Approximating landscape insensitivity regions in solving ill-conditioned inverse problems

Jakub Sawicki¹ · Marcin Łoś¹ · Maciej Smółka¹ · Robert Schaefer¹ · Julen Álvarez-Aramberri^{2,3,4}

Received: 10 August 2017 / Accepted: 27 March 2018 / Published online: 4 April 2018
© The Author(s) 2018

Abstract

Solving ill-posed continuous, global optimization problems is challenging. No well-established methods are available to handle the objective intensity that appears when studying the inversion of non-invasive tumor tissue diagnosis or geophysical applications. The paper presents a complex metaheuristic method that identifies regions of objective function's insensitivity (plateaus). It is composed of a multi-deme hierarchic memetic strategy coupled with random sample clustering, cluster integration, and a special kind of local evolution processes using the multiwinner selection that allows to breed the demes to cover each plateau separately. The final phase consists in a smooth local objective approximation which determines the shape of the plateaus by analyzing the objective level sets. We test the method on benchmarks with multiple non-convex plateaus and in an actual geophysical application of magnetotelluric data inversion.

Keywords Ill-posed global optimization problems · New tournament-like selection · Fitness insensitivity · Objective approximation

The work presented in this paper has been partially supported by Polish National Science Center Grants No. DEC-2015/17/B/ST6/01867 and by the AGH statutory research Grant No. 11.11.230.124.

✉ Jakub Sawicki
jsawicki@agh.edu.pl

Marcin Łoś
marcin.los.91@gmail.com

Maciej Smółka
smolka@agh.edu.pl

Robert Schaefer
schaefer@agh.edu.pl

Julen Álvarez-Aramberri
julen.alvarez.aramberri@gmail.com

¹ AGH University of Science and Technology,
Al. Mickiewicza 30, 30-059 Kraków, Poland

² University of the Basque Country (UPV/EHU), Bilbao, Spain

³ Departamento de Matemática Aplicada, University of the Basque Country UPV/EHU, Barrio Sarriena s/n, 48941 Leioa, Basque, Spain

⁴ BCAM - Basque Center for Applied Mathematics,
Mazarredo 14, E48009 Bilbao, Basque, Spain

1 Introduction

In this paper we focus on inverse problems (IPs) formulated as global optimization ones (GOPs) which consist in finding a set of arguments $\mathcal{S}$ that minimize the objective function f over an admissible set of solutions $\mathcal{D} \subset \mathbb{R}^N$, i.e.:

$$\mathcal{D} \supset \mathcal{S} = \arg \min_{\omega \in \mathcal{D}} \{f(u_0, u(\omega)); A(u(\omega)) = 0\}, \quad (1)$$

where $\omega \in \mathcal{D}$ is an unknown parameter to be found, $u(\omega) \in U$ is the forward solution corresponding to ω , $u_0 \in \mathcal{O}$ is the forward solution observation and $f : \mathcal{O} \times U \rightarrow \mathbb{R}_+$ is the so-called misfit function. Moreover, $A : U \rightarrow U'$ stands for the forward problem operator being the mathematical model of the studied phenomenon, U is the space of forward solutions and U' is its conjugate (see e.g. [1] for details).

We employ the same notation f for the fitness as for the misfit, dropping its dependence on the observation, so that it is shortened from $f(u_0, u(x))$ to $f(x)$ for all $x \in \mathcal{D}$.

If (1) has more than one solution ($\text{card}(\mathcal{S}) > 1$), it becomes *ill-conditioned*, *multimodal*. If $\mathcal{D}$ is embedded in a topological space, and solutions $\mathcal{S}$ fill an open set, then we have a neutral, or *insensitivity region* in the optimization landscape. Such regions are considered in the general study of optimization landscapes in both, discrete and con-

tinuous problems (see e.g. [2]), and they are often called *plateaus*.

