Safe global optimization of expensive noisy black-box functions in the δ-Lipschitz framework

Yaroslav D. Sergeyev, Antonio Candelieri, Dmitri E. Kvasov, Riccardo Perego

Abstract

In this paper, the problem of safe global maximization (it should not be confused with robust optimization) of expensive noisy black-box functions satisfying the Lipschitz condition is considered. The notion “safe” means that the objective function \( f(x) \) during optimization should not violate a “safety” threshold, for instance, a certain a priori given value \( h \) in a maximization problem. Thus, any new function evaluation (possibly corrupted by noise) must be performed at “safe points” only, namely, at points \( y \) for which it is known that the objective function \( f(y) > h \). The main difficulty here consists in the fact that the used optimization algorithm should ensure that the safety constraint will be satisfied at a point \( y \) before evaluation of \( f(y) \) will be executed. Thus, it is required both to determine the safe region \( \Omega \) within the search domain \( D \) and to find the global maximum within \( \Omega \). An additional difficulty consists in the fact that these problems should be solved in the presence of the noise. This paper starts with a theoretical study of the problem, and it is shown that even though the objective function \( f(x) \) satisfies the Lipschitz condition, traditional Lipschitz minorants and majorants cannot be used due to the presence of the noise. Then, a δ-Lipschitz framework and two algorithms using it are proposed to solve the safe global maximization problem. The first method determines the safe area within the search domain, and the second one executes the global maximization over the found safe region. For both methods, a number of theoretical results related to their functioning and convergence is established. Finally, numerical experiments confirming the reliability of the proposed procedures are performed.

Keywords Safe global optimization · Expensive black-box functions · Noise · Lipschitz condition · Machine learning

1 Introduction

Many complex industrial applications are characterized by black-box, multiextremal, and expensive objective functions that should be optimized (see Archetti and Schoen 1984; Daponte et al. 1996; Jones et al. 1998; Kvasov and Sergeyev 2015; Paulavičius et al. 2014; Sergeyev et al. 1999, 2013, 2018b; Sergeyev and Kvasov 2017). The word expensive means here that each evaluation of the objective function \( f(x) \) is a time-consuming operation. Since locally optimal solutions often do not give a sufficiently high level of improvement w.r.t. a currently available solution obtained by engineers using practical reasons, global optimization problems are considered (see Archetti and Schoen 1984; Cavoretto et al. 2019; Floudas and Pardalos 1996; Gergel and Sidorov 2015; Horst and Pardalos 1995; Jones et al. 1998; Pintér 1996; Sergeyev and Kvasov 2017; Strongin and Sergeyev 2000; Vanderbei 1999; Žilinskas and Zhigljavsky 2016; Žilinskas and Žilinskas 2010). Unfortunately, a practical global optimization process is often performed under a limited budget, i.e., the number of allowed evaluations of \( f(x) \) is fixed a priori and is not very high (see a detailed discussion in Sergeyev et al. 2018b) requiring so an accurate development of fast global optimization methods (see, e.g., Barkalov and Gergel 2016; Barkalov and Strongin 2018; Grishagin et al. 2018; Lera and Sergeyev 2010; Paulavičius...
et al. 2014, 2020; Sergeyev and Kvasov 2017; Strongin and Sergeyev 2000; Žilinskas and Zhigljavsky 2016).

Recently, a class of real applications having an additional important constraint on the value of the objective function is under an accurate study (see, e.g., Berkenkamp et al. 2016; García and Fernández 2012; Sui et al. 2015 in the context of Lipschitz optimization, control problems, reinforcement learning, Bayesian optimization, etc.). This constraint requires that the objective function \( f(x) \) during optimization should not violate a “safety” threshold (that should not be confounded with robust optimization, see, e.g., Ben-Tal et al. 2009), for instance, a certain a priori given value \( h \) in a maximization problem. Thus, any new function evaluation must be performed at “safe points” only, namely, at points \( y \) for which it is known that the objective function value will not violate the safety threshold \( h \), i.e., it should be \( f(y) > h \). The main difficulty here consists in the fact that the used optimization algorithm should ensure that the safety constraint will be satisfied at a point \( y \) before evaluation of \( f(y) \) will be executed.

This requirement is very important and represents a key difference w.r.t. traditional constraint problems where constraints can be checked and the algorithm then behaves in dependence on whether a constraint was satisfied or not. In this kind of problems, it is often allowed to evaluate the objective function \( f(x) \) not only at admissible but also at inadmissible points. In contrast, in the safe optimization problems evaluation of \( f(x) \) at inadmissible points is strictly prohibited. Thus, while formulations using global optimization with unknown constraints is well suited for simulation-optimization problems, the safe global optimization formulation is more appropriate for online control and online learning and optimization problems (see, e.g., Berkenkamp et al. 2016; García and Fernández 2012; Sui et al. 2015).

For instance, let us consider, as an example of the safe optimization problem, tuning parameters of a controller used for automatic process control in the manufacturing industry, especially in the pneumatic, electronic, robotic and automotive domains (see, e.g., Fiducioso et al. 2019; Schillinger et al. 2017). One of the most widely used types of controllers is known as PID, according to the three components: Proportional, Integral, and Derivative. A PID controller implements a control loop mechanism based on feedback: depending on the error value, computed continuously as the difference between a desired setpoint and a measured process variable, the PID controller applies a correction based on its proportional, integral, and derivative terms. Although a software model of the system to control can be used to initially design the PID controller, an expensive and prone-to-failures manual tuning phase is needed to set up, on the real-life system, the optimal values of its parameters from where the need to perform this phase “safely” arises. Depending on the characteristics of the system to control, the safety of the controller is evaluated with respect to one or more of the following issues: responsiveness of the controller, overshooting of the desired setpoint by the measured process variable as well as oscillatory behaviour around the setpoint. In particular, overshooting and oscillatory behaviour could lead to malfunctioning or breakage of the system to control. Thus, searching for the optimal tuning of the PID parameters on a real system, without considering “safety” of the system itself, could damage it, in some cases irreparably.

In some sense, it is required both to determine the safe region \( \Omega \) within the search domain \( D \) and to find the global optimum within the safe region \( \Omega \). Usually, at least one safe point is known before the start of the optimization process (e.g., it is taken using the current working configuration of the optimized industrial system). Thus, it is required to invent a “safe expansion” mechanism to extend the current safe region from the starting point in order to find the whole \( \Omega \). It can happen that, if the safe region \( \Omega \) consists of several disjoint subregions, those subregions which do not contain initially provided safety points will be never found. In this case, it is not possible to talk about the global optimum over the whole safe region \( \Omega \) and a global optimum over the current safe subregion should be found.

A further complication that is frequently present in applied optimization problems (see Archetti and Schoen 1984; Floudas and Pardalos 1996; Jones et al. 1998; Molinaro and Sergeyev 2001a; Pintér 1996; Strongin and Sergeyev 2000; Vanderbei 1999; Žilinskas and Zhigljavsky 2016) is the presence of noise affecting evaluations of the objective function \( f(x) \). These evaluations should remain safe even when they are corrupted by noise. It is difficult to underestimate the importance of taking into consideration the presence of noise since it can make invalid many assumptions (e.g., convexity, derivability, Lipschitz continuity, etc.) usually done w.r.t. optimized functions. In spite of its crucial impact, noise is often not considered in detail in safe global optimization problems, whereas the aforementioned approaches (Berkenkamp et al. 2016; García and Fernández 2012; Sui et al. 2015) do it and provide some probability-based considerations on safety. Precisely these papers have stimulated us to study safe global optimization problems with noise.

Since Lipschitz continuity is a quite natural assumption for applied problems (specifically for technical systems, see, e.g., Gillard and Kvasov 2016; Horst and Pardalos 1995; Kvasov and Sergeyev 2013; Pintér 1996; Sergeyev and Kvasov 2017; Strongin and Sergeyev 2000), we consider here objective functions that satisfy the Lipschitz condition over the search domain \( D \). Thus, our problem becomes Lipschitz global optimization problem broadly studied in the literature (see, e.g., Barkalov and Strongin 2018; Gergel et al. 2015; Gillard and Kvasov 2016; Grishagin et al. 2018; Jones et al. 1998; Kvasov and Sergeyev 2013; Pintér 1996; Sergeyev and
Grishagin 2001; Sergeyev and Kvasov 2017; Strongin and Sergeyev 2000 and references given therein). In this paper, we propose a new Lipschitz-based safe global optimization algorithm, specifically designed to work in the noisy setting. It is proved that, in spite of the presence of the noise, our approach does not permit any violation of the safety threshold. The only assumption made with respect to the noise is its boundedness, with a maximal level of the noise known a priori.

The remaining part of the manuscript is structured as follows. Section 2 contains statement of the problem and its analysis. Section 3 presents a theoretical investigation of a reliable expansion of the safe region and proposes an algorithm realizing this expansion. Section 4 introduces a global maximization algorithm working in the presence of the noise over the found safe region. Section 5 proposes three series of numerical experiments confirming theoretical results and showing a reliable performance of the two introduced methods. Section 6 concludes the paper.

## 2 Statement of the problem and its analysis

In order to start, let us present a general formulation of the safe global optimization problem using the Lipschitz framework in one dimension. As was already mentioned, Lipschitz global optimization problems can be very often encountered in applications even when \( f(x) \) is univariate. Nowadays problems of this kind with and without noise are under an intensive study (see, e.g., Calvin et al. 2012; Calvin and Žilinskas 2005; Casado et al. 2003; Daponte et al. 1996; Kvasov and Mukhametzhanov 2018; Kvasov et al. 2009; Kvasov and Sergeyev 2012, 2015; Lera and Sergeyev 2013; Molinaro and Sergeyev 2001a; Pintér 1996; Sergeyev 1995; Sergeyev et al. 1999, 2001, 2016, 2017, 2018a, 2020; Sergeyev and Kvasov 2017).

