the introduced methods and consider an infinite trial sequence $\{x^c\}$ produced by any of the algorithms. First of all, it should be noticed that in the practical case $M < \infty$, the set of limit points of $\{x^c\}$ coincides with the set of limit points of the sequence $\{d_{t(k)}\}$. This is due to the fact that the sequence $\{x^c\}$ is composed by the points of $\{d_{t(k)}\}$ and the additional selected inverse images $\{w_l\}$ inserted at Step 8. Since the points of $\{w_l\}$ belong to the discrete finite (this is because M is finite) uniform grid $p(M, N)$, they cannot produce any additional accumulation point.

Let us study now the behaviour of an infinite sequence $\{d_{t(k)}\}$ of trial points generated by any of the two proposed algorithms, without checking the stopping criterion in Step 5, in the limit case $M = \infty$. The following convergence properties were already proved in [36] for the algorithms employing the piecewise-linear approximation. Now we show that these results hold for the non-univalent approximation to Peano curve, as well.

Theorem 3.1 *Let $f(x) = F(y(x))$ be the objective function which satisfies (4) where $y(x)$ is the Peano curve and let x' be any limit point of $\{d_{t(k)}\}$ generated by GAP1 or GAP2. Then the following assertions hold:*

1. *Convergence to x' is bilateral, if $x' \in (0, 1)$;*
2. *$f(x^c) \geq f(x')$, for any $c \geq 1$;*
3. *If there exists another limit point $x'' \neq x'$ of $\{d_{t(k)}\}$, then $f(x'') = f(x')$;*
4. *If the function $f(x)$ has a finite number of local minima in $[0, 1]$, then the point x' is locally optimal;*
5. *(Sufficient conditions for convergence to a global minimizer.) Let x^* be a global minimizer of $f(x)$. Suppose that there exists an iteration number k^* such that for all $k > k^*$ the inequality*

$$r \cdot H_c > H \quad (12)$$

holds, where H_c calculated at Step 2 at the k th iteration (see (9)) is an estimate of the Hölder constant H from (4) and r is the reliability parameter of the method. Then x^ and all its inverse images will belong to the set of limit points of the sequence $\{d_{t(k)}\}$ and, moreover, any limit point x of $\{d_{t(k)}\}$ is the global minimizer of $f(x)$.*

Proof The proof follows from that of Theorem 8.2 in [33] where analogous results for the piecewise-linear curve with $M = \infty$ and information algorithms have been established (see also [31] where geometric methods are discussed). The difference introduced by the non-univalent approximation consists in the insertion of the inverse images in the trial sequence. In fact, the sequences of all trial points produced by GAP1 and GAP2 contain the points $\{d_{t(k)}\}$ and the additional selected in Step 8 inverse images of the points $\{n_M(d_{t(k)})\}$ chosen with respect to $n_M(x)$. This does not influence the proof of assertions 1–4. From [33] it follows that the global minimizer x^* is the limit point of $\{d_{t(k)}\}$. Since all inverse images of x^* have the same value $f(x^*)$ and condition (12) holds over the whole search region $[0, 1]$, the inverse images and other global minimizers also will be limit points of the sequence. This ensures that assertion 5 is also satisfied. \square

4 Numerical experiments

In this section we discuss results of numerical experiments executed to compare the two new methods with the method MGA from [36] produced by the General Scheme employing the Peano piecewise-linear approximation and with the DIRECT algorithm from [41]. The experiments regarding GAP1, GAP2, and MGA were performed using MATLAB R2020b. The results of experiments using DIRECT were taken from [42]. In GAP1, GAP2, and MGA the technical parameter ξ from (9) was set to 10^{-8} . The value $\varepsilon = 10^{-3}$ was used in Selection rule 3.1 for GAP1. The GKLS-generator described in [43] was used to test the methods since it is very popular in the global optimization community because it provides classes of test functions with *known local and global minima*. These functions are defined by a convex quadratic function (paraboloid) systematically distorted by polynomials. The 8

classes of 100 N -dimensional test functions taken from [42] and used in the experiments are briefly described in Table 1 where the parameters m , f^* , d , and r_g indicate the number of local minima, the value of the global minimum, the distance from the global minimizer to the vertex of the paraboloid and the radius of the attraction region of the global minimizer respectively. An important aspect of the generator is that, for a complete repeatability of the experiments, if the same five parameters (N , m , f^* , d , and r_g) are used, then the same class of functions is exactly produced. Thus, if one compares several methods using the same class of functions then all the methods will optimize *the same* 100 randomly generated functions. Each of them satisfies (2) and is defined on $D = [-1, 1]^N$. The difficulty of a class can be increased either by decreasing the radius r_g or by increasing d , thus for each dimension $N = 2, 3, 4$ we have a simple class and a difficult one.