Formally speaking, a plateau $\mathcal{P}_{\hat{\omega}} \subset \mathcal{S}$ is an insensitivity region that has associated a minimizer $\hat{\omega} \in \mathcal{S}$, and it is defined as the largest nonempty set such that, for each $x \in \mathcal{P}_{\hat{\omega}}$; $x \neq \hat{\omega}$, $f(\hat{\omega}) = f(x)$ there exists an open, connected set $A \subset \mathcal{S}$ so that $x, \hat{\omega} \in A \subset \mathcal{P}_{\hat{\omega}}$ (see [3]). The above definition imposes that a plateau is always an open, nonempty set, that satisfies $\text{meas}(\mathcal{P}_{\hat{\omega}}) > 0$ and $\hat{\omega} \in \mathcal{P}_{\hat{\omega}}$. The set $R_{\mathcal{P}_{\hat{\omega}}} \subset \mathcal{D}$ is a plateau's *set of attraction*, if any steepest and strictly descent local optimization method starting from an arbitrary point in $R_{\mathcal{P}_{\hat{\omega}}}$ converges to some point of $\overline{\mathcal{P}_{\hat{\omega}}}$. By a *basin of attraction* $\mathcal{B}_{\mathcal{P}_{\hat{\omega}}} \subset \mathcal{D}$, of plateau $\mathcal{P}_{\hat{\omega}}$ we mean a connected part of the level set $\{x \in \mathcal{D}; f(x) < h\}$ that includes $\overline{\mathcal{P}_{\hat{\omega}}}$, where $h = \inf\{f(z), z \in \partial R_{\mathcal{P}_{\hat{\omega}}} \cap \mathcal{P}_{\hat{\omega}}\}$.

An effective handling of multimodality and insensitivity regions is essential when solving the parametric IPs that arise, for instance, in the model calibration or tumor diagnosis [4,5], or in the inversion of magnetotelluric (MT) measurements [6].

Traditional approaches to handling multimodality and insensitivity in IP solving rely on regularization methods (see e.g. [1]). These methods supplement the objective function with a regularization term to make it globally or locally convex. Unfortunately, such methods can produce undesirable artifacts and lead to the loss of information regarding the modeled process. Indeed, they can even deliver outright false solutions, forced by the regularization supplement.

There are also some stochastic methods aimed at handling multimodal problems (see, for instance, [7]). These methods typically perform an identification of the basins of attraction based on an evolutionary algorithm (see [8]). Additionally, a proper diversity of the evolving population is important for plateau identification. A review of possible techniques can be found in [9]. Finally, it is also important to increase the *local diversity*. A possibility for doing so is based on the *multiwinner selection* operator defined in [3].

However, to the best of the authors' knowledge, there is no well established strategy of recovering the objective plateau shape. Most of the existing techniques devoted to multimodal GOPs simply stop after finding a single point from attraction basin of each global solution. Such approaches fail to gather all available knowledge about the considered problems. Moreover, they can succeed in computing good-quality approximations of global optima only when supported with a local optimization method and when the objective function is convex in the optima neighborhoods.

The main contribution of this paper is to define a strategy that properly identifies the shapes of the plateau regions of the problem. In the first step, this strategy detects the plateaus using a global search. After that, an approximation of the shape of the plateaus is performed.

The paper is organized as follows: Sect. 2 contains a detailed description of the proposed method, in Sect. 3 we present the results of the performed tests, and Sect. 4 contains the final remarks and conclusions.

Some ideas presented in this paper were already contained in our conference contributions [10,11]. However, here we provide a more thorough description of the proposed method with a special emphasis on algorithmic and mathematical details. We also show results of new tests, including one real-world engineering problem.

2 The proposed memetic strategy

The proposed strategy, that aims at identifying and determining the shapes of the plateau regions of the problem, is summarized in Algorithm 1. It consists of two phases: in the first one, plateaus are detected using the global search, and in the second one, a local approximation of the shape of the plateaus is obtained.

The initial global search is performed with a complex memetic strategy. This search is stopped when most of the basins of attraction are identified. The selected individuals are then subject to density clustering. This way, the clusters located in the same basins of attractions are merged. At this stage, each cluster has to correspond to a single plateau region. We raise the local diversity of each cluster's population using an evolutionary algorithm (EA) with a multiwinner selection. When the diversity in these populations is sufficiently high, in each cluster we build a local approximation of the fitness using the results of the evolution. Appropriate level sets of these approximations serve then as the estimates of plateau regions.

In the remainder of this section we describe the above steps in details.

Discovering multimodality with a hierarchic multi-deme strategy. The goal of the first, global phase is to detect all potential global minima and provide a rough view of their attraction basins. We employ an Hierarchic Memetic Strategy

Algorithm 1: Strategy for covering and approximating plateau regions

```

1 execute HMS and collect individuals from the leaves
2 clusterize individuals, obtaining  $w$  clusters
3 integrate clusters, reducing their number to  $w'$ 
  // the loop below can be executed in parallel
4 for  $i \leftarrow 1$  to  $w'$  do
5   initialize  $\text{LBA}_i$  with  $i$ 'th cluster
6   evolve  $\text{LBA}_i$  until stop condition is reached
7 for  $i \leftarrow 1$  to  $w'$  do
8   approximate fitness and determine shape of the plateau in  $\text{LBA}_i$ 

```

(HMS) [12] because it is able to perform an adapted-accuracy memetic search resulting in a good-quality initial approximation of the global minima attraction basins.