To state the problem formally, let us suppose that a function \( f(x) \) satisfies over a search domain \( D = [a, b] \) the Lipschitz condition

\[
|f(x_1) - f(x_2)| \leq L|x_1 - x_2|, \quad 0 < L < \infty, \quad x_1, x_2 \in D, \tag{1}
\]

with an a priori known Lipschitz constant \( L \). Then, given a safety threshold \( h > 0 \), it is required, in the presence of noise \( \xi(x) \), to find an approximation of the point \( x^* \) and an estimate of the corresponding value \( g(x^*) \) such that

\[
x^* = \arg\max_{x \in \Omega \subseteq D} \quad g(x), \quad g(x) = f(x) + \xi(x), \tag{2}
\]

where \( \Omega \) is the safe region that can consist of several disjoint subregions \( \Omega_j, 1 \leq j \leq m \), and the noise \( \xi(x) \) is bounded by a known value \( \delta \), i.e.,

\[
|\xi(x)| \leq \delta, \quad \delta > 0,
\]

\[
\Omega = \{x : x \in D, g(x) \geq h\}, \quad \Omega = \bigcup_{j=1}^{m}\Omega_j, \quad \Omega_i \cap \Omega_j = \emptyset, \quad i \neq j. \tag{4}
\]

Thus, at each point \( x \) the value \( g(x) \) can belong to the set

\[
G(x) = \{y : y = f(x) + \xi(x), \xi(x) \in [-\delta, \delta]\}. \tag{5}
\]

It should be stressed that in traditional noisy optimization problems (see, e.g., Calvin and Žilinskas 2005) the maximizer of \( f(x) \), not of \( g(x) \), is of interest. This is not the case in safe optimization since the function \( g(x) \) is measured and not \( f(x) \) and, therefore, it is important that precisely \( g(x) \) would be as larger than the threshold \( h \) as possible.

In order to illustrate the problem (1)–(5), let us consider Fig. 1. It shows the Lipschitz function \( f(x) = \sin(x) \cdot \cos(4x) \) having \( L = 4 \) and defined over \( D = [-2, 2] \). The safety threshold is \( h = -0.8 \) and the initial safe point is \( x_1 = -0.8 \). It can be seen that \( f(x_1) \neq g(x_1) \in G(x_1) \), in particular \( g(x_1) < f(x_1) \). The unsafe region is shown in red; it consists of two subregions. Notice that in the right-hand unsafe subregion \( f(x) > h \) but due to the presence of the noise this subregion becomes unsafe. The safe region, \( Q \), consists of three subregions \( Q_1, Q_2, \) and \( Q_3 \). The presence of the safe point \( x_1 \) in the central subregion \( Q_2 \) and the knowledge of the Lipschitz constant \( L \) together with supposition \( g(x_1) < f(x_1) \) allow us to build the following Lipschitz minorant

\[
\psi(x, x_1) = g(x_1) - L|x_1 - x|, \quad x \in D,
\]

and to determine the initial safety region shown in green. It should be expanded during optimization using the Lipschitz information, however, since we have only one initial safe point at \( Q_2 \), only this safe subregion can be explored and the left-hand and right-hand safe subdomains \( Q_1 \) and \( Q_3 \) will remain undiscovered.

A strategy for expanding the initial safe region is illustrated in Fig. 2. Clearly, the maximal possible expansion of the currently found safe region can be obtained by evaluating the objective function at the points \( x_2 \) and \( x_3 \) being its extrema. Unfortunately, if the noise is not taken into account appropriately, there is a risk to overestimate the safe region. In real-life applications, at each safe point \( y \) we do not know whether \( g(y) \leq f(y) \) or \( g(y) > f(y) \). For instance, it can be seen in Fig. 2 that in this example both \( g(x_2) > f(x_2) \) and \( g(x_3) > f(x_3) \). Therefore, functions \( \psi(x, x_2) \) and \( \psi(x, x_3) \) constructed using the Lipschitz constant \( L \) of \( f(x) \) at the points \( x_2 \) and \( x_3 \) are not minorants for \( g(x) \) anymore. In

\[\square\] Springer
Fig. 1 An initial known safe point \( x_1 \), the initial safe region (in green) found using the function \( \psi(x, x_1) \), and two unsafe subregions (in red) (color figure online).

Fig. 2 An example of an error in determining the safe region using functions \( \psi(x, x_1), \psi(x, x_2), \text{and} \ \psi(x, x_3) \) occurring due to an inaccurate consideration of the noise.

particular, this fact results in an error in determining the safe region starting from the point \( x_2 \). It can be seen that the region shown in Fig. 2 in hatched red is unsafe but it is considered by the described procedure of the expansion to be safe. The source of this error is the fact that even though the objective function \( f(x) \) satisfies the Lipschitz condition (1), the function \( g(x) \) is not Lipschitzian as it is shown in Theorem 1 below. To prove it, we need the following definition (notice that similar but slightly different functions have been considered in Vanderbei 1999).

Definition 1 A function \( s(x) \) is called \( \delta \)-Lipschitz over the interval \( D \) if it satisfies the following condition

\[
|s(x_1) - s(x_2)| \leq L|x_1 - x_2| + \delta, \quad 0 < L < \infty, \quad 0 < \delta < \infty, \quad x_1, x_2 \in D.
\] (6)

Theorem 1 Suppose that the function \( f(x) \) satisfies the Lipschitz condition (1) with a constant \( L \), then the following two assertions hold:

i. The function \( g(x) \) from (2) does not satisfy the Lipschitz condition.

ii. The function \( g(x) \) from (2) is \( 2\delta \)-Lipschitzian where \( \delta \) is from (3).

Proof Let us first prove the second assertion. It follows from definition of \( g(x) \) in (2), from the Lipschitz condition (1), and from the boundedness of noise in (3) that

\[
|g(x_1) - g(x_2)| = |f(x_1) + \xi(x_1) - f(x_2) - \xi(x_2)|
\leq |f(x_1) - f(x_2)| + |\xi(x_1)| + |\xi(x_2)|
\leq L|x_1 - x_2| + 2\delta.
\] (7)

Let us now consider the assertion (i). If inequality

\[
|g(x_1) - g(x_2)| \leq L|x_1 - x_2|, \quad 0 < L < \infty, \quad x_1, x_2 \in D.
\] (8)

was satisfied [where \( L \) is from (1)], then it would be that \( \lim_{x_1 \to x_2} |g(x_1) - g(x_2)| = 0 \). However, due to (7), \( \lim_{x_1 \to x_2} |g(x_1) - g(x_2)| = 2\delta \) and, therefore, \( g(x) \) does not satisfy (8). \( \square \)
3 A reliable expansion of the safe region

In order to provide a reliable expansion from the initial safe region, it is necessary to ensure that at a new point \( x \) chosen to evaluate \( g(x) \) condition (4) holds, i.e., \( g(z) \geq h \). Moreover, this condition should be satisfied for any value of noise from (3) and, as a consequence, for any value of \( g(x) \in G(x) \) from (5). Let us start to introduce such a mechanism by supposing that \( g(x) \) has been evaluated at several safe points \( x_i \in \Omega, 1 \leq i \leq k \), and introduce the following functions

\[
\varphi(x, x_i) = g(x_i) - L|x_i - x| - 2\delta, \quad x \in D, \tag{9}
\]

where \( \delta \) is from (3). Then, the following result holds.

**Lemma 1** Function \( \varphi(x, x_i) \) satisfies Lipschitz condition (1).

**Proof** To prove Lemma 1 it is required to show that

\[
|\varphi(x', x_j) - \varphi(x'', x_i)| \leq L|x' - x''|, \quad x', x'' \in D.
\]

For any arbitrary \( x', x'' \in D \), it follows that

\[
|\varphi(x', x_j) - \varphi(x'', x_i)| = |g(x_i) - L|x_i - x'| - 2\delta - g(x_j) + L|x_j - x''| + 2\delta|
\]

\[
= |(L|x_i - x'| - L|x_j - x'\rangle| + L|x_j - x'| + 2\delta|
\]

\[
\leq L |(x_i - x''| - x'\rangle| + |x_j - x'| - |x_i - x'|)|
\]

\[
= L|x' - x''|.
\]

Thus, Lemma 1 has been proven. \( \square \)

**Theorem 2** Let us construct the following function

\[
\Phi_k(x) = \max_{1 \leq i \leq k} \varphi(x, x_i), \quad x \in D, \tag{10}
\]

then for any value of noise \( \xi(x) \in [-\delta, \delta] \) it follows that

\[
\Phi_k(x) \text{ is a minorant for the function } g(x) \text{ satisfying condition (7) over the search region } D, \text{ namely, it follows}
\]

\[
\Phi_k(x) \leq g(x) = f(x) + \xi(x), \quad \forall \xi(x) \in [-\delta, \delta], x \in D. \tag{11}
\]

**Proof** Let us consider an arbitrary point \( x \in D \). In order to prove (11), it is sufficient to show that

\[
\Phi_k(x) \leq \min_{\xi(x) \in [-\delta, \delta]} g(x) = f(x) - \delta.
\]

It follows from (9) that there exists an index \( j, 1 \leq j \leq k \), such that

\[
\Phi_k(x) = \varphi(x, x_j) = g(x_j) - L|x_j - x| - 2\delta. \tag{12}
\]