Since the global minimizers generated by the GKLS generator are known (this is one of the advantages of this generator), for GAP1 and GAP2, instead of the practical stopping rule used in Step 5, we applied the following stopping criterion: if an algorithm generated a point y' which satisfies

$$|y'(i) - y^*(i)| \leq \sqrt[N]{\Delta}(b - a), \quad 1 \leq i \leq N,$$

then the global minimizer y^* was considered found and the method stopped giving y' as an approximation to the solution for the problem (1). The same stopping criterion has been used in MGA and DIRECT, as well.

The reliability parameter r for GAP1, GAP2, and MGA has been chosen as follows: at most two value r_1 , r_2 of this parameter were used for each class. This is due to the fact that it is difficult to use different values of r for each function and, on the other hand, using a unique value of r does not allow the algorithms to show their complete potential. The value r_1 was obtained starting from the initial value 1.1 and it was increased with the step equal to 0.1 until at least the 95% of all test problems were solved and the maximum number of iterations is less than 15000 for Classes 1, 2 and 3, less than 50000 for Classes 4, 5 and 6 and less than 70000 for Classes 7 and 8. Then the parameter r_2 was used for the remaining unsolved problems. It was obtained starting from r_1 and then increased once again with the step equal to 0.1 until the remaining problems were all solved within the previous values of maximum number of iterations. Table 2 shows the values of the parameters r_1 , r_2 used for GAP1, GAP2, MGA and the number of solved problems with r_1 and r_2 (for instance 96(4) means that 96 problems were solved using the value r_1 and 4 problems with r_2). The value $M = 10$ was chosen as the number of partitioning of D both for the construction of Peano non-univalent approximation and for the piecewise-linear approximation. As the objective function is considered hard to evaluate, the number of function evaluation was chosen as the comparison criterion. Figure 5 illustrates the comparison between DIRECT and the methods using approximations of Peano curve on a problem of Class 2. In particular, Fig. 5 a) shows 3025 points of trials executed by DIRECT to find the global minimum of a problem of Class 2, Fig. 5 b), c), and d) present 641, 405, and 528 points of trials executes by MGA, GAP1, and GAP2, respectively, to solve the same problem. As can be seen from the figure, DIRECT remains too “stuck” at a local minimum with respect to its competitors and this implies a slower convergence to the global minimizer.

Table 3 shows the average number of function evaluations performed during minimization of all 100 functions while Table 4 reports the maximum number of function evaluations. The symbol “>” in Table 3 for the DIRECT algorithm means that the method stopped when the maximum number (1000000) of function evaluations had been executed for the particular test class. In these cases, we reported in the table a calculation of a lower estimate of the average. Finally the notation “> 1000000(j)” in Table 4 means that after 1000000 function

Table 1 Parameters setting for GKLS test functions taken from [42]

Class	dimension	m	f^*	d	r_g
1	2	10	−1	0.90	0.20
2	2	10	−1	0.90	0.10
3	3	10	−1	0.66	0.20
4	3	10	−1	0.90	0.20
5	4	10	−1	0.66	0.20
6	4	10	−1	0.90	0.20
7	5	10	−1	0.66	0.30
8	5	10	−1	0.66	0.20