The HMS is a multi-deme strategy with evolutionary populations forming a tree-like structure with a fixed depth. The levels in the tree correspond to the range and accuracy of the search performed by the populations. As the level number increases, the search is more focused and accurate, but, typically, it is computationally more expensive. Hence, the *root* deme performs the least accurate, the most chaotic as well as the cheapest search. This hierarchy starts with the root deme only and is subsequently developed by means of the *sprouting* operation. This operation is an upper-level (closer to the root) deme that detects a region that possibly contains a local minimum and that generates a new lower-level deme to explore the promising region. The output of the global phase is formed through the integration of the leaf-level demes. The result is then subject to the sample clustering. As the leaf demes occupy regions with good fitness value, we expect that the formed clusters provide a rough approximation of local minima attraction basins. For more details about HMS, we refer to [12,13].

Clustering and cluster integration. Here we employ the OPTICS- ξ algorithm, a hierarchical density-based clustering method introduced by Ankerst et al. in [14]. In particular, we use an implementation based on ELKI Data Mining Framework [15].

The clustering algorithm first orders the points based on their vicinity. Each point has assigned a measure of the density in the neighbourhood called *reachability distance*. Based on the order and the measure, the following part determines where the clusters are located. The result is a hierarchical structure of clusters. The root is the sparsest cluster, while the leaves are the densest ones.

We collect the satisfactory clusters from the cluster tree, i.e. the ones with a sufficiently large population that is located in attraction basins of different local minima. To do so, we apply an integration procedure that detects and then merges the clusters occupying common basins. Notice that the denseness is assured by OPTICS- ξ .

In the detection part, we use two complementary methods. The first one exploits the *hill-valley* function described in [16]. Since we consider minimization problems, we use here its *hollow-ridge* version (see Algorithm 2). For two points chosen from different clusters the function returns a positive number if somewhere on the interval between them the objective has value greater than for these points, i.e. there is a „ridge” between „hollows” occupied by the clusters. If this is not the case the function returns 0. Once this function is defined, the overall merge checking procedure states that: two separate clusters C_1 and C_2 can be merged if, for the points $p_1 \in C_1$ and $p_2 \in C_2$ realizing the distance between

Algorithm 2: Hill-valley function (hollow-ridge version)

Require: Points p_1, p_2 from different clusters, number of intermediate points k

```

1  $max \leftarrow \max(fitness(p_1), fitness(p_2))$ 
2 for  $j \leftarrow 1$  to  $k$  do
3   calculate intermediate point  $inter_j$  on the interval  $[p_1, p_2]$ 
4   compute  $fitness(inter_j)$ 
5   if  $max < fitness(inter_j)$  then
6     return  $fitness(inter_j) - max$ 
7 return 0

```

Algorithm 3: Merge check by means of hill-valley function

Require: number of intermediate points k ; clusters $C_1 \neq C_2$

```

1 find points  $p_1 \in C_1$  and  $p_2 \in C_2$  with minimal distance
2 if  $HillValley(p_1, p_2, k) = 0$  then
3    $C_1$  and  $C_2$  can be merged

```

Algorithm 4: Merge check by means of a local optimization method

Require: clusters $C_1 \neq C_2$

```

1 find points  $p_1 \in C_1$  and  $p_2 \in C_2$  with minimal distance
2 run a local optimization method  $local()$  from  $p_1$  and  $p_2$ 
3 if  $local(p_1)$  finishes in  $extension(C_2)$  or  $local(p_2)$  finishes in  $extension(C_1)$  then
4    $C_1$  and  $C_2$  can be merged

```

C_1 and C_2 the hill-valley function returns 0, or, more generally, an insignificantly small positive value.

The second detection procedure is based on local optimization methods and follows the idea of the Clustered Genetic Search algorithm (see [8] and references therein). As above, we take two points p_1 and p_2 representing the distance between separate clusters C_1 and C_2 . The clusters can be merged if the local method started from any of these points converges to an element of the *cluster extension* of the opposite cluster (see Algorithm 4). We consider an appropriate *surrounding* of the cluster that has a nonempty interior. This allows us to easily check if a given point belongs to it. As in [8], we take ellipsoids as cluster extensions: $\{x \in \mathbb{R}^\ell : (x - \bar{x}_C)^T \Sigma_C^{-1} (x - \bar{x}_C) \leq 1\}$, where $\bar{x}_C$ is the center of C and Σ_C is the unbiased covariance matrix of C . Such cluster extensions are computationally inexpensive.