Then, since \( g(x) \) satisfies condition (7), we obtain

\[
g(x_j) - g(x) \leq |g(x_j) - g(x)| \leq L|x_j - x| + 2\delta.
\]

Notice that this estimate holds for any value of \( g(x) \in G(x) \) including, therefore, the minimal possible value of \( g(x) = f(x) - \delta \). As a result we have

\[
g(x_j) - (f(x) - \delta) \leq L|x_j - x| + 2\delta. \tag{13}
\]

By adding and subtracting \( f(x) - \delta \) to (12) and combining it with (13) we get

\[
\Phi_k(x) = g(x_j) - |L|x_j - x| - 2\delta + |f(x) - \delta| - (f(x) - \delta)
\]

\[
\leq f(x) - \delta - |L|x_j - x| - 2\delta + |L|x_j - x| + 2\delta = f(x) - \delta.
\]

Since the point \( x \) has been chosen arbitrarily, the last inequality concludes the proof. \( \square \)

**Corollary 1** Function \( \Phi_k(x) \) from (10) satisfies Lipschitz condition (1).

**Proof** Due to Lemma 1, functions \( \varphi(x, x_i), 1 \leq i \leq k \), satisfy Lipschitz condition (1). Since \( \Phi_k(x) \) is maximum of \( k \) Lipschitz functions satisfying (1), this condition holds for \( \Phi_k(x) \), as well. \( \square \)

Theorems 1 and 2 allow us to substitute unsafe Lipschitz functions \( \psi(x, x_i) \) with safe \( \delta \)-Lipschitz minorants \( \varphi(x, x_i) \) and give a direct suggestion on how the initial safe region should be expanded. The process of expansion is illustrated in Fig. 3. We evaluate \( g(x) \) at the initial safe point \( x_1 \) (the result of evaluation \( g(x_1) \) is shown in Fig. 3 by sign “△”) and construct the minorant \( \varphi(x, x_1) \) (the value \( \varphi(x_1, x_1) = g(x_1) - 2\delta \) is shown by black dot). Then, by using \( \varphi(x, x_1) \), we obtain the initial safe subregion \([x_4, x_3] \). Clearly, since \( \varphi(x, x_1) \leq \psi(x, x_1) \), it follows (cf. Fig. 2) that \([x_4, x_3] \subset [x_2, x_3] \). To expand the safe region, we evaluate \( g(x_4) \) and \( g(x_5) \) and construct \( \varphi(x, x_4) \) and \( \varphi(x, x_5) \). Theorem 2 ensures that the obtained region \([x_6, x_7] \) (shown in green) is safe. The process is repeated and evaluations of \( g(x) \) are iteratively performed at the boundaries of the currently found safe region as follows.

To continue our analysis, let us suppose for simplicity that \( \Omega \) is a simply connected region, i.e., \( m = 1 \) in (4) (if this is not the case then considerations analogous to what follows should be executed for all zones \( \Omega_j, 1 \leq j \leq m \)). Assume that \( g(x) \) has been evaluated at safe points \( x_i, 1 \leq i \leq k \), and let us indicate the points

\[
\hat{x}_k = \min\{x_i : x_i \in \Omega, 1 \leq i \leq k\}, \quad \hat{x}_k = \max\{x_i : x_i \in \Omega, 1 \leq i \leq k\}, \tag{14}
\]

being the current minimal and maximal safe points, respectively. Thus, the current found safe region is \([\hat{x}_k, \hat{x}_k] \). By using functions \( \varphi(x, \hat{x}_k) \) and \( \varphi(x, \hat{x}_k) \), new safe points \( \hat{x}_{k+1} \) and \( \hat{x}_{k+2} \) can be obtained as solutions of equations
evaluations of safe region. Repeated framework reliably expands the safe point correctly starting from the initial taking into account the noise \( \delta \), \( i = 1, 4, 5 \) (compare with Fig. 2).

The introduced in (9) \( \delta \)-Lipschitz minorants allow us to expand the initial safe region using the function evaluation (see Fig.3) is executed at the point \( \xi(x_i) \). If \( \xi(x) \rightarrow \delta \), then the minorant is good, if \( \xi(x) \rightarrow -\delta \), then the minorant is not accurate. This fact can become crucial on the borders of the current safe region. Let us illustrate this difficulty by returning once again to Fig. 3. It can be seen that the value \( \phi(x_4, x_4) \) is very close to the lower boundary of \( G(x_4) \) equal to \( f(x_4) - \delta \) because \( g(x_4) \) is very close to the upper boundary of \( G(x_4) \), namely, to \( f(x_4) + \delta \). In contrast, \( \psi(x_5, x_5) \) is very far away from the lower boundary of \( G(x_5) \) equal to \( f(x_5) - \delta \) because \( g(x_5) \) is very close to the lower boundary of \( G(x_5) \), i.e., to \( f(x_5) - \delta \). Thus, the tightness of the minorant \( \phi(x, x_i) \) depends on the level and sign of noise \( \xi(x) \). If \( \xi(x) \rightarrow \delta \), then the minorant is good, if \( \xi(x) \rightarrow -\delta \), then the minorant is not accurate. This fact can become crucial on the borders of the current safe region. For instance, if the next function evaluation (see Fig. 3) is executed at the point \( x_6 \) and \( g(x_6) \approx f(x_6) + \delta \) then \( \phi(x_6, x_6) \approx f(x_6) - \delta \), the constraint \( \phi(x_6, x_6) > h \) will be satisfied as \( h \) will be less than \( \delta \) and, therefore, it will not be possible to expand the current safe region.

Thus, the introduced \( \delta \)-Lipschitz framework allows us to construct a safe expansion mechanism based on the knowledge of \( L \) and \( \delta \) and to obtain estimates of the left and right margins of the safe region \( \Omega \) (let us call them \( l \) and \( r \), respectively). However, there exists an additional trouble related to situations taking place due to the presence of the noise when the process of expansion approaches borders of the current safe region. Let us illustrate this difficulty by returning once again to Fig. 3. It can be seen that the value \( \phi(x_4, x_4) \) is very close to the lower boundary of \( G(x_4) \) equal to \( f(x_4) - \delta \) because \( g(x_4) \) is very close to the upper boundary of \( G(x_4) \), namely, to \( f(x_4) + \delta \). In contrast, \( \psi(x_5, x_5) \) is very far away from the lower boundary of \( G(x_5) \) equal to \( f(x_5) - \delta \) because \( g(x_5) \) is very close to the lower boundary of \( G(x_5) \), i.e., to \( f(x_5) - \delta \). Thus, the tightness of the minorant \( \phi(x, x_i) \) depends on the level and sign of noise \( \xi(x) \). If \( \xi(x) \rightarrow \delta \), then the minorant is good, if \( \xi(x) \rightarrow -\delta \), then the minorant is not accurate. This fact can become crucial on the borders of the current safe region. For instance, if the next function evaluation (see Fig. 3) is executed at the point \( x_6 \) and \( g(x_6) \approx f(x_6) + \delta \) then \( \phi(x_6, x_6) \approx f(x_6) - \delta \), the constraint \( \phi(x_6, x_6) > h \) will be satisfied as \( h \) will be less than \( \delta \) and, therefore, it will not be possible to expand the current safe region.

This observation suggests that if at a current point, \( z \), that is tested for expanding the safe region using the function
\( \varphi(x, y), y \in \Omega, \) it follows that \( \varphi(z, y) < h, \) then it is necessary to continue to evaluate \( g(z) \) several times trying to obtain a value close to \( f(z) + \delta \) and, therefore, to obtain the best possible minorant. This strategy is illustrated in Fig. 4 where the repeated evaluations of \( g(x) \) at the borders of the current safe region can be clearly seen. These repetitions allow us to establish the borders with a high precision.

In practice, the number of repetitions allowing us to obtain an expansion can be very high and closer we are to the border of the safe region smaller the expansion steps will become. Thus, it is necessary to introduce a parameter \( \nu \) fixing the maximal allowed number of repetitions of evaluations of \( g(x) \) at a point \( z \) during the search of the border and a tolerance for the expansion process. Recall that we know that the level of noise is bounded by \( \delta. \) Thus, if during iterated evaluations of \( g(z) \) two values, \( y_1 \) and \( y_2, y_1 > y_2, \) have been obtained as results of these evaluations, then the following condition using an apriori given accuracy \( \sigma > 0 \) for stopping iterated evaluations of \( g(x) \)

\[
y_1 - y_2 \geq 2\delta - \sigma \quad (16)
\]

can be used.

Let us illustrate the work of the parameters \( \nu \) and \( \sigma \) by the following examples. In the first of them shown in Fig. 4 the safe region for \( h = -0.85 \) has been identified using parameters \( \nu = 15 \) and \( \sigma = 0.05 \cdot 2\delta. \) The search of the safe region has been stopped after 15 repetitions at the lower and 15 repetitions at the upper margin of the current safe region, i.e., the criterion on \( \nu \) has stopped the search.

In Fig. 5, the same values of \( h \) and \( \nu \) have been used but \( \sigma = 0.1 \cdot \delta \) has been taken. Function evaluations at the lower and upper boundaries of the safe region were, respectively, 13 and 10. Namely, at both sides, the search was stopped due to the criterion on \( \sigma. \) Notice that the function \( f(x) \) is the same in both experiments shown in Figs. 4 and 5. However, due to the presence of the noise the resulting functions \( g(x) \) are different.