Table 2 Values of the parameter r used in the experiments

Class	Method	r_1	r_2	Solved problems with r_1 (r_2)
1	MGA	1.5	1.8	96(4)
	GAP1	1.6	1.7	98(2)
	GAP2	1.6	1.7	97(3)
2	MGA	2	2.1	98(2)
	GAP1	2	2.1	99(1)
	GAP2	2.1	2.2	98(2)
3	MGA	1.1	1.2	95(5)
	GAP1	1.1	1.2	96(4)
	GAP2	1.1	1.2	96(4)
4	MGA	1.2	1.3	98(2)
	GAP1	1.3	1.4	98(2)
	GAP2	1.2	1.3	96(4)
5	MGA	1.2	1.3	99(1)
	GAP1	1.2	—	100(0)
	GAP2	1.1	1.2	99(1)
6	MGA	1.3	—	100(0)
	GAP1	1.1	1.2	97(3)
	GAP2	1.3	1.4	99(1)
7	MGA	1.1	1.2	98(2)
	GAP1	1.1	—	100(0)
	GAP2	1.1	—	100(0)
8	MGA	1.1	1.2	96(4)
	GAP1	1.1	—	100(0)
	GAP2	1.1	—	100(0)

evaluations the method under consideration was not able to solve j problems. The best results for each class are shown in bold in the tables.

As it can be seen from tables the advantage of the proposed methods becomes more evident when the dimension of problems increases and the problems become harder, allowing to reduce the average number and the maximum number of function evaluations. Figure. 6 shows the behavior of the four methods through the operational characteristics (see [44]) constructed on Class 6 (left picture) and Class 8 (right picture) that are graphs showing

