UNIVERSITÀ DELLA CALABRIA

Department of Computer Engineering,

Modelling, Electronics and Systems Science

Ph.D. Thesis in
Information and Communication Technologies
Academic Discipline: MAT/08 - Numerical Analysis

Global Optimization and Fractal Curves

Maria Chiara Nasso

MARIA
CHIARA
NASSO
13.07.2022
14:16:49
GMT+00:00

Scientific Advisor
Prof. Yaroslav D. Sergeyev

YAROSLAV
SERGEEV
13.07.2022
15:21:05 UTC

Coordinator
Prof. Felice Crupi

CRUPI FELICE
13.07.2022 16:03:08
GMT+00:00

Academic Year 2020-2021

Contents

Introduction

This work is mainly dedicated to global optimization and in particular to numerical global optimization methods based on fractal constructions. This research makes contributions at various levels with the development of new efficient algorithms and powerful acceleration techniques in the framework of Lipschitz optimization and with the theoretical study on the existence and construction of area-filling curves.

Giuseppe Peano defined in 1890 a curve parametrized by a continuous and surjective function from the interval $[0, 1]$ to the interval $[0, 1]^2$. This kind of curve are indicated with "Peano's curves" or "Space filling curves" and they are usually obtained by uniform limit of a sequence of poligonal curves. Space filling curves are very useful for optimization of functions with several extremal points, especially when the number of these points can be high. These kinds of problems are known as multiextremal problems or global optimization problems and are very important in Machine Learning and in many practical applications (technological processes, engineering design, economic models, biology studies, etc.). Moreover, global optimization's aim is to find algorithms to solve optimization problems where the objective function has not a known analytic expression, it is hard to evaluate and there are not strong information like convexity or differentiability. In other words, in practical global optimization, the objective function can be often a *black-box* function. This means that there is a "black-box" that, given in input a value, returns the corresponding function evaluation. This evaluation can be a really computationally challenging problem in engineering. For example, in automatics and robotics, even one of these evaluations may be obtained

by executing a set of experiments and may take many hours.

For these reasons innovative approaches and acceleration techniques are of particular interest.

In this work methods with a fractal approach to reduce dimensionality of problems using space-filling curves and methods that are based on local tuning on the functions behavior are considered.

In Chapter 1 after a brief introduction on Peano's curve and other fractal constructions, we will consider diagonals algorithms. In these algorithms, the search multidimensional domain is partitioned adaptively into a set of hyperintervals, and the objective function is evaluated only at the vertices corresponding to the main diagonal of each sub-hyperinterval. Then, it will be described a new, recently developed, technique unifying the ideas of the Peano curves and ones of the diagonal methods using an efficient diagonal partition. It turns out to be a construction of a sequence of *adaptive diagonal curves* which are similar to Peano's curves but become more dense in the vicinity of the global minimizers of the objective function if the selection of hyperintervals for partitioning is realized appropriately. The new partition strategy can significantly outperform (especially, when the problem dimension increases) traditional schemes in terms of the number of functions evaluations.

In Chapter 2 area-filling curves, i.e. continuous and injective mappings defined on $[0, 1]$ whose graph has positive measure are discussed. Despite the fact that current literature calls them "Osgood curves", we will show that Lebesgue invented such curves one year before Osgood and we suggest to call them "area-filling curves".

Sierpiński and Knopp criticized the fact that in Osgood's construction some arcs have area zero and they proposed two different examples of area-filling curves γ which have only arcs of strictly positive area. We will call curves with this property *positive* curves. Stromberg and Tseng constructed "homogeneous" area-filling curves. In their paper [118] they presented a construction of a positive area-filling curve which has an additional property. They wrote: "there is a 'homogeneity' feature which is desirable but not guaranteed by Knopp's and Sierpiński's construction: given $\beta \in]0, 1[$, one

may ask that for every subset E of $[0, 1]$ the image $\gamma(E)$ have area equal to β times the 'length' of E" but we show that an appropriate variant of Knopp's construction attains the same homogeneity result. We will discuss the existence of an "invasive" curve, i.e. a continuous and injective mapping from the half-open interval $[0, 1[$ to the unit square, whose image has measure 1, and also several aspects of the Lance-Thomas curve, connecting it with another construction due to Stromberg and Tseng.

Moreover part of the research topic studied in this work is the acceleration of algorithms thanks to the *local tuning technique*. This new approach allows to build global optimization algorithms that adapt their behavior to the one of the objective function on the various areas of the search region. This kind of research is useful for problems where the objective function under consideration has bounded slopes, in other words, it satisfies the Lipschitz property. An important subclass in Lipschitz global optimization consists of functions with the first derivative satisfying the Lipschitz condition.

In Chapter 3 our attention is on two univariate Lipschitz global optimization problems. In the first problem the objective function satisfies the Lipschitz condition while, in the second one, the first derivative of the objective function satisfies the Lipschitz condition.

For the first problem we will discuss geometric and information statistical frameworks for construction of global optimization algorithms, for the second one smooth piece-wise quadratic support functions are used in the presented methods which are implemented both in the traditional floating-point arithmetic and in the *Infinity Computing* framework allowing one to efficiently compute exact derivatives in the case where the optimized function is given as black box. In each framework we will describe several Local Tuning techniques which are able to estimate local Lipschitz constants in different areas of the search region allowing to accelerate the search. Convergence conditions of the methods are established and numerical experiments presenting advantages of these techniques are discussed.

In Chapter 4 multi-dimensional global optimization problems are considered, where the objective function is supposed to be Lipschitz continuous, multiextremal and without a known analytic expression. Different approxi-

mations of Peano curve to reduce the problem to a univariate one satisfying the Hölder condition are discussed. Two new multi-dimensional geometric algorithms employing a Non-Univalent approximation of Peano curve are introduced and their convergence conditions are established. Numerical experiments executed on 900 test functions show a promising performance of the algorithms.

Obtained scientific results have been presented on 5 international conferences. Moreover, 2 papers have been published in the international journals and 1 paper has been accepted for publication with minor revisions, 1 contribution to the book and 2 papers in proceedings of the international conferences have been also published [49, 58, 69, 70, 108, 109].

Chapter 1

Peano's curve and other fractal constructions

In this chapter we will see the Peano curve and other fractal constructions. Giuseppe Peano was the first to build a curve defined on the interval $[0, 1]$ whose image covers $[0, 1]^2$. Not long after Hilbert proposed another construction (which is more popular today) and the two giants were followed by Lebesgue, Sierpiński and others. As noticed in [86] we will see that the previous Peano's (or space-filling) curves can be obtained as limit of a sequence of *approximating polygons*. We will briefly discuss fractal constuctions and their history. In the last part of the chapter the *Diagonal Approach* introduced in [78, 79, 80, 81, 91] and the *Non-redundant Diagonal Partition Strategy* [89, 91] which produces a sequence of curves similar to traditional space-filling curves is discussed. Figures presented in Section 1.2 are taken from [54] with the permission of the authors.

1.1 Space-filling curves and Classical Fractals

A curve which passes through all the points of a square appeared for the first time in 1890 in a note by Giuseppe Peano [77]. The Peano curve is a curve parameterized by a continuous and surjective function from the interval $[0, 1]$

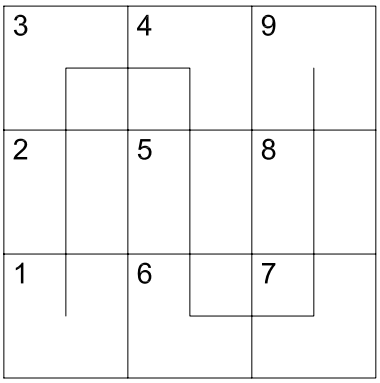

Figure 1.1. $\gamma_1 : Q_1 \to Q_2$.

to the square $[0,1]^2$ and it is obtained as the uniform limit of a sequence of curves.

However, Peano's construction, was purely analytical, it was David Hilbert in [42], the year after, who gave a geometric construction of a curve that has similar properties.
Peano's idea was the following: we initially consider the curve $\gamma_1 : Q_1 = [0,1] \to Q_2 = [0,1]^2$, shown in Figure (1.1).

The interval $[0,1]$ is divided into 9 congruent intervals and for each $j = 1,..,9$, the function γ_1 maps the interval $[\frac{j-1}{9}, \frac{j}{9}]$ piecewise-linearly onto the segments of the polygonal chain contained in the $j-$th square. Each subinterval $[\frac{j-1}{9}, \frac{j}{9}]$, with $j = 1,..,9$ is divided into nine congruent subintervals and each square into nine congruent subsquares. The function γ_2 maps each of these new subintervals to one of the nine new squares continuously. The previous procedure is iterated. We report in Figure (1.2) the second step of the construction.

It follows that, at the $n-$th step, Q_1 and Q_2 are divided into 3^{2n} congruent replicas of themselves.

Since the "grid" of the construction of the n-th step of the curve is composed of squares which diameter tends to 0 for $n \to \infty$, the functions γ_n satisfy the Cauchy condition for uniform convergence and are continuous. Thus γ_n converges uniformly on $[0,1]$ to a continuous curve γ and since the image $\gamma([0,1])$ is closed and dense in Q_2, $\gamma([0,1]) = Q_2$. The function γ is

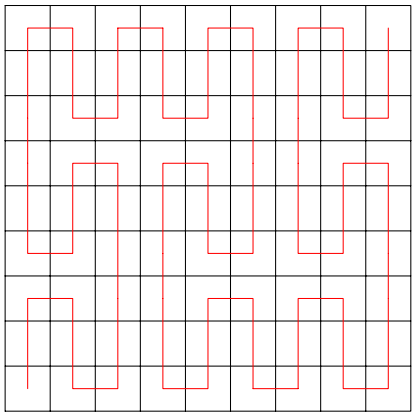

Figure 1.2. γ_2.

A	-A*	A
A*	-A	A*
A	-A*	A

Figure 1.3. Construction pattern of γ_2.

the Peano curve.

It should be noticed that Figure 1.2 follows the scheme shown in Figure 1.3. In other words, γ_2 is a polygonal chain which is obtained by joining the curves drawn on nine pieces of a puzzle. We can identify four different types of these curves which we indicate with A, A^*, $-A$ and $-A^*$, where the presence of the minus sign indicates that we had to walk on the curve starting from the top to the bottom of the piece. Also note that A^* is obtained from the reflection that mirrors all the points of A which are above the central horizontal axis of the piece, with respect to the same axis, (see Figure 1.4).

The idea of use some replacements rules of a particular initial pattern lead to many other self-similar curves and many interesting fractal constructions.

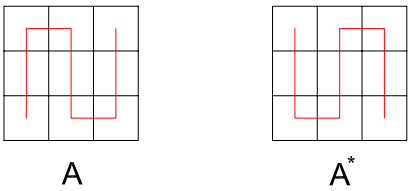

Figure 1.4. Curves A and A^*.

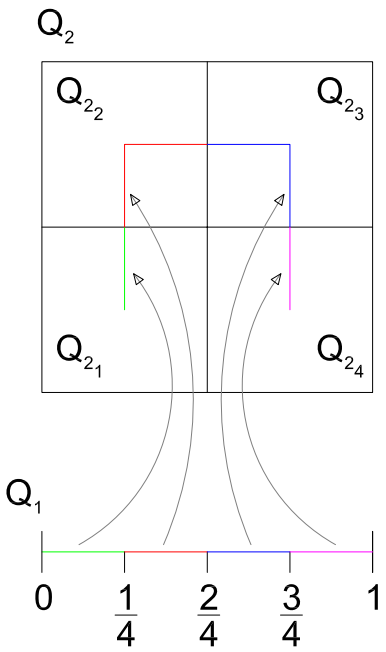

Figure 1.5. $\gamma_1 : Q_1 \to Q_2$.

Later, other authors (Hilbert, Sierpiński, Lebesgue and others) construct curves which passes through every point of a square. We will call these curves "Peano's curves" or "Space filling curves". Here we report Hilbert's construction. The idea is the following: we initially consider the curve parameterized by $\gamma_1 : Q_1 = [0,1] \to Q_2 = [0,1]^2$, shown in Figure 1.5.

Each of the sub-intervals $[\frac{i-1}{4}, \frac{i}{4}]$, with $i = 1,..,4$ is divided into four congruent subintervals and each of the four squares Q_{2_i} with $i = 1,..,4$ into four congruent squares. The function γ_2 maps each of these new subintervals to one of the four new squares continuously.

In their turn, each of the subintervals can be divided into four congruent

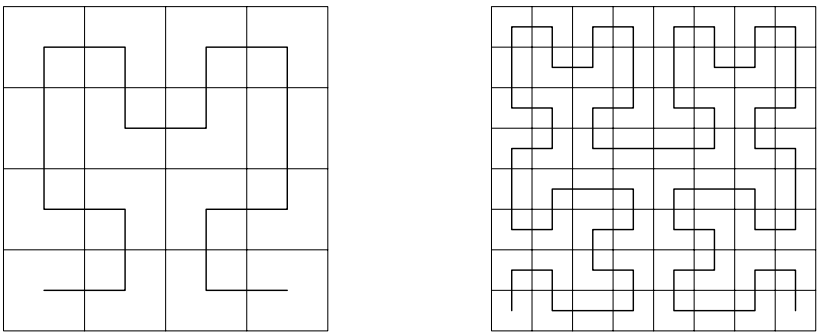

Figure 1.6. γ_2 and γ_3.

subintervals and each of the four squares into four congruent squares and the previous procedure is iterated. We report in Figure 1.6 the second and third construction steps.

It follows that, at the $n-$th step, Q_1 and Q_2 are divided into 2^{2n} congruent replicas of themselves.

Moreover, adjacent subintervals correspond to adjacent squares with an edge in common so that the inclusion relationships are preserved: if an interval corresponds to a square, then its four subsquares correspond to the subintervals of the same interval.

Also in this case, since the grid of the construction of the $n-$th step of the curve is composed of squares which diameter tends to 0 for $n \to \infty$, the function γ_n satisfy the Cauchy condition for uniform convergence and are continuous. Thus γ_n converges uniformly on $[0,1]$ to a continuous curve γ and since the image $\gamma([0,1])$ is closed and dense in Q_2, $\gamma([0,1]) = Q_2$. The function γ is a Peano's curve.

As noticed in [86] the previous space-filling curves can be obtained as limit of a sequence of *approximating polygons* $\{p_n\}$.

Definition 1.1. The polygonal line that joins the points

$$\gamma(0), \gamma\left(\frac{1}{3^{2n}}\right), \gamma\left(\frac{2}{3^{2n}}\right), \ldots, \gamma\left(\frac{3^{2n-1}}{3^{2n}}\right), \gamma(1),$$

is called the $n-$th approximating polygon for the original Peano curve γ.

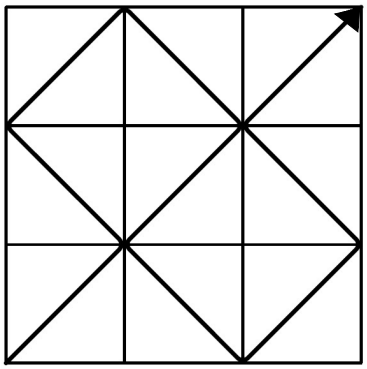

Figure 1.7. Fist approximating polygon for the Peano curve.

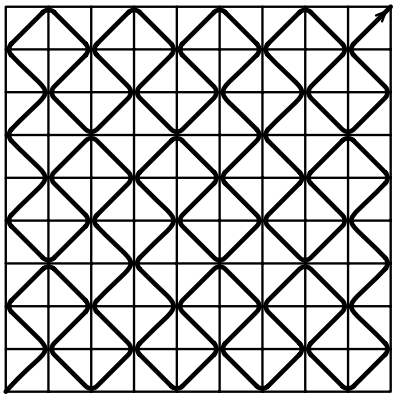

Figure 1.8. Second approximating polygon for the Peano curve.

Thus the approximating polygon $p_n : [0,1] \to [0,1]^2$ is parametrized by

$$p_n(t) = 3^{2n}\left(t - \frac{k}{3^{2n}}\right)\gamma\left(\frac{k+1}{3^{2n}}\right) - 3^{2n}\left(t - \frac{k+1}{3^{2n}}\right)\gamma\left(\frac{k}{3^{2n}}\right),$$

for $k/3^{2n} \leq t \leq (k+1)/3^{2n}, k = 0, 1, 2, 3, \ldots 3^{2n} - 1$ and

$$||\gamma(t) - p_n(t)|| \leq \frac{\sqrt{2}}{3^n}, \text{for all } t \in [0,1].$$

Fig 1.7 and Fig 1.8 present the approximating polygons for $n = 1, 2$ where the corners were slightly rounded in order to have a more clear illustration. For $n \to \infty$, the Peano curve is obtained.

Similarly, we can introduce the following

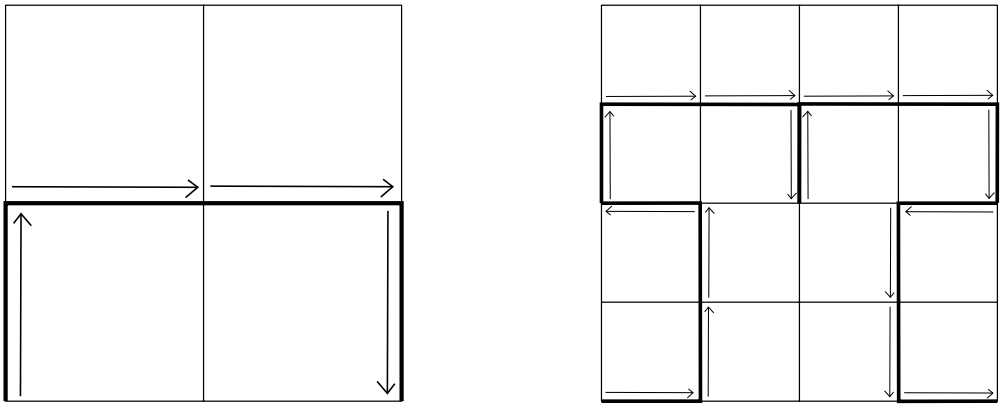

Figure 1.9. First and second approximating polygons for the Hilbert curve.

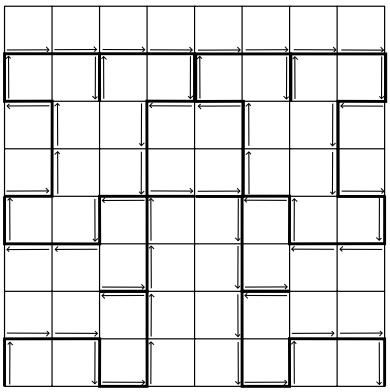

Figure 1.10. Third approximating polygon for the Hilbert curve.

Definition 1.2. The polygonal line that joins the points

$$\gamma(0), \gamma\left(\frac{1}{2^{2n}}\right), \gamma\left(\frac{2}{2^{2n}}\right), \ldots, \gamma\left(\frac{2^{2n-1}}{2^{2n}}\right), \gamma(1),$$

is called the $n-$th approximating polygon for the Hilbert curve γ.

Fig 1.9 and Fig 1.10 show the first three elements of the sequence of the approximating polygons for the Hilbert curve. The arrows indicate the progression of the curves.

As showed above, the Peano curve and the Hilbert curve exhibit similar patterns, this property is called *self-similarity* and it is typical in fractal

structures. Fractals are geometrical objects which appear the same at different scales, therefore by enlarging any part of them, a figure similar to the original is obtained. Let us now introduce a general construction for fractals which uses self-similar sets.

Let D be a closed subset of $\mathbb{R}^n$. A mapping $S : D \to D$ is called a *contraction* on D if there is a number c , $0 < c < 1$ such that

$$|S(x) - S(y)| \leq c|x - y|, \quad \forall x, y \in D.$$

If equality holds then S is called a *similarity* and transforms sets into geometrically similar ones.

Definition 1.3. Let $S_1, S_2, \ldots, S_m$ be contractions. A subset F of D is invariant for the trasformations S_i if

$$F = \bigcup_{i=1}^{m} S_i(F).$$

This sets are often fractals ([22]).

Let us define with $\mathcal{S}$ the class of all non-empty compact subsets of D and if $A \in \mathcal{S}$, the set $A_\delta = \{x \in D : |x - a| \leq \delta \text{ for some } a \in A\}$ which consists of points within distance δ of A, is called $\delta-$*parallel body* of A.

Recall that the Hausdorff metric between $A, B \in \mathcal{S}$ is defined by $d(A, B) = \inf\{\delta : A \subset B_\delta \text{ and } B \subset A_\delta\}$. Then $(\mathcal{S}, d)$ is a metric space. As shown in [22] the following theorem holds.

Theorem 1.1.1. *Let $S_1, \ldots, S_m$ be contractions on $D \subset \mathbb{R}^n$ so that*

$$|S_i(x) - S_i(y)| \leq c_i|x - y| \quad (x, y \in D)$$

with $c_i < 1$ for each i. Then there exists a unique non-empty compact set F that is invariant for the S_i, i.e. which satisfies

$$F = \bigcup_{i=1}^{m} S_i(F).$$

Moreover, if we define a transformation S on the class $\mathcal{S}$ of non-empty compact sets by

$$S(E) = \bigcup_{i=1}^{m} S_i(E)$$

and write S^k *for the* $k-th$ *iterate of* S *given by* $S^0(E) = E, S^k(E) = S(S^{k-1}(E))$ *for* $k \geq 1$, *then*

$$F = \bigcap_{k=1}^{\infty} S^k(E)$$

for any set E *in* $\mathcal{S}$ *such that* $S_i(E) \subset E$ *for each* i.

Thus if E is an initial set in $\mathcal{S}$ and F is a fractal, the $S^k(E)$ are increasingly good approximations to F which are often called *pre-fractals* for F.

When $S_1, \ldots S_k : \mathbb{R}^n \to \mathbb{R}^n$ are similarities and a set F is invariant for these transformations, F is the union of a number of smaller copies of itself and it is called strictly self similar set. The most famous examples of this situation are the middle third Cantor set and the von Koch curve.

In the middle third Cantor set F we have that the fundamental self-similarities are $S_1, S_2 : \mathbb{R} \to \mathbb{R}$ defined by

$$S_1(x) = \frac{1}{3}x,$$

$$S_2(x) = \frac{1}{3}x + \frac{2}{3}.$$

Then $F = S_1(F) \cup S_2(F)$ and if E is the interval $[0, 1]$, as in Theorem 1.1.1 we can define the transformations S on the class $\mathcal{S}$ of non-empty compact sets by

$$S(E) = \bigcup_{i=1}^{2} S_i(E),$$

and the pre-fractals for F by $S^0(E) = E$, $S^k(E) = S(S^{k-1}(E))$ for $k \geq 1$, see Fig. 1.11.

Another fractal curve (and one of the earliest fractals to have been described) appeared in 1904 in [120] written by the Swedish mathematician Helge von Koch.

The Koch curve is generated by the following procedure. Begin with the straight line $[0, 1]$,

- divide the line segment into three segments of equal length,

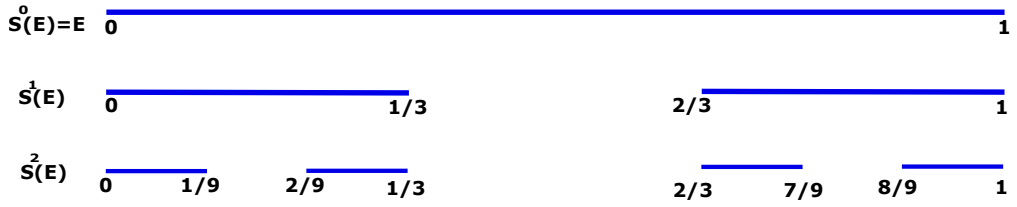

Figure 1.11. Generation of the middle third Cantor set.

- replace the middle segment by the two sides of an equilateral triangle of the same length,
- remove the line segment that is the base of the new constructed triangle.

The Koch curve is the limit approached as the above steps are followed indefinitely for each line segment. It easy to see that the previous procedure is obtained considering the following four similarity transformations $S_i : \mathbb{R}^2 \to \mathbb{R}^2$, $i = 1, 2, 3, 4$:

$$S_1(x) = \frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x.$$

$$S_2(x) = \frac{1}{3}\begin{pmatrix} 1/2 & -\sqrt{3}/2 \\ \sqrt{3}/2 & 1/2 \end{pmatrix} x + \frac{1}{3}\begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

$$S_3(x) = \frac{1}{3}\begin{pmatrix} 1/2 & \sqrt{3}/2 \\ -\sqrt{3}/2 & 1/2 \end{pmatrix} x + \frac{1}{3}\begin{pmatrix} 3/2 \\ \sqrt{3}/2 \end{pmatrix}.$$

$$S_4(x) = \frac{1}{3}\begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} x + \frac{1}{3}\begin{pmatrix} 2 \\ 0 \end{pmatrix}.$$

If we apply S_i, $i = 1, 2, 3, 4$ to the interval $[0, 1]$ we obtain four copies of the initial straight line, each scaled by the factor $\frac{1}{3}$. In particular, $S_1([0, 1])$ is the segment AB in Fig 1.12 (a), $S_2([0, 1]) = BC$ and $S_3([0, 1]) = CD$ are the two segments rotated by an angle of 60°, one counterclockwise and one clockwise respectively along with the required translations and finally $S_4([0, 1]) = DE$, (see Fig. 1.12 (a)). Applying S_i, $i = 1, 2, 3, 4$ to the preceding configuration we obtain Fig 1.12 (b) and so on. The fixed attractor of this iterated functions

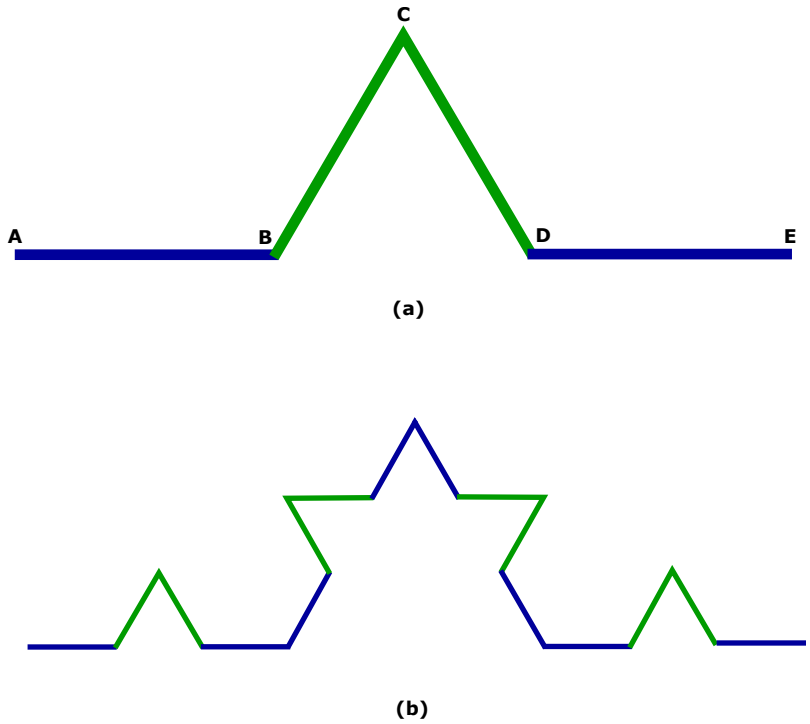

Figure 1.12. Generation of the von Koch curve.

system is the Koch curve. If we start with a triangle instead of a simple straight line we obtain the Koch snowflake. Fig 1.13 shows the 7-th step of the construction of the Koch snowflake.

It should be noticed that the Koch curve is a non-differentiable curve, indeed differentiation can only be applied to functions whose graphs look like straight lines in the vicinity of the point at which you want to differentiate while the Koch curve exhibits self-similarity, so it is identical at all scales. At that time fractals were only thought as mathematical oddities. Richard F. Voss called the von Koch curve "one of the early mathematical monsters".

As reported in `https://it.wikipedia.org/wiki/Curva_di_Koch` in 1975 Benoît Mandelbrot in his book [66] proposed the Koch curve as a model of the coast of an island. It is a famous figure that the Italian mathematician Ernesto Cesàro described in the following way:

"It is this similarity between the whole and its parts, even the infinitesimal ones, that leads us to consider the Koch curve as a marvelous line among all.