It should be stressed that, even if the objective function evaluations are repeatedly performed at the boundaries of the estimated safe region to better define them, there is no way to go through unsafe zones to other safe subregions (that could contain the desired global maximum) if these subregions do not contain originally provided safe points. In fact, in this specific example the safe subregion obtained starting from the initial safe point \( x = 0.5 \) shown in Fig. 5 in green does not coincide with the entire safe region \( \Omega. \) As a result, the identified safe subregion does not contain the global maximum over the whole \( \Omega \) since it is located within the unfound subregion of \( \Omega. \)

The importance of the strategy consisting in insisting on repeated evaluations on the borders of the current search region is shown even better in Fig. 6 where the same function \( f(x) \) is considered but the safety threshold is decreased to \( h = -1.1. \) In this case, the threshold \( h \) limits the safe region on the left-hand side only, whereas the right border of \( \Omega \) coincides with the right margin of the search space \( D. \)

It can be seen from Fig. 6 that the safe expansion mechanism is able to correctly estimate the part of the safe region \( \Omega \) containing the global optimum. This happens in spite of the fact that the initial safe point (shown in Fig. 6 in green) is located quite far from the upper margin of \( \Omega \) and a local minimum of \( f(x) \), in between, is quite close to the safety threshold \( h. \) At a neighborhood of this local minimizer a huge number of function evaluations is performed trying to observe a function value which would guarantee a reliable expansion of the current safe region. Since in these subregion \( G(x) \) is very close to \( h, \) the size of each new step of the expansion is quite small. In this example, the safe subregion has been identified using \( \nu = 15 \) and \( \sigma = 0.1 \cdot 2\delta. \) The executed function evaluations at the lower boundary of the safe region were 6 (i.e., the process of the expansion on this side was stopped due to the criterion on \( \sigma. \) The expansion on the upper boundary of \( \Omega \) was stopped because the right margin of the search region \( D \) has been reached.

A detailed description of Algorithm 1 executing the described above procedure of the reliable expansion of the safe region can be found in “Appendix”.

4 Global maximization over the found safe region

Let us suppose now that the \( \delta \)-Lipschitz expansion procedure described in the previous section has stopped after evaluating \( g(x) \) at \( k \) trial points and a desired approximation \( \hat{\Omega}_k \) (indicated just \( \hat{\Omega} \) hereinafter) of the safe region \( \Omega \) has been obtained. After that, the second phase of the algorithm, namely, global maximization over the found safe region \( \hat{\Omega}, \) starts. For simplicity we suppose here, as was done in the previous section, that \( \hat{\Omega} \) is a simply connected region (all the subsequent considerations in this section can easily be extended to the case of disjoint subregions composing \( \hat{\Omega}). \) Notice also that the maximization problem (2) over \( \Omega \) is substituted by its approximation, i.e., by the problem

\[
x^* = \arg\max_{x \in \hat{\Omega}} g(x), \quad g(x) = f(x) + \xi(x), \quad (17)
\]

The resulting problem (1), (17), (3)–(5) is solved by using all previously executed evaluations of \( g(x) \) at the points \( x_i \in \hat{\Omega}, 1 \leq i \leq k \) (see Algorithm 1 in “Appendix”). The maximization procedure constructs a majorant \( \Gamma_k(x), x \in \hat{\Omega}, \) for \( g(x) \) similarly to the construction of the minorant \( \Phi_k(x) \) from (10). Recall that during the safe expansion procedure there could be produced points \( x_i \) where \( g(x) \) has
The search of the safe region stops using the criterion on $\sigma$ from (16) with $\sigma = 0.1 \cdot 2^8$. It can be seen by comparing with Fig. 4 that the density of evaluations at the borders of the safe region is lower.

Fig. 6 The initial safe point (shown in green) is located quite far from the upper margin of $\Omega$ and a local minimum of $f(x)$, in between, is quite close to the safety threshold $h$. At a neighborhood of this local minimizer a huge number of function evaluations is performed trying to observe a value of $g(x)$ which would guarantee a reliable expansion of the current safe region. Since in this subregion $G(x)$ is very close to $h$, the size of each new step of the expansion is quite small in this zone (color figure online).

been evaluated $\nu(x_i)$ times (where $\nu(x_i)$ in any case is not superior to the maximal allowed number, $\nu$, of repeated evaluations of $g(x)$ at a single point). This means that different values $g_j(x_i)$ could be obtained during the $j$th evaluation, $1 \leq j \leq \nu(x_i) \leq \nu$ of $g(x)$ at a point $x_i$, $1 \leq i \leq k$. Recall, that it was necessary to use values

$$\hat{g}(x_i) = \max_{1 \leq j \leq \nu(x_i)} g_j(x_i), \quad 1 \leq i \leq k,$$

(18)

in order to build a better minorant $\Phi_k(x)$. Analogously, in order to build a better majorant $\Gamma_k(x)$, we shall use values

$$\hat{\gamma}(x_i) = \min_{1 \leq j \leq \nu(x_i)} g_j(x_i), \quad 1 \leq i \leq k.$$

(19)

Then, the function $\Gamma_k(x)$ is built as follows:

$$\Gamma_k(x) = \min_{1 \leq i \leq k} \gamma(x, x_i), \quad x \in \hat{\Omega},$$

(20)

$$\gamma(x, x_i) = \hat{g}(x_i) + L|\xi_i - x| + 2\delta, \quad x \in \hat{\Omega},$$

(21)

where $\delta$ is from (3). Both functions, $\Phi_k(x)$ and $\Gamma_k(x)$, are presented in Fig. 7 using the data from Fig. 6.

The following result ensures that

$$\Gamma_k \geq g(x) = f(x) + \xi(x), \quad \forall \xi(x) \in [-\delta, \delta], \quad x \in \hat{\Omega}. \tag{22}$$

Theorem 3 For any value of noise $\xi(x) \in [-\delta, \delta]$ the function $\Gamma_k(x)$ from (20) is a majorant for the objective function $g(x)$ satisfying (7) over the search region $\Omega$, i.e., condition (22) holds.

Proof The theorem follows immediately from Theorem 2. □

Corollary 2 Function $\Gamma_k(x)$ from (20) satisfies Lipschitz condition (1).

Proof It follows from considerations analogous to proof of Lemma 1 that functions $\gamma(x, x_i)$, $1 \leq i \leq k$, satisfy Lipschitz condition (1). Since $\Gamma_k(x)$ is minimum of $k$ Lipschitz functions satisfying (1), this condition holds for $\Gamma_k(x)$, as well. □

In order to estimate the global maximizer of $g(x), x \in \hat{\Omega}$, we propose a procedure that generalizes to the case of maximization of noisy functions the algorithm of Piyavskij (see Piyavskij 1972) proposed for Lipschitz optimization and
working without any noise (this algorithm is called PA hereinafter). The PA works constructing a majorant $\Lambda_k(x)$ similar to $\Gamma_k(x)$ at each iteration using previously executed evaluations of the Lipschitz objective function $f(x)$. The absence of the noise allows the PA to improve $\Lambda_k(x)$ after each evaluation of $f(x)$ that is executed at a point $x_{k+1} = \arg\max_{x \in D} A_k(x)$.

In other words, due to the Lipschitz condition (1) it follows that

$$f(x_{k+1}^{\Lambda_k}) \leq \max_{x \in D} A_k(x), \quad A_{k+1}(x) \leq A_k(x), \quad x \in D,$$

and, as a consequence, after each new evaluation of $f(x)$ the majorant $A_k(x)$ becomes closer and closer to the objective function $f(x)$ allowing the PA to obtain a good approximation of the global maximum.

In contrast, in the noisy framework we deal with, a condition for $\Gamma_k(x)$ similar to (23) cannot be written down since the noise (see Fig. 6) can provoke a situation where

$$g(x_{k+1}^{\Lambda_k}) > \max_{x \in \hat{\Omega}} \Gamma_k(x), \quad x_{k+1} = \arg\max_{x \in \hat{\Omega}} \Gamma_k(x). \quad (24)$$

In cases where the condition (24) holds, the evaluation of $g(x_{k+1}^{\Lambda_k})$ does not improve $\Gamma_k(x)$ and leads to the equivalence $\hat{\Gamma}_{k+1}(x) = \Gamma_k(x)$. Let us recall that we have already faced similar situations before, when we got results of evaluation of $g(x)$ that did not allow us to enlarge the current safe region. Therefore, the remedy will be the same, i.e., execution of $\nu(x_{k+1}^{\Lambda_k})$ repeated evaluations of $g(x)$ at the point $x_{k+1}^{\Lambda_k}$ from (24) until either the number of possible repetitions $\nu$ will be reached or a value less than $\max_{x \in \hat{\Omega}} \Gamma_k(x)$ will be produced. In the former case the search stops and in the latter one [similarly to (19)] the value

$$\tilde{g}(x_{k+1}^{\Lambda_k}) = \min_{1 \leq j \leq \nu(x_{k+1}^{\Lambda_k})} g_j(x_{k+1}^{\Lambda_k}) < \max_{x \in \hat{\Omega}} \Gamma_k(x) \quad (25)$$

is accepted and $\tilde{\Gamma}_{k+1}(x) \leq \Gamma_k(x)$ is built.