Local phase of evolution. Each cluster has assigned a separate Local Basin Agent (LBA) to raise the local diversity of the samples and to cover the plateau regions. We use a $(\mu + \lambda)$ evolutionary scheme with multiwinner selection (see Algorithm 5).

This algorithm stops when a desired coverage of the region is attained. Since the region is unknown a priori, we stop the local evolution if the density represented by the mean

Algorithm 5: Evolutionary algorithm utilizing multiwinner selection.

Notation:

$P_t = \langle x_t^{(1)}, \dots, x_t^{(\mu)} \rangle \in \mathcal{D}^\mu \leftarrow$ the population in the t -th epoch,
 $\text{MWS}() \leftarrow$ multiwinner selection procedure.

```

1 Sample the initial population  $P_0$ 
2 Evaluate  $P_0$ 
3  $t \leftarrow 0$ 
  // the main loop over the epochs
4 while  $\text{Stopping\_Condition}(P_t)$  do
5    $\text{Offspring} \leftarrow \emptyset$ 
6   for  $i \leftarrow 1$  to  $\mu$  do // the Vose's scheme
7      $x \leftarrow \text{select}(P_t)$ 
8      $x' \leftarrow \text{mutate}(x)$ 
9      $\text{Offspring} \leftarrow \text{Offspring} \cup \{x'\}$ 
10   $C \leftarrow P_t \cup \text{Offspring}$ 
11   $t \leftarrow t + 1$ 
12   $P_t \leftarrow \text{MWS}(C, \mu)$ 
13  Evaluate  $P_t$ 
14 output  $P_t$ 
  
```

distance to the neighboring individuals is stabilized in the whole population.

A single iteration of this algorithm consists of two main steps. First, the offspring is created with a simple evolutionary algorithm, that employs a proportional selection, no crossover, and Gaussian mutation (for every selected individual). This disturbed offspring is merged with the parental population and reduced to the desired size with a single run of multiwinner selection. Such design ensures uniform coverage of the region of interest.

Multiwinner selection. It employs a strategy based on multiwinner elections. In general, an election consists of a set of voters and a set of candidates. In our case, we take these sets to be the same. Each voter $x^{(i)}$ has a preference list of candidates, ordered by a utility function of the form:

$$u_i(x^{(j)}) = h(f(x^{(j)}))(1 + d(x^{(i)}, x^{(j)}))^{-1}, \quad (2)$$

where $x^{(j)}$ corresponds to an other candidate than $x^{(i)}$, $d: \mathcal{D} \times \mathcal{D} \rightarrow \mathbb{R}_+ \cup \{0\}$ is the distance metric, $f: \mathcal{D} \rightarrow \mathbb{R}_+ \cup \{0\}$ is the fitness function, and $h: \mathbb{R}_+ \cup \{0\} \rightarrow \mathbb{R}_+$ corresponds to the inversion function that we use: $h(x) = 1/(1+x)$. Such utility function ensures three things: a better fitness is preferred, the individuals that are closer are promoted, and the overlapping individuals are not excessively promoted.

We determine the set of winners based on the preference lists. In particular, we employ a greedy implementation of the Chamberlin-Courant voting rule. A justification of choosing this particular algorithm can be found in paper [3].

Fitness approximation inside the basin of attraction. The approximation of the objective function is a common tech-

nique in the evolutionary search area (see, e.g. a survey paper [17]). Typically, the approximation is used as a *fitness surrogate* if the original one is computationally expensive. Here, its level sets form estimates of the objective plateaus.

We propose two methods originally invented for computer graphics and solving PDEs, one using only values of the fitness function and the other utilizing the fitness gradient as well. In both cases the process of constructing the approximation consists of two stages: first, we construct a non-smooth auxiliary approximation and then, we approximate it with B -splines.

Let us denote by $P_{\hat{\omega}}$, $\text{card}(P_{\hat{\omega}}) = n < +\infty$ the set of points contained in the basin of attraction $\mathcal{B}_{P_{\hat{\omega}}}$, obtained from the local phase of evolution associated with the particular plateau $\mathcal{P}_{\hat{\omega}}$ described in Sect. 1. The local evolution delivers also the fitness values $\{(f)^x\}_{x \in P_{\hat{\omega}}}$.