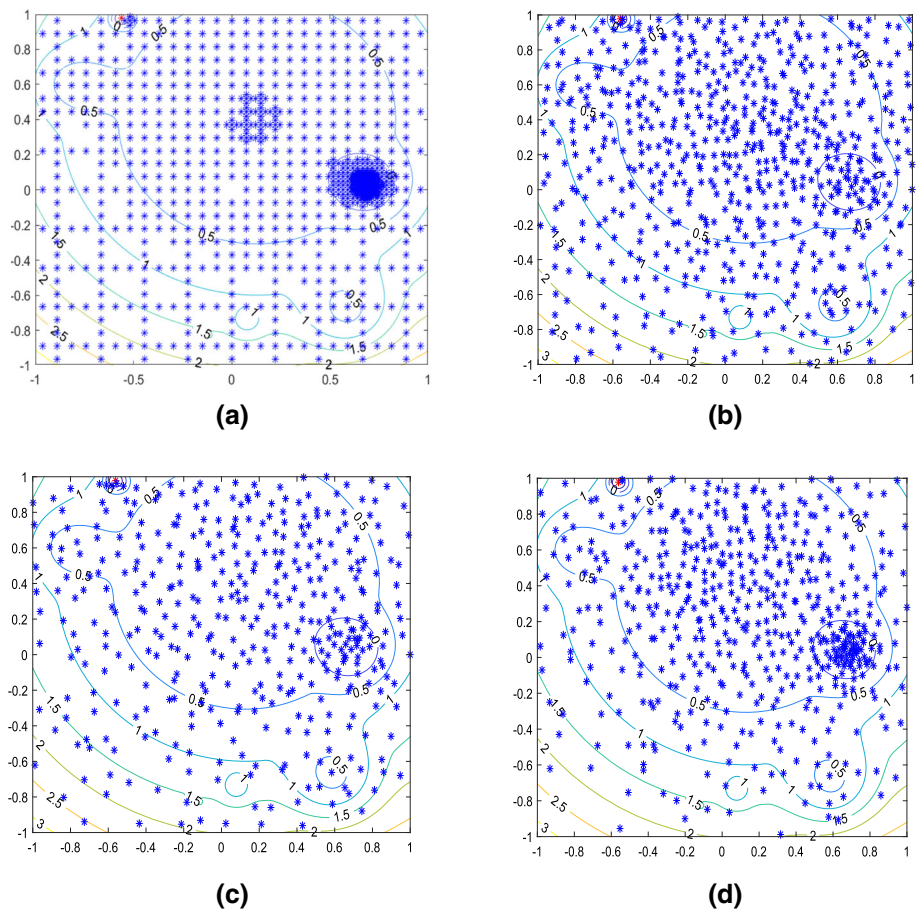


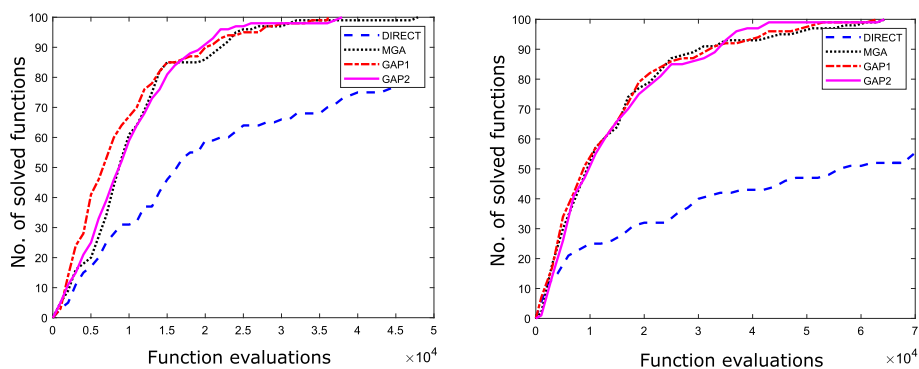
Fig. 5 Contour lines of a test function and trial points generated by DIRECT (a), MGA (b), GAP1 (c) and GAP2 (d) during their work. Global minimizer is marked by the red symbol “*”

Table 3 Average number of function evaluations for GKLS test functions

N	Class	Δ	DIRECT	MGA	GAP 1	GAP 2
2	1	10^{-4}	198.89	249.02	248.66	274.89
2	2	10^{-4}	1063.78	699.39	796.02	683.51
3	3	10^{-6}	1117.70	1311.31	1248.3	1261.17
3	4	10^{-6}	>42322.65	2413.01	2618.65	2671.25
4	5	10^{-6}	>47282.89	4504.33	4149.93	4339.02
4	6	10^{-6}	>95708.25	10360.63	8854.58	9889.38
5	7	10^{-7}	>16057.46	5941.37	5408.88	5403.33
5	8	10^{-7}	>217215.58	13650.57	13217.57	13525.18

Table 4 Maximum number of function evaluations for GKLS test functions

N	Class	Δ	DIRECT	MGA	GAP1	GAP2
2	1	10^{-4}	1159	723	825	1707
2	2	10^{-4}	3201	2525	4872	2326
3	3	10^{-6}	12507	12550	3580	6924
3	4	10^{-6}	>1000000 (4)	7206	8324	8049
4	5	10^{-6}	>1000000 (4)	18923	16313	29609
4	6	10^{-6}	>1000000 (7)	47908	36094	37950
5	7	10^{-7}	> 1000000 (1)	40469	32770	22248
5	8	10^{-7}	>1000000 (16)	64444	62932	64296

**Fig. 6** DIRECT, MGA, GAP1 and GAP2 on Class 6, $N = 4$ (left) and on Class 8, $N = 5$ (right)

the number of solved problems in dependence on the number of executed evaluations of the objective function. It can be seen that MGA, GAP1, and GAP2 are competitive in dependence of the available budget of function evaluations.

5 Conclusions

In this paper, the multi-dimensional global optimization problem of functions satisfying Lipschitz condition has been considered. In problems of this kind each evaluation of the objective function $f(x)$ is often a time consuming operation. The Geometric Algorithm MGA using the traditional Peano piecewise-linear approximation and two new multidimensional geometric algorithms GAP1, GAP2 have been considered. The methods GAP1 and GAP2 allow one to pass from the search domain in N dimension to the interval $[0, 1]$ using a not very well known approximation of the Peano curve, called *Peano non-univalent approximation*. This approximation denoted by $n_M(x)$ reflects the property of the Peano curve: a point y in D can have different inverse images x', x'' in $[0, 1]$ such that $n_M(x') = n_M(x'') = y$. The maximal possible number of inverse images of a point $y \in D$ is 2^N . Thus, the introduced algorithms GAP1 and GAP2 give the opportunity to evaluate only once the objective function $F(n_M(x))$ and introduce the same outcome to all multiple inverse images, leading to the chance of reducing the total number of expensive objective function evaluations. Numerical comparison of GAP1, GAP2, and MGA with the well-known DIRECT algorithm on 800

test problems constructed using the GKLS generator show a promising performance of all methods using Peano curves and, in particular, of the new algorithms.