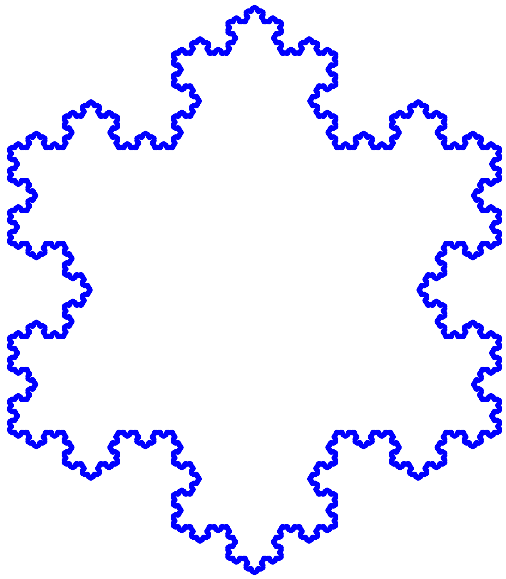

Figure 1.13. Iteration number 7 of the construction of the von Koch snowflake.

If it were endowed with life, it would not be possible to annihilate it without suppressing it at the first blow, since otherwise it would be reborn incessantly from the depths of its triangles, like life in the universe".

Paul Lévy also wrote:

"Undoubtedly our intuition foresaw that the absence of tangent and the infinite length of the curve was linked to infinitely small hairpin bends that one cannot think of drawing. But we are confused by the fact that our imagination cannot even go beyond the first steps in the construction of these infinitely small hairpin bends". Along the lines of Lévy, Hugo Stainhaus wrote:

"We are approaching reality, considering that most of the arcs encountered in nature are non-rectifiable. This statement contrasts with the belief that non-rectifiable arcs are an invention of mathematicians, and that natural arcs are rectifiable: the opposite is true".

Charles Hermite, bound to a certain idea of purity of the geometric function in front of the Koch curve declared that he would

"withdraw with fear and horror from this lamentable plague of functions

that have not derivative".

Only when Mandelbrot in 1982 published [67] a real fractal geometry was born. Fractal geometry is an irreplaceable tool for studying many complex natural phenomena.

Peano's curves and fractal constuctions have had surprising applications in global optimization in recent years. We will see some of these applications in the following chapters. On the other hand, next section is dedicated to a global optimization approach which produces during its work a series of curves similar to those illustrated above. This approach is called *Diagonal Approach.*

1.2 Diagonal approach

Let us consider the following global optimization problem.

Let $D = \{x \in \mathbb{R}^N : a(j) \leq x(j) \leq b(j), 1 \leq j \leq N\}$, $a, b \in \mathbb{R}^N$ and $f : D \to \mathbb{R}$ a multidimensional multiextremal black-box function that satisfies the Lipschitz condition:

$$|f(x') - f(x'')| \leq L||x' - x''||, \quad \forall\, x', x'' \in D,$$

with an unknown Lipschitz constant L, $0 < L < \infty$. We want to find the point $x^* \in D$ such that

$$f^* = f(x^*) = \min f(x), \quad x \in D. \tag{1.1}$$

In [78, 79, 80, 81, 91] some *diagonal algorithms* were introduced. These algorithms subdivide the search domain D into adaptively generated hyperintervals $D_i = [a_i, b_i]$ where a_i, b_i are the vertices of the main diagonal connecting them. Let us denote with $\delta(D_i)$ the boundary of D_i, $M(k)$ the number of hyperintervals D_i at the beginning of the $k-$th iteration and $\Delta M(k) > 1$ is the number of new hyperintervals produced during the subdivision of a hyperinterval at the current $k-$th iteration. Then the diagonal partition $\{D^k\}$ of the search domain at the k-th iteration satisfies

$$D^k = \bigcup_{i=1}^{M(k)+\Delta M(k)-1} D_i, \quad D_i \cap D_j = \delta(D_i) \cap \delta(D_j), \quad i \neq j.$$

The diagonal approach combines the idea of approximating the function $f(x)$ on the basis of the evaluations performed over each subset D_i of a partition of the search domain D and the idea to reduce the dimension of the problem. In particular, the general scheme of the diagonal algorithms in [78, 79, 80, 81, 91] is described as follows. Let $\{x^{p(k)}\}$ the sequence of trial points generated by the algorithm in k iterations and $\{z^{p(k)}\}$ the corresponding objective function evaluations.

Step 1. The first two trials are performed at the vertices $x^1 = a, x^2 = b$ of the search domain D. The current partition of D is $\{D^1\} = \{D\}$, the number of functional evaluation is $p(1) = 2$ and the current number of generated hyperintervals is $M(1) = 1$. After executing k iterations the iteration $k + 1$ is done as follows:

Step 2. For each hyperinterval $D_i = [a_i, b_i]$, $1 \leq i \leq M(k)$ of the partition $\{D^k\}$ compute the characteristic R_i :

$$R_i = R(D_i^k, \{x^{p(k)}\}, \{z^{p(k)}\}, \nu) \quad 1 \leq i \leq M(k), \tag{1.2}$$

with ν vector of parameters of the algorithm.

Step 3. Find the hyperinterval $D_t = [a_t, b_t]$ with the maximal characteristic

$$t = arg \max_{1 \leq i \leq M(k)} R_i.$$

Step 4. If

$$||a_t - b_t|| > \varepsilon ||a - b||,$$

where $\varepsilon > 0$ is a given search accuracy, then subdivide the hyperinterval D_t into $\Delta M(k)$ new hyperintervals by means of a partition strategy P,

$$\{D^{k+1}\} = P(\{D^k\}, \{x^{p(k)}\}, \{z^{p(k)}\}, \nu),$$

in such a way that

$$D_{i(k)} = D_{i(k+1)} \quad i(k) \neq t,\ i(k+1) \neq t,$$

$$D_{t(k)} = D_{t(k+1)} \cup \left(\bigcup_{M(k)+1}^{M(k)+\Delta M(k)-1} D_{i(k+1)} \right),$$

and go to Step 5. Otherwise, calculate an estimate of the minimum as

$$z_k^* = \min\{z : z \in z^{p(k)}\}$$

attained at the point

$$x_k^* = arg\min\{f(x^j) : x^j \in x^{p(k)}\}$$

and Stop.

Step 5. Execute the next $\Delta p(k)$ trials at the vertices corresponding to the main diagonal of the new generated hyperintervals. Set $p(k+1) = p(k) + \Delta p(k)$, $M(k+1) = M(k) + \Delta M(k) - 1$, increase the iteration counter $k = k + 1$ and go to Step 2.

Notice that this procedure is a natural generalization of many one-dimensional algorithms since the main diagonals of the hyperintervals can be seen as one-dimensional intervals and the behaviour of the function $f(x)$ is described through its evaluation at the endpoints of these intervals.

Usually the partition strategy P in Step 4 works with a point selection function

$$S_t = S(\{D_t^k\}, \{x^{p(k)}\}, \{z^{p(k)}\}, \nu) \tag{1.3}$$

which chooses a point of D_t and subdivide it by means of some hyperplanes parallel to its boundary hyperplanes and passing through the chosen point. Normally this point is on the main diagonal of D_t. We can obtain, for example, a generalization of Piyaskij-Shubert's method [82, 112] (in Chapter 3 we will see some accelaration techniques for this global optimization method), letting the characteristic R_i from (1.2) as

$$R_i = -\frac{f(a_i + f(b_i))}{2} + \overline{L}\frac{||b_i - a_i||}{2}, \quad 1 \leq i \leq M(k),$$

and the point selection function

$$S_i = \frac{a_i + b_i}{2} - \frac{f(b_i) - f(a_i)}{2\overline{L}} \times \frac{b_i - a_i}{||b_i - a_i||}, \; 1 \leq i \leq M(k),$$

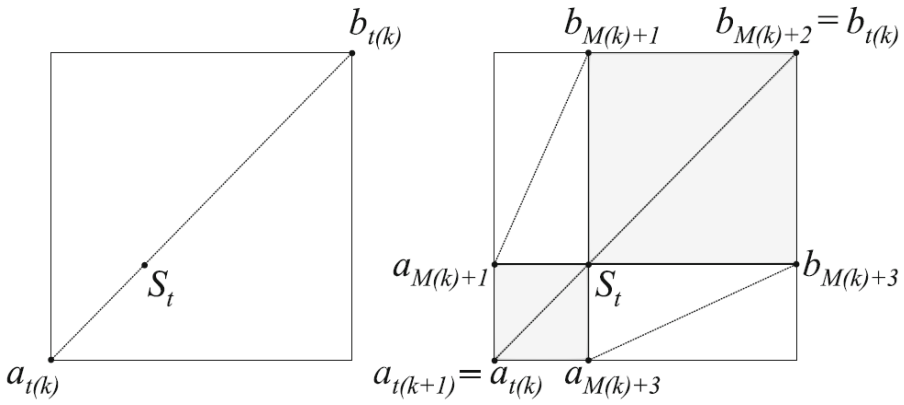

Figure 1.14. Partition of the two-dimensional hyperinterval D_t performed by the 2^N- Partition strategy.

with $\overline{L}$ an overestimate of the Lipschitz constant L.

Traditionally there are two strategies for the hyperinterval partitioning P used at the Step 4: the 2^N-Partition and the Bisection Partition. In this Section we summarize them, while in the last part we will illustrate the *efficient diagonal partition strategy* from [54].

1.2.1 Traditional Partition Schemes

In the 2^N- Partition strategy the point selection function from (1.3) choose a point S_t on the main diagonal of the hyperinterval $D_{t(k)}$, then the partition operates as follows. First, $D_{t(k)}$ is split into 2^N hyperintervals using N hyperplanes which pass through the point S_t and are parallel to the boundary hyperplanes of D_t. Then the hyperinterval D_t is replaced by 2^N new hyperintervals, so that $\Delta M(k) = 2^N$ and

$$D_{t(k)} = D_{t(k+1)} \cup D_{M(k)+1} \cup \cdots \cup D_{M(k)+\Delta M(k)-1}.$$

The objective function $f(x)$ is evaluated at the vertices of the main diagonals a_j, b_j of all new hyperintervals. In this way we have $\Delta p(k) = 2^{N+1} - 3$ new trial points because $S_t = a_{M(k)+\Delta M(k)-1}$ coincide with the point $b_{t(k+1)}$ and the objective function $f(x)$ has already been evaluated at the vertices $a_{t(k)}$

and $b_{t(k)}$ of the hyperinterval $D_{t(k)}$ in the previous iterations.

Example 1.1. In Fig. 1.14 we have $N = 2$ and the hyperinterval D_t is splitted into 2^2 new hyperintervals such that

$$D_{t(k)} = D_{t(k+1)} \cup D_{M(k)+1} \cup D_{M(k)+2} \cup D_{M(k)+3}.$$

The objective function $f(x)$ has already been evaluated at the vertices $a_{t(k)}$ and $b_{t(k)}$ before splitting (see Fig. 1.14, left picture). The new trials are performed at $\Delta p(k) = 2^3 - 3 = 5$ points which are $S_t, a_{M(k)+1}, b_{M(k)+1}, a_{M(k+3)}$, $b_{M(k)+3}$ (see Fig. 1.14, right picture).

In the Bisection Partition, as in the previous strategy, the selected point S_t lies on the main diagonal of the hyperinterval D_t but the subdivision is executed by one hyperplane which is orthogonal to the longest edge of D_t and passes through the point S_t. The hyperinterval D_t is replaced by 2 new hyperintervals (not necessarily of the same volume) $D_{t(k+1)}, D_{M(k)+1}$, so that $\Delta M(k) = 2$ and

$$a_{t(k+1)} = a_{t(k)},$$

$$b_{t(k+1)} = (b(1), b(2), \ldots, b(i-1), S_t(i), b(i+1), \ldots, b(N)),$$

$$a_{M(k)+1} = (a(1), a(2), \ldots, a(i-1), S_t(i), a(i+1), \ldots, a(N)),$$

$$b_{M(k)+1} = b_{t(k)},$$

where $a(j), b(j)$, $1 \leq j \leq N$ are the $j - th$ coordinates of the vectors $a_{t(k)}$ and $b_{t(k)}$ and

$$i = arg \min \max_{1 \leq j \leq N} |b(j) - a(j)|.$$

The objective function $f(x)$ is evaluated at the vertices of the two new hyperintervals. In this way we have $\Delta p(k) = 2$ new trial points because $f(x)$ has already been evaluated at the vertices $a_{t(k)}$ and $b_{t(k)}$ of the hyperinterval $D_{t(k)}$ in the previous iterations.

Example 1.2. In Fig 1.15 we have $N = 2$ and the hyperinterval D_t is splitted into 2 new hyperintervals $D_{t(k+1)}, D_{M(k)+1}$. The objective function $f(x)$ has already been evaluated at the vertices $a_{t(k)}$ and $b_{t(k)}$ before splitting (see Fig.

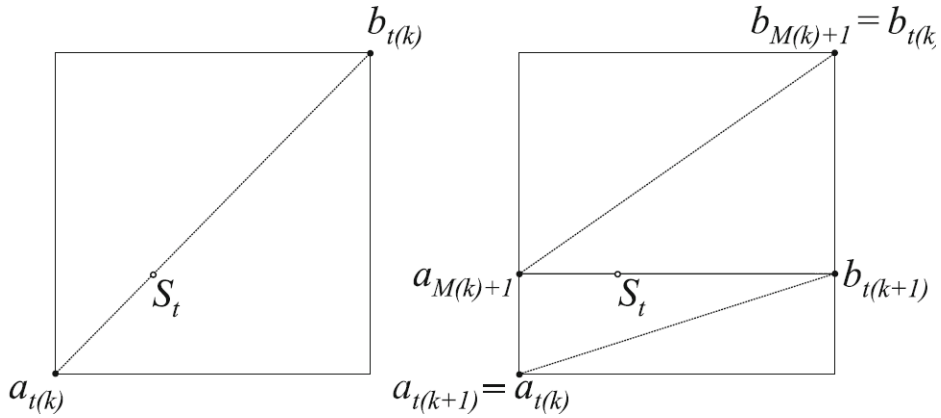

Figure 1.15. Partition of the two-dimensional hyperinterval D_t performed by the Bisection strategy.

1.15, left picture). The new trials are performed at $b_{t(k+1)}$ and $a_{M(k)+1}$ (Fig. 1.15, right picture).

Usually, each evaluation of the objective function $f(x)$ from (1.1) is often a time consuming operation and the aim of the diagonal approach is to reduce the computational efforts needed to solve the global optimization problem. However, we can notice that the 2^N- Partition Strategy performing $2^N - 3$ new function evaluations at each iteration can be computationally very demanding especially when the dimension of the problem N increases. In contrast, the number of trials executed during each iteration by the Bisection Partition does not depend on the problem dimension N. Nevertheless, both strategies perform too many redundant functional evaluations during the search. For example, as we can see from Fig. 1.14, 1.15 they do not ensure the fact that the function $f(x)$ is evaluated at only two vertices of each hyperinterval even if only two trial points are used to calculate the characteristic R_i from (1.2). Another problem is an unnecessary evaluation of $f(x)$ at close points generated during different iterations (in the worst case the objective function is evaluated twice at the same point).

1.2.2 Non-redundant Diagonal Partition Strategy and Adaptive Diagonal Curves

This strategy introduced in [91], (see also [89]) is able to produce a partition where each hyperinterval has exactly two vertices at which the evaluation of $f(x)$ is performed. Moreover it avoids the duplication of trial points. For these reasons it is called *non-redundant* diagonal partition strategy. Let us summarize its general scheme. Without loss of generality, let D be an hypercube of $\mathbb{R}^N$ and denote with $D_t = D_{t(k)} = [a_{t(k)}, b_{t(k)}]$ the hyperinterval chosen for partitioning at the iteration number k.

Step 1. Determine points u abd v as follows:

$$u = (a_t(1), \ldots, a_t(i-1), a_t(i) + \frac{2}{3}(b_t(i) - a_t(i)), a_t(i+1), \ldots, a_t(N)), \tag{1.4}$$

$$v = (b_t(1), \ldots, b_t(i-1), b_t(i) + \frac{2}{3}(a_t(i) - b_t(i)), b_t(i+1), \ldots, b_t(N)), \tag{1.5}$$

with

$$i = arg \min \max_{1 \le j \le N} |b_t(j) - a_t(j)|. \tag{1.6}$$

Evaluate or read from a database, the values of the objective function $f(x)$ at the points u and v, (further details will be given later).

Step 2. Consider $\Delta M(k) = 3$ new equal hyperintervals obtained subdividing D_t by two parallel hyperplanes which are perpendicular to the longest edge i of D_t and passing through the point u and v.

Step 3. Construct the new partition $\{D^{k+1}\}$ replacing D_t with the three new generated hyperintervals $D_{t(k+1)}, D_{M(k)+1}, D_{M(k)+2}$ which are determined by the vertices of their main diagonals:

$$a_{t(k+1)} = a_{M(k)+2} = u, \quad b_{t(k+1)} = b_{M(k)+1} = v,$$

$$a_{M(k)+1} = a_{t(k)}, \quad b_{M(k)+1} = v,$$

$$a_{M(k)+2} = u, \quad b_{M(k)+2} = b_{t(k)}.$$

Set $M(k+1) = M(k) + \Delta M(k) - 1$ and $k = k + 1$.

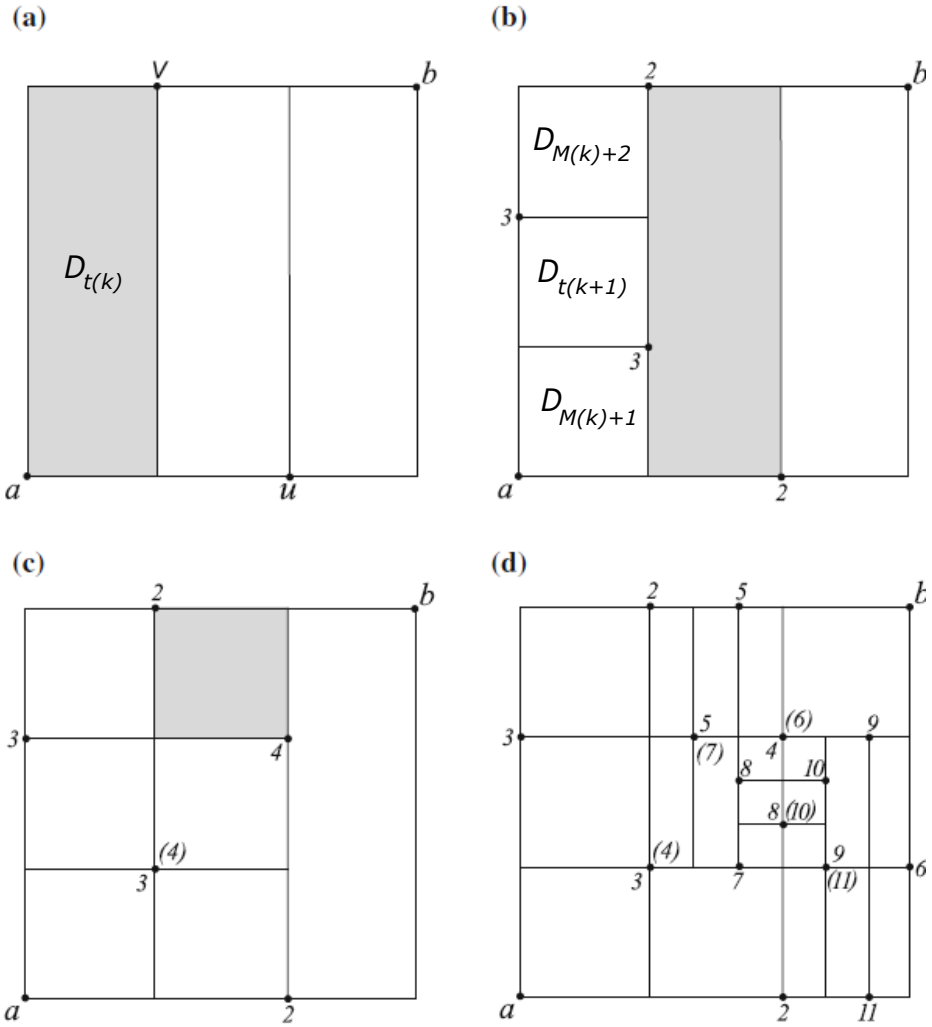

Figure 1.16. Partition of a two-dimensional search interval D executed by a diagonal algorithm using the non-redundant partition strategy. Numbers in brackets indicate the iterations at which the corresponding function values are not evaluated but retrieved from the previous iterations.

Example 1.3. In Fig 1.16 (a), (b) we can see that the hypercube $D_{t(k)}$ contained in $\mathbb{R}^2$ is subdivided into three new hyperintervals $D_{t(k+1)}, D_{M(k)+1}$, $D_{M(k)+2}$ and that the objective function $f(x)$ is evaluated at exactly two vertices for each of these hyperintervals. Supposing now that the hyperinteval shown in gray in Fig 1.16 (b) is chosen for the next partitioning. We can

notice that also in this case the objective function $f(x)$ is evaluated exactly at two points for each hyperintervals of the current partition. In particular the number around the points in brackets, see Fig 1.16 (c) and (d), indicate the iterations at which the corresponding function evaluations are not performed but simply retrieved from the previous iterations.

As shown in [91] the vertices of the generted hyperintervals are produced in order to establish links between hyperintervals having common facets but generated during different iterations avoiding so redundant trial points. In particular the following two results hold.

Theorem 1.2.1. *Every hyperinterval D_t having $f(x)$ evaluated only at the vertices a_t and b_t is split by using the non-redundant diagonal partition strategy into three hyperintervals with exactly two vertices where the function $f(x)$ is evaluated.*

Theorem 1.2.2. *There exists such an indexation of the hyperintervals D_i, obtained in the course of subdivisions by using the non-redundant diagonal partition strategy, that the index of a hyperinterval D_i can be used to calculate the coordinates of its vertices a_i and b_i where the function $f(x)$ is evaluated.*

These results allows to construct specialized databases, presented in details in [48]. Each vertex where the objective function is evaluated can belong to several (up to 2^N) hyperintervals. Moreover, the vertices u and v from (1.4) and (1.5) differ from the vertices a_t, b_t of the hyperinterval D_t chosen for partitioning only in the i-th coordinate, where i is from (1.6). The special indexation of the hyperintervals avoids the need to store the coordinates a_i and b_i for a hyperinterval D_i in the part of memory regarding D_i , since it is possible to calculate these coordinates by knowing the index of D_i. For this reason, in order to construct in an efficient way a *vertex database* which contains the coordinates of the vertices along with the corresponding functional information the data are separated into two parts. The first part is a *list of hyperintervals* which is a *linear list* (see [15] for further details on linear lists), each element of which correspond to a hyperinterval D_i and contains

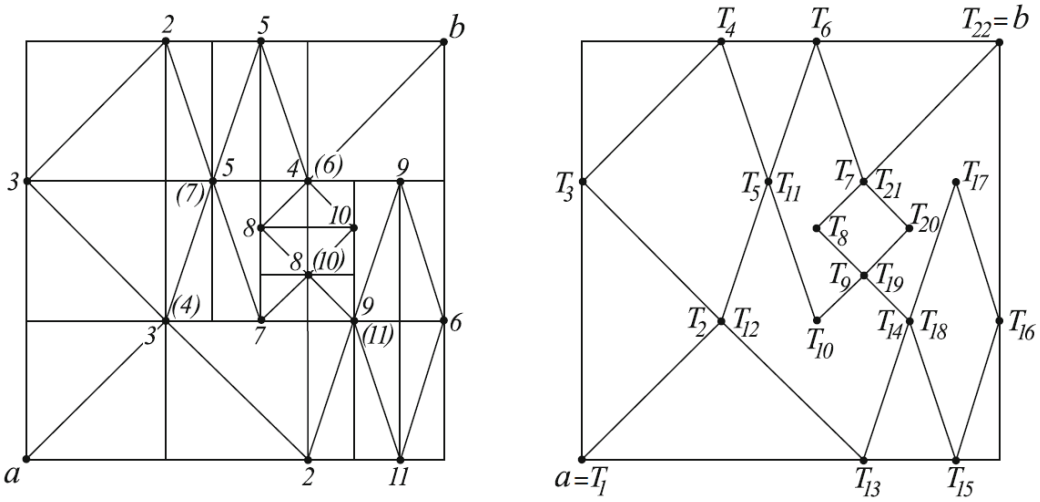

Figure 1.17. Partition of a two-dimensional interval D performed by the nonredundant diagonal partition strategy (left picture) and the corresponding adaptive diagonal curve (right picture). The curve starts at the point $a = T_1$ and connect the points $T_2, T_3, ..., T_{22}$.

its information (characteristic, number of subdivision, etc.) and two pointers A_i and B_i, pointing to the elements of the *set of vertices* which is the second part of the vertex database. The set of vertices contains information regarding the coordinates of the vertices and their functional information. It is organized in N bidirectional different lists, one for each coordinate because in order to check if the points u and v coincide with points at which $f(x)$ has been already evaluated it is sufficient to compare the coordinate $u(i)$ or $v(i)$ with the $i-$th coordinates of the points belonging to the line $x(i) = a_t(i)$ or $x(i) = b_t(i)$ until these values are bounded by $b_t(i)$ or $a_t(i)$. The reason for which each list is bidirectional is that the orientation of the main diagonal of the hyperintervals D_i may be different but the use of $2N$ pointers allows an efficient searching and inserting procedure.

It is interesting to observe how the non-redundant Diagonal Partition Strategy can be also viewed as a process which generates a sequence of curves, hereinafter called *adaptive diagonal curves* which are similar to the sequence

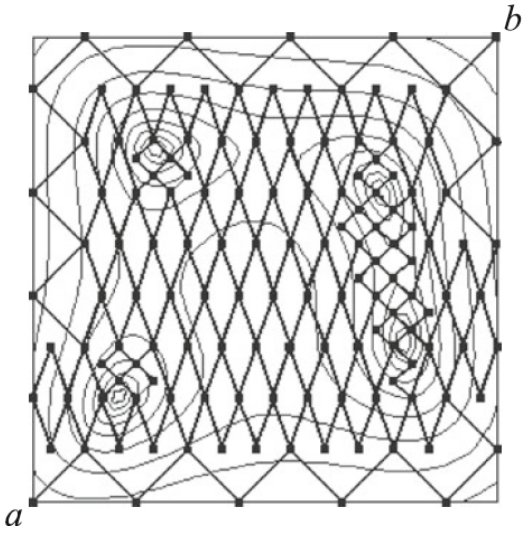

Figure 1.18. Level curves of an objective function $f(x)$ and the diagonal curve generated during its minimization through a diagonal method.

of approximations to the Peano curves considered in Section 1.

These curves are continuous and join the vertices a and b of D. The trace of the first element γ_1 of this sequence of curves simply coincide with the main diagonal of the hyperinterval D. Then, for $k \geq 1$ in order to obtain the trace of γ_{k+1}, the trace of γ_k is considered where the main diagonal $[a_{t(k)}, b_{t(k)}]$ of the hyperinterval $D_{t(k)}$ is replaced with the polygonal line connecting the points $a_{t(k)}, u, v, b_{t(k)}$ with u and v from (1.4) and (1.5). Thus, the two curves γ_{k+1} and γ_k differ only within $D_{t(k)}$. It should be noticed that while in the approximations to Peano curves described in Section 1 there is a uniform level of subdivision over the whole region, in an adaptive diagonal curve, the level of partition within each sub-hyperinterval of D is determined by the characteristic R from (1.2). For this reason, if the selection for partitioning is realized judiciously the subdivision becomes deeper in the vicinity of the global minimizer of the objective function $f(x)$.

In Fig 1.17 (left picture) a partition of an hyperinterval D in the case where $N = 2$ is shown while Fig 1.17 (right picture) shows the corresponding adaptive diagonal curve constructed on this partition. In Fig 1.18 the level

curves of an objective function $f(x)$ and the diagonal curve generated during its minimization through a diagonal method are presented. It can bee seen that the curve condenses in the neighborhood of the four global minimizers.

Chapter 2

Area-filling curves

In this chapter we study area-filling curves, i.e. continuous and injective mappings defined on $[0,1]$ whose graph has positive measure. Current literature calls them "Osgood curves", but their invention is due to H. Lebesgue. Stromberg and Tseng constructed "homogeneous" area-filling curves. We show that an appropriate variant of Knopp's construction attains the same homogeneity result. We will discuss briefly the existence of an "invasive" curve, i.e. a continuous and injective mapping from the half-open interval $[0,1[$ to the unit square, whose image has measure 1. In the last section we discuss several aspects of the Lance-Thomas curve, connecting it with the other construction due to Stromberg and Tseng.

2.1 Early history

This chapter is dedicated to some questions raised by Sagan's entertaining book "Space-filling curves" [86] and the deep and analytically highly non trivial paper [114].
As already seen in Chapter 1, Peano, Hilbert ([77, 42]) and other authors constructed examples of *surjective* continuous mappings $f : [0,1] \to [0,1]^2$ which are called space-filling curves. It is well-known that such curves cannot be injective by a result of Netto [71].