In practice, the operations of building $\Gamma_k(x)$ from the trial data $(x_i, \tilde{g}(x_i)), 1 \leq i \leq k$, provided by Algorithm 1 are executed as follows. To each trial point $x_i$, we associate a value...
\[
z_i = \min_{1 \leq j \leq k} \gamma(x_i, x_j), \quad 1 \leq i \leq k.
\]

Then, it follows from the Lipschitz condition (1) that
\[
\max_{x \in [x_{i-1}, x_i]} \Gamma_k(x) = \max_{x \in [x_{i-1}, x_i]} \min_{1 \leq j \leq k} \{z_{i-1} + L(x - x_{i-1}), z_i + L(x_i - x)\} = R_i = 0.5(z_{i-1} + z_i) + 0.5L(x_i - x_{i-1}), \quad 2 \leq i \leq k,
\]
and it is reached at the point
\[
\tilde{x}_i = 0.5(x_i + x_{i-1}) + 0.5(z_i - z_{i-1})/L,
\]
where hereinafter the value \(R_i\) is called **characteristic** of the interval \([x_{i-1}, x_i]\). As a result, it follows that
\[
\max_{x \in \tilde{\Omega}} \Gamma_k(x) = \max_{2 \leq i \leq k} R_i
\]
and, therefore, the point \(x^{k+1}\) from (24) can be calculated as
\[
x^{k+1} = \tilde{x}_t, \quad t = \arg \max_{2 \leq i \leq k} R_i.
\]

If the interval \([x_{t-1}, x_t]\) is larger than the preset accuracy \(\varepsilon\) (i.e., the stopping rule is not satisfied), the process of global maximization tries to update the majorant \(\Gamma_k(x)\) by obtaining a value \(\hat{g}(x^{k+1})\) satisfying (25). In case this is not possible after \(v\) evaluations of \(g(x^{k+1})\), the algorithm stops. Otherwise, the interval \([x_{t-1}, x_t]\) is subdivided in two subintervals \([x_{t-1}, x^{k+1}]\) and \([x^{k+1}, x_t]\), the number of points where \(g(x)\) has been evaluated becomes \(k+1\), the intervals having numbers larger than \(t\) are renumbered and the two new intervals become \([x_{t-1}, x_t]\) and \([x_t, x_{t+1}]\), respectively. Then, the value \(\hat{g}(x^{k+1})\) is assigned to \(z_t\) and all the values \(z_i\) are renewed as follows:
\[
z_i = \min\{z_i, \gamma(x_i, x_t)\}, \quad 1 \leq i \leq k + 1.
\]

This concludes construction of the new majorant \(\Gamma_{k+1}(x)\) and the optimization process can be repeated. A detailed description of Algorithm 2 executing the global maximization over \(\tilde{\Omega}\) is given in “Appendix” (the maximization is performed separately at each subregion of \(\tilde{\Omega}\), but it can also be executed simultaneously over all disjoint subregions of the safe region, thus accelerating the search).

After satisfaction of one of the two stopping rules of Algorithm 2, a lot of information regarding the problem (1), (2), (3)–(5) is returned. First of all, the found safe region \(\tilde{\Omega}\) is provided. Then, the largest found value \(g^*_k\) and the corresponding point \(x^*_k\), namely,
\[
g^*_k = \max\{\hat{g}(x_i) : \quad 1 \leq i \leq k\},
\]
\[
x^*_k = \arg \max\{\hat{g}(x_i) : \quad 1 \leq i \leq k\}
\]
can be taken as an estimate of the global maximum for the safe optimization problem (1), (17), (3)–(5). Then, the minorant \(\Phi_k(x)\) and the majorant \(\Gamma_k(x)\) for functions \(f(x)\) and \(g(x)\) are supplied, i.e.,
\[
\Phi_k(x) \leq f(x) \leq \Gamma_k(x), \quad \Phi_k(x) \leq g(x) \leq \Gamma_k(x), \quad x \in \tilde{\Omega},
\]
where both \(\Phi_k(x)\) and \(\Gamma_k(x)\) are Lipschitz piece-wise linear functions (see Fig. 7). Their simple structure allows one to delimit easily areas of \(\tilde{\Omega}\) where the global maximizers of \(g(x)\) and \(f(x)\) cannot be located. In fact, the set \(N^k_g = \{x : \Gamma_k(x) < g^*_k\}\) cannot contain the global maximizer of \(g(x)\) and \(N^k_f = \{x : \Gamma_k(x) < g^*_k - \delta\}\) cannot contain the global maximizer of \(f(x)\) (see Fig. 8 for illustration). The following lemma establishes that each new trial point produced by Algorithm 2 is chosen with the tentative to reduce the area \(\tilde{\Omega} \setminus N^k_g\) containing the global maximizer of \(g(x)\). This reduction becomes effective in case the condition (24) does not hold at the chosen point \(x^{k+1}\).

**Lemma 2** All trial points produced by Algorithm 2 at any iteration number \(k > 1\) will belong to the area \(\tilde{\Omega} \setminus N^k_g\).

**Proof** Due to its definition, the area \(N^k_g\) contains points \(x\) for which \(\Gamma_k(x) < g^*_k\). It follows from (29) and (30) that the new trial point \(x^{k+1}\) will be chosen by Algorithm 2 at the point corresponding to \(\max_{x \in \tilde{\Omega}} \Gamma_k(x)\). The fact that this value is strictly larger than \(g^*_k\) concludes the proof.

The introduced Algorithm 2 works with the function \(g(x)\) and looks for its global maximizer \(x^*\) from (17). However, sometimes the interest of the person executing the optimization is related to the original function \(f(x)\). Clearly, due to the noise, the global maxima of \(g(x)\) and \(f(x)\) can be different both in value and in location. The following theorem establishes conditions where Algorithm 2 can localize not only the area \(\tilde{\Omega} \setminus N^k_g\) containing the global maximum of the function \(g(x)\) but also can indicate a neighborhood of the global maximizer \(X^*\) of \(f(x)\) over \(\tilde{\Omega}\).

**Theorem 4** Suppose that:

i. The function \(f(x)\) satisfies the Lipschitz condition (1).

ii. The function \(f(x)\) has the only global maximizer \(X^*\) over \(\tilde{\Omega}\) and for any local maximizer \(x'\) of \(f(x)\) such that \(x' \neq X^*\), \(x' \in \tilde{\Omega}\), it follows that
\[
f(X^*) > f(x') + 2\delta + \Delta,
\]
where $\Delta > 0$ is a fixed finite number and $\delta$, as usual, is the maximal level of the noise from \((3)\).

iii. Parameters of Algorithm 2 are chosen as follows: $\varepsilon = 0$, $\nu = \infty$.

Then, during the work of Algorithm 2 there exists an iteration number $d^*$ such that for $k > d^*$ the only local maximizer of $f(x)$ located at the region $\bar{\Omega} \backslash N^k_{\delta}$ is $X^*$.

**Proof** Supposition (iii) regarding parameters of Algorithm 2 means that the majorant $\bar{I}_k(x)$ will be improved at each iteration at each new trial point $x^{k+1}$ because the number of allowed repeated evaluations of $g(x^{k+1})$ is $\nu = \infty$ and, therefore, the algorithm will not stop due to this stopping rule (see lines 48–49 of Algorithm 2). Then, since we have $\varepsilon = 0$ in the second stopping rule, the algorithm will not stop due to this rule (see lines 43 and 70 of Algorithm 2), as well. Thus, supposition (iii) ensures that the sequence of trial points generated by Algorithm 2 is infinite and the majorant $\bar{I}_k(x)$ is improved at each iteration.

Since $f(x)$ satisfies the Lipschitz condition \((1)\) and due to the finiteness of $\Delta$ it follows from supposition (ii) and condition (33) that there exists a neighborhood $\omega(X^*)$ of the point $X^*$ such that

\[
\omega(X^*) = (X^* - \omega, X^* + \omega), \quad \omega \geq (f(X^*) - \Delta)/L, \quad f(x) > f(x') + 2\delta, \quad x \in \omega(X^*), \quad (34)
\]

and there are no other local maximizers of $f(x)$ belonging to $\omega(X^*)$.

By taking into account the definition of $g(x)$ in (2) and (3), condition (34) can be re-written in the form

\[
g(x) \geq f(x) - \delta > f(x') + \delta \geq g(x'), \quad x \in \omega(X^*), \quad x' \in \bar{\Omega},
\]

meaning that for any point $x \in \omega(X^*)$ and any local maximizer $x' \neq X^*, x' \in \bar{\Omega}$, it follows that the value of $f(x)$ decreased by the maximal level of the noise is larger than the value of $f(x')$ increased by the highest level of the noise and, as a result, we have

\[
g(x) > g(x'), \quad x \in \omega(X^*), \quad x' \in \bar{\Omega}, \quad x' \neq X^*. \quad (35)
\]

On the other hand, Algorithm 2 works in such a way that the majorant $\bar{I}_k(x)$ is improved at each iteration and the point of its improvement corresponds to its maximum [see (29) and (30)]. This means that $\max_{x \in \bar{\Omega}} \bar{I}_k(x)$ decreases with the increase of the iteration number $k$ (notice that this decrease is not strict since there can be situations where $\max_{x \in \bar{\Omega}} \bar{I}_k(x)$ is reached at several points). The decrease of $\max_{x \in \bar{\Omega}} \bar{I}_k(x)$ and the presence of a finite neighborhood $\omega(X^*)$ where all points $x \in \omega(X^*)$ satisfy (35) mean that there will exist an iteration number $\tilde{d}$ such that the new point $x^{\tilde{d}}$ will belong to $\omega(X^*)$ and, therefore, the value of the function $g(x^{\tilde{d}})$ will satisfy

\[
g(x^{\tilde{d}}) > \max_{x' \in \bar{\Omega}} g(x')
\]

for all local maximizers $x' \neq X^*, x' \in \bar{\Omega}$. Due to Lemma 2, new trials with the numbers $k > \tilde{d}$ will be executed at regions where $I_k(x) > g(x^{\tilde{d}})$. Therefore, since (36) holds and due to the already mentioned decrease of $\max_{x \in \bar{\Omega}} I_k(x)$, after a finite number $\tilde{d}$ of iterations of Algorithm 2 new trials will not be placed outside $\omega(X^*)$. In fact, suppose that there exists a local maximizer $x' \in \bar{\Omega}$ that does not belong to $\omega(X^*)$, at the $k$-th iteration $x' \in [x_{i(k)-1}, x_{i(k)}]$, and new trials are placed infinitely many times within this interval. It follows from (28) and footnote 1 (see Sect. 2) that in this case

\[
\lim_{k \to \infty} x_{i(k)} - x_{i(k)-1} = 0
\]

and, therefore,

\[
\lim_{k \to \infty} x_{i(k)-1} = x', \quad \lim_{k \to \infty} x_{i(k)} = x'.
\]

As a result, for the characteristic $R_{i(k)}$ from (27) of this interval we have

\[
\lim_{k \to \infty} R_{i(k)} \leq g(x')
\]

where the sign $\leq$ here is due to the rule (31). However, since $g(x^{\tilde{d}}) > g(x')$, the new trial cannot be placed within the interval $[x_{i(k)-1}, x_{i(k)}]$ and, therefore, our supposition that new trial points will be placed in this interval infinitely many times is false.