Funding Open access funding provided by Università della Calabria within the CRUI-CARE Agreement.

Data Availability Data sharing is not applicable to this article since no datasets were generated or analysed during the current study.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Audet, C., Hare, W.: *Derivative-Free and Blackbox Optimization*. Springer, Natural Computing Series (2017)
2. Pardalos, P.M., Rosen, J.B.: *Constrained Global Optimization: Algorithms and Applications*. Springer Lecture Notes In Computer Science, vol. 268. Springer-Verlag, New York (1987)
3. Paulavičius, R., Sergeyev, Y.D., Kvasov, D.E., Žilinskas, J.: Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. *Expert Syst. Appl.* **144**, 113052 (2020)
4. Rios, L.M., Sahinidis, N.V.: Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.* **56**, 1247–1293 (2013)
5. Sergeyev, Y.D.: Efficient strategy for adaptive partition of n -dimensional intervals in the framework of diagonal algorithms. *J. Optim. Theory Appl.* **107**(1), 145–168 (2000)
6. Sergeyev, Y.D., Mukhametzhano, M.S., Kvasov, D.E., Lera, D.: Derivative-free local tuning and local improvement techniques embedded in the univariate global optimization. *J. Optim. Theory Appl.* **171**(1), 186–208 (2016)
7. Sergeyev, Y.D., Nasso, M.C., Mukhametzhano, M.S., Kvasov, D.E.: Novel local tuning techniques for speeding up one-dimensional algorithms in expensive global optimization using Lipschitz derivatives. *J. Comput. Appl. Math.* **383**, 113134 (2021)
8. Strongin, R.G., Barkalov, K., Bevzok, S.: Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints. *Soft. Comput.* **24**(16), 11853–11865 (2020)
9. Zhigljavsky, A., Žilinskas, A.: *Stochastic Global Optimization*. Springer, New York (2008)
10. Zhigljavsky, A., Žilinskas, A.: *Bayesian and High-Dimensional Global Optimization*. Springer, New York (2021)
11. Archetti, F., Candelieri, A.: *Bayesian Optimization and Data Science*. Springer, New York (2019)
12. Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., Sysoyev, A., Zolotykh, N.: Tuning hyperparameters of a svm-based water demand forecasting system through parallel global optimization. *Comput. Oper. Res.* **106**, 202–209 (2019)
13. Cavoretto, R., De Rossi, A., Mukhametzhano, M.S., Sergeyev, Y.D.: On the search of the shape parameter in radial basis functions using univariate global optimization methods. *J. Global Optim.* **79**, 305–327 (2021)
14. Daponte, P., Grimaldi, D., Molinaro, A., Sergeyev, Y.D.: An algorithm for finding the zero-crossing of time signals with Lipschitz derivatives. *Measurement* **16**(1), 37–49 (1995)
15. Famularo, D., Pugliese, P., Sergeyev, Y.D.: Global optimization technique for checking parametric robustness. *Automatica* **35**(9), 1605–1611 (1999)
16. Floudas, C.A., Pardalos, P.M.: *State of the Art in Global Optimization*. Kluwer Academic Publishers, Dordrecht (1996)
17. Kvasov, D.E., Sergeyev, Y.D.: Lipschitz global optimization methods in control problems. *Autom. Remote. Control.* **74**(9), 1435–1448 (2013)
18. Kvasov, D.E., Sergeyev, Y.D.: Deterministic approaches for solving practical black-box global optimization problems. *Adv. Eng. Softw.* **80**, 58–66 (2015)