First, we perform the Voronoi tessellation of the points $P_{\hat{\omega}}$, so that its area $V_{\hat{\omega}}$ satisfies $P_{\hat{\omega}} \subset V_{\hat{\omega}} \subset \mathcal{B}_{P_{\hat{\omega}}}$. Next, we obtain a Delaunay decomposition of $V_{\hat{\omega}}$ to the family of simplexes associated with the Voronoi tessellation. The Lagrange approximation $\tilde{f}_{\hat{\omega}} \in C^0(V_{\hat{\omega}})$ of the fitness $f|_{V_{\hat{\omega}}}$ is a composition of 1st degree polynomials on each simplex, non-smooth on their interfaces, satisfying $\tilde{f}_{\hat{\omega}}(x) = (f)^x, \forall x \in P_{\hat{\omega}}$ (see, e.g. Ciarlet [18]).

If the values of fitness gradient $\{(Df)^x\}_{x \in P_{\hat{\omega}}}$ are also available, we can compute the non-smooth approximation $\tilde{Df}_{\hat{\omega}} \in (C^0(V_{\hat{\omega}}))^N$ of its components in the same manner.

In order to obtain a smooth fitness approximation we define a proper Hilbert space of smooth functions $\mathcal{V}_{\hat{\omega}}$ and we take a projection of $\tilde{f}_{\hat{\omega}}$ (and also the components of $\tilde{Df}_{\hat{\omega}}$ if available) on it. To satisfy all these requirements, we use a tensor product of B -splines. These splines are computationally inexpensive to evaluate and they guarantee global C^1 approximation.

Let us denote by $Q_{\hat{\omega}}$ the minimal hyperrectangle with edges parallel to the coordinate axes that contains $V_{\hat{\omega}}$, but still is contained in $\mathcal{B}_{P_{\hat{\omega}}}$. We assume that $\tilde{f}_{\hat{\omega}}$ has the continuous extension $g_{\hat{\omega}} \in C^0(Q_{\hat{\omega}})$ so that $g_{\hat{\omega}}|_{V_{\hat{\omega}}} = \tilde{f}_{\hat{\omega}}$. Similarly, if the fitness gradients are considered, we assume that there exists $d_{\hat{\omega}} \in (C^0(Q_{\hat{\omega}}))^N$, so that $d_{\hat{\omega}}|_{V_{\hat{\omega}}} = \tilde{Df}_{\hat{\omega}}$.

Furthermore, let $\{B_i^{(1)}\}_{i=1}^{K_1}, \dots, \{B_i^{(N)}\}_{i=1}^{K_N}$ denote one-dimensional B -spline bases corresponding to each dimension of $Q_{\hat{\omega}}$, where K_i is the number of basis B -splines used for each dimension. The space $\mathcal{V}_{\hat{\omega}}$ is spanned by the basis functions of the form

$$B_I(x_1, \dots, x_N) = B_{i_1}^{(1)}(x_1) B_{i_2}^{(1)}(x_1) \cdots B_{i_N}^{(N)}(x_N), \quad (3)$$

where $I = (i_1, \dots, i_N)$ are multi-indices ranging over the one-dimensional bases.

Since $\mathcal{V}_{\hat{\omega}}$ is a finite dimensional subspace of the Hilbert space $L^2(Q_{\hat{\omega}})$, then the best B -spline approximation of $g_{\hat{\omega}}$

is the unique $L^2(Q_{\hat{\omega}})$ orthogonal projection of this function onto $\mathcal{V}_{\hat{\omega}}$ (see, e.g. [19, Theorem 5.24]). Therefore, we are looking for $\bar{f}_{\hat{\omega}} = \sum_I u_I B_I \in \mathcal{V}_{\hat{\omega}}$ such that $(g_{\hat{\omega}} - \bar{f}_{\hat{\omega}}) \perp \mathcal{V}_{\hat{\omega}}$ in $L^2(Q_{\hat{\omega}})$, which is equivalent to $(g_{\hat{\omega}} - \bar{f}_{\hat{\omega}}, v)_{L^2(Q_{\hat{\omega}})} = 0 \ \forall v \in \mathcal{V}_{\hat{\omega}}$. To compute it, we solve the following linear system of equations

$$Mu = F; M = \{(B_I, B_J)_{L^2(Q_{\hat{\omega}})}\}, \quad F = \left\{ \int_{Q_{\hat{\omega}}} B_J g_{\hat{\omega}} dx \right\}. \quad (4)$$

As we mentioned above, $\bar{f}_{\hat{\omega}}$ is unique and

$$\| \bar{f}_{\hat{\omega}} - g_{\hat{\omega}} \|_{L^2(Q_{\hat{\omega}})} \leq \| v - g_{\hat{\omega}} \|_{L^2(Q_{\hat{\omega}})}, \quad \forall v \in \mathcal{V}_{\hat{\omega}}. \quad (5)$$

The system (4) can be solved efficiently using the Alternating Direction Solver (ADS) (cf. [20]) thanks to the tensor product structure of the chosen basis.