Nowadays, space-filling curves have several useful and surprising applications in global optimization as we will see in the following chapters.
In analogy we shall call *area-filling curves* those continuous and *injective* mappings from $[0,1]$ to $[0,1]^2$ which cover a set of positive measure.
To our knowledge, there are at least ten constructions of area-filling curves. Moreover there are two existence theorems. Lebesgue and Osgood proposed two constructions which are in essence equivalent. The others are due to Sierpiński [113], Knopp [47], Gelbaum and Olmsted (who offered two constructions) [27], Lance and Thomas [55] (with the variant due to Sagan [85] and another one due to Ravishankar and Salas [83]), and two different constructions are due to Stromberg and Tseng [114]. The last paper contains also a non-constructive existence proof suggested to the authors by Sadahiro Saeki. Another interesting non-constructive existence proof was given in [5] by Balcerzak and Kharazishvili using the Denjoy-Riesz theorem. All the results we discuss are variants of the following

Theorem 2.1.1. *Given a regular planar set Q having measure 1, for every $\beta \in]0,1[$ there exists a continuous injective curve $\gamma_\beta : [0,1] \to Q$ such that $\lambda_2(\gamma_\beta([0,1])) = \beta$.*

We indicate with λ_m the m-dimensional Lebesgue measure. In [113] and [47] Q is a triangle, in [57] it is a sector of a circular crown. In all the other constructions Q is the unit square. Sierpiński's construction has the drawback that it does not allow one to choose *an arbitrary* $\beta \in]0,1[$.
Current literature consistently calls such curves "Osgood curves", but we suggest to call them "area-filling curves", since we will show that Lebesgue invented such curves one year before Osgood. On the other hand we cannot call them Lebesgue curves, since there exists already a space-filling curve named after him.

A curve is a continuous mapping from an interval $I \subset \mathbb{R}$ into $\mathbb{R}^m$. Hereinafter I will be a bounded closed interval, except in the last part of this chapter, where topological circumstances will suggest us to take it half-open.

If φ and ψ are two curves defined on $[a,b]$ and $[a',b']$, respectively, with

values in $\mathbb{R}^m$, we will call them *equivalent* if there exists an increasing homeomorphism $s(\cdot)$ from $[a,b]$ to $[a',b']$ such that $\varphi(\cdot) = \psi(s(\cdot))$ for every $t \in [a,b]$.

Sometimes it is convenient to identify curves with their equivalence classes, while in other cases it is important to distinguish between two equivalent curves.

This ambivalent approach can be seen already in the papers which are at the origin of space- and area- filling curves: Peano provides the analytic expression of his curve (and shows no picture), while Hilbert only shows pictures, saying that they lead trivially to their parametric representation.

Area-filling curves are not part of the traditional program offered to students of mathematics in spite of the simplicity of some constructions and the influential mathematicians who contributed to their study. Area-filling curves tend to be forgotten and rediscovered. Sagan reports in [86] the funny anecdote about a well-known topologist who on the pages of the American Mathematical Monthly asked in 1982 (E2975) if there exist Jordan curves with positive Lebesgue measure. Only four years later the proposer himself answered positively the question quoting the popular book of Gelbaum and Olmsted [27], which is on the shelves of every decent mathematical library and has been published more than twenty years earlier.

We think, on the contrary, that they should belong to the permanent display of amusing objects aimed to arouse the interest of students (like the Peano and Hilbert curve, the snowflake curve, the Cantor set and the Cantor function, the Sierpiński triangle, the Möbius strip, Klein's bottle and the rest of the zoo). The origin itself of the area-filling curves is curious. We will discuss their early history. In Section 2.2 we will discuss the *homogeneous constructions* of area-filling curves, a concept introduced in [114]. We will show that an appropriate (and simpler) variant of Knopp's construction also has the homogeneous property. Moreover we will discuss a related question, the existence of a continuous and injective mapping $\varphi : [0,1[\to \mathbb{R}^m$ such that $\lambda_m(\varphi([0,1[)) = 1$. In the last section we will generalize the construction of Lance and Thomas [55] and construct a class of area-filling curves which depend (for any given β) on continuously many parameters. They will be parametrized on symmetric Cantor sets (see [119]) which will turn out to be

homeomorphic to the *essential trace* of the curve.

H. Lebesgue, in his famous thesis [56], considered a measure-theoretic problem which can be reformulated in the following way: *Suppose A is the interior of a Jordan curve. Is A Peano-Jordan-measurable?*
Putting it in other words, the question is if there are Jordan curves whose image (sometimes called trace) has positive Jordan measure.

This was unknown at that time and Lebesgue sets to focus the problem. On page 17 of his thesis (footnote (**)) he writes:

There exist Jordan curves whose image are not Jordan measurable, since there exist curves whose image is a square. To construct an injective non Jordan measurable curve it is sufficient to modify slightly the method used by Hilbert [42] [recte: Peano [77]]. We replace each square which shows up in Hilbert's [Peano's] construction with a polygon contained in the square having a large area and such that the boundaries of two such polygons have at most one point in common, if it exists, through which the curve passes from one to the other.

Osgood [72], one year after the publication of Lebesgue's thesis, constructed independently an example of an injective curve having positive area, based on the construction of the Peano curve.

In the same year [57] Lebesgue returned to his measure-theoretic question, elaborating a particular instance of his general idea and proposed a similar construction, using again Peano's idea.

It is curious that nobody noticed in this 120 years that Lebesgue in [57] (quoted in [114], for instance) refers to [56]. Also the little measure-theoretic question, which motivated it, seems to have been forgotten.

2.2 Homogeneity and invasive curves

Sierpiński and Knopp criticized the fact that in Osgood's construction some arcs have area zero and they proposed two similar constructions which eliminate this drawback, constructing two different examples of area-filling curves γ which have only arcs of strictly positive area. We will call curves with this property *positive* curves. Another positive curve has been constructed by Gelbaum and Olmsted (their first example).

Stromberg and Tseng [114] presented a construction of a positive area-filling curve which has an additional property.

They wrote: "there is a 'homogeneity' feature which is desirable but not guaranteed by Knopp's and Siepiński's construction: given $\beta \in]0,1[$, one may ask that for every subset E of $[0,1]$ the image $\gamma(E)$ have area equal to β times the 'length' of E".

Here we summarize their construction.
Let $\{\varepsilon_n\}$ a sequence of real numbers such that $0 < \varepsilon_n < \frac{1}{6}$ and define a family of *directed* parallelograms (in a directed parallelogram one of the four vertices is the initial point, the diagonally opposite vertex is the terminal point):

$$\mathcal{P} = \{P(n,k) : n \in \mathbb{N}, k = 0,1,2,..,9^n - 1\}.$$

This family is such that each $P(n, 9k+j)$ with $k, j \in \{0,1,..,8\}$ is a subset of $P(n-1,k)$ and the parallelograms in $\{P(n,9k+j)\}$ are pairwise disjoint except that the terminal point of $P(n,9k+j)$ is the initial point of $P(n,9k+j+1)$. Figure 2.1 shows an example of nine directed parallelograms $P(1,k)$, $k = 0,1,\ldots,8$ and the construction of the parallelograms $P(2,j)$ with $j \in \{0,1,..,8\}$ in the interior of the parallelogram $P(1,0)$. In order to define $\gamma : [0,1] \to [0,1] \times [0,1]$, let $\gamma(1) = (1,1)$ and consider any $t \in [0,1[$. For $n \geq 1$, we can find $k_n(t)$ the unique integer such that

$$\frac{k_n(t)}{9^n} \leq t < \frac{k_n(t)+1}{9^n}$$

and since $\{P(n,k_n(t))\}$ is a decreasing sequence of closed sets with diameters tending to 0, we can define $\gamma(t) = \bigcap_{n=1}^{\infty} P(n,k_n(t))$.

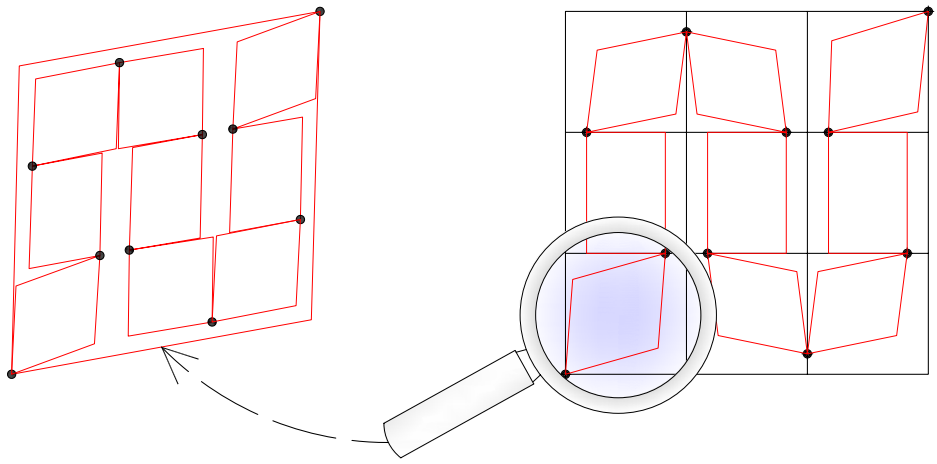

Figure 2.1. Parallelograms $P(1,k)$, $k = 0,1,\ldots,8$ (rigth picture) and parallelograms $P(2,j)$ with $j \in \{0,1,..,8\}$ constructed in the interior of the parallelogram $P(1,0)$ (left picture).

Let $\mathcal{P}_n = \{P(n,k) : k = 0,1,...,9^n - 1\}$, let $C_n = \bigcup \mathcal{P}_n$ and $C = \bigcap C_n$, γ is a homeomorphism of $[0,1]$ onto C and

$$\lambda_2\left(\gamma\left(\left[\frac{h}{9^m}, \frac{h+1}{9^m}\right]\right)\right) = \frac{1}{9^m} \cdot \prod_j^{\infty}(1 - 6\varepsilon_j).$$

So if we choose any strictly decreasing sequence $\{\beta_j\}_{j=0}^{\infty}$ such that $\beta_0 = 1$ and $\beta_j \to \beta$ as $j \to \infty$ we can define $\{\varepsilon_j\}$ by $\varepsilon_j = \frac{1}{6}(1 - \frac{\beta_j}{\beta_{j-1}})$ and γ satisfies the "homogeneity" feature stated by authors.

Remark 2.2.1.
At the first step of the construction the area of each $P(1,i)$, $i = 0,\ldots,8$, is $\frac{1}{9}(1 - 6\varepsilon_1)$ and hence the total area for all 9 directed parallelograms will be $(1 - 6\varepsilon_1)$. At the second step the area of each $P(2,i)$, $i = 0,\ldots,80$, is $\frac{1}{81}(1 - 6\varepsilon_1)(1 - 6\varepsilon_2)$, then the total area of $\cup_i P(2,i)$ is $(1 - 6\varepsilon_1)(1 - 6\varepsilon_2)$. At the $n-$th step the total area of $\bigcup_{i=0}^{9^n-1} P(n,i)$ is $\prod_{j=1}^{n}(1-6\varepsilon_j)$ and for $n \to \infty$ the total area of the directed parallelograms will be $\prod_{j=1}^{\infty}(1 - 6\varepsilon_j)$. So if we choose $0 < \beta < 1$ at pleasure it sufficient let $\{\varepsilon_n\}_n$ any sequence such that $\prod_{j=1}^{\infty}(1 - 6\varepsilon_j) = \beta$.

Remark 2.2.2.
Let us give an alternative proof for the continuity of the function γ.
The set C can be represented as a set of points in base 9, in the sense that we can write $C = \{0, a_1a_2...a_n.. : a_j = 0, 1, ..8\}$ and $z = 0, a_1a_2...a_n.. \in C$ means that

$$z \in P(1, a_1) \cap ... \cap P(n, 9a_{n-1} + a_n) \cap ...$$

We have to observe that some points can be represented in two different ways. For instance, the terminal point of $P(1, 0)$ is the initial point of $P(1, 1)$ and so it allows the following representations:

$$0, 0\overline{8}$$

and

$$0, 1$$

However, we can decide to choose the smaller representation with respect to lexicographical order.
In this way a bijective representation $\varphi : [0, 1] \to C$ is established and φ is continuous.
Let $\varepsilon > 0$ and $x_o = 0, a_1a_2...a_k.. \in [0, 1]$. We know that $\text{diam}P(n, i) < \frac{1}{2^n}$ for each $n \in \mathbb{N}$ and we can find $m \in \mathbb{N}$ such that $\frac{1}{2^m} < \varepsilon$.
If $y_o = 0, b_1b_2...b_k.. \in C$ is such that $a_i = b_i$ with $i \leq m$, then $\varphi(x_o)$ and $\varphi(y_o)$ are in the same parallelogram $P(m, 9a_{m-1} + a_m)$, so

$$|x_o - y_o|_9 < \frac{1}{9^m} \to |\varphi(x_o) - \varphi(y_o)| < \frac{1}{2^m} < \varepsilon.$$

Here it is important to distinguish between curves as functions and curves as equivalence classes as we will see.

But first we want to show that an appropriate variant of the Knopp construction, which seems to us simpler, provides also this homogeneity feature.

Theorem 2.2.3. *For every $\beta \in]0, 1[$ there exists a homogeneous construction of a Knopp curve γ with $\lambda_2(\gamma([0, 1])) = \beta$.*

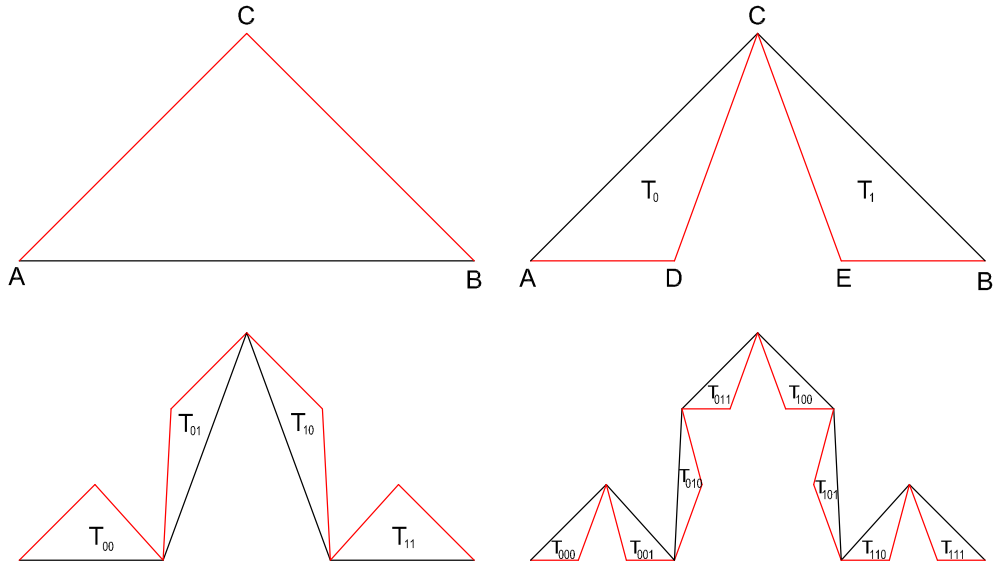

Figure 2.2. First four steps of the generation of Knopp homogeneous area-filling curve.

Proof. Take a triangle $\triangle ABC$ having area 1. At the first step subtract from it a triangle with vertex in C and base DE on the segment AB to obtain two closed triangles $T_0 = \triangle ADC$ and $T_1 = \triangle EBC$ having equal area (details will be given later). See Figure 2.2.

Repeat now this construction subtracting from T_0 a triangle with vertex in D and base a segment on AC, and from T_1 a triangle with vertex in E and base a segment on BC. This can be done obtaining a chain of four triangles (so that they have equal area) T_{00}, T_{01}, T_{10} and T_{11}, which connects A to B.

Iterating this procedure, we get nested chains V_n of triangles having the same area connecting A to B. The indices of $T_{\varepsilon_1\varepsilon_2\ldots\varepsilon_n}$, with $\varepsilon_k \in \{0,1\}$, are chosen so to establish an ordered correspondence between the triangles and the binary n-digit fractions $0.\varepsilon_1\varepsilon_2\ldots\varepsilon_n$.

$V = \bigcap_{n=1}^{\infty} V_n$ is a compact connected set and for any $\beta \in]0,1[$, it is possible to chose (in many ways) the areas of the subtracted triangles so that $\lambda_2(V) = \beta$.

The crucial step in Knopp's construction is to prove that the diameters

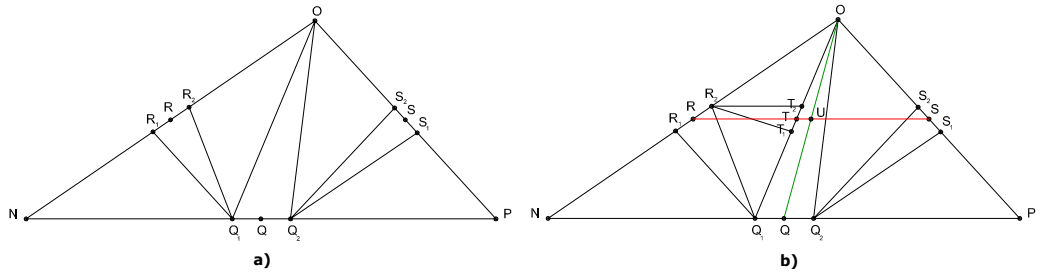

Figure 2.3. Triangle of V_n and construction at $(n+2)$th step (left) and $(n+3)$th step (right).

of the triangles $T_{\varepsilon_1\varepsilon_2\ldots\varepsilon_n}$ tend to zero. This technical step proves that there exists a continuous and injective mapping γ from $[0,1]$ onto V which assigns to every real number written in the binary form $0.\varepsilon_1\varepsilon_2\ldots\varepsilon_n\ldots$ the uniquely determined point $v \in V$ which belongs to the intersection $\bigcap_{n=1}^{\infty} T_{\varepsilon_1\varepsilon_2\ldots\varepsilon_n}$.

Our choice of the triangles is different, so we have to prove that our construction produces triangles whose diameters tend to zero, so we can use the last part of Knopp's proof.

Let $\beta \in]0,1[$ and take a sequence of real numbers $\{r_k\}$ such that $r_k \in]0,1[$ and $\prod_{k=1}^{\infty}(1-r_k) = \beta$.
At the first step let T_0 and T_1 have area $\frac{1-r_1}{2}$.
At the next step let T_{00}, T_{01}, T_{10} and T_{11} have area $\frac{(1-r_1)(1-r_2)}{4}$.
In general, let the 2^n triangles of V_n have area $\frac{1}{2^n}\prod_{k=1}^{n}(1-r_k)$.

Since the sequence $\{r_k\}$ tends to zero, there exists n_0 such that $r_n < \frac{1}{8}$ for $n \geq n_0$. Take an $n \geq n_0$ and consider a triangle in the chain V_n with vertices N, P and O.

We are going to show that in three steps the diameters of the triangles are reduced by a factor at most $\frac{3}{4}$.
Denote by Q and R the midpoints of the segments NP and NO respectively. The triangle $\triangle NQR$ (see Figure 2.3 a)) is similar to $\triangle NPO$ with similarity ratio $\frac{1}{2}$. Since $\triangle NQ_1R_1 \subset \triangle NQR$, this triangle and all the triangles of the successive chains will have diameter less than $\frac{1}{2}$ times the diameter of $\triangle NPO$. The same argument applies to $\triangle Q_2PS_1$.

Let us now turn to $\triangle R_2Q_1O$ and the operations of the next step (see

Figure 2.3 b)). $\triangle R_2Q_1O$ generates $\triangle R_2T_2O$ and $\triangle R_2T_1Q_1$. The first one is contained in $\triangle ORS$ and its diameter is smaller than one half of the diameter on $\triangle NPO$. $\triangle R_2Q_1T_1$ offers greater resistance. But it is easy to see that it is contained in the image of $\triangle NPO$ under a homothety which shrinks it by a factor $\frac{3}{4}$ and keeps N fixed. This is due to the fact that $r_n < \frac{1}{8}$.

Now Knopp's argument applies and it follows that V is the image of a continuous injective mapping.

Let us prove now the homogeneity property of γ. Let $T = T_{\varepsilon_1 \dots \varepsilon_n}$ be a triangle of the chain V_n and denote by X and Y its vertices which connect it to the remaining part of the chain. Possibly $X = A$ or $Y = B$. The points A, X, Y and B are in that order on V. We have that $\{X\} = \bigcap_{m=0}^{\infty} T_{\varepsilon_1 \dots \varepsilon_n 0 \dots 0}$, where ε_n is followed by m 0's. Therefore $X = \gamma(0.\varepsilon_1 \dots \varepsilon_n)$.

On the other hand $\{Y\} = \bigcap_{m=0}^{\infty} T_{\varepsilon_1 \dots \varepsilon_n 1 \dots 1}$, where ε_n is followed by m 1's. Therefore $Y = \gamma(0.\varepsilon_1 \dots \varepsilon_n \bar{1})$.

It follows that the interval $I \subset [0,1]$, mapped by γ onto the arc $\overset{\frown}{XY}$, has length $\frac{1}{2^n}$.

On the other hand T has area

$$\prod_{i=1}^{n} \frac{(1-r_i)}{2} = \frac{1}{2^n} \prod_{i=1}^{n} (1-r_i)\,.$$

The 2^m triangles belonging to V_{n+m} and contained in T have total area

$$2^m \frac{1}{2^{n+m}} \prod_{i=1}^{n+m} (1-r_i) = \frac{1}{2^n} \prod_{i=1}^{n+m} (1-r_i)\,.$$

Taking the limit for $m \to \infty$, it follows that

$$\lambda_2(\overset{\frown}{XY}) = \frac{1}{2^n} \prod_{i=1}^{\infty} (1-r_i) = \beta \frac{1}{2^n} = \beta \lambda_1(I)\,.$$

The equation $\lambda_2(\gamma(E)) = \beta \lambda_1(E)$, for every measurable set E contained in $[0,1]$, follows now from the density of binary intervals and their finite unions among measurable sets. □

We may speak about a homogeneous parametrization of a curve. But curves alone cannot be considered homogeneous and we will show that any

positive curve can be reparametrized so to get a homogeneous parametrization.

Theorem 2.2.4. *Let γ be a positive area-filling curve parametrized on $[0,1]$ (with $\lambda_2(\gamma([0,1])) = \beta$). Then there exists an increasing homeomorphism h from $[0,1]$ to $[0,1]$ such that $\gamma_1 = \gamma \circ h$ is homogeneously parametrized.*

Proof. The function $h^{-1}(t) = \beta^{-1}\lambda_2(\gamma([0,t]))$ is an increasing homeomorphism from $[0,1]$ to $[0,1]$. Let us denote its inverse with h.

Since $h^{-1}(h(s)) = s = \beta^{-1}\lambda_2(\gamma([0,h(s)])$, we have

$$\lambda_2(\gamma_1(s)) = \lambda_2(\gamma([0,h(s)]) = \beta\, s\,.$$

as required. □

On the other hand, if we take the homogeneous representation of the Knopp curve just constructed and consider the increasing homeomorphism $s(\cdot) : [0,1] \to [0,1]$ $s(t) = t^2$, we notice that $\gamma_1(t) = \gamma(t^2)$ is such that $\lambda_2(\gamma_1([0,\frac{1}{2}])) = \frac{1}{4}\beta$ and $\lambda_2(\gamma_1([\frac{1}{2},1])) = \frac{3}{4}\beta$, so γ_1 is not homogeneously parametrized.

The natural question whether there exists an injective continuous curve γ with image in $[0,1]^2$ such that $\lambda_2(\gamma([0,1])) = 1$ has an immediate negative answer: $\gamma([0,1])$ is compact and the only compact subset of $[0,1]^2$ with measure 1 is the square itself. But this is excluded by Netto's result.

We will mention here two results related to this question which can be deduced or found in [114] and [39], respectively.

Sadahiro Saeki contributed to [114] with a theorem which is not constructive, since it uses the deep result due to von Neumann, Oxtoby and Ulam, the Homeomorphic Measures Theorem (HMT) (see [73] and [32]). He proved just the existence of non-intersecting area-filling curves. Looking at his proof, however, it is easy to draw the following (non-constructive) conclusion.

Theorem 2.2.5. *There exist sequences $\{\gamma_n\}$ of non-intersecting injective curves defined on $[0,1]$ such that $\sum_{n=1}^{\infty} \lambda_2(\gamma_n([0,1])) = 1$.*

A result which gets closer to an answer to the question posed at the beginning of the section is contained in [39]. The Authors do not state it explicitly, but it can be found in the discursive part of their article (p. 176, lines -6, -5):

Theorem 2.2.6. *There exists a continuous and injective mapping γ from $[0,1[$ to $[0,1]^2$ such that $\lambda_2(\gamma([0,1[)) = 1$.*

It is an easy consequence of their nice main result, the Thread Theorem.

2.3 Lance-Thomas curves

In the introduction of this chapter we mentioned non constructive existence proofs. One is contained in [5]. It is not explicitly stated, but is described in the text of the proof of their Theorem 2.2. It uses the Denjoy-Riesz theorem, which asserts that

Theorem 2.3.1. *Any bounded totally disconnected set in the plane can be covered by a Jordan arc.*

The existence of an area-filling curve follows easily from the previous result: fix $\beta > 0$, take a planar totally disconnected set of measure β contained in $[0,1]^2$ and cover it with a Jordan arc. So one obtains an area-filling curve of measure at least β (since the Jordan arc may contain other area-filling arcs). The easiest way of producing a totally disconnected compact set in the plane of positive measure is to take a Cantor set $P \subset [0,1]$ of positive measure and to consider then $P \times P$. Having the Denjoy-Riesz theorem in mind, the proofs of Lebesgue, Osgood and the second construction by Gelbaum and Olmsted can be immediately related to this scene, but there is no doubt that the direct inspiration came from Peano's and Hilbert's set.

It should be noted however that the first of the two constructions from [114] can also be related to this trace: fixed a $\beta > 0$, they let $\alpha = \sqrt{\beta}$ and constructed a symmetric Cantor set $P \subset [0,1]$ having measure α. Then they exhibited explicitly segments which connect points of $P \times P$ and form a Jordan arc covering $P \times P$. Its measure is β, since the segments have

measure zero. The construction is explicit, but very complicated (it takes pages 34-43) and it does not present any evident symmetry. They do not provide any picture of their set and also our attempts to obtain a suggestive figure failed.

It should be noted that [5] has been published five years later, so there is no direct connection between the two papers, and we just imagine how the authors have argued.

We will now make a change to the construction of the Lance-Thomas curve to fit into this line of thought. Let $\beta \in]0,1[$ and consider a sequence $\{a_n\}$ of real numbers, called the reduction coefficients, such that $0 < a_n < 1$ and $\lim_{n\to\infty}(a_1 \cdot a_2 \cdot \cdots \cdot a_n)^2 = \beta$.

Our construction of the Lance-Thomas curve differs from the previous ones ([55], [86] and [85]) by the parametrization.
Note that Sagan parametrized it on the ternary Cantor set and did not follow [55] who simply divided $[0,1]$ into seven equal parts.

To construct the first approximation, consider (following the original idea) four disjoint squares, having side length $\frac{a_1}{2}$, placed at the corners of $[0,1]^2$ and joined by three segments (joints) as in Figure 2.4. The curve γ_1 is obtained adding three joints to the the diagonals of the squares to get a polygonal line which connects the left lower corner with the right upper corner of the square. Denote by A_1 the union of those four closed squares and these joints.

We choose to subdivide $[0,1]$ into seven subintervals so that the first, the third, the fifth and the seventh have length $(\frac{a_1}{2})^2$. Denote by C_2 the union of these four intervals. The second and sixth intervals have length $\frac{a_1}{2}(1-a_1)$, while the central has length $(1-a_1)$. The function γ_1 maps linearly the even intervals (second, fourth, sixth) in the segments that connect the squares of A_1 and the remaining subintervals in the natural order into the diagonals of the squares as shown in Figure 2.4.

There is no mystery in the choice of the length of the intervals: they are equal to the areas of the seven rectangles crossed by the seven parametrized segments.

The curve γ_2 is constructed analogously, operating now in the four squares of the first generation using the reduction coefficient a_2. Denote by A_2 the

set made up of sixteen squares obtained putting into each of the four squares of A_1 four smaller squares, placed at the vertices, of side length $\frac{a_1}{2}\frac{a_2}{2}$ and their junctions as in Figure 2.4.

The four intervals of C_2 are subdivided as in the first step using the reduction coefficient a_2. Each interval of C_2 is subdivided into seven parts: the first, third, fifth and seventh have length $(\frac{a_1}{2}\frac{a_2}{2})^2$.