Thus, by taking $d^* = \tilde{d} + \tilde{d}$ and reminding that $\omega(X^*)$ does not contain other local maximizers we conclude the proof. \hfill $\Box$

## 5 Numerical experiments

The experiments have been organized on three test cases, where the parameters of the $\delta$-Lipschitz framework have been set as follows: both $\delta$ and $h$ have been set at 10% of the min-
Fig. 9 Graphical results of the δ-Lipschitz framework on Test case 1
Safe global optimization of expensive noisy black-box functions in the $\delta$-Lipschitz framework.

Fig. 10  Graphical results of the $\delta$-Lipschitz framework on Test case 2, Part 1
max range of $f(x)$, $\epsilon = 0.001$, $\nu = 15$, and $\sigma = 0.10$. For all test cases, initial safe points have been chosen randomly.

- **Test case 1** This test case consists of four test problems. The safe expansion phase can lead to the identification of a unique safe region. Table 1 presents the structure of the test problems used in this test case. Figure 9 shows for each test problem the final state of each phase in optimization processes (safe expansion—phase 1 and global maximization—phase 2).

- **Test case 2** This test case consists of six test problems. The safe expansion phase can lead to the identification of the maximum of two disconnected safe regions. Table 2 presents the structure of the test problems used in this test case. Figures 10 and 11 show for each test problem the final state of each phase in optimization processes (safe expansion—phase 1 and global maximization—phase 2).

- **Test case 3** This test case consists of eight test problems. The safe expansion phase can lead to the identification of multiple disconnected safe regions. Table 3 presents the structure of the test problems used in this test case. Figures 12 and 13 show for each test problem the final state of each phase in optimization processes (safe expansion—phase 1 and global maximization—phase 2).

Table 4 presents results of the numerical experiments, and Figs. 9, 10, 11, 12 and 13 illustrate them. Figures related to Phase 1 show initial safe points by green dots and results of evaluations by symbol ‘x’; they present also the constructed minorant $\Phi_k(x)$ and the found safe region. Figures related to Phase 2 show results of evaluations executed during the global maximization by blue symbols ‘x’. The best found point is shown by symbol ‘+’ and the global maximizer by symbol ‘*’. These figures show also the constructed majorant $\Gamma_k(x)$ together with the functions $\gamma(x, x_i)$ from (21) used to build $\Gamma_k(x)$. The black dots above the majorant represent values that either have not improved the majorant or better values [see (31)] than these ones have been obtained during the maximization process.
Safe global optimization of expensive noisy black-box functions in the $\delta$-Lipschitz...

Fig. 12  Graphical results of the $\delta$-Lipschitz framework on Test case 3, Part 1
Fig. 13 Graphical results of the $\delta$-Lipschitz framework on Test case 3, Part 2
Safe global optimization of expensive noisy black-box functions in the $\delta$-Lipschitz…

Table 1  Test case 1

| Problem | Lipschitz Function | Interval | $L$ | Threshold | References |
|---------|--------------------|----------|-----|-----------|------------|
| 1       | $f(x) = -\frac{1}{5}x^6 + \frac{52}{17}x^5 -\frac{59}{16}x^4 -\frac{71}{18}x^3 + \frac{79}{19}x^2 + x - \frac{1}{10}$ | $[-1.5, 11]$ | 13870 | 2974.180 | Sergeyev et al. (2016) |
| 2       | $f(x) = -\sin x^3 - \cos x^3$ | $[0, 6.28]$ | 2.2 | -0.800 | Molinaro and Sergeyev (2001b) |
| 3       | $f(x) = x - \sin 3x + 1$ | $[0, 6.5]$ | 4 | 1.202 | Molinaro and Sergeyev (2001b) |
| 4       | $f(x) = x^2 - x^4$ | $[-5, 5]$ | 6.5 | 0.671 | Sergeyev et al. (2016) |

Table 2  Test case 2

| Problem | Lipschitz function | Interval | $L$ | Threshold | References |
|---------|--------------------|----------|-----|-----------|------------|
| 5       | $f(x) = -\sin x - \sin \frac{10}{x}$ | $[2.7, 7.5]$ | 4.29 | -0.609 | Hansen et al. (1992) |
| 6       | $f(x) = (-3x + 1.4) \sin 18x$ | $[0, 1.2]$ | 36 | -1.271 | Hansen et al. (1992) |
| 7       | $f(x) = (x + \sin x)e^{-x^2}$ | $[-10, 10]$ | 2.5 | -0.659 | Hansen et al. (1992) |
| 8       | $f(x) = -\sin x - \sin \frac{1}{3}x$ | $[3.1, 20.4]$ | 1.7 | -1.483 | Hansen et al. (1992) |
| 9       | $f(x) = e^{-x} \sin 2\pi x$ | $[0, 4]$ | 6.5 | -0.347 | Hansen et al. (1992) |
| 10      | $f(x) = -e^{-x} \sin 2\pi x + 0.5$ | $[0, 4]$ | 6.5 | -0.154 | Molinaro and Sergeyev (2001b) |

Table 3  Test case 3

| Problem | Lipschitz function | Interval | $L$ | Threshold | References |
|---------|--------------------|----------|-----|-----------|------------|
| 11      | $f(x) = \sum_{i=1}^{5}(i+1)x_i + 3$ | $[-10, 10]$ | 67 | -24.335 | Molinaro and Sergeyev (2001b) |
| 12      | $f(x) = \cos x - \sin 5x + 1$ | $[0, 7]$ | 5.951 | -0.545 | Molinaro and Sergeyev (2001b) |
| 13      | $f(x) = \begin{cases} \cos 5x, & x \leq \frac{5}{2}
\cos x, & \text{otherwise} \end{cases}$ | $[0, 18]$ | 4.999 | -0.800 | Molinaro and Sergeyev (2001b) |
| 14      | $f(x) = \begin{cases} \sin x, & x \leq \pi
\sin 5x, & \text{otherwise} \end{cases}$ | $[-10, 10]$ | 4.999 | -0.800 | Molinaro and Sergeyev (2001b) |
| 15      | $f(x) = \sum_{i=1}^{5}(i+1)x_i$ | $[-10, 10]$ | 18.119 | -4.229 | Molinaro and Sergeyev (2001b) |
| 16      | $f(x) = |x| \sin x + 6$ | $[-10, 10]$ | 9.632 | -0.332 | Molinaro and Sergeyev (2001b) |
| 17      | $f(x) = |x \sin x| - 1.5$ | $[-10, 10]$ | 9.632 | -0.709 | Molinaro and Sergeyev (2001b) |
| 18      | $f(x) = \begin{cases} \sin x, & x > \cos x
\cos x, & \text{otherwise} \end{cases}$ | $[-10, 10]$ | 1 | -0.519 | Molinaro and Sergeyev (2001b) |

Table 4  The number of points at which the objective function has been evaluated and the number of these evaluations executed by Algorithms 1 and 2 on the three test cases

| Problem | Safe expansion  | Global maximization | Total |
|---------|------------------|---------------------|-------|
| Points  | Evaluations      | Points              | Evaluations |
| 1       | 15 41            | 20 31               | 35 72 |
| 2       | 10 43            | 82 178              | 92 221 |
| 3       | 16 44            | 10 19               | 26 63 |
| 4       | 40 95            | 39 99               | 79 194 |
| 5       | 52 191           | 29 33               | 81 224 |
| 6       | 59 92            | 27 30               | 86 122 |
| 7       | 150 209          | 128 209             | 278 418 |
| 8       | 53 245           | 68 169              | 121 414 |
| 9       | 380 1002         | 393 403             | 773 1495 |
| 10      | 71 95            | 35 44               | 106 139 |
| 11      | 210 485          | 170 196             | 380 681 |
| 12      | 57 184           | 57 97               | 114 281 |
| 13      | 99 441           | 55 74               | 154 515 |
| 14      | 101 441          | 35 61               | 136 502 |
| 15      | 93 255           | 89 108              | 182 363 |
| 16      | 54 136           | 46 81               | 100 217 |
| 17      | 59 249           | 74 169              | 133 418 |
| 18      | 27 191           | 29 35               | 56 226 |
6 Conclusions

The problem of safe global maximization of an expensive black-box function $f(x)$ satisfying the Lipschitz condition has been considered in this paper. The word “expensive” means here that each evaluation of the objective function $f(x)$ is a time-consuming operation. With respect to the traditional formulations used in global optimization, the problem under consideration here has two important distinctions:

i. The first difficulty consists in the presence of the noise. As a result, instead of the function $f(x)$ a function $g(x)$ distorted by the noise is optimized.

ii. The second difficulty is related to the notion of safe optimization. The term “safe” means that the objective function $g(x)$ during optimization should not violate a “safety” threshold, that in our case is a certain a priori given value $h$ in the maximization problem. Any new function evaluation must be performed at “safe points” only, namely, at points $y$ for which it is known that the objective function $g(y) \geq h$. Clearly, the main difficulty here consists in the fact that the used optimization algorithm should ensure that the safety constraint will be satisfied at a point $y$ before evaluation of $f(y)$ will be executed.