The second strategy is similar to the first one, but it does not discard information about the fitness gradient values. In addition to $\bar{f}_{\hat{\omega}} \in C^0(V_{\hat{\omega}})$, we utilize the approximations of the components of the fitness gradient $\bar{D}f_{\hat{\omega}} \in (C^0(V_{\hat{\omega}}))^N$. Unfortunately, these approximations are not necessarily coherent in the sense that the distributional derivative $D\bar{f}_{\hat{\omega}}$ coincides with $\bar{D}f_{\hat{\omega}}$ almost everywhere in $V_{\hat{\omega}}$, but we can nevertheless use $\bar{f}_{\hat{\omega}}$ and $\bar{D}f_{\hat{\omega}}$ as the approximations of the fitness and its gradient to compute their H^1 projection onto $\mathcal{V}_{\hat{\omega}}$.

If the functions $f|_{Q_{\hat{\omega}}}$ and $Df|_{Q_{\hat{\omega}}}$ are known, the H^1 projection of $f|_{Q_{\hat{\omega}}}$ onto $\mathcal{V}_{\hat{\omega}}$ leads to finding $b = \sum_I w_I B_I \in \mathcal{V}_{\hat{\omega}}$ satisfying $(f|_{Q_{\hat{\omega}}} - b, v)_{H^1(Q_{\hat{\omega}})} = 0 \ \forall v \in H^1(Q_{\hat{\omega}})$.

Our approach consists in replacing the functions $f|_{Q_{\hat{\omega}}}$, $Df|_{Q_{\hat{\omega}}}$ by the extensions of their Lagrange approximations $g_{\hat{\omega}}$, $d_{\hat{\omega}}$. We are looking for $\bar{f}_{\hat{\omega}} = \sum_I \tilde{w}_I B_I \in \mathcal{V}_{\hat{\omega}}$, so that

$$\begin{aligned} \mathcal{M} \tilde{w} &= \tilde{\mathcal{F}}; \\ \mathcal{M} &= \{(B_I, B_J)_{H^1(Q_{\hat{\omega}})}\}, \\ \tilde{\mathcal{F}} &= \left\{ \int_{Q_{\hat{\omega}}} B_J g_{\hat{\omega}} dx + \int_{Q_{\hat{\omega}}} \nabla B_J \cdot d_{\hat{\omega}} dx \right\}. \end{aligned} \quad (6)$$

In this case, we can only guarantee that there exists a unique $\bar{f}_{\hat{\omega}}$ satisfying (6) (because $\mathcal{M}$ is the matrix of isomorphism) and $\| \bar{f}_{\hat{\omega}} - g_{\hat{\omega}} \|_{L^2(Q_{\hat{\omega}})} + \| D\bar{f}_{\hat{\omega}} - d_{\hat{\omega}} \|_{(L^2(Q_{\hat{\omega}}))^N} \leq \| v - g_{\hat{\omega}} \|_{L^2(Q_{\hat{\omega}})} + \| Dv - d_{\hat{\omega}} \|_{(L^2(Q_{\hat{\omega}}))^N} \ \forall v \in \mathcal{V}_{\hat{\omega}}$.

Unfortunately, the matrix $\mathcal{M}$ does not possess a tensor product structure and the ADS algorithm cannot be applied. To the best of authors' knowledge, no linear time solver exists for such systems.

3 Testing the strategy

We show the performance of the proposed method to: (a) its ability to cover the plateau region with points up to the local approximation phase (i.e. the final stage of the local phase), and (b) to estimate the error in the plateau shape. To that end, we present three benchmarks and a single engineering application.

The first two benchmarks have 2D fitness functions with a single non-convex plateau region. The subsequently 3D benchmark shows the performance in a more complex case, where the visualization is not obvious. Finally, the engineering application gives insight into how the scheme may perform in a real-life scenario.

We will be happy to share the implementation of the method and the benchmarks with whoever contacts us by email.

We execute each test 20 times. We consider a plateau any region with fitness values below 0.1 for the definition of the "plateau coverage". A rectangular grid is imposed on the region, and for each grid-point, it is checked if any point from the sample lies closer than some prescribed threshold distance from it. If this is the case, the grid-point is covered. By calculating the ratio of the covered grid-points to the total number of grid-points one arrives at the coverage value for the sample. The threshold distance is determined based on the volume of the plateau region and the number of points in the sample.