Let us denote by C_4 the union of these four disjoint closed subintervals. The second and sixth have length $(\frac{a_2}{2})^2 \cdot \frac{a_2}{2}(1-a_2)$ and the fourth has measure $(\frac{a_2}{2})^2 \cdot (1-a_2)$.

The function γ_2 maps linearly the intervals of C_4 into the diagonals of the A_2 squares and the remaining subintervals are mapped in the junction segments, as in Figure 2.4, forming so a connected polygonal.

We iterate the construction, putting into each of the squares of A_{n-1} four squares and three junctions as in step one and two, using at each step the reduction factor a_n. Iterating, we have $\lambda_2(A_n) = (a_1 \cdot a_2 \dots a_n)^2$ and the set C_{2n} (which is the $2n^{th}$ step of the construction of a symmetric Cantor set C) consists of 4^n disjoint closed intervals having one-dimensional measure equal to the two-dimensional measure of the rectangle having as diagonal the segment in which the interval is mapped, so $\lambda_1(C_{2n}) = (a_1 \cdot a_2 \dots a_n)^2$.

Note that $\gamma_{n+1} = \gamma_n$ on the complement of C_{2n}. On the other hand, for every k and each $x \in C_{2n}$, $||\gamma_{n+k}(x) - \gamma_n(x)|| \leq \left(\frac{a_1}{2} \cdot \frac{a_2}{2} \cdot \dots \cdot \frac{a_n}{2}\right) \cdot \sqrt{2}$, which tends to 0. Thus the sequence $\{\gamma_n\}$ converges uniformly to a continuous curve γ such that

$$\lambda_2(\gamma([0,1])) = \lambda_2\left(\bigcap_n A_n\right) = \lim_{n\to\infty}(a_1 \cdot a_2 \dots a_n)^2 = \beta\,.$$

Remark 2.3.2. If we denote with B_n the set which consists of the union of squares of A_n for each n, the restriction of the limiting curve γ to C is a homeomorphism between C and $B = \cap_n B_n$ that preserves the measure. We call the compact set B the *essential image* of $[0,1]$ under γ.

It is easy to see that for any Borel set $E \subset [0,1]$, $\lambda_2(\gamma(E \cap C)) = \lambda_1(E \cap C)$, so our parametrization is sort of a "homogeneous" one.

Note also that B is a Cartesian product of two symmetric Cantor sets

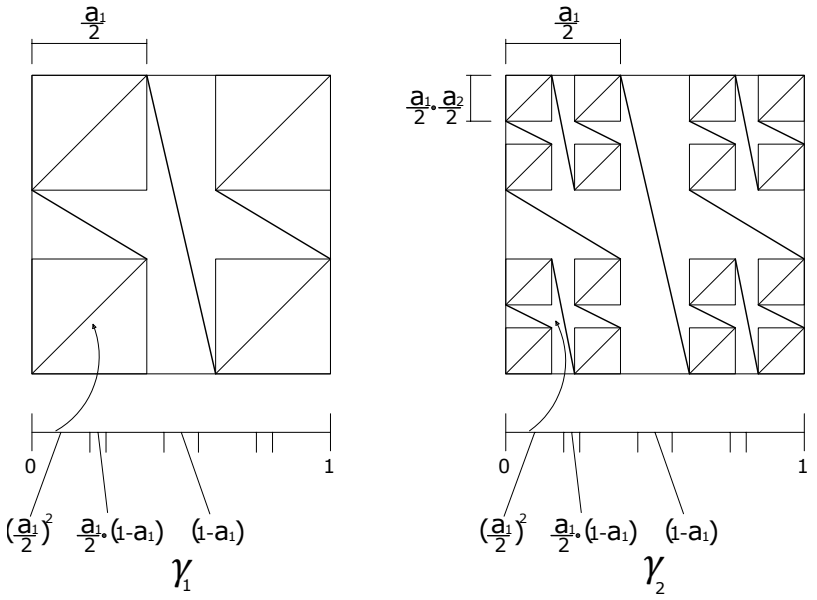

Figure 2.4. Construction of γ_1 and γ_2.

contained in $[0,1]$ and that the curve γ is a Jordan arc containing B. So this construction is an illustration of the Riesz-Denjoy theorem.

The previous construction can also be generalized to higher dimensions.

Let us start with dimension $N = 3$ and then we will proceed in the technically demanding case $N > 3$.

Fix $\beta \in \ ]0,1[$ and consider a sequence of reduction coefficients a_n, such that $0 < a_n < 1$ and

$$\lim_{n\to\infty} (a_1 \cdot a_2 \cdot ... \cdot a_n)^3 = \beta. \tag{2.1}$$

Let us construct 8 disjoint congruent cubes S_i with $i = 1, .., 8$, having side length $\frac{a_1}{2}$, placed at the corners of $[0,1]^3$. The first approximation is defined as follows: divide $[0,1]$ into 15 subintervals. The first, the third,.., the fifteenth interval (the "odd" ones) have length $(\frac{a_1}{2})^3$. The second, sixth, tenth and fourteenth interval have length $(\frac{a_1}{2})^2(1-a_1)$ and finally the central interval has length $(1-a_1)$.
The function γ_1 maps linearly the interval $[0, (\frac{a_1}{2})^3]$ in the diagonal of the cube S_1 starting from the point $(0,0,0)$, the second interval is linearly mapped into

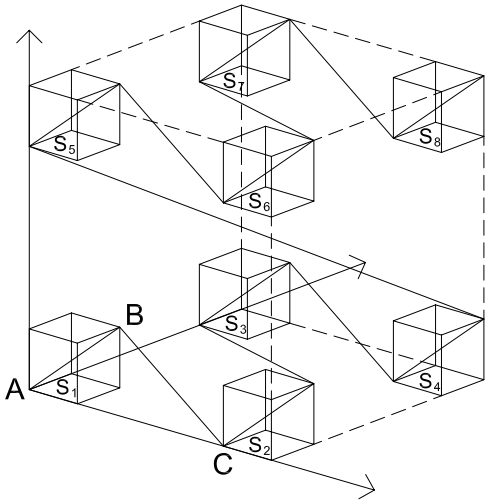

Figure 2.5. Construction of γ_1.

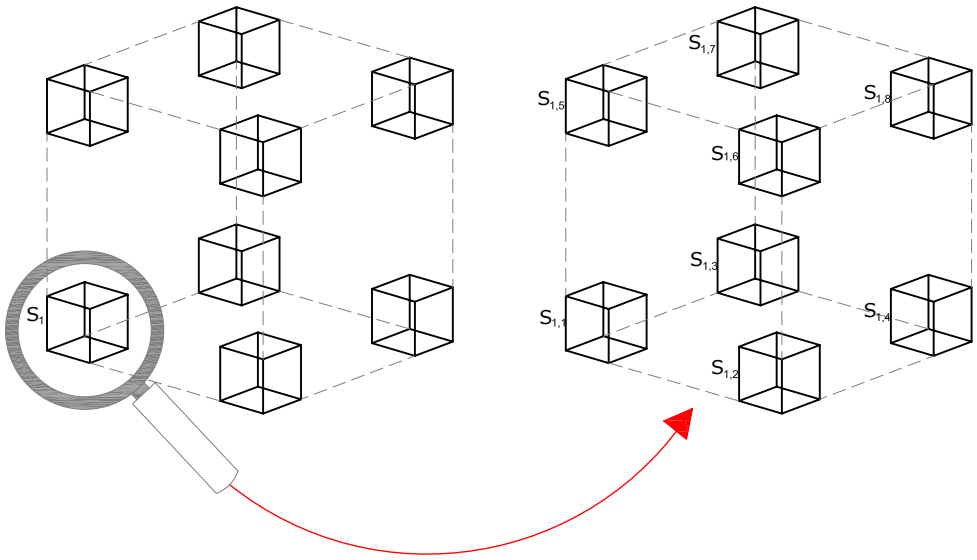

Figure 2.6. Construction of cubes $S_{i,j}$.

the segment that joins the point B with the point C as in Figure 2.5, the third interval into the diagonal of S_2 starting from the point C and so on.

So the odd intervals are mapped, in increasing order, into the diagonals of the cubes S_{i+1} for $i = 0, 1, .., 7$, while the remaining seven subintervals are

sent into the segments joining the cube S_i with the cube S_{i+1}. In this way, each interval has length equal to the volume of the parallelepiped having as diagonal the segment in which the interval is mapped.
In order to define $\gamma_2 : [0,1] \to [0,1]^3$, in each S_i we construct 8 cubes $S_{i,j}$ $j = 1,..,8$ having side length equal to $\frac{a_1}{2}\frac{a_2}{2}$, placed at the corners of S_i, along with their joints (see Fig. 2.6).
The interval $[0,1]$ is divided in such a way that the intervals mapped by γ_1 in the junctions remain unaltered and on these intervals γ_2 coincides with γ_1. The remaining part is divided as done in the previous step using the reduction coefficient a_2.

The curve γ_2 consists of the diagonals of the cubes $S_{i,j}$ and the segments which connect them, so that in each S_i we have a polygonal line which is analogous to the graph of γ_1 constructed in the previous step. Again, each interval has length equal to the volume of the corresponding parallelepiped. Iterating, we construct the sequence $\{\gamma_n\}$, considering at each step cubes having side length $\frac{a_1}{2}\frac{a_2}{2}..\frac{a_n}{2}$. The uniform convergence of $\{\gamma_n\}$ is proved as in the previous case.
Let us denote by V_n the union of the 8^n cubes constructed at the n-th step. From (2.1), we have that

$$\lambda_3(\gamma([0,1])) = \lim_{n\to\infty} \lambda_3(V_n) = \lim_{n\to\infty} \frac{(a_1 a_2..a_n)^3}{2^{3n}} \cdot 2^{3n} = \beta.$$

Remark 2.3.3. A possible choice for the sequence $\{a_n\}$ is $a_n = \beta^{\frac{1}{2^n \cdot 3}}$.
Moreover, as in the previous considerations, the limit curve γ is a homeomorphism between a symmetric Cantor set C and $V = \bigcap_n V_n$ that preserves the measure.

The construction can be generalized to any dimension $N \geq 2$. It is possible to define a sequence $\gamma_n^N : [0,1] \to [0,1]^N = Q_N = Q_{N-1} \times [0,1]$, so that the curve $\lim_{n\to\infty} \gamma_n^N =: \gamma^N$ is such that $\lambda_N(\gamma^N([0,1])) = \beta$, with $\beta \in]0,1[$.
Let us consider for example, the reduction sequence $\{a_n^N\} = \beta^{\frac{1}{2^n \cdot N}}$ and we proceed by recursion. For $N = 2$ we have the previous two-dimensional area-

filling curve with respect to the sequence $\{a_n^2\} = \beta^{\frac{1}{2^{n+1}}}$. For $N \geq 3$, suppose we have constructed the sequence γ_n^{N-1} and consider 2^N Cartesian hypercubes of side $\frac{a_1^N}{2}$ placed at the 2^N corners of the hypercube $[0,1]^N$. The first approximation of N-dimensional area-filling curve consists of two curves and a segment that connects them. Both curves have as projection on $[0,1]^{N-1}$ the first approximation of $(N-1)$-dimensional area-filling curve constructed considering the sequence $\{a_n^N\}$, instead of the sequence $\{a_n^{N-1}\}$.
The first curve starts from $(0,0,..,0)$ and continues in a zigzag pattern bouncing between the "floor" of $[0,1]^N$ ($x_N = 0$) and the "ceiling of the first level" ($x_N = \frac{a_1^N}{2}$). In analogous way, the second curve bounces between the floor of the second level ($x_N = 1 - \frac{a_1^N}{2}$) and the ceiling of $[0,1]^N$ ($x_N = 1$).
The segment joining the points $(1,1,...,1,\frac{a_1^N}{2})$ and $(0,0,...,0,1-\frac{a_1^N}{2})$ connects the two "floors".
In order to parameterize γ_1^N, $[0,1]$ is divided in $2^{N+1}-1$ intervals such that each interval has one-dimensional measure equal to the N-dimensional measure of the parallelepiped having as diagonal the segment in which the interval is mapped.
At the second step each of the 2^N hypercubes is "filled" by a transformation which is analogous to that of the first step but using a different reduction coefficient.
Outside these hypercubes, the construction of the second approximation of the area-filling curve is the same of the first one and the interval $[0,1]$ is divided in such a way that the intervals mapped by γ_1^N in the joints remain unchanged, while for the remaining part we proceed as done in the previous step using the reduction coefficient a_2^N.
Denote by V_n the union of the 2^{Nn} cubes which are constructed at the n^{th} step. We observe that

$$\lambda_N(V_n) = \frac{\left(a_1^N a_2^N ..a_n^N\right)^N}{2^{Nn}} \cdot 2^{Nn} = \left(a_1^N a_2^N ..a_n^N\right)^N .$$

So $\lambda_N(V_n)$ tends to β, having chosen $\{a_n^N\} = \beta^{\frac{1}{2^{n \cdot N}}}$. Furthermore also in this case, the limiting curve γ is a homeomorphism between a symmetric Cantor set C and $V = \bigcap_n V_n$ that preserves the measure.

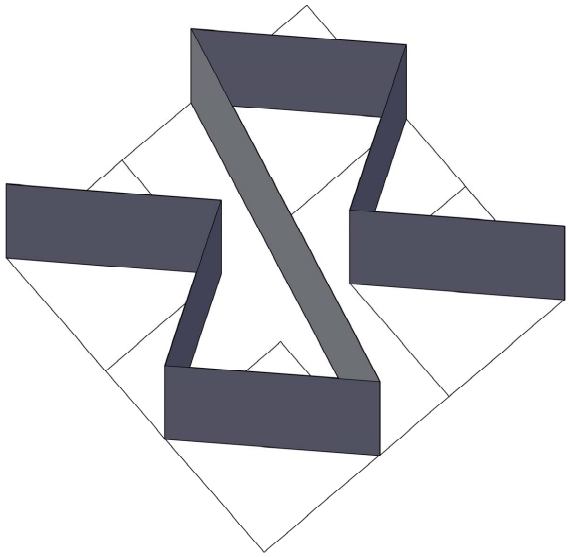

Figure 2.7. First approximation of an area-filling surface.

Using these constructions we can also obtain the following result.

Theorem 2.3.4. *For any $\beta \in]0,1[$ and any pair of integers $N < M$ there exists a continuous and injective mapping $g_{N,M} : [0,1]^N \to [0,1]^M$ such that $\lambda_M(g_{N,M}([0,1]^N)) = \beta$.*

This kind of application can be called *area-filling surface*.

We can define this application as follows. We indicate with $g_N : [0,1] \to [0,1]^N$, the area-filling curve constructed as before such that $\lambda_N(g_N([0,1])) = \beta$ and let

$$g_{N,M}(x_1, x_2, ...x_N) = (g_{M-N+1}(x_1), x_2, ..., x_N).$$

We observe that $g_{N,M}$ can be seen as Cartesian product of functions:

$$g_{N,M} : [0,1] \times [0,1]^{N-1} \to [0,1]^{M-N+1} \times [0,1]^{N-1}$$

$$g_{N,M}(t,x) := (g_{M-N+1}(t), Id(x)).$$

Since g_{M-N+1} and the identity function are injective and continuous also $g_{N,M}$ is injective and continuous.
Moreover

$$\lambda_M(g_{N,M}([0,1]^N)) = \lambda_M(g_{M-N+1}([0,1])) \cdot \lambda_M([0,1]^{M-1}) = \beta \cdot 1 = \beta.$$

Chapter 3

Global one-dimensional Lipschitz optimization

In this chapter univariate box-constrained global optimization problems are considered. In particular, two Lipschitz Global Optimization Problems are discussed. In the first problem the objective function satisfies the Lipschitz condition while, in the second one, the first derivative of the objective function satisfies the Lipschitz condition. In both problems it is assumed that the function to be optimized is multiextremal over the search interval, its analytical representation is unknown (the function is given as a "black-box") and even one of its evaluation is a computationally expensive procedure. For this reasons developing acceleration techniques is of great interest. For the first problem we will discuss geometric and information statistical frameworks for construction of global optimization algorithms, for the second one smooth piece-wise quadratic support functions are used in the presented methods which are implemented both in the traditional floating-point arithmetic and in the Infinity Computing framework allowing one to efficiently compute exact derivatives in the case where the optimized function is given as black box. In each framework we will describe several Local Tuning techniques which are able to estimate local Lipschitz constants in different areas of the search region allowing to accelerate the search. Convergence conditions of the methods are established. Finally, we discuss numerical experiments pre-

senting advantages of these techniques.

Global optimization problems characterized by the presence of numerous local minima and maxima arise in many real-life applications. It is sufficient to mention such fields as engineering, electronics, machine learning, optimal decision making, computational biology, etc. (see, e.g., [1, 12, 16, 24, 53, 54, 68, 81, 96, 97, 123] and references given therein). Very often the function to be optimized is characterized by a very high time required to execute each evaluation obtained by running some codes, i.e., an analytical representation of the objective function is not available. In these cases the optimization is called *black-box* and *expensive* (see, e.g., [12, 75, 96, 106]). The necessity to find the best (in other words, *global*) solution in the situation where a high number of local extrema is present explains the continuously increasing interest of researchers to global optimization algorithms looking for global minimum (or maximum). One of the important methodologies developed to attack this problem is Lipschitz global optimization (see, e.g., [29, 31, 37, 41, 52, 53, 81, 82, 115, 118]). It uses a natural assumption on the global optimization problem supposing that the objective function under consideration has bounded slopes, in other words, it satisfies the Lipschitz property and in the same moment it can be multiextremal and each evaluation can be a very time-consuming operation. An important subclass in Lipschitz global optimization consists of functions with the first derivative satisfying the Lipschitz condition (see [6, 10, 16, 28, 52, 63, 90, 118], etc.).

Problems belonging to Lipschitz global optimization are extremely difficult even in the one-dimensional case and are under an intensive study at least for two reasons. First, there exists a huge number of applications where problems of this kind arise (see, e.g.,[9, 10, 16, 40, 88, 92, 96, 97, 98, 99, 104]). The second reason is that one-dimensional schemes are broadly used for constructing multi-dimensional global optimization methods (see, e.g., [53, 75, 81, 92, 98, 99, 118], etc.).

An additional practical difficulty consists in the fact that it is supposed that there exists a code for computing $f(x)$ only and a code for the derivative $f'(x)$ is not available.

In this chapter, our attention is devoted to the following two global optimization problem.

Let $D = [a, b]$ be a real interval and $f : D \to \mathbb{R}$ a black-box function. The aim is to find a point x^* belonging to D and the value $f^* = f(x^*)$ such that

$$f^* = f(x^*) = \min f(x), \quad x \in D, \tag{3.1}$$

where either the objective function $f(x)$ or its first derivative $f'(x)$ satisfy the Lipschitz condition, i.e., either

$$|f(y) - f(x)| \leq L|(x - y)|, \quad x, y \in D, \tag{3.2}$$

or

$$|f'(y) - f'(x)| \leq M|(x - y)|, \quad x, y \in D, \tag{3.3}$$

with Lipschitz constants $0 < L < \infty$, $0 < M < \infty$.

Moreover, in Lipschitz global optimization (see, for example, [52, 53, 81, 118]) the usage of the global (i.e., the same for the whole search region D) Lipschitz constant or its global estimate can slow down the search. In order to overcome this difficulty, a number of local tuning techniques automatically controlling the exploitation-exploration trade-off have been proposed in [31, 36, 49, 63, 88, 90, 107]. These techniques adaptively estimate *local* Lipschitz constants over different subregions of D constructing auxiliary functions that are closer to $f(x)$ with respect to those using the global Lipschitz constant (or its estimates). As a result, methods using local tuning techniques are significantly faster than algorithms working with the global Lipschitz constant.

The present chapter illustrate a number of contributions for acceleration of the global search with a special attention to the situation where a code for computing the derivative $f'(x)$ is not available.

In Section 3.1 we describe algorithms for solving the problem (3.1), (3.2) while in Section 3.2 we deal with the problem (3.1), (3.3) through methods which use piece-wise quadratic support functions. For both problems numerical results are presented and discussed. In particular, in Section 3.3

the Infinity Computing framework is also used in order to compute exact (up to the machine precision) derivatives during the work of the algorithms presented.

3.1 Derivative-Free Univariate Lipschitz Global Optimization Methods

The fact that in the problem (3.1),(3.2) the objective functions are often non-differentiable explains the continuous interest of researches in derivative-free univariate Lipschitz global optimization methods (see, e.g., [8, 41, 88, 90, 99, 102, 105, 110]).

Thus, let us consider geometric and information statistical classes of algorithms which have their origins in the methods of Piyasvskij (see [82]) and Strongin (see [115]), respectively. These methods have been chosen for this study because they have shown their efficacy on several classes of problems (see, e.g., [81, 88, 90, 102, 110, 118]) and it is also known that they can be improved with some powerful acceleration techniques (see [50, 87, 88, 102]). These two classes of methods have a different nature: Piyasvskij's method requires the knowledge of an overestimate of the Lipschitz constant L and uses geometric ideas based on the Lipschitz property whereas in the Information approach, introduced by Strongin, an adaptive estimate of L calculated during the search is used in a statistical model.

If a one dimensional Lipschitz function $f(x)$ has been evaluated at two points x' and x'' from the Lipschitz condition:

$$|f(x') - f(x'')| \leq L|x' - x''|, \quad x', x'' \in D,$$

the following four inequalities take place,

$$f(x) \leq f(x') + L(x - x'), \quad f(x) \geq f(x') + L(x - x') \quad x \geq x';$$

$$f(x) \leq f(x'') + L(x - x''), \quad f(x) \geq f(x'') + L(x - x'') \quad x \geq x''.$$

Thus, the function graph over the interval $[x', x'']$ lies inside the area limited by four lines with the slopes L as in Figure 3.1.

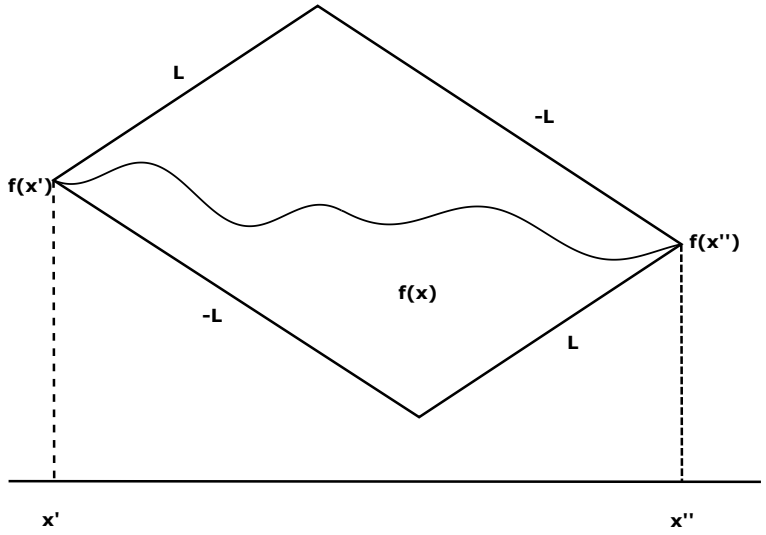

Figure 3.1. The function graph over the interval $[x', x'']$ lies inside the parallelogram.

Thus the main idea of geometric algorithms is to construct a minorant $\varphi_i(x)$ for $f(x)$ over each subinterval $[x_{i-1}, x_i]$ of the search region D, where x_i, $i = 1, \ldots, k$, are so-called *trial* points, i.e., points where the values $z_i = f(x_i)$ have been evaluated. If we suppose that $\hat{L}$ is an overestimate of L, then it follows

$$f(x) \geq \varphi_i(x) = \max\{z_{i-1} - \hat{L}(x - x_{i-1}), z_i + \hat{L}(x - x_i)\}, \;\; x \in [x_{i-1}, x_i],$$

and the minimal value of $\varphi_i(x)$, $x \in [x_{i-1}, x_i]$, denoted by R_i, is called *characteristic* of the interval $[x_{i-1}, x_i]$ (see, e.g., [99]),

$$R_i = 0.5(z_{i-1} + z_i - \hat{L}(x_i - x_{i-1})). \tag{3.4}$$

In contrast, the information approach uses the Bayesian ideas and considers the objective function from a stochastic point of view. The characteristic R_i of the Strongin information algorithm associated to each subinterval $[x_{i-1}, x_i]$ is

$$R_i = 2(z_i + z_{i-1}) - \overline{L}_k(x_i - x_{i-1}) - (z_i - z_{i-1})^2(\overline{L}_k(x_i - x_{i-1}))^{-1}, \tag{3.5}$$

where $\overline{L}_k$ is an adaptive estimate of the global (i.e., valid for the whole search region D) Lipschitz constant L during the search

$$\overline{L}_k = r \cdot \max\{H^k, \xi\}, \tag{3.6}$$

$$H^k = \max\{H_i : i = 2, \ldots, k\}, \quad H_i = |z_i - z_{i-1}|/(x_i - x_{i-1}), \tag{3.7}$$

and $\xi > 0$ is a small technical parameter allowing the correct work of the method.

It has been shown in [87, 88, 90, 107] that the usage of global estimates of L can slow down the search significantly. However, for both the methodologies, geometric and information, the so-called *local tuning approach* introduced in [87, 88] can be used to accelerate the global search. It allows one to tune the behavior of the algorithm according to behavior of the objective function at different subintervals using adaptive estimates of the local Lipschitz constants. In fact, when subinterval $[x_{i-1}, x_i]$ is narrow, only the local information near trial points x_{i-1}, x_i has a decisive influence on the method. In contrast, when the subintervals is wide, the local information becomes less reliable.

In order to introduce the local tuning techniques let us denote as $\{x_i\}_1^k$ the ordered trial points and $k \geq 2$ the number of iterations of the algorithm (for $k = 2$: $x_1 = a$ and $x_2 = b$). Let $r > 1$ the reliability parameter. In the local tuning approach, we compute estimates l_i of local Lipschitz constants for each interval $[x_{i-1}, x_i], i = 2, \ldots, k$, in one of the three following procedures:

1. "Maximum" Local Tuning

$$l_i = r \cdot \max\{\lambda_i, \gamma_i, \xi\}, \tag{3.8}$$

$$\lambda_i = \max\{H_{i-1}, H_i, H_{i+1}\}, \quad \gamma_i = H^k \frac{(x_i - x_{i-1})}{\max\{x_i - x_{i-1} : i = 2, \ldots, k\}},$$

where H_i, H^k are from (3.7) (when $i = 2$ and $i = k$ only H_2, H_3 and H_{k-1}, H_k should be considered, respectively) and ξ is the technical parameter.

2. "Maximum-Additive" Local Tuning

$$l_i = r \cdot \max\{H_i, 0.5(\lambda_i + \gamma_i), \xi\}, \tag{3.9}$$

where r, ξ, H_i, λ_i, and γ_i have the same meaning as above.

3. "Mixed" Local Tuning

$$l_i = 0.5\left(r\eta + H_i^2(r\eta)^{-1}\right), \quad \eta = \max\{H^k, \xi\}. \tag{3.10}$$

Let us give an explanation of the last procedure. It has been observed in [118] that (3.5) can be rewritten in the form

$$R_i = 2(z_i + z_{i-1}) - (x_i - x_{i-1})\left(\overline{L}_k + H_i^2 \cdot (\overline{L}_k)^{-1}\right),$$

so it can be interpreted as an auxiliary piecewise-linear function with local slopes

$$0.5\left(\overline{L}_k + H_i^2 \cdot (\overline{L}_k)^{-1}\right).$$

Therefore, the stochastic model has a geometric interpretation and, as we shall see, the local estimates (3.10) can be very useful when used together with the characteristic (3.5).

The second acceleration technique used hereinafter is the *local improvement technique* (see [61, 63, 107]). We distinguish the "optimistic" and the "pessimistic" approaches. The optimistic method alternates local steps with the global ones until a local stopping rule is satisfied. In its turn, the pessimistic strategy continues the global search after the local search accuracy has been achieved and stops if the global stopping rule is satisfied (see [107] for details). In both cases the information obtained during the local searches is used in the global search, too.

By mixing the above procedures we obtained the following 19 methods:

- **GEOM-AL, GEOM-GL, GEOM-LTM, GEOM-LTMA** that are **G**eometric methods (in the sense that (3.4) is used) which use, respectively: **A** priori given **L**ipschitz constant; **G**lobal estimate of the **L**ipschitz constant; **M**aximum **L**ocal **T**uning and **M**aximum-**A**dditive **L**ocal **T**uning. Each of these methods does not perform local improvement.