In our approach, the problem under investigation has been split in two parts (phases):

1. During the first phase, it is required to find an approximation $\hat{\Omega}$ of the safe region $\Omega$ within the search domain $D$ by learning from the information received from evaluations of $g(x)$.

2. Then, during the second phase, it is necessary to find an approximation of the global maximum of $g(x)$ over $\hat{\Omega}$.

A theoretical study of the problem has been performed, and it has been shown that even though the objective function $f(x)$ satisfies the Lipschitz condition, traditional Lipschitz minorants and majorants cannot be used in this context due to the presence of the noise. In fact, a counterexample showing that the usage of simple Lipschitz ideas can lead to erroneous evaluations of $g(x)$ at unsafe points has been presented.

Then, a $\delta$-Lipschitz framework has been introduced and an algorithm determining the safe area within the search domain $D$ has been proposed. It has been theoretically proven that this method is able to construct a minorant ensuring that all executed evaluations of $g(x)$ performed by the method will be safe. It has been shown that the introduced procedure allows one to expand reliably the safe region from initial safe points and to obtain the desired approximation $\hat{\Omega}$ of the safe region $\Omega$.

After that the second algorithm executing the safe global maximization over the found safe region $\hat{\Omega}$ has been proposed. It has been theoretically investigated and conditions allowing one to make conclusions with respect to not only the noisy function $g(x)$ but also regarding the original function $f(x)$ have been established. It should be stressed that the introduced algorithm is able not only to provide lower and upper bounds for the global maxima of $g(x)$ and $f(x)$ but to restrict significantly the area of a possible location of the respective global maximizers, as well.

Finally, numerical experiments executed on a series of problems showing the reliability of the proposed procedures have been performed.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.
Appendix

Algorithm 1 A reliable expansion of the safe region

1: INPUT:
2: \( D = [a, b] \) search space
3: \( L \) Lipschitz constant
4: \( \delta \) maximal quantity of noise: \( |\xi(x)| \leq \delta \)
5: \( h \) safety threshold
6: \( S_{i=1,n} \) the set of \( n \) initial safe points, with \( S_i = [a_i, b_i] \), where
7: \( x_i = b_i = x_i. \) Safe expansion starts from every initial safe point
8: leading to the identification of an equal or lower number of safe
9: regions
10: \( r, r_h \) the sets of noisy function values observed at \( x_i \) and \( \beta_j \), respect-
11: ively, namely repetitions
12: \( \varepsilon \) minimal expansion (if the expansion induced by the new function
13:  evaluation is lower than \( \varepsilon \), then the function evaluation is considered
14:  as a repetition at the current point)
15: \( \sigma \) tolerance (stop criterion for expansion
16: depends on \( \varepsilon \))
17: OUTPUT:
18: \( \Omega = \cup_j \Omega_j \) approximation of the safe region \( \Omega \)
19: \( P \) the set of trial points (for the Global Maximization (Algorithm 2))
20: INITIALIZATION:
21: \( A^k, B^k = \emptyset \), where: \( A^k = \{i = 1, \ldots, n : a_i^k\} \) cannot be expanded
22: any longer and \( B^k = \{i = 1, \ldots, n : b_i^k\} \) cannot be expanded any
23: longer
24: \( k \leftarrow 0 \) (current iteration)
25: \( r_{a_i} \leftarrow r_{b_i} \leftarrow \emptyset \)
26: \( P \leftarrow S_{i=1,n} \)
27: for \( i = 1 : n \) do
28: \# merging overlapping safe regions
29: if \( \exists S_j^i : a_i^k < b_j^k, j = 1, ..., n, j \neq i \) then
30: \( A^k \leftarrow A^k \cup \{i\} \)
31: end if
32: if \( \exists S_j^i : b_j^k > \alpha_i^k, j = 1, ..., n, j \neq i \) then
33: \( B^k \leftarrow B^k \cup \{i\} \)
34: end if
35: \# safe expansion towards extreme \( a \)
36: if \( g(\alpha_i^k) - 2\delta > h \) then
37: \( \alpha_i^{k+1} \leftarrow \alpha_i^k + \frac{|g(\alpha_i^k) - 2\delta - h|}{L} \)
38: if \( |\alpha_i^{k+1} - \alpha_i^k| < \varepsilon \) then
39: \( r_{a_i} \leftarrow r_{a_i} \cup \{g(\alpha_i^k)\} \)
40: else
41: \( r_{a_i} \leftarrow \emptyset \)
42: \( P \leftarrow P \cup \max(a, \alpha_i^{k+1}) \)
43: end if
44: else
45: \( \alpha_i^{k+1} \leftarrow \alpha_i^k \)
46: \( r_{a_i} \leftarrow r_{a_i} \cup \{g(\alpha_i^k)\} \)
47: end if
48: end if
49: if \( |r_{a_i}| = \nu \max_i(r_{a_i}) - \min_i(r_{a_i}) > 2\delta - \sigma \alpha_i^k < a \) then
50: \( A^{k+1} \leftarrow A^k \cup \{i\} \)
51: end if
52: end for
53: end if
54: end if
55: end if
56: end if
57: end if
58: \( \Omega = \cup_j \Omega_j \) where \( \Omega \) is the found approximation of the safe
59: region \( \Omega \), which can consist of several disjoint safe subregions. \( P \),
the set of trial points

Algorithm 2 Global Maximization

1: INPUT:
2: \( L \) Lipschitz constant
3: \( \delta \) the maximal quantity of noise (as in Algorithm 1), see (3)
4: \( \varepsilon \) accuracy of the global maximization (as in Algorithm 1)
5: \( \sigma \) the maximal number of allowed repetitions of evaluations of \( g(x) \)
(stopping criteria for expansion)
6: \( P \), the set of safe trial points from Algorithm 1
7: \( \Omega = \cup_j \Omega_j \) the safe region found by Algorithm 1
8: \( \Omega_j = [\alpha_j, \beta_j] \) the \( j \)-th disconnected safe subregion, with \( j = 1, \ldots, N_{\Omega} \)
9: \# finally, merge possible overlapping safe regions
10: \( S_i \leftarrow S_i^j \)
11: \( A = \{a_i\} \cup \{a_i : a_i > \beta_i, l = 2, \ldots, n\} \)
12: \( B = \{b_i\} \cup \{b_i : b_i < \alpha_i, l = n - 1, \ldots, 1\} \)
13: \( \Omega = [\alpha_j, \beta_j] \) \( j \)-th disconnected safe subregion, with \( j = 1, \ldots, N_{\Omega} \)
14: \# Construction of \( \Gamma(x) \) on points from Algorithm 1
15: \( \Omega = \emptyset \) when \( \delta \) is defined in (27)
16: \( x_i, x_j \) as defined in (3), (30)
17: \( \gamma(x_i, x_j), i = 1 : k, j = 1 : k \) as defined in (21)
18: \( \hat{\gamma}(x_i) \) and \( \hat{\gamma}(x_j) \), \( i = 1 : k, j = 1 : k \), as defined in (18) and (19), respectively.
19: \( \forall q = 1, \ldots, k \Rightarrow x_q \geq \alpha_j \wedge x_q \leq \beta_j \)
20: \# Construction of \( \Gamma(x) \) on points from Algorithm 1
21: \( \forall q = 1, \ldots, k \Rightarrow x_q \geq \alpha_j \wedge x_q \leq \beta_j \)
for $l = 1 : k$ do 
  $z_l \leftarrow \gamma(x_l, x_1)$
end for 
for $i = 2 : k$ do 
  if $\gamma(x_i, x_1) < z_1$ then 
    for $l = 1 : k$ do 
      if $\gamma(x_i, x_l) < z_l$ then 
        $z_l \leftarrow \gamma(x_i, x_l)$ 
      end if 
    end if 
  end if 
end for 

# Construction of $\Gamma_{k+1}(x)$ from $\Gamma_k(x)$ 
$R_{\text{max}} \leftarrow R_2$ 
$t \leftarrow 2$ 
$v_{\text{var}} \leftarrow 0$
for $i = 3 : k$ do 
  $R_i \leftarrow 0.5(z_{i-1} + z_i) + 0.5L(x_i - x_{i-1})$
  if $(R_{\text{max}} < R_i)$ then 
    $R_{\text{max}} \leftarrow R_i$
  end if 
  $t \leftarrow i$
end for 
end if 
end for 
if $(x_i - x_{i-1} > \varepsilon)$ then 
  $x^{k+1} \leftarrow x_i$
else 
  $v_{\text{var}} \leftarrow v_{\text{var}} + 1$
end if 
GOTO 46 (Re-evaluate $g(x^{k+1})$)
else 
for $i = k : t - 1$ do 
  $x_{i+1} \leftarrow x_i$
  $z_{i+1} \leftarrow z_i$
end for 
for $i = 1 : k$ do 
  $x_i \leftarrow x^{k+1}$
  $z_i \leftarrow g(x^{k+1})$
end if 
if $(z_i > \gamma(x_i, x_1))$ then 
  $z_i \leftarrow \gamma(x_i, x_1)$
end if 
end if 
GOTO 18 (Continue with the next iteration within the same safe region $\tilde{\Omega}_j$)
end if 
STOP (The required accuracy $\varepsilon$ has been reached)
end if 