The Table 1 shows global phase configuration. Each level of HMS tree is characterized by three sets of parameters presented in separate lines: EA, sprouting, and local stopping condition. EA parameters are: population size, mutation probability, mutation standard deviation, and arithmetic crossover probability. Sprouting: fitness threshold, minimal distance to other demes, and standard deviation of the created sample. The local stopping condition can be: ineffective (no improvement for the number of epochs) or no-sprout (if no sprout was performed for a number of epochs). The global stopping condition can be: calls (fitness evaluation call number bound) or no-deme (when all demes die). In Table 2, local phase configuration is described, where the columns hold following meaning. Clustering: (OPTICS- ξ) maximal reachability distance, MinPts, steepness factor, (DFS reduction) cluster size threshold, (Hill-Valley integration) intermediate points, maximal distance, and HV tolerance. LBA: population size and mutation standard deviation. Stopping condition: epoch (stops after the number of epochs). The HMS exhibits self-tuning of the sampling measure. However, the setting of its basic parameters (see Tables 1, 2) is problem-dependent and can follow the experimental experience (see e.g. [11–13]).

In the local approximation phase, we compare three methods: a L^2 projection, a H^1 semiprojection and Simple

Table 1 Global phase configuration

Benchmark	Level 1	Level 2	Level 3	Stop condition
C-shaped	20/0.5/1.0/0.05	20/0.1/0.1/0.5	–	Calls/500
	0.1/0.5/0.5	–		
	–	Ineffective/2		
2D X-shaped	20/0.5/2.0/0.05	5/0.1/0.1/0.5	–	Calls/500
	0.1/1.0/0.5	–		
	–	Ineffective/1		
3D X-shaped	20/0.5/2.0/0.05	10/0.1/0.1/0.5	–	No-deme
	0.1/1.0/0.5	–		
	No-sprout/5	Ineffective/1		
MT	20/0.4/3.0/0.3	10/0.1/0.3/0.5	6/0.1/0.015/0.5	No-deme
	10^{-8} /1.2/0.6	10^{-9} /0.06/0.06	–	
	No-sprout/5	No-sprout/5	Ineffective/3	

Table 2 Local phase configuration

Benchmark	Clustering	LBA	Stop condition
C-shaped	2.0/20/0.02/50/3/2.0/0.1	100/0.5	Epoch/4
2D X-shaped	10/10/0.02/10/3/–/0.1	100/0.5	Epoch/4
3D X-shaped	5.0/10/0.02/10/3/10/0.1	100/0.5	Epoch/4
MT	3.0/10/0.02/10/3/1.0/10 ^{–8}	30/0.1	Epoch/4

Kriging with constant trend and exponential semivariogram model—statistical method based on modeling approximated function by Gaussian processes and computing best unbiased prediction of the intermediate values (see [17] and references therein). For the projections, we use B-splines of order 2 and a grid with 40 elements in each dimension. While using more elements increases the accuracy of the projection, our experiments established the Lagrange approximation as by far the most significant source of error thus using very fine mesh is not worth the increased computational cost. In each run, we employ evaluations from all the epochs combined. The quality of the approximation in the case of benchmark problems is measured by distance from the exact solution in L^2 and H^1 norms and Hausdorff distance between the plateau determined by the approximate solution and by the exact function. The engineering problem is discussed separately.

3.1 Benchmark function primer

The benchmarks are based on Gaussian functions:

$$g_{x^0}^r(x) = 1 - \exp\left(-(x - x^0)^T S(x - x^0)\right), \quad (7)$$

where $r \in \mathbb{R}_+^N$ and S is a diagonal matrix with $S_{i,i} = 1/r_i$. In what follows we consider products of such functions which are flattened at the bottom by applying the following transformation:

$$\text{flat}_s(x) = \max\{2s(x) - 1, 0\}. \quad (8)$$

3.2 C-shaped plateau

The first benchmark function is defined on $\mathcal{D} = [-3, 3]^2$ given by flat_{f_C} , where

$$f_C(x) = g_{(0,1.5)}^{(1,0.5)}(x) \cdot g_{(1.5,0)}^{(0.5,1)}(x) \cdot g_{(0,-1.5)}^{(1,0.5)}(x). \quad (9)$$

The plateau of this benchmark function f_C forms a C-shaped valley comprising 6.2% of the whole domain.

For the plateau coverage computations, we employ a threshold distance of 0.3. No LBA cluster is obtained for a single run. Then, only two runs or more are considered for the sample. The remaining cases are characterized by mean coverage $75 \pm 16\%$, with 1-sigma normal confidence level. The minimal coverage is 44% and maximal is 100%. This shows that the scheme is able to cover the plateau region robustly when we run our scheme in a single plateau scenario.

Notice that the existence of certain individuals outside the plateau would allow us to determine its shape.