- **INF-AL, INF-GL, INF-LTM, INF-LTMA, INF-LTMI** that are **I**nformation methods ((3.5) is used) which use, respectively : **A** priori given **L**ipschitz constant; **G**lobal estimate of the **L**ipschitz constant; **M**aximum **L**ocal **T**uning; **M**aximum-**A**dditive **L**ocal **T**uning and **Mi**xed **L**ocal **T**uning. Each of these methods does not perform local improvement.

- **GEOM-LTIMO, GEOM-LTIMAO** are **G**eometric methods which use
  Maximum **L**ocal **T**uning and **M**aximum-**A**dditive **L**ocal **T**uning. Each of these methods uses the **O**ptimistic strategy of the local improvement.

- **INF-LTIMO, INF-LTIMAO, INF-LTIMIO** are **I**nformation methods which use **M**aximum **L**ocal **T**uning; **M**aximum-**A**dditive **L**ocal **T**uning and **Mi**xed **L**ocal **T**uning. Each of these methods uses the **O**ptimistic strategy of the local improvement.

- **GEOM-LTIMP, GEOM-LTIMAP** are **G**eometric methods which use
  Maximum **L**ocal **T**uning and **M**aximum-**A**dditive **L**ocal **T**uning. Each of these methods uses the **P**essimistic strategy of the local improvement.

- **INF-LTIMP, INF-LTIMAP, INF-LTIMIP** are **I**nformation methods which use **M**aximum **L**ocal **T**uning; **M**aximum-**A**dditive **L**ocal **T**uning and **Mi**xed **L**ocal **T**uning. Each of these methods uses the **P**essimistic strategy of the local improvement.

All the methods presented in this section have been compared on two classes of functions: 100 Shekel (see [118]) type test functions (Class 1) and the opposite of 100 Shekel type test functions selected so that $x^* \neq a \wedge x^* \neq b$, where x^* is from (3.1). Notice that given a function $f : \mathbb{R} \to \mathbb{R}$, we denoted by "the opposite of f" the function $g : \mathbb{R} \to \mathbb{R}$ defined as $g(x) = -f(x)$.

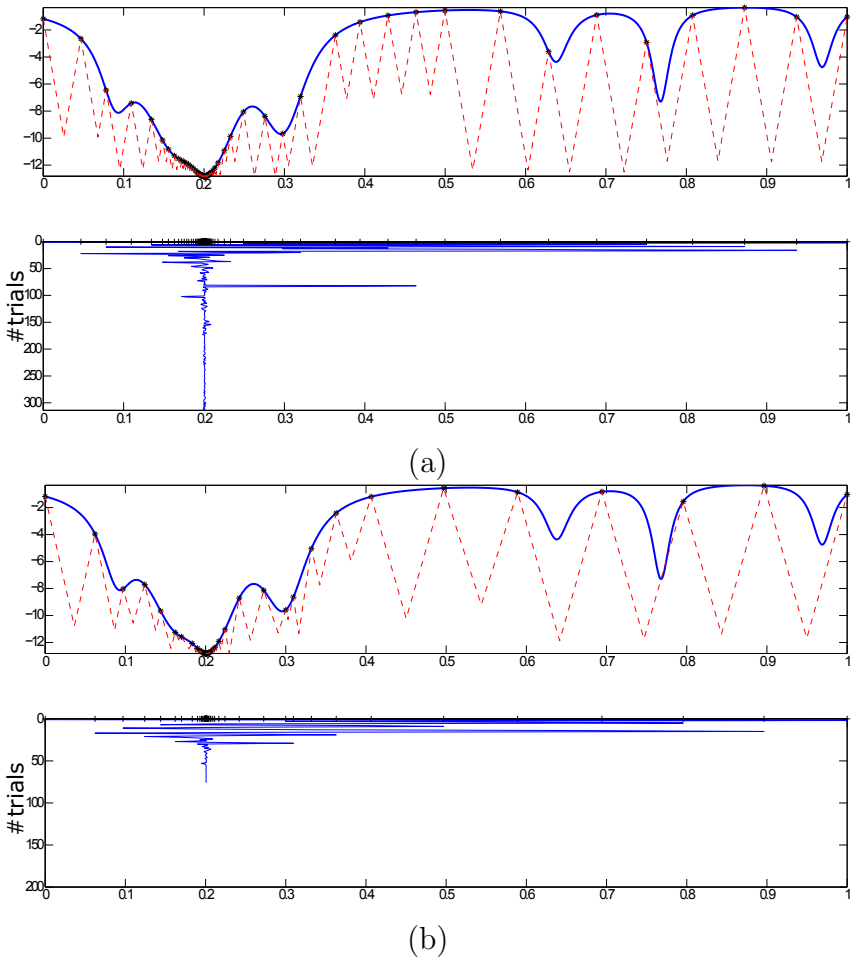

Figure 3.2. Function 3 from Class 1 and trial points generated by GEOM-AL (a) and GEOM LTMA (b).

Functions from Class 1 were generated as follows

$$\varphi(x) = -\sum_{i=1}^{10} \left[k_i^2(10x - a_i)^2 + c_i\right]^{-1}, \quad 0 \le x \le 1, \tag{3.11}$$

where $1 \le k_i \le 3$, $0.1 \le c_i \le 0.3$, $0 \le a_i \le 10$, $1 \le i \le 10$, and all the parameters are supposed to be the pseudorandom numbers in the corresponding ranges. For each method the technical parameter ξ from (4.4) was set to 10^{-8}. The initial values $r = 1.1$ and $r = 2$ were used respectively for the geometric and information methods without optimistic local improvement over all the classes of test functions and they were increased with step equal

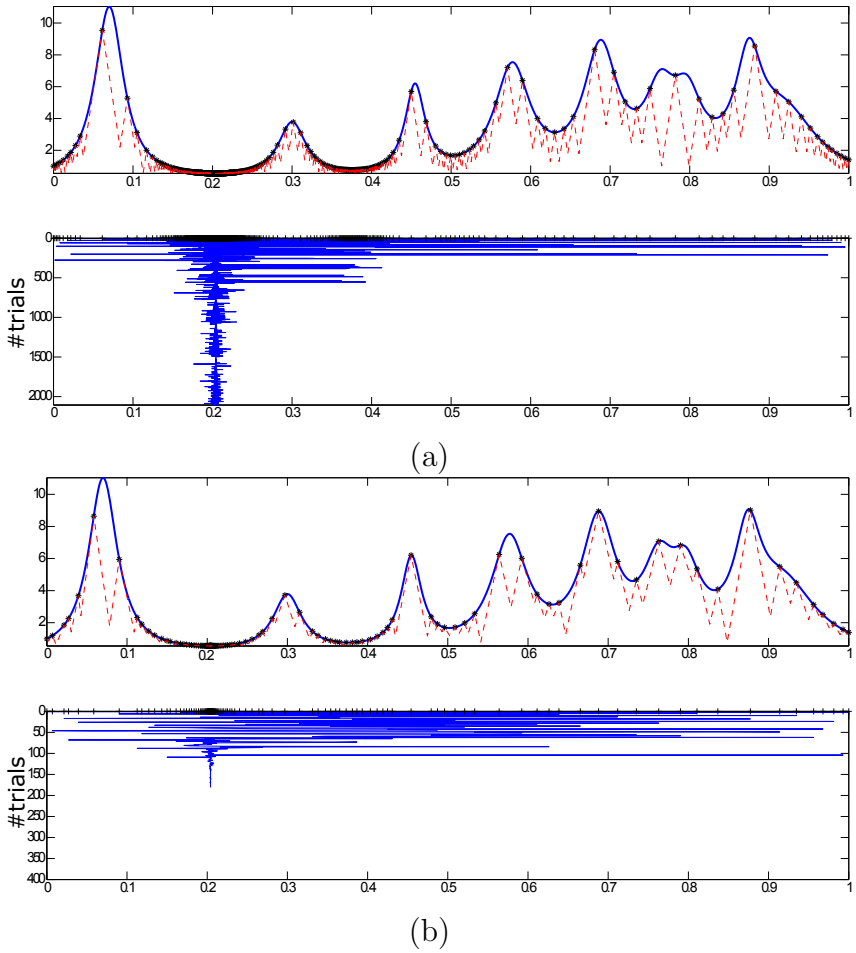

Figure 3.3. Function 85 from Class 2 and trial points generated by GEOM AL (a) and GEOM LTMA (b).

to 0.1 until all test problems were solved, i.e., the tested algorithm has generated a point x^k after k trials such that $|x^k - x^*| \leq \varepsilon$ with $\varepsilon = 10^{-5}$. Fig.3.2 and Fig.3.3 show two examples of application of the methods respectively on Class 1 and on Class 2. In the same figures appears the auxiliary function and trial points generated by the methods. As the objective functions $f(x)$ are considered to be hard to evaluate, the number of trials was chosen as the comparison criterion. We reported in Table 3.1 the averages of trials for each method on both classes. The best results are shown in bold.

Since the majority of test problems in both classes can be solved with

Table 3.1. Results of numerical experiments without the optimistic local improvement.

	Class 1		**Class 2**	
Method	**r**	**Average**	**r**	**Average**
GEOM-AL	1.1	190.28	1.1	3756.54
GEOM-GL	1.4	205.42	1.3	2905.17
GEOM-LTM	1.7	93.85	1.3	217.11
GEOM-LTMA	2.5	97.92	1.5	196.00
INF-AL	2.0	188.79	2.0	3709.51
INF-GL	2.8	211.64	2.0	2211.69
INF-LTM	3.7	103.00	2.1	158.12
INF-LTMA	4.0	**78.05**	2.3	130.71
INF-LTMI	5.4	208.09	3.4	1913.82
GEOM-LTIMP	1.7	98.04	1.1	153.07
GEOM-LTIMAP	2.3	93.56	1.5	202.15
INF-LTIMP	3.7	107.98	2.1	160.84
INF-LTIMAP	4.2	86.01	2.1	**114.76**
INF-LTMIP	5.6	215.10	3.4	1892.04

smaller values of the parameter r w.r.t those reported in Table 3.1, the use of a common value of this parameter for the whole classes of test functions can increase considerably the number of trials. For this reason, for the geometric and information methods with the use of optimistic local improvement applied to Class 1 we have not increased r until all test problems were solved to further appreciate the speed of this approach. We have chosen to stop until at least 90 of the 100 problems were solved and for the remaining ones we obtained a local minimum. In Table 3.2 we reported the averages of trials using the optimistic local improvement on both classes and the percentage of problems solved.

Thus, it has been shown that the two acceleration techniques, described

Table 3.2. Results of numerical experiments with the optimistic local improvement.

	Class 1			Class 2		
Method	**r**	**Average**	**Success**	**r**	**Average**	**Success**
GEOM-LTIMO	1.7	55.16	92%	1.8	55.40	100%
GEOM-LTIMAO	1.6	**47.90**	91%	1.7	55.74	100%
INF-LTIMO	2.3	54.48	91%	3.1	51.70	100%
INF-LTIMAO	4.0	61.34	95 %	3.1	56.78	100%
INF-LTMIO	5.5	51.78	90 %	3.4	**50.66**	100%

in this section, increase considerably the speed of geometric and information methods, especially when they are tested on Class 2. Mixing the considered procedures we obtained 19 different derivative-free Lipschitz Global Optimization Methods. In particular, the use of the geometric interpretation of the stochastic model led to the 3 recently introduced methods (INF-LTIMI, INF-LTIMIO, INF-LTIMIP).

3.2 Expensive global optimization using Lipschitz derivatives

Let us now study in this section the problem

$$f^* = f(y^*) = \min f(x), \quad x \in D, \tag{3.12}$$

where $D = [a, b]$ is a real interval and and $f(x)$ is the objective black-box function that satisfies

$$|f'(y) - f'(x)| \leq M|x - y|, \quad x, y \in D, \tag{3.13}$$

with $0 < M < \infty$.

This problem is under a close scrutiny since the nineties of the last century. Breiman and Cutler (see [6]) have proposed a method to solve the problem (3.12) where the constant M from (3.13) is a priori known whereas Gergel in [28] has proposed a global optimization method that estimates M in the course of the search. Both methods use in their work auxiliary non-smooth support functions that are less or equal to $f(x), x \in D$ (see Fig. 3.4). Since the objective function $f(x)$ is differentiable over the search region D, in [63, 90] there have been introduced methods constructing smooth support functions that are closer to the objective function $f(x)$ with respect to non-smooth ones providing so a notable acceleration in comparison with the methods [6, 28].

Global optimization methods introduced in [90] for solving the problem (3.12), (3.13) construct a smooth support $\psi_i(x)$ at each subinterval $[x_{i-1}, x_i]$, $2 \leq i \leq k$, of the search region D for the objective function $f(x)$ using the fact that the maximal possible curvature of $f(x)$ is determined by the Lipschitz constant M from (3.13). The functions

$$\psi_i(x) \leq f(x), \quad x \in [x_{i-1}, x_i], \ 2 \leq i \leq k, \tag{3.14}$$

are shown in Figures 3.5 and 3.6 and can be built as follows.

Let us consider the following three functions

$$\pi_i(x) = 0.5Mx^2 + b_i x + c_i, \tag{3.15}$$

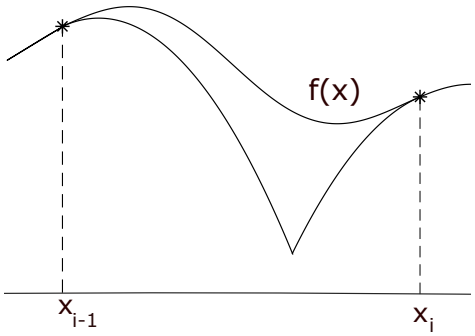

Figure 3.4. Non-smooth auxiliary function.

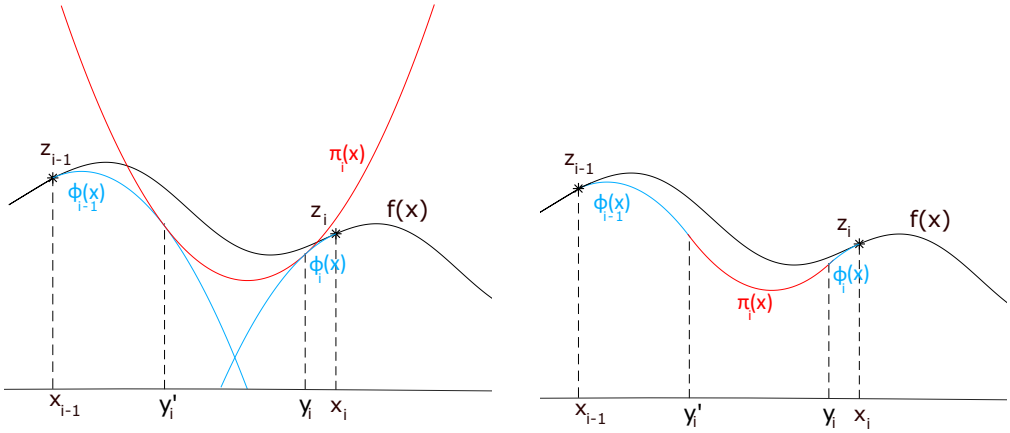

Figure 3.5. Construction of smooth support function $\psi_i(x)$ from (3.18), the case where the vertex of $\pi_i(x)$ is inside the interval $[y_i', y_i]$.

$$\phi_{i-1}(x) = z_{i-1} + z'_{i-1}(x - x_{i-1}) - 0.5M(x - x_{i-1})^2, \tag{3.16}$$

$$\phi_i(x) = z_i - z'_i(x_i - x) - 0.5M(x_i - x)^2, \tag{3.17}$$

where $z_i = f(x_i)$, $z'_i = f'(x_i)$, and b_i, c_i are two parameters to be determined. Notice that $\phi_{i-1}(x)$ and $\phi_i(x)$ are support functions for $f(x)$ and are obtained from the Taylor formulae based at the points x_{i-1} and x_i, respectively. The meaning of the parabola $\pi_i(x)$ will be explained in a minute. It has been shown in [6, 28] that

$$\max\{\phi_{i-1}(x), \phi_i(x)\}, \quad x \in [x_{i-1}, x_i], \ 2 \le i \le k,$$

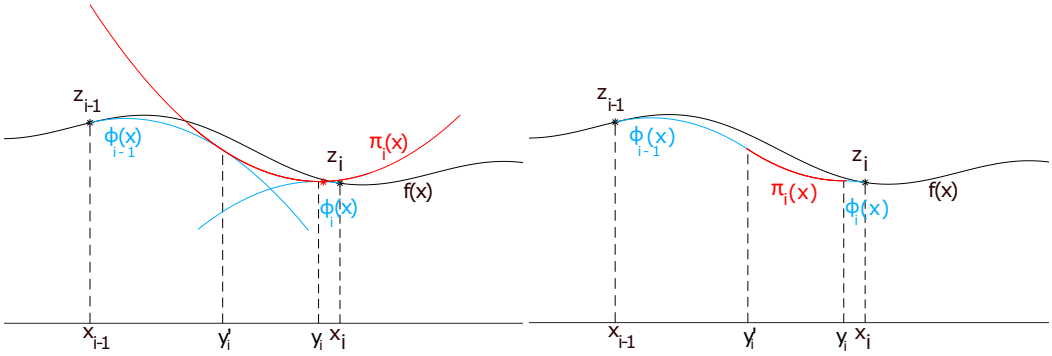

Figure 3.6. Construction of support function $\psi_i(x)$ from (3.18), the case where the vertex of $\pi_i(x)$ is outside the interval $[y'_i, y_i]$.

is a non-smooth minorant for $f(x)$ over $[x_{i-1}, x_i]$. Then, the key observation made in [90] in order to construct a smooth minorant consists in the fact that due to the boundedness of the curvature of $f(x)$, it cannot be below the parabola $\pi_i(x)$ over the interval $[y'_i, y_i]$ that can be found by asking that the piece-wise quadratic function

$$\psi_i(x) = \begin{cases} \phi_{i-1}(x), & x \in [x_{i-1}, y'_i], \\ \pi_i(x), & x \in [y'_i, y_i], \\ \phi_i(x), & x \in [y_i, x_i], \end{cases} \tag{3.18}$$

is a smooth support function for $f(x)$ over $[x_{i-1}, x_i]$. The values y_i, y'_i and b_i, c_i from (3.15) can be determined by "gluing" the functions $\phi_{i-1}(x)$, $\pi_i(x), \phi_i(x)$ and their first derivatives. In other words, the following system of equations should be solved:

$$\begin{cases} \phi_{i-1}(y'_i) = \pi_i(y'_i) \\ \phi_i(y_i) = \pi_i(y_i) \\ \phi'_{i-1}(y'_i) = \pi'_i(y'_i) \\ \phi'_i(y_i) = \pi'_i(y_i) \end{cases} \tag{3.19}$$

As was shown in [90], its solution is:

$$y_i = \frac{x_i - x_{i-1}}{4} + \frac{z'_i - z'_{i-1}}{4M} + \frac{z_{i-1} - z_i + z'_i x_i - z'_{i-1} x_{i-1} + 0.5M(x_i^2 - x_{i-1}^2)}{M(x_i - x_{i-1}) + z'_i - z'_{i-1}}, \tag{3.20}$$

$$y'_i = -\frac{x_i - x_{i-1}}{4} - \frac{z'_i - z'_{i-1}}{4M} + \frac{z_{i-1} - z_i + z'_i x_i - z'_{i-1} x_{i-1} + 0.5M(x_i^2 - x_{i-1}^2)}{M(x_i - x_{i-1}) + z'_i - z'_{i-1}}, \tag{3.21}$$

$$b_i = z'_i - 2My_i + Mx_i, \tag{3.22}$$

$$c_i = z_i - z'_i x_i - 0.5Mx_i^2 + My_i^2. \tag{3.23}$$

Taking in mind that we construct $\psi_i(x)$ in order to solve the global optimization problem (3.12), (3.13) we are interested in finding the point

$$p_i = \min\{\psi_i(x) : x \in [x_{i-1}, x_i]\}$$

and the respective value $R(i) = \psi_i(p_i)$ called *characteristic* of the interval $[x_{i-1}, x_i]$. The term characteristic is due to the class of characteristic methods introduced in [35] (see also [38]) for solving Lipschitz global optimization problems. The same terminology is used in Divide the Best methods (see [111]) to determine a goodness of each subregion of the search domain. In fact, the methods under consideration belong to the class of Divide the Best algorithms.

Then the following three situations can take place:

1. The first one corresponds to the case illustrated in Fig. 3.5 where $\pi'_i(y'_i) < 0$ and $\pi'_i(y_i) > 0$. In this case

$$R_i = \min\{f(x_{i-1}), \psi_i(\overline{x_i}), f(x_i)\}, \tag{3.24}$$

   where $\overline{x_i}$ is the vertex of $\pi_i(x)$, namely,

$$\overline{x_i} = 2y_i - M^{-1}z'_i - x_i. \tag{3.25}$$

2. The second situation is shown in Fig. 3.6 and it corresponds to the situation where $\pi'_i(y'_i) < 0$ and $\pi'_i(y_i) \leq 0$. In this case

$$R_i = \min\{f(x_{i-1}), f(x_i)\}. \tag{3.26}$$

3. In the third case where $\pi'_i(y'_i) \geq 0$ and $\pi'_i(y_i) > 0$ the value R_i can be computed analogously to the previous case.

3.2.1 Local tuning for accelerating the search

We start this subsection by presenting two traditional strategies for choosing an estimate for the global value M from (3.13). These methods are used for a numerical comparison that is required for checking a speed up that a local tuning can provide. After that a known local tuning from [90] and two recent local tuning techniques from [109] will be introduced. In order to describe these procedures let us denote as $\{x_i\}^k$ the ordered points where the objective function $f(x)$ has been evaluated. The operation of evaluation of $f(x)$ and $f'(x)$ at a point x is called *trial* hereinafter and points $\{x_i\}^k$ are called *trial points*, where $k \geq 2$ is the number of iterations of the algorithm (for $k = 2$: $x_1 = a$ and $x_2 = b$). Let $r > 1$ be the reliability parameter of the methods.

For each interval $[x_{i-1}, x_i], 2 \leq i \leq k$, its local Lipschitz constant for $f'(x)$ is estimated by values m_i in one of the following five ways:

1. A priori given Lipschitz constant. Set

$$m_i = M, \quad 2 \leq i \leq k. \tag{3.27}$$

In this case, an algorithm uses the same exact a priori known value M from (3.13) of the Lipschitz constant of $f'(x)$ for each subinterval $[x_{i-1}, x_i], 2 \leq i \leq k$.

2. Global estimate. Compute estimates

$$m_i = r \cdot \max\{\xi, H^k\}, \quad 2 \leq i \leq k, \tag{3.28}$$

where $\xi > 0$ is a technical parameter (a small number greater than 0) reflecting the supposition that $f'(x)$ is not constant over the interval $[x_{i-1}, x_i]$. Then,

$$H^k = \max\{v_i : 2 \leq i \leq k\}, \tag{3.29}$$

with

$$v_i = \frac{|2(z_{i-1} - z_i) + (z'_{i-1} + z'_i)(x_i - x_{i-1})| + d_i}{(x_i - x_{i-1})^2} \tag{3.30}$$

and

$$d_i = \sqrt{|2(z_{i-1} - z_i) + (z'_{i-1} + z'_i)(x_i - x_{i-1})|^2 + (z'_i - z'_{i-1})^2 (x_i - x_{i-1})^2}. \tag{3.31}$$

Notice that this adaptive estimate of the global Lipschitz constant is obtained by imposing that the upper bound

$$\phi_{i-1}^{+}(x) = z_{i-1} + z'_{i-1}(x - x_{i-1}) + \frac{1}{2}m_i(x - x_{i-1})^2,$$

based on the point x_{i-1} is equal or greater than the lower bound

$$\phi_{i}^{-}(x) = z_{i} + z'_{i}(x - x_{i}) - \frac{1}{2}m_i(x - x_{i})^2,$$

based on the point x_i, over the interval $[x_{i-1}, x_i]$, i.e.,

$$\phi_{i-1}^{+}(x) \geq \phi_{i}^{-}(x), \quad x \in [x_{i-1}, x_i], \tag{3.32}$$

and the upper bound

$$\phi_{i}^{+}(x) = z_{i} + z'_{i}(x - x_{i}) + \frac{1}{2}m_i(x - x_{i})^2,$$

based on the point x_i is equal or greater than the lower bound

$$\phi_{i-1}^{-}(x) = z_{i-1} + z'_{i-1}(x - x_{i-1}) - \frac{1}{2}m_i(x - x_{i-1})^2,$$

based on the point x_{i-1}, over the interval $[x_{i-1}, x_i]$ i.e.,

$$\phi_{i}^{+}(x) \geq \phi_{i-1}^{-}(x), \quad x \in [x_{i-1}, x_i]. \tag{3.33}$$

As shown in [118], in order to satisfy (3.32), (3.33) it is sufficient that the following inequality holds for m_i:

$$m_i \geq \tau(x), \quad x \in [x_{i-1}, x_i],$$

where

$$\tau(x) = 2\frac{|z_i - z_{i-1} + z'_i(x - x_i) - z'_{i-1}(x - x_{i-1})|}{(x - x_i)^2 + (x - x_{i-1})^2}.$$

It can be proved that the values v_i from (3.30) are such that

$$v_i = \max\{\tau(x) : x \in [x_{i-1}, x_i]\}.$$

As was already mentioned, strategies 1 and 2 described above can slow down the global search. In order to overcome this problem, the following local tuning technique has been introduced in [90].

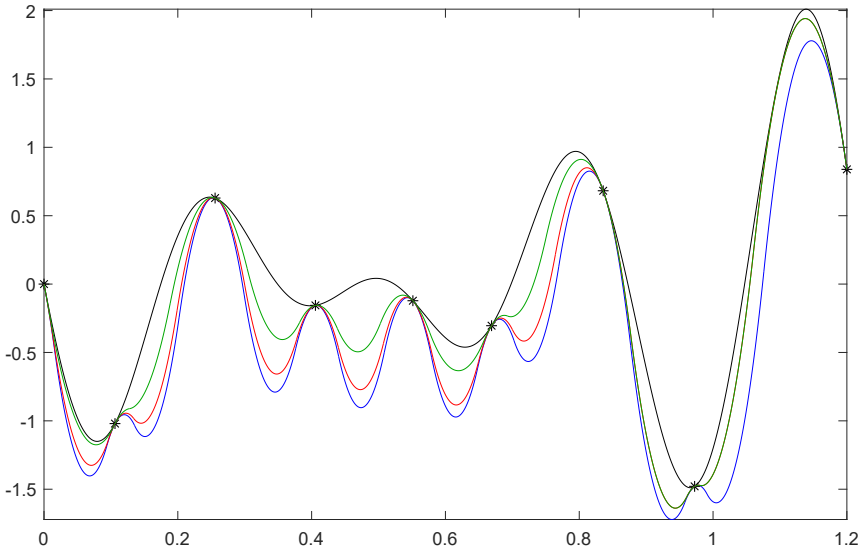

Figure 3.7. Smooth support functions constructed by using the A priori given Lipschitz constant (blue), the Global estimate (red) and the Maximum Local Tuning (green).

3. Maximum Local Tuning. Compute estimates

$$m_i = r \cdot \max\{\lambda_i, \gamma_i, \xi\}, \tag{3.34}$$

where

$$\lambda_i = \max\{v_{i-1}, v_i, v_{i+1}\}, \quad 3 \le i \le k-1, \tag{3.35}$$

with v_i as in (3.30); when $k = 2$ only v_2 should be considered and when $i = 2$ and $i = k$ we consider only v_2, v_3 and v_{k-1}, v_k, respectively.

The value γ_i is calculated as follows:

$$\gamma_i = H^k \frac{(x_i - x_{i-1})}{X^{max}}, \tag{3.36}$$

$$X^{max} = \max\{(x_i - x_{i-1}), \quad 2 \le i \le k\}. \tag{3.37}$$

Notice that these adaptive local estimates of local Lipschitz constants are obtained during the search balancing through (3.34) local and global information obtained in the course of the previous iterations. Indeed, when

the interval $[x_{i-1}, x_i]$ is small than the local information managed by λ_i has a decisive rule; in contrast, when the interval $[x_{i-1}, x_i]$ is wide the global information represented by γ_i is used.

Figure 3.7 shows support functions built using the three strategies described above. It can be seen that $\psi_i(x)$ constructed using the local tuning technique is significantly closer to $f(x)$ with respect to the other two strategies. It should be stressed that in the case of the a priori given Lipschitz constant, condition (3.13) ensures that $\psi_i(x)$ from (3.18) is a minorant for $f(x)$. Since strategies 2 and 3 *estimate* Lipschitz constants, the resulting functions $\psi_i(x)$ can violate the inequality $\psi_i(x) \leq f(x)$. This fact can lead to the loss of the global solution during the search. Conditions ensuring convergence to global solutions of optimization methods using global and local estimates of Lipschitz constants for $f'(x)$ will be established in Section 3.2.2.