19. Lera, D., Posypkin, M., Sergeyev, Y.D.: Space-filling curves for numerical approximation and visualization of solutions to systems of nonlinear inequalities with applications in robotics. *Appl. Math. Comput.* **390**, 125660 (2021)
20. Sergeyev, Y.D., Candelieri, A., Kvasov, D.E., Perego, R.: Safe global optimization of expensive noisy black-box functions in the δ -Lipschitz framework. *Soft. Comput.* **24**(23), 17715–17735 (2020)
21. Sergeyev, Y.D., Daponte, P., Grimaldi, D., Molinaro, A.: Two methods for solving optimization problems arising in electronic measurements and electrical engineering. *SIAM J. Optim.* **10**(1), 1–21 (1999)
22. Sergeyev, Y.D., Kvasov, D.E., Mukhametzhanov, M. S.: A generator of multiextremal test classes with known solutions for black-box constrained global optimization. *IEEE Transactions on Evolutionary Computation*. <https://doi.org/10.1109/TEVC.2021.3139263>, in press
23. Gergel, V.P., Grishagin, V.A., Israfilov, R.: Multiextremal optimization in feasible regions with computable boundaries on the base of the adaptive nested scheme. In: *Numerical Computations: Theory and Algorithms – NUMTA 2019*, volume 11974 of LNCS, pages 112–123. Springer, 2020
24. Grishagin, V., Israfilov, R., Sergeyev, Y.D.: Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes. *Appl. Math. Comput.* **318**, 270–280 (2018)
25. Kvasov, D.E., Sergeyev, Y.D.: Multidimensional global optimization algorithm based on adaptive diagonal curves. *Comput. Math. Math. Phys.* **43**(1), 40–56 (2003)
26. Paulavičius, R., Žilinskas, J.: *Simplicial Global Optimization*. Springer, New York (2014)
27. Sergeyev, Y.D.: On convergence of “Divide the Best” global optimization algorithms. *Optimization* **44**, 303–325 (1998)
28. Sergeyev, Y.D., Grishagin, V.A.: A parallel method for finding the global minimum of univariate functions. *J. Optim. Theory Appl.* **80**, 513–536 (1994)
29. Sergeyev, Y.D., Kvasov, D.E.: *Diagonal Global Optimization Methods*. FizMatLit, Moscow (2008). **(In Russian)**
30. Sergeyev, Y.D., Kvasov, D.E.: *Deterministic Global Optimization: An Introduction to the Diagonal Approach*. SpringerBriefs in Optimization. Springer, New York (2017)
31. Sergeyev, Y.D., Strongin, R.G., Lera, D.: *Introduction to Global Optimization Exploiting Space-Filling Curves*. Springer, New York (2013)
32. Strongin, R.G., Sergeyev, Y.D.: Global multidimensional optimization on parallel computer. *Parallel Comput.* **18**(11), 1259–1273 (1992)
33. Strongin, R.G., Sergeyev, Y.D.: *Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms*. Kluwer Academic Publishers, Dordrecht (2000)
34. Peano, G.: Sur une courbe, qui remplit toute une aire plane. *Math. Ann.* **36**, 157–160 (1890)
35. Hilbert, D.: Über die stetige abbildung einer linie auf ein flächenstück. *Math. Ann.* **38**, 459–460 (1891)
36. Lera, D., Sergeyev, Y.D.: Lipschitz and Hölder global optimization using space-filling curves. *Appl. Numer. Math.* **60**(1), 115–129 (2010)
37. Lera, D., Sergeyev, Y.D.: Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants. *Commun. Nonlinear Sci. Numer. Simul.* **23**(1–3), 328–342 (2015)
38. Lera, D., Sergeyev, Y.D.: GOSH: derivative-free global optimization using multi-dimensional space-filling curves. *J. Global Optim.* **71**(1), 193–211 (2018)
39. Gourdin, E., Jaumard, B., Ellaia, R.: Global optimization of Hölder functions. *J. Global Optim.* **8**, 323–348 (1996)
40. Lera, D., Sergeyev, Y.D.: Global minimization algorithms for Hölder functions. *BIT* **42**(1), 119–133 (2002)
41. Jones, D.R., Perttunen, C.D., Stuckman, B.E.: Lipschitzian optimization without the Lipschitz constant. *J. Optim. Theory Appl.* **79**, 157–181 (1993)
42. Sergeyev, Y.D., Kvasov, D.E.: Global search based on efficient diagonal partitions and a set of Lipschitz constants. *SIAM J. Optimization* **16**(3), 910–937 (2006)
43. Gaviano, M., Lera, D., Kvasov, D.E., Sergeyev, Y.D.: Algorithm 829: 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)
44. Grishagin, V.: Operational characteristics of some global search algorithms. *Problems of Stochastic Search* **7**, 198–206 (1978)