GOTO (Continue the search within the next safe region) 

return the values $x_{\text{opt}}^j$ and $g_{\text{opt}}^j$; the majorant $\Gamma_j(x)$, the regions $N_k^j$ and $N_{\text{var}}^j$

References

Archetti F, Schoen F (1984) A survey on the global optimization problem: general theory and computational approaches. Ann Oper Res 1(2):87–110
Barkalov KA, Gergel VP (2016) Parallel global optimization on GPU. J Glob Optim 66(1):3–20
Barkalov KA, Strongin RG (2018) Solving a set of global optimization problems by the parallel technique with uniform convergence. J Glob Optim 71(1):21–36
Ben-Tal A, El Ghaoui L, Nemirovski A (2009) Robust optimization. Princeton University Press, Princeton
Berkenkamp F, Schoellig AP, Krause A (2016) Safe controller optimization for quadrotors with Gaussian processes. In: 2016 IEEE International Conference on Robotics and Automation (ICRA). IEEE, pp 491–496. https://doi.org/10.1109/icra.2016.7487170
Calvin JM, Žilinskas A (2000) One-dimensional P-algorithm with convergence rate $O(n - 3 + \delta)$ for smooth functions. J Optim Theory Appl 106(2):297–307
Calvin JM, Žilinskas A (2005) One-dimensional global optimization for observations with noise. Comput Math Appl 44(10–11):157–169
Calvin JM, Chen Y, Žilinskas A (2012) An adaptive univariate global optimization algorithm and its convergence rate for twice continuously differentiable functions. J Optim Theory Appl 155(2):628–636
Casado L, García I, Sergeyev YD (2003) Interval algorithms for finding the minimal root in a set of multivariate one-dimensional nondifferentiable functions. SIAM J Sci Comput 24(2):359–376
Cavoretto R, De Rossi A, Mukhametzhanov MS, Sergeyev YD (2019) On the search of the shape parameter in radial basis functions using univariate global optimization methods. J Glob Optim. https://doi.org/10.1007/s10898-019-00853-3
Daponte F, Grimaldi D, Molinaro A, Sergeyev YD (1996) Fast detection of the first zero-crossing in a measurement signal set. Measurement 19(1):29–39
Fiducioso M, Curi S, Schumacher B, Gwerder M, Krause A (2019) Safe contextual Bayesian optimization for sustainable room temperature PID control tuning. In: Proceedings of the twenty-eighth international joint conference on artificial intelligence. IJCAI, pp 5850–5856. https://doi.org/10.24963/ijcai.2019/811
Floudas CA, Pardalos PM (1996) State of the art in global optimization. Kluwer Academic Publishers, Dordrecht
García J, Fernández F (2012) Safe exploration of state and action spaces in reinforcement learning. J Artif Intell Res 45:515–564
Gergel VP, Sidorov SV (2015) A two-level parallel global search algorithm for solution of computationally intensive multitemplet optimization problems. In: Myshlyshkin V (ed) Parallel computing technologies (PaCT 2015), LNCS, vol 9251. Springer, Cham, pp 505–515
Gergel VP, Grishagin VA, Israfilov RA (2015) Local tuning in nested scheme of global optimization. Procedia Comput Sci 51:865–874
Gillard JW, Kvasov DE (2016) Lipschitz optimization methods for fitting a sum of damped sinusoids to a series of observations. Stat Interface 10(1):59–70
Grishagin VA, Israfilov RA, Sergeyev YD (2018) Convergence conditions and numerical comparison of global optimization methods based on dimensionality reduction schemes. Appl Math Comput 318:270–280
Hansen P, Jaumard B, Lu SH (1992) Global optimization of univariate Lipschitz functions: II. New algorithms and computational comparison. Math Program 55(1–3):273–292
Horst R, Pardalos PM (eds) (1995) Handbook of global optimization, vol I. Kluwer Academic Publishers, Dordrecht
Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. J Glob Optim 13(4):455–492
Kvasov DE, Mukhamezhanov MS (2018) Metaheuristic vs. deterministic global optimization algorithms: the univariate case. Appl Math Comput 318:245–259
Kvasov DE, Sergeyev YD (2012) Univariate geometric Lipschitz global optimization algorithms. Numer Algebra Control Optim 2(1):69–90
Kvasov DE, Sergeyev YD (2013) Lipschitz global optimization methods in control problems. Autom Remote Control 74(9):1435–1448
Kvasov DE, Sergeyev YD (2015) Deterministic approaches for solving practical black-box global optimization problems. Adv Eng Softw 80:58–66
Kvasov DE, Mukhamezhanov MS, Nasso MC, Sergeyev YD (2020) On acceleration of derivative-free univariate Lipschitz global optimization methods. In: Sergeyev Y., Kvasov D. (eds) Numerical computations: theory and algorithms. NUMTA 2019. Lecture notes in computer science, vol 11974. Springer, Cham, pp 413–421
Lera D, Sergeyev YD (2010) Lipschitz and Hölder global optimization using space-filling curves. Appl Numer Math 60:115–129
Lera D, Sergeyev YD (2013) Acceleration of univariate global optimization algorithms working with Lipschitz functions and Lipschitz first derivatives. SIAM J Optim 23(1):508–529
Molinaro A, Sergeyev YD (2001a) An efficient algorithm for the zero-crossing detection in digitized measurement signal. Measurement 30(3):187–196
Molinaro A, Sergeyev YD (2001b) Finding the minimal root of an equation with the multietremal and nondifferentiable left-hand part. Numer Algorithms 28(1–4):255–272
Paulavičius R, Sergeyev YD, Kvasov DE, Žilinskas J (2014) Globally-biased DISIMPL algorithm for expensive global optimization. J Glob Optim 59(2–3):545–567
Paulavičius R, Sergeyev YD, Kvasov DE, Žilinskas J (2020) Globally-biased DIRECT algorithm with local accelerators for expensive global optimization. Expert Syst Appl 144:113052
Pintér JD (1996) Global optimization in action (continuous and Lipschitz optimization: algorithms, implementations and applications). Kluwer Academic Publishers, Dordrecht
Piyavskii SA (1972) An algorithm for finding the absolute extremum of a function. USSR Comput Math Math Phys 12(4):57–67 (In Russian: Zh. Vychisl. Mat. Mat. Fiz., 12(4) (1972), pp 888–896)
Schillinger M, Hartmann B, Skalecki P, Meister M, Nguyen-Tuong D, Nelles O (2017) Safe active learning and safe Bayesian optimization for tuning a PI-controller. IFAC-PapersOnLine 50(1):5967–5972 20th IFAC World Congress
Sergeyev YD (1995) A one-dimensional deterministic global minimization algorithm. Comput Math Math Phys 35(5):553–562
Sergeyev YD, Grishagin VA (2001) Parallel asynchronous global search and the nested optimization scheme. J Comput Anal Appl 3(2):123–145
Sergeyev YD, Kvasov DE (2017) Deterministic global optimization: an introduction to the diagonal approach. Springer, New York
Sergeyev YD, Daponte P, Grimaldi D, Molinaro A (1999) Two methods for solving optimization problems arising in electronic measurements and electrical engineering. SIAM J Optim 10(1):1–21
Sergeyev YD, Famularo D, Pugliese P (2001) Index branch-and-bound algorithm for Lipschitz univariate global optimization with multi-extremal constraints. J Glob Optim 21(3):317–341
Sergeyev YD, Strongin RG, Lera D (2013) Introduction to global optimization exploiting space-filling curves. Springer, New York
Sergeyev YD, Mukhamezhanov MS, Kvasov DE, Lera D (2016) Derivative-free local tuning and local improvement techniques embedded in the univariate global optimization. J Optim Theory Appl 171(1):186–208
Sergeyev YD, Kvasov DE, Mukhamezhanov MS (2017) Operational zones for comparing metaheuristics and deterministic one-dimensional global optimization algorithms. Math Comput Simul 141:96–109
Sergeyev YD, Kvasov DE, Mukhamezhanov MS (2018a) On strong homogeneity of a class of global optimization algorithms working with infinite and infinitesimal scales. Commun Nonlinear Sci Numer Simul 59:319–330
Sergeyev YD, Kvasov DE, Mukhamezhanov MS (2018b) On the efficiency of nature-inspired metaheuristics in expensive global optimization with limited budget. Sci Rep 8:1–9
Sergeyev YD, Nasso MC, Mukhamezhanov MS, Kvasov DE (2020) Novel local tuning techniques for speeding up one-dimensional algorithms in expensive global optimization using Lipschitz derivatives. J Comput Appl Math (submitted)
Strongin RG, Sergeyev YD (2000) Global optimization with non-convex constraints: sequential and parallel algorithms. Kluwer Academic Publishers, Dordrecht
Sui Y, Gotovos A, Burdick JW, Krause A (2015) Safe exploration for optimization with Gaussian processes. In: Bach F, Blei D (eds) Proceedings of the 32nd international conference on machine learning, PMLR, vol 37. Lille, France, pp 997–1005
Vanderbei RJ (1999) Extension of Piyavskii’s algorithm to continuous global optimization. J Glob Optim 14(2):205–216
Žilinskas A, Zhigljavsky A (2016) Stochastic global optimization: a review on the occasion of 25 years of Informatica. Informatica 27(2):229–256
Žilinskas A, Žilinskas J (2010) Interval arithmetic based optimization in nonlinear regression. Informatica 21(1):149–158

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.