In order to introduce the Maximum-Additive Local Tuning for derivatives let us notice that the Maximum Local Tuning technique described above balances local and global information about $f'(x)$ computing the maximum among values λ_i, γ_i, and ξ (see (3.34)). As we saw in the previous section, see (3.9), it has been recently shown in [107] that for functions $g(x)$ satisfying the Lipschitz condition (3.2) with a constant L, $0 < L < \infty$, it makes sense to use a maximum-additive convolution of values representing local and global information collected during the search (that, clearly, are collected in a different way since functions satisfying (3.2) can be non-differentiable whereas λ_i and γ_i from (3.35) and (3.36) estimate Lipschitz constants for derivatives). Let us propose a maximum-additive convolution of (3.35) and (3.36) for our problem (3.12)–(3.13).

4. Maximum-Additive Local Tuning for derivatives. Compute estimates

$$m_i = r \cdot \max\{v_i, \frac{1}{2}(\lambda_i + \gamma_i), \xi\}, \quad 2 \leq i \leq k, \tag{3.38}$$

where values v_i, λ_i, and γ_i are from (3.30), (3.35), and (3.36), respectively.

This estimate provides an additive mixture of λ_i and γ_i with the equal usage of local and global information. The presence of v_i in (3.38) is explained by the fact that for small intervals the estimate γ_i can be very small (see

(3.36)) leading so to the prohibited situation $v_i > 0.5(\lambda_i + \gamma_i)$. If v_i was not in (3.38), this could lead to the possibility that points y_i' and y_i can be generated outside the interval $[x_{i-1}, x_i]$ and this could produce errors in the work of algorithm. The presence of v_i in (3.38) avoids this case. Notice also that (3.38) can be generalized to the case of a weighted usage of local and global estimates depending on a parameter $0 < \rho < 1$ as follows

$$m_i = r \cdot \max\{v_i, \rho\lambda_i + (1-\rho)\gamma_i, \xi\}.$$

The idea for the last local tuning technique we are going to introduce also comes from algorithms developed to deal with problems satisfying Lipschitz condition (3.2). As we already said in the previous section, methods using the local tuning technique (3.10), which have been tested in [49], showed a promising performance on a broad class of test problems satisfying (3.2). In this section, we introduce the following Mixed Local Tuning technique for derivatives evolving the idea of the convolution from algorithms working with functions satisfying (3.2) to methods that can be used to solve our problem (3.12)–(3.13).

5. Mixed Local Tuning for derivatives. Compute values

$$m_i = \max\{rv_i, 0.5\left(r\eta + v_i^2(r\eta)^{-1}\right)\}, \quad \eta = \max\{H^k, \xi\}, \quad 2 \le i \le k, \tag{3.39}$$

where H^k is from (3.29). In this local tuning the local information is represented by v_i and the global one is represented by η.

Let us now make a remark regarding the correct construction of the functions $\psi_i(x)$ from (3.18). As was already mentioned above, if over an interval $[x_{i-1}, x_i]$ the value m_i is underestimated, the points y_i' and y_i can be generated outside the interval $[x_{i-1}, x_i]$ and this can produce errors in the work of the algorithms. It has been proven in [90, 118] that strategies (3.27), (3.28), and (3.34) ensure that the points y_i' and y_i are inside the interval $[x_{i-1}, x_i]$. An analogous result can be proven for the strategies (3.38) and (3.39).

Theorem 3.2.1. *If β is a finite number and the following condition*

$$v_i < m_i \le \beta < \infty, \tag{3.40}$$

takes place for the strategies (3.38) and (3.39), then for functions $\psi_i(x)$ constructed by these algorithms

$$y'_i \in (x_{i-1}, x_i), \quad y_i \in (x_{i-1}, x_i).$$

Moreover, it follows

$$y'_i - x_{i-1} \geq \frac{(\frac{m_i}{v_i} - 1)^2}{4\frac{m_i}{v_i}(\frac{m_i}{v_i} + 1)}(x_i - x_{i-1}), \tag{3.41}$$

$$x_i - y_i \geq \frac{(\frac{m_i}{v_i} - 1)^2}{4\frac{m_i}{v_i}(\frac{m_i}{v_i} + 1)}(x_i - x_{i-1}). \tag{3.42}$$

Proof. The proof is analogous to the proof of Theorem 4.11 from [118] with the remark that strategies (3.38) and (3.39) ensure that the inequality (3.40) is satisfied automatically since the parameter r is a finite number $r > 1$. □

3.2.2 General scheme and convergence conditions

We start this section by describing decision rules of five global optimization algorithms using the strategies described above to estimate Lipschitz constants. The methods have a similar structure that is described by the following General Scheme incorporating the five algorithms using Derivatives (GSD).

STEP 0. The first two trials are executed at the points $x^1 = a$ and $x^2 = b$. For $k \geq 2$ we choose the point x^{k+1} using the following steps:

STEP 1. Renumber the points $x^1, ..., x^k$ and the corresponding function values $z^1, ..., z^k$ of the previous iterations by subscripts so that

$$a = x_1 < ... < x_k = b, \quad z_i = f(x_i), \quad 1 \leq i \leq k.$$

STEP 2. Calculate the current estimate m_i of the Lipschitz constants of $f'(x)$ over the intervals $[x_{i-1}, x_i], 2 \leq i \leq k$, in one of the ways previously described in (3.27),(3.28),(3.34),(3.38), and (3.39).

STEP 3. Initiate the index sets $I = \emptyset$, $Y' = \emptyset$, and $Y = \emptyset$. Set the index of the current interval $i = 2$.

STEP 3.1. If for the subinterval $[x_{i-1}, x_i]$ it follows that

$$\pi_i'(y_i') \cdot \pi_i'(y_i) \geq 0 \tag{3.43}$$

then go to STEP 3.3, otherwise go to STEP 3.2, where π_i, y_i', and y_i are from (3.15), (3.20) and (3.21).

STEP 3.2. Calculate for $[x_{i-1}, x_i]$ its characteristic R_i using (3.24) where $\overline{x_i}$ is from (3.25). Include i in the set I and go to the STEP 3.4.

STEP 3.3. Calculate for $[x_{i-1}, x_i]$ its characteristic R_i using (3.26). If $f(x_{i-1}) < f(x_i)$ include the index i in Y' otherwise include i in the set Y. Go to the STEP 3.4.

STEP 3.4. If $i < k$, set $i = i + 1$ and go to the STEP 3.1, otherwise go to the STEP 4.

STEP 4 Select the interval (x_{t-1}, x_t) corresponding to the minimal characteristic, i.e, such that

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

STEP 5 If the stopping rule is not satisfied, i.e.,

$$|x_t - x_{t-1}| > \varepsilon,$$

where ε is the accuracy of the search, then execute the next trial at the point

$$x^{k+1} = \begin{cases} y_t', & \text{if } t \in Y,' \\ \overline{x_t}, & \text{if } t \in I, \\ y_t, & \text{if } t \in Y. \end{cases} \tag{3.44}$$

Using the procedures proposed to estimate M in the previous section and the general scheme described above we obtained the following 5 algorithms:

- **DKC**: The method using the first **D**erivatives and the a priori **K**nown Lipschitz **C**onstant M, see (3.27).
- **DGE**: The method using the first **D**erivatives and the **G**lobal **E**stimate of the Lipschitz Constant M, see (3.28).

- **DML**: The method using the first **D**erivatives and the **M**aximum **L**ocal Tuning, see (3.34).

- **DMAL**: The method using the first **D**erivatives and the **M**aximum-**A**dditive **L**ocal Tuning, see (3.38).

- **DMXL**: The method using the first **D**erivatives and the **Mix**ed **L**ocal Tuning, see (3.39).

Global convergence properties of the methods DKC, DGE, and DML have been studied in [90] (see also [118]). In order to extend these results to the new methods DMAL and DMXL let us consider an infinite trial sequence $\{x^k\}$ generated by an algorithm belonging to the general scheme GSD with the accuracy $\varepsilon = 0$. The following results describing convergence properties of the algorithms belonging to GSD hold:

Theorem 3.2.2. *Let the point $\overline{x}$, ($\overline{x} \neq a$, $\overline{x} \neq b$) be a limit point of the sequence $\{x^k\}$ generated by an algorithm belonging to the general scheme GSD during the course of minimizing a function $f(x)$, $x \in [a, b]$. If the values m_i satisfy conditions (3.40) then the point $\overline{x}$ will be a local minimizer of the function $f(x)$.*

Proof. The proof is analogous to the proof of Theorem 4.12 from [118] with the remark that the values m_i from (3.38) for DMAL, and from (3.39) for DMXL ensure that the inequality (3.40) is satisfied. □

Theorem 3.2.3. *Let $\overline{x}$, ($\overline{x} \neq a, \overline{x} \neq b$) be a limit point of the sequence $\{x^k\}$ generated by an algorithm belonging to the general scheme GSD during the course of minimizing a function $f(x)$, $x \in [a, b]$. Then, if condition (3.40) is fulfilled for the intervals containing $\overline{x}$, there exist two subsequences of $\{x^k\}$ converging to $\overline{x}$, one from the left, the other from the right.*

Proof. To prove this result it is necessary to consider two cases: (i) $\overline{x} \neq x^k, k \geq 1$; (ii) there exists an iteration number q such that $\overline{x} = x^q = x_{j(k)}, k \geq q$. In the first case we have only one interval containing $\overline{x}$ and in the second one there are two intervals $\overline{x}$ belongs to. The first case is proved analogously

to Theorem 4.13 from [118] whereas the second one follows the proof of Theorem 4.14 from [118]. □

Theorem 3.2.4. *Let $\overline{x}$ be a limit point of the sequence $\{x^k\}$ generated by an algorithm belonging to the general scheme GSD and condition (3.40) be fulfilled. We then have for trial points x^k*

$$f(x^k) \geq f(\overline{x}), \quad k \geq 1. \tag{3.45}$$

Proof. The proof follows the argumentation proving Theorem 4.15 from [118]. □

Corollary 3.2.5. *If, given the conditions of the theorem, alongside $\overline{x}$ there exists another limit point x' of the sequence $\{x^k\}$ then $f(\overline{x}) = f(x')$.*

Proof. The proof is trivial and so omitted. □

Let us denote by M_j the local Lipschitz constant of $f'(x)$ over an interval $[x_{j-1}, x_j]$, by $\overline{X}$ the set of limit points of $\{x^k\}$ and by X^* the set of global minimizers of $f(x)$. The following theorem then takes place.

Theorem 3.2.6. *Let x^* be a global minimizer of $f(x)$ and $[x_{j-1}, x_j], j = j(k)$, be an interval containing this point during the course of k-th iteration of GSD. If there exists an iteration number s such that, for all $k \geq s$ for the values $m_j, j = j(k)$, the inequality*

$$M_j \leq m_j \leq \beta \tag{3.46}$$

takes place for $[x_{j-1}, x_j]$ where β is a finite number and (3.40) holds for all the other intervals, then the point x^* is a limit point of the sequence $\{x^k\}$.

Proof. The proof is analogous to the reasoning for Theorem 4.16 from [118]. □

Corollary 3.2.7. *Given the conditions of the theorem, all limit points of $\{x^k\}$ are global minimizers of $f(x)$.*

Proof. The proof follows immediately from Corollary 3.2.5. □

Corollary 3.2.8. *If, given the condition of the theorem, (3.46) holds for all points $x^* \in X^*$, then $\overline{X} = X^*$.*

Proof. The proof is obvious and is omitted. □

3.3 Derivatives with the Infinity Computer and Numerical Experiments

At the beginning of the XXIth century a new computational paradigm called *Infinity Computing* has been proposed (see a recent survey [94]) in order to give people the possibility to work numerically with a huge number of different infinities and infinitesimals. From the foundational point of view, this computational methodology[1] is based on the notion of *grossone* introduced as an infinite unit of measure equal to the number of elements of the set $\mathbb{N}$ of natural numbers and denoted by the numeral ①. It acts as a basic element of a powerful numeral system allowing one to express not only finite but also different infinite and infinitesimal quantities (analogously, the numeral 1 is a basic element allowing one to express a variety of finite quantities). The numeral ① is introduced by describing its properties postulated by the Infinite Unit Axiom consisting of three parts: Infinity, Identity, and Divisibility.

The Infinite Unit Axiom The infinite unit of measure is introduced as the number of elements of the set $\mathbb{N}$, of natural numbers. It is expressed by the numeral ① called grossone and has the following properties:

- **Infinity** Any finite natural number n is less than grossone, i.e., $n < ①$.
- **Identity** The following relations link ① to identity elements 0 and 1

$$0\cdot① = ①\cdot 0 = 0, \quad ①-① = 0, \quad \frac{①}{①} = 1, \quad ①^0 = 1, \quad 1^{①} = 1, \quad 0^{①} = 0.$$

- **Divisibility** For any finite natural number n sets $\mathbb{N}_{k,n}$, $1 \le k \le n$, being the nth parts of the set $\mathbb{N}$, of natural numbers have the same number of elements indicated by the numeral $\frac{①}{n}$ where

$$\mathbb{N}_{k,n} = \{k, k+n, k+2n, k+3n, \dots\}, \quad 1 \le k \ \le n, \quad \bigcup_{k=1}^{n} \mathbb{N}_{k,n} = \mathbb{N}.$$

This axiom is added to axioms for real numbers and so it is also postulated that usual associative and commutative properties of multiplication and

[1] Notice that it is not related to non-standard analysis (see [95] for a detailed discussion).

addition, distributive property of multiplication over addition, existence of inverse elements with respect to addition and multiplication hold for grossone as for finite numbers (see [94] for a detailed discussion).

From the practical point of view, this methodology has given rise both to a new supercomputer patented in several countries and called the *Infinity Computer* and to a variety of applications in different research areas of mathematics and computer science including infinite series, fractals, Turing machines, cellular automata, probability, hyperbolic geometry, percolation, etc. (for the references see [94]). In the context of this work dealing with optimization and numerical computations it is worthwhile to mention the following texts: [2, 13, 14, 17, 18, 19, 20, 21, 25, 43, 44, 93, 105].

In order to start, let us recall the basics of the positional numeral system with the radix ① used at the Infinity Computer (very often numbers written in this system are called *grossnumbers*). A grossnumber A being a number expressed in this positional numeral system is expressed by the numeral

$$A = d_{p_m}①^{p_m}...d_{p_1}①^{p_1}d_{p_0}①^{p_0}...d_{p_{-k}}①^{p_{-k}}, \tag{3.47}$$

representing the quantity

$$A = d_{p_m}①^{p_m} + ... + d_{p_1}①^{p_1} + d_{p_0}①^{p_0} + ... + d_{p_{-k}}①^{p_{-k}},$$

where *grossdigits* $d_i \neq 0$ are finite (positive or negative, fractional or integer) numbers and *grosspowers* p_i can be finite, infinite, or infinitesimal such that

$$p_m > p_{m-1} > \ldots > p_1 > p_0 > p_{-1} > \ldots p_{-(k-1)} > p_{-k},$$

with $p_0 = 0$. Since ① is a number, it follows (see [94]) that $①^0 = 1$. This means, in particular, that a finite number d is represented as $d①^0$ and, therefore, if a number A written in the form (3.47) has $d_{p_0} \neq 0$, then A has a finite part. A number is called *purely finite* if it is finite and does not contain infinitesimal parts.

This numeral system allows one to represent numbers that can have different infinite and infinitesimal parts. In particular, terms having finite positive

grosspowers represent the simplest infinite parts of the number A. Analogously, terms having negative finite grosspowers represent the simplest infinitesimal parts of A. For example, the number

$$A_1 = 13.4①^{-1.5} + 16.7①^{-34.9}$$

has two infinitesimal parts, whereas the number

$$A_2 = -16.2 \cdot 10^{-4}①^{34.2} - 56.2 \cdot 10^{6}①^{15.7} + 2①^{-10.3} - 104.4①^{-22.7}$$

has two infinite and two infinitesimal parts.

In the context of this work, our interest in the Infinity Computing is due to the fact that, as it has been shown in [93], this methodology allows one to compute easily exact (the words *exact* means: up to the machine precision) derivatives for black-box functions. Thus, this opportunity can be used for solving the global optimization problem (3.12), (3.13) by evaluating derivatives of $f(x)$ during the work of the algorithms introduced above without the necessity to have a code for evaluating $f'(x)$.

Let P be a procedure implemented to calculate the function $f(x)$, which uses implementations of elementary functions involved in computing $f(x)$, the argument x and finite constants connected by four arithmetical operations executed with the numerals (3.47). We are interested in calculating $f'(x)$ by using P. The following theorem (see [93]) explains how the exact derivatives can be computed using the Infinity Computing methodology:

Theorem 3.3.1. *Suppose that:*
(i) for a function $f(x)$ calculated by a procedure implemented at the Infinity Computer there exists an unknown Taylor expansion over an interval $[a, b]$, with a and b purely finite numbers, containing a purely finite point y;
(ii) $f(x)$, $f'(x), f^{(2)}(x), \ldots f^{(k)}(x)$ assume purely finite values or are equal to zero for purely finite points $x \in [a, b]$;
(iii) $f(x)$ has been evaluated at a point $y + ①^{-1} \in [a, b]$. Then the Infinity Computer returns the result of this evaluation in the positional numeral system with the infinite radix ① in the following form

$$f(y + ①^{-1}) = c_0①^0 c_{-1}①^{-1} c_{-2}①^{-2} \ldots c_{-(k-1)}①^{-(k-1)} c_{-k}①^{-k}, \qquad (3.48)$$

where

$$f(y) = c_0,\ f'(y) = c_{-1},\ f^{(2)}(y) = 2! \cdot c_{-2},\ \ldots\ f^{(k)}(y) = k! \cdot c_{-k}. \quad (3.49)$$

Proof. Let us consider the Taylor expansion for $f(x)$ with $h > 0$

$$f(x+h) = f(x) + f'(x)h + f^{(2)}(x)\frac{h^2}{2} + \ldots$$

by assuming $x = y$ and $h = ①^{-1}$. Obviously, the Taylor expansion of $f(x)$ is unknown for the Infinity Computer. Due to the rules of its operation, while calculating $f(y + ①^{-1})$, different exponents of ① are simply collected in independent groups with finite grossdigits. Since functions $f(x)$, $f'(x), f^{(2)}(x), \ldots f^{(k)}(x)$ assume purely finite values or are equal to zero over the interval $[a, b]$ with a, b which are also purely finite, the highest grosspower in the number (3.48) is necessary less or equal to zero. Thus, the number that the Infinity Computer returns can have only a finite and infinitesimal parts. The fact that four arithmetical operations (see [94]) executed by the Infinity Computer with the operands having finite integer grosspowers in the form (3.47) produce only results with finite integer grosspowers concludes the proof. □

This result can be used to calculate the values $z_i = f(x_i), z_i' = f'(x_i)$ for each method belonging to the general scheme described in Subsection 3.2.2. To do this, it is sufficient to evaluate $f(x)$ at the point $x = x_i + ①^{-1}$. The result is a number in the form (3.47) from where we obtain easily the values of $f(x_i)$ and $f'(x_i)$ using d_0 and d_{-1} as shown in Theorem (3.3.1).

For instance, let us illustrate the application of the Theorem 3.3.1 by the following example which requires to calculate the first 3 derivatives of the following function

$$f(x) = \frac{e^x}{x+1}$$

at the point $x^* = 1$ and evaluate the function at the same point. The exact values are:

$$f(1) = \frac{e}{2}, \quad f'(1) = \frac{e}{4},$$
$$f''(1) = \frac{e}{4}, \quad f'''(1) = \frac{e}{8},$$

which can be approximated by

$$f(1) \approx 1.359140914229523,\ f'(1) \approx 0.679570457114761,$$

$$f''(1) \approx 0.679570457114761,\ f'''(1) \approx 0.339785228557381.$$

Let us calculate now these values on the Infinity Computer (all computations have been performed in MATLAB version R2016b using the software simulator of the Infinity Computer, all the numbers have been rounded according to the standard "double" floating-point format). First, the value $f(1+①^{-1})$ is calculated:

$$f(1+①^{-1}) \approx 1.359140914229523①^0 + 0.679570457114761①^{-1} +$$

$$+0.339785228557381①^{-2} + 0.056630871426230①^{-3} + ...,$$

from where one can obtain that

$$\begin{array}{rcrcl} f(1) & \approx & 1.359140914229523, & & \\ f'(1) & \approx & 0.679570457114761 \cdot 1! & = & 0.679570457114761, \\ f''(1) & \approx & 0.339785228557381 \cdot 2! & = & 0.679570457114762, \\ f'''(1) & \approx & 0.056630871426230 \cdot 3! & = & 0.339785228557380. \end{array}$$

The results of these multiplications by 2! and 3! differ from the approximations of the exact values in the last digit only because the intermediate values have been rounded according to the standard "double" floating-point format.

The five methods discussed in Subsection 3.2.2 were implemented both in the traditional floating-point arithmetic and in the Infinity Computing framework allowing one to efficiently compute exact derivatives in the case where the optimized function is given as a black box. In the traditional floating-point arithmetic analytical formulae for the first derivative have been used providing so exact values of $f'(x)$. In the Infinity Computing implementation the derivatives have been calculated numerically also giving exact values of $f'(x)$. Both implementations have given identical results in the numerical experiments described below confirming so that the Infinity Computing opens very interesting horizons in numerical global optimization in situations

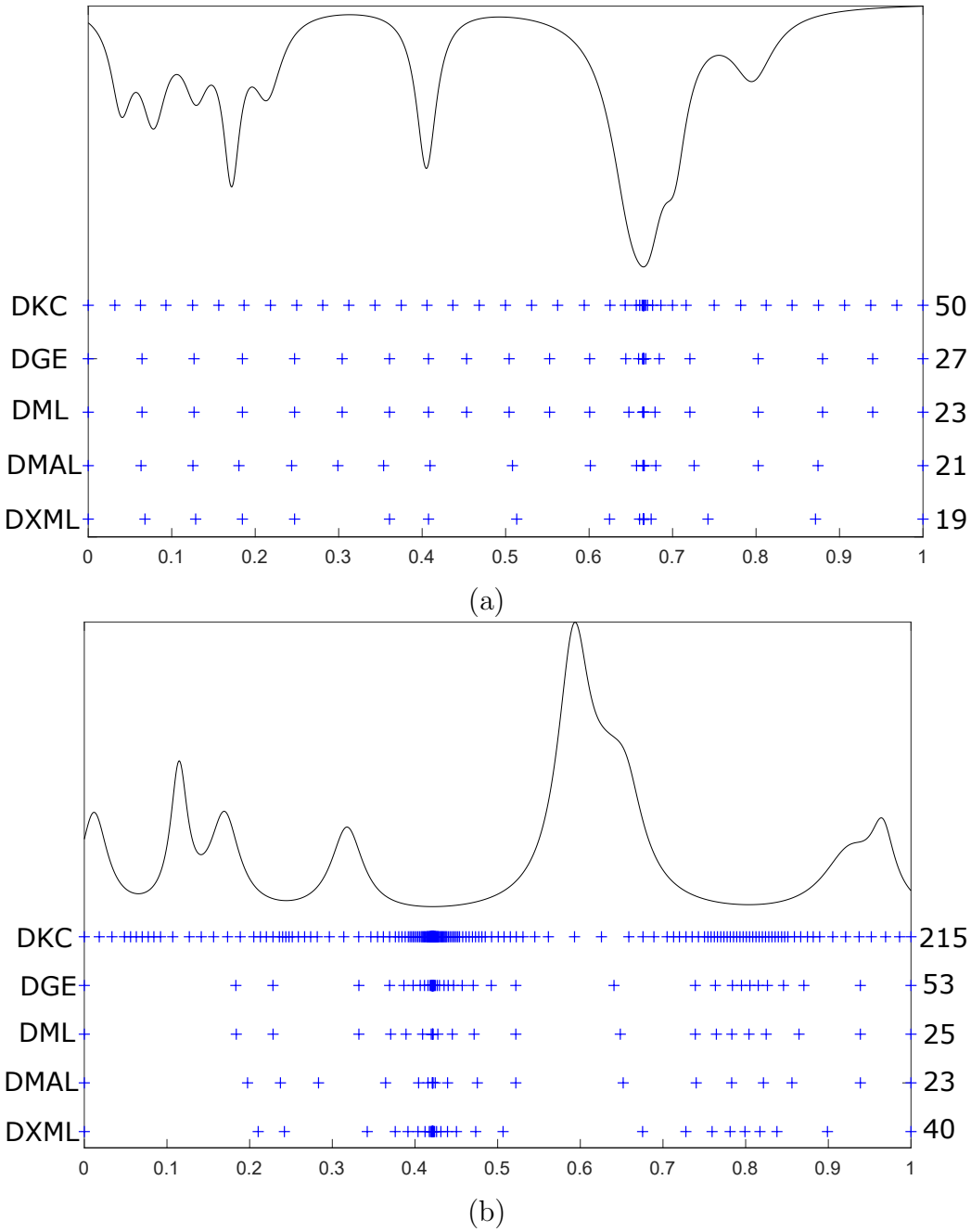

Figure 3.8. Graphs of a function from Class 1 (a) and from Class 2 (b) and trial points generated by the five methods during their work.

where the objective function is given as a black box and a code for computing derivatives is absent.

We compared all the methods on the two classes of randomly generated test functions presented in Section 3.1 : 100 Shekel type test functions (Class 1) and the opposite of 100 Shekel type test functions (Class 2). For both classes, as the global solution x^* there was taken a value obtained after evaluating test functions using a mesh with the step 10^{-7}.

These two classes of functions have been chosen for tests because, in general, they are difficult for global optimization algorithms. In fact, functions from Class 1 have pronounced narrow peaks at many minima and, therefore, algorithms have troubles in finding the global optimum. In their turn, functions from Class 2 have a high Lipschitz constant but are relatively flat at many minima and this fact slows down the search significantly. Moreover, functions of this class usually contain local minima with values very close to the value of the global minimum. Clearly, this complicates finding the global solution.

Three series of experiments have been executed. For each method the technical parameter ξ from (3.28) was set to 10^{-8}. As concerns the first two series, the reliability parameter r_1 was obtained starting from the initial value 1.1 and it was increased with step equal to 0.1 until at least the 90% of all test problems were solved, i.e., the tested algorithm has generated a point x^k after k trials such that $|x^k - x^*| \leq \varepsilon$ with $\varepsilon = 10^{-4}$ for the first series and $\varepsilon = 10^{-6}$ for the second series of experiments. For the remaining unsolved problems the parameter r_2 was used. It was obtained starting from r_1 and it was increased with step equal to 0.1 until the remaining problems were all solved. Figures 3.8(a) and 3.8(b) show two examples of application of the methods respectively on a function from Class 1 and on a function from Class 2 together with trial points generated by the five methods during their work with these functions.

As the objective functions $f(x)$ are considered to be hard to evaluate, the number of trials was chosen as the comparison criterion. We reported in Tables 3.3–3.6, for each method on both classes, the parameters r_1, the averages of trials (AVG 1) and the percentages of test problems solved using r_1 in the columns 2–4, respectively; the parameters r_2, the averages of trials (AVG 2) to solve the remaining problems and the weighted averages

(Weighted AVG) in the last columns 5–7, where

$$\text{Weighted AVG} = \frac{\text{AVG } 1 \cdot n_1 + \text{AVG } 2 \cdot (100 - n_1)}{100},$$

and n_1 is the number of problems solved using r_1. Best results are shown in all tables in bold.

As can be seen from Tables 3.3–3.6, the algorithms DML, DMAL, and DMXL using the local tuning strategies show the most promising behavior. Due to this reason, the third series of experiments was carried out with these three methods only. This series of experiments has been done starting from the following observation regarding unsolved problems of the first two series of experiments with the parameter r_1. It has been realized that the main reason for the failure was that algorithms did not have enough information about the local Lipschitz constants during the first iterations and used therefore significantly smaller values of the estimates m_i w.r.t. the real values causing a premature stop of the methods after a very small number of iterations being so an indicator of this abnormal situation.

To overcome this problem, the parameter $r = r(k)$ dependent on the number of trials k has been chosen. The main idea was to choose a value $n \geq 2$ of trials such that for $k \leq n$, the parameter $r_1(k)$ is relatively high in order to obtain a sufficient information on the behavior of the objective function, while for $k > n$ the parameter was stated to the previously tested value, i.e., $r_1(k) = 1.1$. For the remaining unsolved problems the parameter r_2 equal to the previously used value of $r(k), k \leq n$, has been taken. Two values, $n = 5$ and $n = 10$, have been applied and the best result was included in Table 3.7. The choice of the parameter $r_1(k)$ has been done using the values r_2 from Table 3.3, i.e., $r_1(k) = r_2, k \leq n$, for all the cases but DMXL on Class 2 where $r_1(k) = r_2 + 0.2$ has been taken. Accuracy $\varepsilon = 10^{-4}$ was used in all the experiments. As can be seen from Table 3.3 and Table 3.7, using a high value of the parameter r at the initial iterations improves performance of the algorithms.

In conclusion the experiments have shown a promising behavior of both global optimization methods using the introduced local tuning techniques for speeding up the process of the global search. In particular, the DMXL

Table 3.3. Results of numerical experiments with $\varepsilon = 10^{-4}$ on Class 1.

Method	r_1	**AVG 1**	**Success**	r_2	**AVG 2**	**Weighted AVG**
DKC	-	41.32	100%	-	-	41.32
DGE	1.1	36.64	92%	1.5	56.50	38.23
DML	1.1	34.42	90%	1.8	60.80	37.06
DMAL	1.2	29.92	90 %	2.4	52.90	32.22
DMXL	1.1	26.78	90 %	2.8	59.20	**30.02**

Table 3.4. Results of numerical experiments with $\varepsilon = 10^{-4}$ on Class 2.

Method	r_1	**AVG 1**	**Success**	r_2	**AVG 2**	**Weighted AVG**
DKC	-	237.36	100%	-	-	237.36
DGE	1.1	98.72	97%	1.3	94.67	98.60
DML	1.1	38.61	96%	1.3	31.00	38.31
DMAL	1.1	31.11	95 %	1.3	38.40	**31.47**
DMXL	1.3	73.55	95 %	1.5	74.40	73.59

algorithm behaved better than its competitors for functions having narrow pronounced peaks and the method DMAL worked better for functions having wide zones of attractions of local minima with low Lipschitz constants.

Table 3.5. Results of numerical experiments with $\varepsilon = 10^{-6}$ on Class 1.

Method	r_1	**AVG 1**	**Success**	r_2	**AVG 2**	**Weighted AVG**
DKC	-	45.79	100%	-	-	45.79
DGE	1.1	40.25	92%	1.5	61.13	41.92
DML	1.1	37.39	90%	1.8	65.90	40.24
DMAL	1.2	33.37	90 %	2.4	59.20	35.95
DMXL	1.1	29.82	90 %	2.8	66.40	**33.48**

Table 3.6. Results of numerical experiments with $\varepsilon = 10^{-6}$ on Class 2.

Method	r_1	**AVG 1**	**Success**	r_2	**AVG 2**	**Weighted AVG**
DKC	-	377.24	100%	-	-	377.24
DGE	1.1	156.18	97%	1.3	158.33	156.24
DML	1.1	41.49	96%	1.3	35.00	41.23
DMAL	1.1	34.09	95 %	1.3	42.60	**34.52**
DMXL	1.3	115.06	95 %	1.5	115.40	115.08

Table 3.7. Results for the third series of experiments.

	Class 1			**Class 2**		
Method	DML	DMAL	DMXL	DML	DMAL	DMXL
$\mathbf{r_1(k)}$ **for** $\mathbf{k \le n}$	1.8	2.4	2.8	1.3	1.3	1.7
n	10	10	10	5	5	5
$\mathbf{r_1(k)}$ **for** $\mathbf{k > n}$	1.1	1.1	1.1	1.1	1.1	1.1
AVG 1	34.91	28.56	27.12	36.55	29.06	60.06
Success	92%	90%	91%	100%	100%	97%
$\mathbf{r_2}$	1.8	2.4	2.8	-	-	1.7
AVG 2	60.00	52.00	58.56	-	-	98.00
Weighted AVG	36.92	30.90	**29.95**	36.55	**29.06**	61.20

Chapter 4

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

In this chapter, 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 900 randomly generated test functions taken from the literature show a promising performance of algorithms employing Peano curve approximations w.r.t. their direct competitors.

A wide variety of real-life applications requires a deep study of global

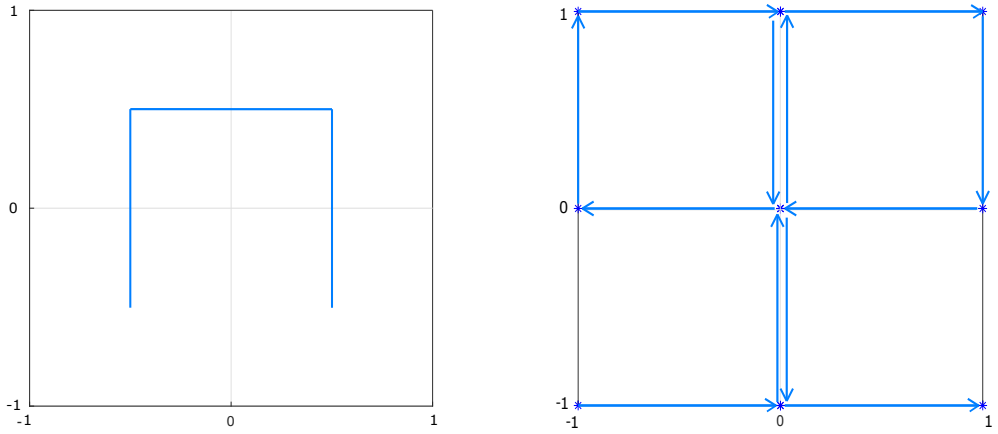

Figure 4.1. Peano piecewise-linear approximation (left) and Peano non-univalent approximation (right) for $N = 2$, $M = 1$.

optimization problems with many local minima and maxima (see. e.g. [4, 7, 74, 75, 84, 91, 107, 109, 116, 121, 122]). In particular, applications in engineering, machine learning, electronics, optimal control, etc. (see, e.g., [3, 11, 12, 16, 23, 24, 46, 53, 54, 59, 96, 97, 103]) are interested 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

$$F^* = F(y^*) = \min F(y), \quad y \in D, \tag{4.1}$$

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.,

$$|F(y') - F(y'')| \leq L\|y' - y''\|, \quad y', y'' \in D, \tag{4.2}$$

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. [30, 37, 51, 76, 99, 101, 102, 110, 111, 117, 118]).

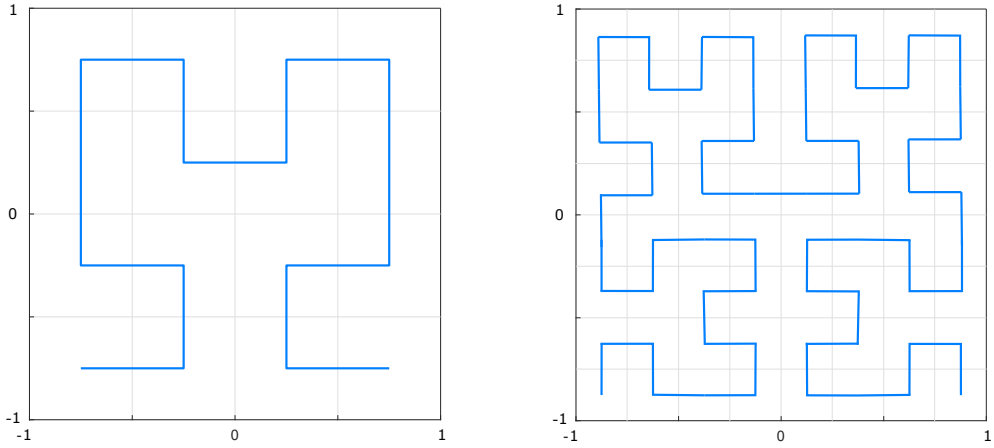

Figure 4.2. Peano piecewise-linear approximation for $N = 2$ and $M = 2, 3$.

It has been shown (see, e.g., [118]) that solving the problem (4.1), (4.2) is equivalent to solve the following one-dimensional problem:

$$f(x^*) = F(y(x^*)) = \min F(y(x)), \quad x \in [0, 1], \tag{4.3}$$

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 [77] passes through every point of $[a, b]^N$ (in the present paper, Hilbert's version of a Peano curve proposed in [42] will be used). In addition, it can be also proved (see [118]) that the function $f(x)$ from (4.3) satisfies the Hölder condition

$$|f(x') - f(x'')| \leq H|x' - x''|^{1/N}, \quad x', x'' \in [0, 1], \tag{4.4}$$

with the constant

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

Thus, we can use one-dimensional algorithms proposed in [110, 118] to solve (4.3), (4.4). Obviously, computable approximations to the Peano curve should be used in numerical algorithms. In several works (see, e.g., [62, 64, 65, 110, 118]) 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 $2^{-M}(b - a)$.

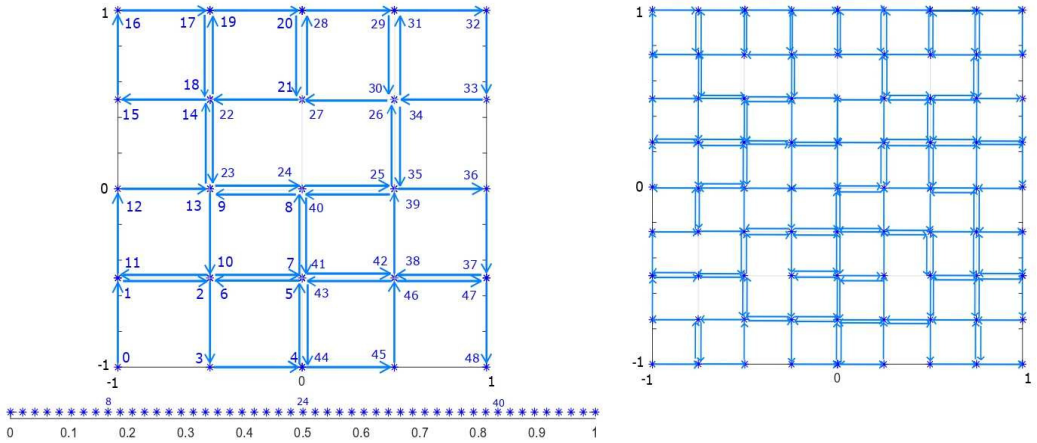

Figure 4.3. Peano non-univalent approximation for $N = 2$ and $M = 2, 3$.

In this chapter, 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 [118]). Fig. 4.1 compares $l_1(x)$ (left picture) with $n_1(x)$ (right picture) and Fig. 4.2 shows $l_2(x)$ and $l_3(x)$ with $D = [-1, 1]^2$ whereas Fig. 4.3 presents $n_2(x)$ and $n_3(x)$. In [118], $n_M(x)$ was applied in so-called information methods. In contrast, in this work, 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 [62, 64, 65, 110, 118]). 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 [118]) 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

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

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 $F(n_M(\overline{x}))$ for a point $\overline{x}$ of the uniform grid in $[0,1]$ allows one to know the values $F(n_S(\overline{x_i}))$, where $S \geq M$ and the s, $1 \leq s \leq 2^N$, points $\overline{x_i} \in [0,1]$, $1 \leq i \leq s$, are the *inverse images* of $n_M(\overline{x})$ with respect to the correspondence $n_M(x)$.

In Fig. 4.3.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 $M \geq 2$.

In the next section, theorems discussing solution to the problem (4.1), (4.2) using Peano non-univalent approximations are given. In Section 4.2, two new geometric optimization algorithms are introduced and their convergence conditions are established. Finally in Section 4.3, we discuss results of some numerical experiments.

4.1 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 [62] 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. 4.1–4.3) 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 4.1.1. *Let U_M^* be a lower bound along $n_M(x)$ for a multidimen-*

sional 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

$$U = U_M^* - 2^{-(M+1)} L\sqrt{N}(b-a), \tag{4.5}$$

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

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

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

$$|F(y) - F(\alpha(y))| \leq L\|y - \alpha(y)\|,$$

$$F(y) \geq F(\alpha(y)) - L\|y - \alpha(y)\|.$$

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

$$F(y) \geq U_M^* - Ld_M, \tag{4.6}$$

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

$$d_M = 2^{-(M+1)}\sqrt{N}(b-a). \tag{4.7}$$

Thus, from (4.6) and (4.7) we obtain

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

that concludes the proof. □

Corollary 4.1.2. *The following condition holds*

$$\min\{F(n_M(x)) : x \in p(M,N)\} - F^* \leq 2^{-(M+1)} L\sqrt{N}(b-a), \qquad (4.8)$$

where F^* from (4.1) 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 (4.5) 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 (4.8). □

The theorem and its corollary mean, in particular, that, given the required accuracy to solve the problem (4.1), (4.2), 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.

4.2 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 (4.1), (4.2). 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.

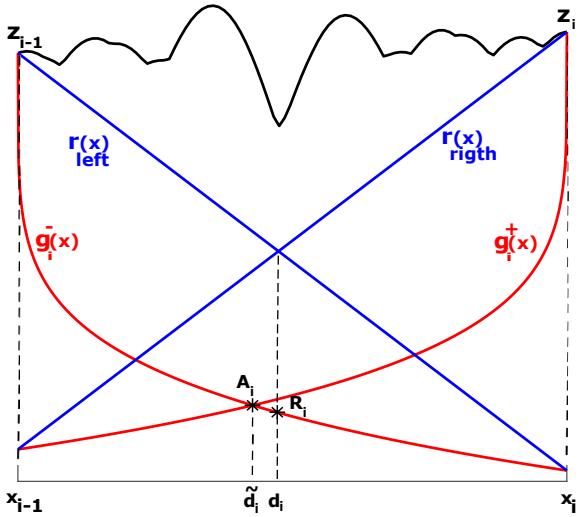

Figure 4.4. The point $(\tilde{d}_i, A_i)$ is approximated by the point (d_i, R_i).

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

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

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

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 [33] that $G^c(x)$ is a minorant over $[0,1]$ for the function $f(x)$ which satisfies (4.4), (see Fig. 4.4). In [33], 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. 4.4). However, as it has been shown in [60], 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 [60, 62, 110] 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 [60] 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

$$H_c = \max\{\xi, M_c\}, \tag{4.9}$$

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

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

with

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

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

$$d_i = 0.5(x_i + x_{i-1}) - \frac{z_i - z_{i-1}}{2rH_c(x_i - x_{i-1})^{\frac{1-N}{N}}}, \tag{4.10}$$

is found as intersection of the following lines (see an illustration in Fig. 4.4)

$$r_{left}(x) = -rH_c(x_i - x_{i-1})^{\frac{1-N}{N}}x + rH_c(x_i - x_{i-1})^{\frac{1-N}{N}}x_{i-1} + z_{i-1},$$

$$r_{right}(x) = rH_c(x_i - x_{i-1})^{\frac{1-N}{N}}x - rH_c(x_i - x_{i-1})^{\frac{1-N}{N}}x_i + z_i,$$

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

$$R_i = \min\{z_{i-1} - rH_c(d_i - x_{i-1})^{1/N}, z_i - rH_c(x_i - d_i)^{1/N}\}, \tag{4.11}$$

is found as the minimum value of the auxiliary functions $g_i^-(x), g_i^+(x)$ evaluated at the point d_i. Fig. 4.4 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 $r_{left}(x), r_{rigth}(x)$.

Thus, for each interval $[x_{i-1}, x_i]$, $2 \le i \le c$, 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

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

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 $w_1, ..., w_j$, 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 $\{w_1, ..., w_j\}$ 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 $x_1, x_2, ..., x_c$. 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=width of the grid p(M,N)
$W_2(1) = W_1(1)$
for $i = 1 : length(W_1)$ **do**
  if $abs(W_1(i) - W_2(end)) > s$ **then**
    $W_2(end + 1) = W_1(i)$
  end
  else if $0 < (x^{c+1} - W_2(end)) < s$ **then**
    $W_2(end) = x^{c+1}$
  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 algoritms:

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 points $x^1, x^2, .., x^c$, and the corresponding function values $z^1, z^2, ..., z^c$, of the previous c trial points by subscripts so that

$$0 = x_1 < x_2 < ... < x_c = 1,$$

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

Step 2. *Estimates of the Hölder constant.* Calculate the current estimates H_c of the Hölder constant using (4.9).

Step 3. *Calculation of characteristics.* For each interval $[x_{i-1}, x_i]$, $2 \leq i \leq c$, compute the point d_i from (4.10) and the characteristic R_i from (4.11).

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 $\tilde{x} = \arg\min\{F(n_M(x_i)), i = 1, ..., c.\}$ 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 $w_1, ..., w_j$, 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 $x^{c+1} \cup \{w_1, ..., w_j\}$. 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 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 accumulation.

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 [62] 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 4.2.1. *Let $f(x) = F(y(x))$ be the objective function which satisfies (4.4) 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*

$$r \cdot H_c > H \tag{4.12}$$

holds, where H_c calculated at Step 2 at the kth iteration (see (4.9)) is an estimate of the Hölder constant H from (4.4) 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 [118] where analogous results for the piecewise-linear curve with $M = \infty$ and information algorithms have been established (see also [110] 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 [118] 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 (4.12) 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. □

4.3 Numerical experiments

In this section we discuss results of two series of numerical experiments executed to compare the two new methods with the method MGA from [62] produced by the General Scheme employing the Peano piecewise-linear approximation and with the DIRECT algorithm from [45]. The first series of experiments regarding GAP1, GAP2, and MGA were performed using MATLAB R2020b. The results of experiments using DIRECT were taken from [100]. In GAP1, GAP2, and MGA the technical parameter ξ from (4.9) was set to 10^{-8}. The value $\varepsilon = 10^{-3}$ was used in Selection rule 3.1 for GAP1. The GKLS-generator described in [26] was used to test the methods since it is very popular in the global optimization community generator which 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 [100] and used in the experiments are briefly described in Table 4.1 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 (4.2) 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

$$|y'(i) - y^*(i)| \leq \sqrt[N]{\Delta}(b - a), \quad 1 \leq i \leq N, \tag{4.13}$$

then the global minimizer y^* was considered found and the method stopped giving y' as an approximation to the solution for the problem (4.1). 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 iteration is less than 15.000 for Classes 1, 2 and 3, less than 50.000 for Classes 4, 5 and 6 and less than 70.000 for Classes 7 e 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 4.2 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. Fig. 4.5 illustrates the comparison between DIRECT and the methods using approximations of Peano curve on a problem of Class 2. In particular, Fig. 4.5 a) shows 3025 points of trials executed by DIRECT to find the global minimum of a problem of Class 2, Fig. 4.5 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 4.3 shows the average number of function evaluations performed during minimization of all 100 functions while Table 4.4 reports the maximum number of function evaluations. The symbol ">" in Table 4.3 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 4.4 means that after 1000000 function 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 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. Fig. 4.6 shows the behavior of the four methods through the operational characteristics (see [34]) constructed on Class 6 (left picture) and Class 8 (right picture) that are graphs showing 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.

In the second series of experiments we performed our experiments using Grishagin's class of test functions which provides 100 two-dimensional multiextremal randomly generated test functions with known global minima y^*

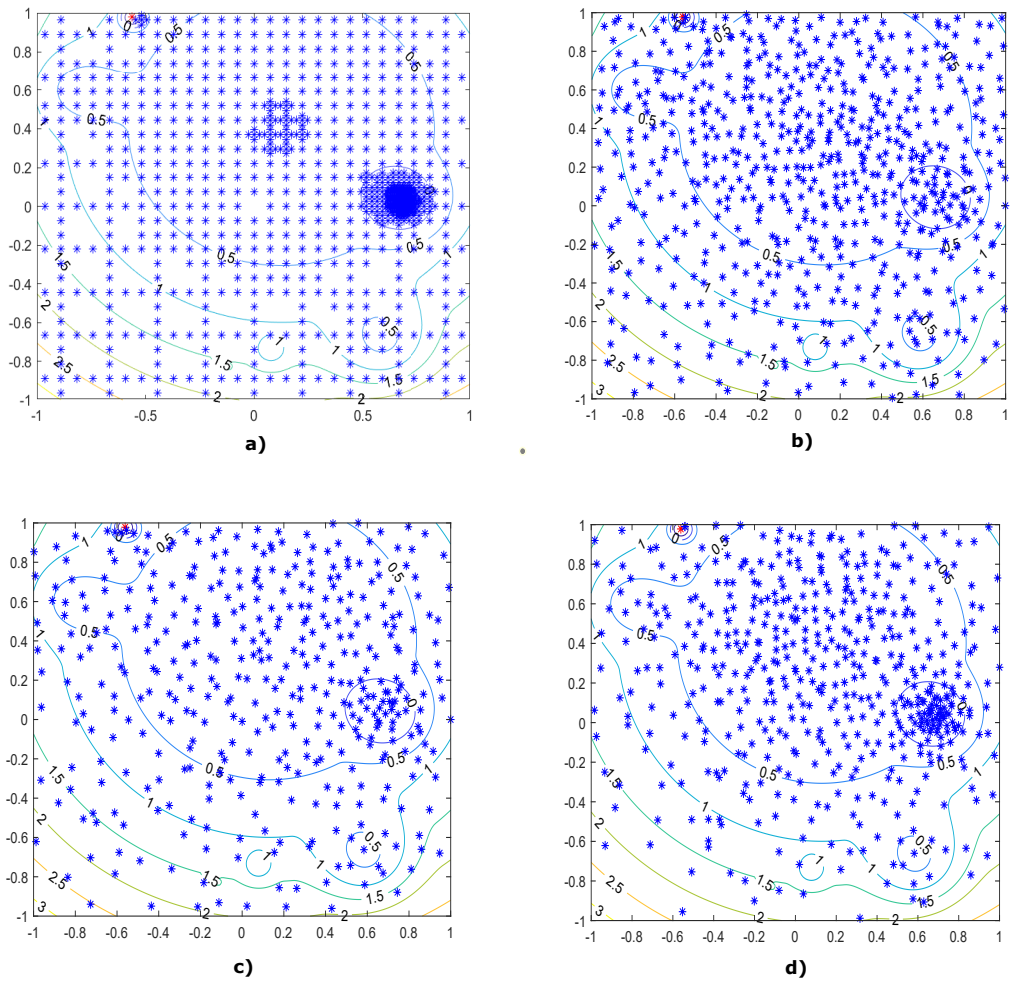

Figure 4.5. Contour lines of a test function and trial points generated by DIRECT (a), MGA (b), GAP1 (c) and GAP2 (d) during their work. Global minimizer is marked by the red symbol "*".

(see [36]) and MATLAB R2020b. An example of a Grishagin's function can be seen in Fig.4.7.

The stopping criterion (4.13) has been used for each algorithm. Tab. 4.5 and Fig. 4.8 show the comparison between DIRECT and the methods using approximations of level 10 of Peano curve. The reliability parameter $r = 1.1$ was used for GAP1, GAP2 and MGA. In particular Figure 4.8 a) shows 1045 points of trials executed by DIRECT to find the global minimum of the

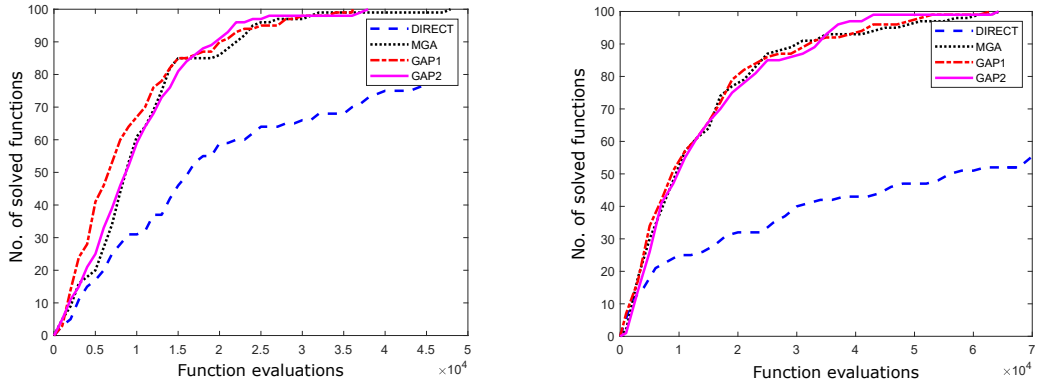

Figure 4.6. DIRECT, MGA, GAP1 and GAP2 on Class 6, $N = 4$ (left) and on Class 8, $N = 5$ (right).

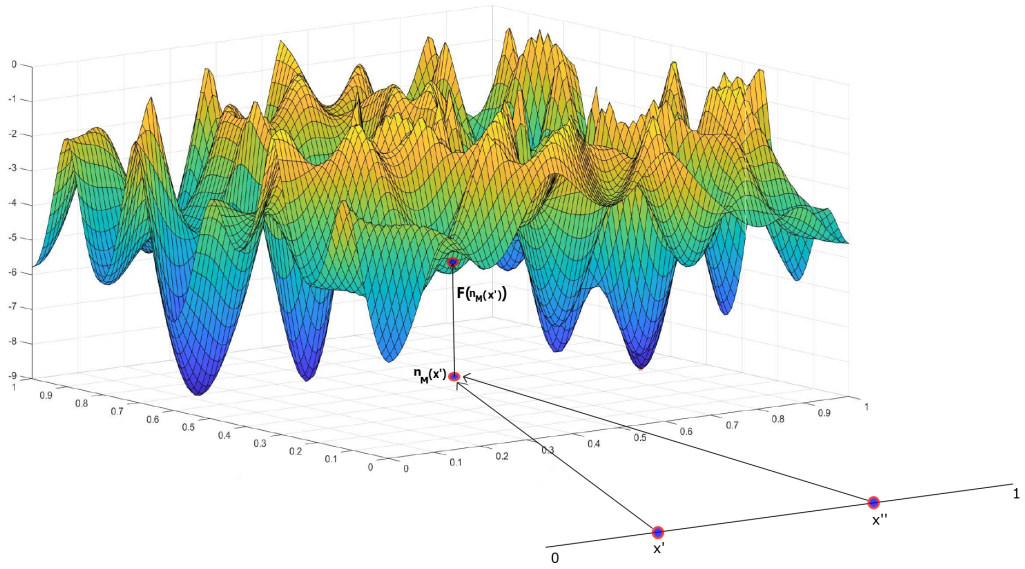

Figure 4.7. A Grishagin's test function. The value $F(n_M(x'))$ was assigned to both inverse images, x' and x''.

problem and Fig. 4.8 b) presents 232 points of trials executed by the GAP2 to solve the same problem.

Thus the introduced algorithms GAP1 and GAP2 giving the opportunity to evaluate only once the objective function $F(n_M(x))$ and introduce

Table 4.1. Parameters setting for GKLS test functions taken from [100]

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	5	10	-1	0.90	0.20
7	5	10	-1	0.66	0.30
8	5	10	-1	0.66	0.20

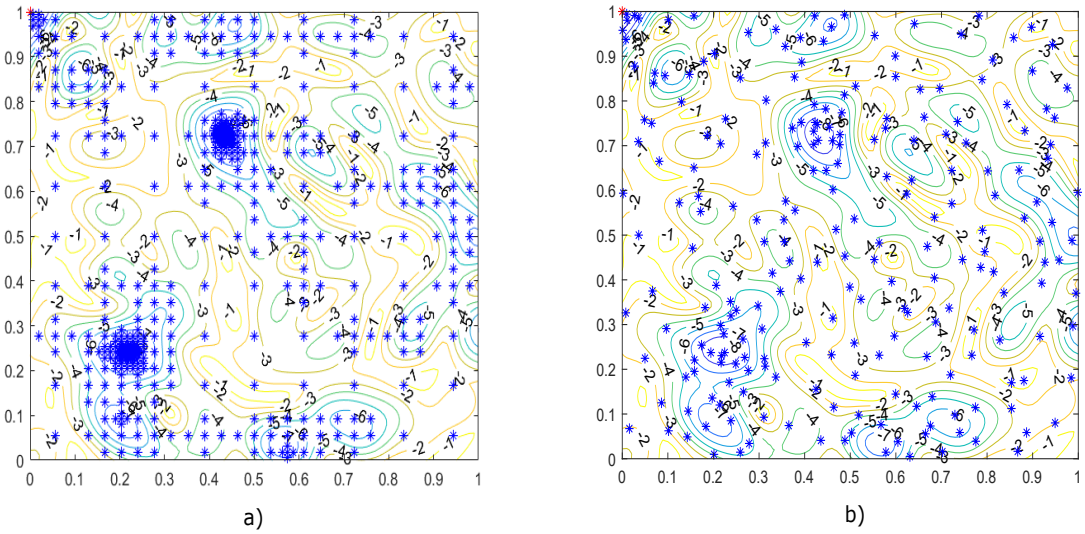

Figure 4.8. Contour lines of a test function and trial points generated by DIRECT a) and GAP2 b) during their work. Global minimizer is marked by the red symbol "*".

the same outcome to all multiple inverse images, led to the chance of reducing the total number of expansive objective function evaluations. Numerical comparison of GAP1, GAP2 with the Geometric Algorithm MGA using the traditional Peano piecewise-linear approximation and with DIRECT algo-

Table 4.2. 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)

Table 4.3. Average number of function evaluations for GKLS test functions

N	**Class**	Δ	**DIRECT**	**MGA**	**GAP1**	**GAP2**
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 4.4. 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

Table 4.5. Average and maximum number of trials in numerical experiments on Grishagin's test functions

	DIRECT	**MGA**	**GAP1**	**GAP2**
average	184.48	206.33	188.87	190.74
max	1045	694	691	550

rithm on 900 randomly generated test functions taken from the literature show a promising performance of all methods using Peano curves and, in particular, of the new algorithms.

Conclusions

The attention of research was mainly devoted to the following:

- Theorical study of Area-filling curves;
- One-dimensional global optimization problem with objective black-box function satisfying the Lipschitz condition over the search region;
- One-dimensional global optimization problem with objective black-box function whose derivative satisfies the Lipschitz condition over the search region;
- Multi-dimensional global optimization problem with the objective black-box function satisfying the Lipschitz condition.

The main obtained results of this thesis consist of the following:

- Variant of Knopp's construction attaining the homogeneity result has been proposed. Several aspects of existence and construction of Area-filling curve are considered. In particular, we generalized the construction of Lance and Thomas and constructed a class of area-filling curves which depend on continuously many parameters. These curves are homeomorphisms between a symmetric Cantor set and their essential image and preserve the measure;
- New acceleration techniques (local tuning techniques) have been proposed for both one-dimensional optimization problems considered. All of them have been studied theoretically and numerically compared on test classes taken from the literature. Infinity Computing framework

allowing one to efficiently compute exact derivatives in the case where the optimized function is given as a black box was also used;

- Different approximations of Peano curve to reduce a multi-dimensional global optimization problem to a univariate one were considered. Two new multi-dimensional geometric algorithms employing a Non-Univalent approximation of Peano curve are introduced and their convergence conditions were established. Numerical experiments executed on 900 test functions show a promising performance of the algorithms.

Based on the obtained results, 2 papers have been published in the international journals and 1 paper has been accepted for publication with minor revisions, 1 contribution to the book and 2 papers in proceedings of the international conferences have been also published:

1. M. C. Nasso, A. Volčič, "Area-filling curves", Arkiv der Mathematik, 2022, `https://doi.org/10.1007/s00013-022-01704-6`;

2. D. Lera, M. C. Nasso, Y. D. Sergeyev, "Non-Univalent Approximation of Peano Curve for Global Optimization" accepted in AIP Conference Proceedings of ICNAAM 2021;

3. M. C. Nasso, Y. D. Sergeyev, Chapter: "Exact Numerical Differentiation on the Infinity Computer and Applications in Global Optimization", in 'Numerical Infinities and Infinitesimals in Optimization', Editors: Yaroslav D. Sergeyev, Renato De Leone, Springer, 2022, in press.

4. Y. D. Sergeyev, M. C. Nasso, M. S. Mukhametzhanov; D. E. Kvasov, "Novel local tuning techniques for speeding up one-dimensional algorithms in expensive global optimization using Lipschitz derivatives', Journal of Computational and Applied Mathematics, Volume 383, February 2021, 113134;

5. D. E. Kvasov, M. S. Mukhametzhanov, M. C. Nasso, Yaroslav D. Sergeyev, "On Acceleration of Derivative-Free Univariate Lipschitz Global

Optimization Methods", Sergeyev Y., Kvasov D. (eds) Numerical Computations: Theory and Algorithms. NUMTA 2019. Lecture Notes in Computer Science, Springer, Cham, 11974:413-421, 2020;

6. Y. D. Sergeyev, M. C. Nasso, D. Lera, "Numerical methods using two different approximations of space-filling curves for black-box global optimization" (accepted with minor revisions).

The obtained scientific results have been present on the following international conferences:

1. 19th International Conference of Numerical Analysis and Applied Mathematic, ICNAAM 2021, Rhodes, Greece, 20-26 Sept. 2021;

2. Biennal Congress SIMAI 2020+2021, Parma 30 Aug-3 Sep 2021;

3. World Congress on Global Optimization, WCGO 2021, Athens, Greece, July 7-10, 2021;

4. International Conference Mathematical Optimization Theory and Operations Research MOTOR 2021, Irkutsk, Russia, July 5-10 2021;

5. Mathematical modeling and supercomputer technologies, Lobachevsky State University of Nizhny Novgorod, 23-25 November 2020;

6. CAE Conference 2019: "The International CAE Conference and Exhibition 2019", Vicenza, Italy, 28-29 October 2019;

7. The 3nd International Conference and Summer School "Numerical Computations: Theory and Algorithms NUMTA-2019", Le Castella Village (VV), Italy, 15-21 June 2019.

Recognition and awards:

- Student Paper Award at the World Congress on Global Optimization, WCGO 2021, Athens, Greece, July 7-10, 2021 for the presentation of "Lipschitz and Hölder global optimization using space-filling curves";

- Highly cited paper (Web of Science) January/February and September/October 2021 for the paper: Y. D. Sergeyev, M. C. Nasso, M. S. Mukhametzhanov; D. E. Kvasov, 'Novel local tuning techniques for speeding up one-dimensional algorithms in expensive global optimization using Lipschitz derivatives', Journal of Computational and Applied Mathematics, Volume 383, February 2021, 113134;

- Finalist of the CAE International Poster Award 2019 (poster "Global Optimization, Infinity Computer and Applications") of "the International CAE Conference and Exhibition 2019", Vicenza, Italy, 28-29 October 2019.

Acknowledgements

I would like to give my warmest thanks to my supervisor Prof. Yaroslav Sergeyev for his guidance, patience, empathy and advice during my Ph.D. course. He has taught me the methodology to carry out the research and to present the research works as clearly as possible. It was a great privilege and honor to work and study under his guidance.

My special gratitude to Prof. Aljoša Volčič for his enthusiasm, immense knowledge, his contribution and help during my study.

I would like to extend my sincere thanks to Prof. Dmitri Kvasov, Dr. Marat Mukhametzhanov and Prof. Daniela Lera for their insightful comments and suggestions.

Bibliography

[1] M.O. Ahmed, S. Vaswani, and M. Schmidt. Combining Bayesian optimization and Lipschitz optimization. *Machine Learning*, 109:79–102, 2020.

[2] P. Amodio, F. Iavernaro, F. Mazzia, M. S. Mukhametzhanov, and Ya. D. Sergeyev. A generalized Taylor method of order three for the solution of initial value problems in standard and infinity floating-point arithmetic. *Mathematics and Computers in Simulation*, 141:24–39, 2017.

[3] F. Archetti and A. Candelieri. *Bayesian Optimization and Data Science*. Springer, New York, 2019.

[4] C. Audet and W. Hare. *Derivative-Free and Blackbox Optimization*. Natural Computing Series. Springer, 2017.

[5] M. Balcerzak and A. Kharazishvili. On uncountable unions and intersections of measurable sets. *Georgian Math. J.*, 6(3):201–212, 1999.

[6] L. Breiman and A. Cutler. A deterministic algorithm for global optimization. *Mathematical Programming*, 58(1-3):179–199, 1993.

[7] S. Busygin, S. Butenko, and P. Pardalos. A heuristic for the maximum independent set problem based on optimization of a quadratic over a sphere. *J. Comb. Optim.*, 6:287–297, 2002.

[8] J. M. Calvin and A. Žilinskas. One-dimensional global optimization for observations with noise. *Comput. Math. Appl.*, 50(1–2):157–169, 2005.

[9] J.M. Calvin, Y. Chen, and A. Žilinskas. An adaptive univariate global optimization algorithm and its convergence rate for twice continuously differentiable functions. *J. Optimization Theory and Applications*, 155(2):628–636, 2012.

[10] J.M. Calvin and A. Žilinskas. One-dimensional P-algorithm with convergence rate O(n-3+δ) for smooth functions. *J. Optimization Theory and Applications*, 106(2):297–307, 2000.

[11] A. Candelieri, I. Giordani, F. Archetti, K. Barkalov, I. Meyerov, A. Polovinkin, A. Sysoyev, and N. Zolotykh. Tuning hyperparameters of a svm-based water demand forecasting system through parallel global optimization. *Computers and Operations Research*, 106:202–209, 2019.

[12] R. Cavoretto, A. De Rossi, M.S. Mukhametzhanov, and Ya. D. Sergeyev. On the search of the shape parameter in radial basis functions using univariate global optimization methods. *Journal of Global Optimization*, 79(2):305–327, 2021.

[13] M. Cococcioni, A. Cudazzo, M. Pappalardo, and Ya. D. Sergeyev. Solving the lexicographic multi-objective mixed-integer linear programming problem using branch-and-bound and grossone methodology. *Communications in Nonlinear Science and Numerical Simulation*, 84:105177, 2020. doi:10.1016/j.cnsns.2020.105177.

[14] M. Cococcioni, M. Pappalardo, and Ya. D. Sergeyev. Lexicographic multi-objective linear programming using grossone methodology: Theory and algorithm. *Applied Mathematics and Computation*, 318:298–311, 2018. doi:10.1016/j.amc.2017.05.058.

[15] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms, 2nd edn.* MIT Press and McGraw-Hill, New York, 2001.

[16] P. Daponte, D. Grimaldi, A. Molinaro, and Ya. D. Sergeyev. An algorithm for finding the zero-crossing of time signals with Lipschitzean derivatives. *Measurement*, 16(1):37–49, 1995.

[17] R. De Leone. Nonlinear programming and grossone: Quadratic programming and the role of constraint qualifications. *Applied Mathematics and Computation*, 318:290–297, 2018.

[18] R. De Leone, G. Fasano, M. Roma, and Ya. D. Sergeyev. Iterative grossone-based computation of negative curvature directions in large-scale optimization. *Journal of Optimization Theory and Applications*, 186(2):554–589, 2020.

[19] R. De Leone, G. Fasano, and Ya. D. Sergeyev. Planar methods and grossone for the conjugate gradient breakdown in nonlinear programming. *Computational Optimization and Applications*, 71:73–93, 2018.

[20] A. Falcone, A. Garro, M.S. Mukhametzhanov, and Ya. D. Sergeyev. Representation of grossone-based arithmetic in Simulink and applications to scientific computing. *Soft Computing*, 24(23):17525–17539, 2020.

[21] A. Falcone, A. Garro, M.S. Mukhametzhanov, and Ya.D. Sergeyev. A Simulink-based infinity computer simulator and some applications. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 11974 LNCS:362–369, 2020. doi:10.1007/978-3-030-40616-5_31.

[22] K. J. Falconer. *Fractal Geometry - Mathematical Foundations and Applications*. Wiley, 1990.

[23] D. Famularo, P. Pugliese, and Ya. D. Sergeyev. Global optimization technique for checking parametric robustness. *Automatica*, 35(9):1605–1611, 1999.

[24] C. A. Floudas and P. M. Pardalos. *State of the Art in Global Optimization*. Kluwer Academic Publishers, Dordrecht, 1996.

[25] M. Gaudioso, G. Giallombardo, and M. S. Mukhametzhanov. Numerical infinitesimals in a variable metric method for convex nonsmooth optimization. *Applied Mathematics and Computation*, 318:312–320, 2018.

[26] M. Gaviano, D. Lera, D. E. Kvasov, and Ya. D. Sergeyev. 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.

[27] B. R. Gelbaum and J. M. H. Olmsted. *Counterexamples in Analysis*. Holden-Day, 1964.

[28] V. P. Gergel. A global search algorithm using derivatives. In Yu.I. Neymark, editor, *Systems Dynamics and Optimization*, pages 161–178. N. Novgorod University Press, 1992.

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

[30] V. P. Gergel, V. A. Grishagin, and R. Israfilov. 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.

[31] V. P. Gergel, V. A. Grishagin, and R. A. Israfilov. Local tuning in nested scheme of global optimization. *Procedia Comput. Sci.*, 51:865–874, 2015.

[32] C. Goffman and G. Pedrick. A proof of the homeomorphism of lebesgue-stieltjes measure with lebesgue measure. *Proc. Amer. Math. Soc.*, 52:196–198, 1975.

[33] E. Gourdin, B. Jaumard, and R. Ellaia. Global optimization of Hölder functions. *Journal of Global Optimization*, 8:323–348, 1996.

[34] V. Grishagin. Operational characteristics of some global search algorithms. *Problems of Stochastic Search*, 7:198–206, 1978.

[35] V. A. Grishagin. On convergence conditions for a class of global search algorithms. In *Numerical methods of nonlinear programming*, pages 82–84, Kharkov, 1979. KSU.

[36] V. A. Grishagin and R. A. Israfilov. Global search acceleration in the nested optimization scheme. In T. Simos and C. Tsitouras, editors, *Proc. of the Int. Conf. of Numer. Analys. and Applied Math. (ICNAAM 2015)*, volume 1738, page 400010. AIP Publishing, NY, 2016. doi:10.1063/1.4952198.

[37] V. A. Grishagin, R. A. Israfilov, and Ya. D. Sergeyev. Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes. *Appl. Math. Comput.*, 318:270–280, 2018.

[38] V.A. Grishagin, Ya. D. Sergeyev, and R.G. Strongin. Parallel characteristic algorithms for solving problems of global optimization. *J. Global Optim.*, 10(2):185–206, 1997.

[39] P. Hajłasz and P. Strzelecki. How to measure volume with a thread. *Amer. Math. Monthly*, 112:176–179, 2005.

[40] K. Hamacher. On stochastic global optimization of one-dimensional functions. *Physica A: Statistical Mechanics and its Applications*, 354(15 August 2005):547–557, 2005.

[41] P. Hansen, B. Jaumard, and H. Lu. Global optimization of univariate Lipschitz functions: 1–2. *Mathematical Programming*, 55:251–293, 1992.

[42] D. Hilbert. Ueber die stetige abbildung einer linie auf ein flächenstück. *Mathematische Annalen*, 38(3):459–460, 1891.

[43] F. Iavernaro, F. Mazzia, M. S. Mukhametzhanov, and Ya. D. Sergeyev. Conjugate-symplecticity properties of Euler–Maclaurin methods and their implementation on the Infinity Computer. *Applied Numerical Mathematics*, 155 September:58–72, 2020. doi:10.1016/j.apnum.2019.06.011.

[44] F. Iavernaro, F. Mazzia, M. S. Mukhametzhanov, and Ya. D. Sergeyev. Computation of higher order Lie derivatives on the Infinity Computer. *Journal of Computational and Applied Mathematics*, 383(113135), 2021.

[45] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79:157–181, 1993.

[46] V. A. Kalyagin, P. M. Pardalos, O. Prokopyev, and I. Utkina. *Computational Aspects and Applications in Large-Scale Networks*. NET 2017, June 2017, Springer Proceedings in Mathematics & Statistics, Nizhny Novgorod, Russia, 2017.

[47] K. Knopp. Einheitliche erzeugung und darstellung der kurven von peano. *Arch. Math. Phys.*, (26):103–115, 1917.

[48] D. E. Kvasov. Algoritmi diagonali di ottimizzazione globale lipschitziana basati su una efficiente strategia di partizione. *Bollettino U.M.I. 10-A (Serie VIII)(2), 225-258*, 2007.

[49] D. E. Kvasov, M.S. Mukhametzhanov, M.C. Nasso, and Ya. D. Sergeyev. On acceleration of derivative-free univariate Lipschitz global optimization methods. *Sergeyev Y., Kvasov D. (eds) Numerical Computations: Theory and Algorithms. NUMTA 2019. Lecture Notes in Computer Science, Springer, Cham*, 11974:413–421, 2020.

[50] D. E. Kvasov, C. Pizzuti, and Ya. D. Sergeyev. Local tuning and partition strategies for diagonal GO methods. *Numer. Math.*, 94(1):93–106, 2003.

[51] D. E. Kvasov and Ya. D. Sergeyev. Multidimensional global optimization algorithm based on adaptive diagonal curves. *Computational Mathematics and Mathematical Physics*, 43(1):40–56, 2003.

[52] D. E. Kvasov and Ya. D. Sergeyev. A univariate global search working with a set of Lipschitz constants for the first derivative. *Optimization Letters*, 3(2):303–318, 2009.

[53] D. E. Kvasov and Ya. D. Sergeyev. Lipschitz global optimization methods in control problems. *Automation and Remote Control*, 74(9):1435–1448, 2013.

[54] D. E. Kvasov and Ya. D. Sergeyev. Deterministic approaches for solving practical black-box global optimization problems. *Adv. Eng. Softw.*, 80:58–66, 2015.

[55] T. Lance and E. Thomas. Arcs with positive measure and a space-filling curve. *Amer. Math. Monthly*, 98(2):124–127, 1991.

[56] H. Lebesgue. *Intégrale, longueur, aire, These.* Faculté des Sciences de Paris, 1902.

[57] H. Lebesgue. Sur le problème des aires. *Bull. Soc. Math. France*, 31:197–203, 1903.

[58] D. Lera, M. C. Nasso, and Sergeyev Ya. D. Non-Univalent Approximation of Peano Curve for Global Optimization. In *Proc. of the Int. Conf. of Numer. Analys. and Applied Math. (ICNAAM 2021)*. in press.

[59] D. Lera, M. Posypkin, and Ya.D. Sergeyev. Space-filling curves for numerical approximation and visualization of solutions to systems of nonlinear inequalities with applications in robotics. *Applied Mathematics and Computation*, 390:125660, 2021.

[60] D. Lera and Ya. D. Sergeyev. Global minimization algorithms for Hölder functions. *BIT*, 42(1):119–133, 2002.

[61] D. Lera and Ya. D. Sergeyev. An information global minimization algorithm using the local improvement technique. *J. Global Optim.*, 48(1):99–112, 2010.

[62] D. Lera and Ya. D. Sergeyev. Lipschitz and Hölder global optimization using space-filling curves. *Applied Numerical Mathematics*, 60(1):115–129, 2010.

[63] D. Lera and Ya. D. Sergeyev. Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. *SIAM J. Optim.*, 1(23):508–529, 2013.

[64] D. Lera and Ya. D. Sergeyev. Deterministic global optimization using space-filling curves and multiple estimates of Lipschitz and Hölder constants. *Communications in Nonlinear Science and Numerical Simulation*, 23(1-3):328–342, 2015.

[65] D. Lera and Ya. D. Sergeyev. GOSH: derivative-free global optimization using multi-dimensional space-filling curves. *Journal of Global Optimization*, 71(1):193–211, 2018.

[66] B. B. Mandelbrot. *Les objets fractals: forme, hasard et dimension.* Flammarion, Paris, 1975.

[67] B. B. Mandelbrot. *The Fractal Geometry of Nature.* W. H. Freeman and Co., United States, 1982.

[68] V. Y. Modorskii, D. F. Gaynutdinova, V. P. Gergel, and K. A. Barkalov. Optimization in design of scientific products for purposes of cavitation problems. In *Proc. of the Int. Conf. of Numer. Analys. and Applied Math. (ICNAAM 2015)*, volume 1738, page 400013. AIP Publishing, New York, 2016.

[69] M. C. Nasso and Ya. D. Sergeyev. Chapter: Exact Numerical Differentiation on the Infinity Computer and Applications in Global Optimization, in 'Numerical Infinities and Infinitesimals in Optimization'. Editors: Yaroslav D. Sergeyev, Renato De Leone, Springer, 2022, in press.

[70] M.C. Nasso and A. Volčič. Area-filling curves. *Archiv der Mathematik*, 2022. `https://doi.org/10.1007/s00013-022-01704-6`.

[71] E. Netto. Beitrag zur mannigfaltigkeitslehre. *J. Reine Angew. Math.*, 86:263–268, 1879.

[72] W. F. Osgood. A Jordan curve of positive area. *Trans. Amer. Math. Soc.*, 4(1):107–112, 1903.

[73] J. C. Oxtoby and S. M. Ulam. Measure-preserving homeomorphisms and metrical transitivity. *Ann. of Math.*, 2(42):874–920, 1941.

[74] P. M. Pardalos and J. B. Rosen. *Constrained Global Optimization: Algorithms and Applications*, volume 268 of *Springer Lecture Notes In Computer Science*. Springer–Verlag, New York, 1987.

[75] R. Paulavičius, Ya. D. Sergeyev, D. E. Kvasov, and J. Žilinskas. Globally-biased BIRECT algorithm with local accelerators for expensive global optimization. *Expert Systems with Applications*, 144:113052, 2020.

[76] R. Paulavičius and J. Žilinskas. *Simplicial Global Optimization*. Springer, New York, 2014.

[77] G. Peano. Sur une courbe, qui remplit toute une aire plane. *Math. Ann.*, 36(1):157–160, 1890.

[78] J. D. Pintér. A unified approach to globally convergent one-dimensional optimization algorithms. *Technical Report No. IAMI-83.5, Istituto per le Applicazioni della Matematica e dell'Informatica, CNR, Milan, Italy*, 1983.

[79] J. D. Pintér. Extended univariate algorithms for N-dimensional global optimization. *Computing*, 36(1-2):91–103, 1986.

[80] J. D. Pintér. Convergence qualification of adaptive partition algorithms in global optimization. *Mathematical Programming*, 56(1-3):343–360, 1992.

[81] J. D. Pintér. *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications).* Kluwer, Dordrecht, 1996.

[82] S. A. Piyavskij. An algorithm for finding the absolute extremum of a function. *USSR Comput. Math. Math. Phys.*, 12(4):57–67, 1972.

[83] K. Ravishankar and H. Salas. On the existence of locally heavy arcs. *Rev. Un. Mat. Argentina*, 36:101–110, 1990.

[84] L. M. Rios and N. V. Sahinidis. Derivative-free optimization: A review of algorithms and comparison of software implementations. *J. Global Optim.*, 56:1247–1293, 2013.

[85] H. Sagan. A geometrization of lebesgue's space-filling curve. *Math. Intelligencer*, 15(4):37–43, 1993.

[86] H. Sagan. *Space-filling curves.* Universitext, New York, 1994.

[87] Ya. D. Sergeyev. An information global optimization algorithm with local tuning. *SIAM J. Optim.*, 5(4):858–870, 1995.

[88] Ya. D. Sergeyev. A one-dimensional deterministic global minimization algorithm. *Comput. Math. Math. Phys.*, 35(5):705–717, 1995.

[89] Ya. D. Sergeyev. A two-points-three-intervals partition of the N-dimensional hyperinterval. *Technical Report 10, ISI-CNR, Institute of Systems and Informatics, Rende(CS), Italy*, 1995.

[90] Ya. D. Sergeyev. Global one-dimensional optimization using smooth auxiliary functions. *Math. Program.*, 81(1):127–146, 1998.

[91] Ya. D. Sergeyev. Efficient strategy for adaptive partition of N-dimensional intervals in the framework of diagonal algorithms. *J. Optim. Theory Appl.*, 107(1):145–168, 2000.

[92] Ya. D. Sergeyev. Univariate global optimization with multiextremal non-differentiable constraints without penalty functions. *Computational Optimization and Applications*, 34(2):229–248, 2006.

[93] Ya. D. Sergeyev. Higher order numerical differentiation on the Infinity Computer. *Optimization Letters*, 5(4):575–585, 2011.

[94] Ya. D. Sergeyev. Numerical infinities and infinitesimals: Methodology, applications, and repercussions on two Hilbert problems. *EMS Surveys in Mathematical Sciences*, 4:219–320, 2017.

[95] Ya. D. Sergeyev. Independence of the grossone-based infinity methodology from non-standard analysis and comments upon logical fallacies in some texts asserting the opposite. *Foundations of Science*, 24(1):153–170, 2019.

[96] Ya. D. Sergeyev, A. Candelieri, D. E. Kvasov, and R. Perego. Safe global optimization of expensive noisy black-box functions in the δ-Lipschitz framework. *Soft Computing*, 2020. published on line. doi:10.1007/s00500-020-05030-3.

[97] Ya. D. Sergeyev, P. Daponte, D. Grimaldi, and A. Molinaro. Two methods for solving optimization problems arising in electronic measurements and electrical engineering. *SIAM J. Optim.*, 10(1):1–21, 1999.

[98] Ya. D. Sergeyev, D. Famularo, and P. Pugliese. Index branch-and-bound algorithm for lipschitz univariate global optimization with multiextremal constraints. *J. Global Optimization*, 21(3):317–341, 2001.

[99] Ya. D. Sergeyev and V. A. Grishagin. A parallel method for finding the global minimum of univariate functions. *J. Optimiz. Theory Appl.*, 80(3):513–536, 1994.

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

[101] Ya. D. Sergeyev and D. E. Kvasov. *Diagonal Global Optimization Methods*. FizMatLit, Moscow, 2008. In Russian.

[102] Ya. D. Sergeyev and D. E. Kvasov. *Deterministic Global Optimization: An Introduction to the Diagonal Approach*. Springer, New York, 2017.

[103] Ya. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhanov. A generator of multiextremal test classes with known solutions for black-box constrained global optimization. *IEEE Transactions on Evolutionary Computation*, page in press.

[104] Ya. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhanov. Operational zones for comparing metaheuristic and deterministic one-dimensional global optimization algorithms. *Mathematics and Computers in Simulation*, 141:96–109, 2017.

[105] Ya. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhanov. On strong homogeneity of a class of global optimization algorithms working with infinite and infinitesimal scales. *Commun. Nonlinear Sci.*, 59:319–330, 2018.

[106] Ya. D. Sergeyev, D. E. Kvasov, and M. S. Mukhametzhanov. On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. *Scientific Reports*, 8(1):453, 2018.

[107] Ya. D. Sergeyev, M. S. Mukhametzhanov, D. E. Kvasov, and D. Lera. Derivative-free local tuning and local improvement techniques embedded in the univariate global optimization. *Journal of Optimization Theory and Applications*, 171(1):186–208, 2016.

[108] Ya. D. Sergeyev, M. C. Nasso, and D. Lera. Numerical methods using two different approximations of space-filling curves for black-box global optimization. accepted with minor revisions.

[109] Ya. D. Sergeyev, M. C. Nasso, M. S. Mukhametzhanov, and D. E . Kvasov. Novel local tuning techniques for speeding up one dimensional algorithms in expensive global optimization using Lipschitz derivatives. *Journal of Computational and Applied Mathematics*, 383(113134), 2021.

[110] Ya. D. Sergeyev, R. G. Strongin, and D. Lera. *Introduction to Global Optimization Exploiting Space-Filling Curves.* Springer, New York, 2013.

[111] Ya.D. Sergeyev. On convergence of "divide the best" global optimization algorithms. *Optimization*, 44(3):303–325, 1998.

[112] B.O. Shubert. A sequential method seeking the global maximum of a function. *SIAM Journal on Numerical Analysis*, 9(3):379–388, 1972.

[113] W. Sierpiński. Sur une courbe non quarrable (sci. math. et nat., série a). *Bull. Acad. Sci. de Cracovie*, pages 254–263, 1913.

[114] K. Stromberg and S. Tseng. Simple plane Arcs of positive Area. *Expo. Math.*, 12:31–52, 1994.

[115] R. G. Strongin. On the convergence of an algorithm for finding a global extremum. *Eng. Cybern.*, 11:549–555, 1973.

[116] R. G. Strongin, K. Barkalov, and S. Bevzuk. Global optimization method with dual Lipschitz constant estimates for problems with non-convex constraints. *Soft Computing*, 24(16):11853–11865, 2020.

[117] R. G. Strongin and Ya. D. Sergeyev. Global multidimensional optimization on parallel computer. *Parallel Computing*, 18(11):1259–1273, 1992.

[118] R. G. Strongin and Ya. D. Sergeyev. *Global Optimization with Non-convex Constraints: Sequential and Parallel Algorithms.* Kluwer Academic Publishers, Dordrecht, 2000.

[119] R. W. Vallin. *The Elements of Cantor Sets with applications*. Wiley, 2013.

[120] H. von Koch. Sur une courbe continue sans tangente, obtenue par une construction géométrique élémentaire. *Arkiv for Matematik*, 1:681–704.

[121] A. Zhigljavsky and A. Žilinskas. *Stochastic Global Optimization*. Springer, New York, 2008.

[122] A. Zhigljavsky and A. Žilinskas. *Bayesian and High-Dimensional Global Optimization*. Springer, New York, 2021.

[123] A. Zhigljavsky and A. Žilinskas. Stochastic global optimization: A review on the occasion of 25 years of Informatica. *Informatica*, 27(2):229–256, 2016.