The L^2 projection approximation retains much of the irregularities exhibited by the Lagrange interpolation, while the H^1 projection matches the smoothness of exact fitness significantly better. Similar results are obtained for the Kriging approximation. The L^2 projection error has large variation and depends heavily on the distribution of evaluation points. This dependence is weaker in the case of H^1 projection, hence the variation is also smaller. Kriging approximation exhibits a slight relative increase of the error near the boundary of the plateau.

The H^1 projection is more accurate considering the mean L^2 and H^1 errors (Table 3). H^1 projection and Kriging have similar error level.

We compare the obtained approximated plateaus with the exact one in Fig. 1. Both L^2 and H^1 projections produce

Table 3 Errors for C-shaped plateau

Test no.	Method	L^2 error	H^1 error	Plateau error
Presented	L^2 projection	0.1360	0.9408	0.2803
	H^1 projection	0.0692	0.4408	0.6892
	Kriging	0.0767	0.4203	0.1367
Average	L^2 projection	0.1481 ± 0.05	0.6104 ± 0.18	1.161 ± 0.74
	H^1 projection	0.1358 ± 0.05	0.4491 ± 0.09	1.627 ± 0.86
	Kriging	0.1306 ± 0.05	0.4405 ± 0.09	1.161 ± 0.79

Fig. 1 C-shaped plateau approximation comparison

a plateau that closely matches the exact one, except for the lower left part. The Kriging approximation behaves better in this region. In the presented example, the L^2 projection provides better overall plateau approximation than H^1 projection (Table 3) although its border exhibits more irregularities than the one obtained from H^1 projection. On average, the plateaus produced by the L^2 projections and Kriging method have similar Hausdorff distance to the exact plateau. Additionally, it is significantly smaller than in the case of H^1 projections.

3.3 2D and 3D X-shaped plateaus

The next test involves a cross shape plateau region. We define the benchmark fitness function on $\mathcal{D} = [-10, 10]^2$ as flat_{f_X} , where

$$f_X(x) = g_{(0,0)}^{(5,0,5)}(x) \cdot g_{(0,0)}^{(0,5,5)}(x). \quad (10)$$

The plateau comprises 3.8% of the whole domain.

We compute the plateau coverage with a threshold distance of 0.5 combining all the populations (excluding the first) from the LBAs.

The coverage value is 0.84 ± 0.12 (at $1 - \sigma$ confidence level) with a minimum at 0.40 and a maximum at 0.97. We

obtain an unlikely event as the minimal value, which on the far edges of the Gaussian function. The second worst result corresponds to a coverage of 0.74.

We also compare the results of the global phase to the local one. The first global phase only manages to achieve a coverage of 0.63 ± 0.090 (at $1 - \sigma$ confidence level), being significantly less efficient than the following phase.

We present the local approximations for two specific cases: one with a slightly uneven point distribution that difficulties the plateau estimation and one with a very good distribution.

The first case is presented in Fig. 2. Most of the plateau is covered, except for the leftmost part and a small region in the right part. The shape of the H^1 projection based approximation seems smoother and more regular, and close to the original fitness function, which is reflected by lower L^2 and H^1 errors (Table 4). The plateau approximation based on H^1 projection is disconnected and generally worse than the one obtained by using the L^2 projection.

The second case is presented in Fig. 3. The points are distributed evenly without significant uncovered regions. The quality of both L^2 and H^1 approximations is higher than in the previous examples and the H^1 projection yields more accurate plateau approximation (Table 4). The plateau approximation are worse than the one produced by L^2 approximation in the previous example. In both cases Kriging method yields lower L^2 and H^1 errors.

On average, both L^2 and H^1 error metrics favor H^1 projection based approximation, but L^2 projections yield better plateau approximations. It appears that successful application of the more accurate H^1 projection requires achieving a higher plateau coverage during the preceding phases. Kriging method gives lower L^2 and H^1 errors than both and produces plateau approximations with quality between that of L^2 and H^1 projections.

The benchmark function $\text{flat}_{f_{3D}}$ is defined on $\mathcal{D} = [-10, 10]^3$ as

$$f_{3D}(x) = g_{(0,0,0)}^{(0.5,5,5)}(x) \cdot g_{(0,0,0)}^{(5,0.5,5)}(x) \cdot g_{(0,0,0)}^{(5,5,0.5)}(x). \quad (11)$$

This function employs a 3D variant of the function, and therefore, this benchmark is more difficult to solve than the previous one. The plateau comprises 1.5% of the whole domain.

Fig. 2 Results of test no. 7. **a** L^2 projection. **b** H^1 projection. **c** Plateau approximations

The plateau coverage metric is computed with a threshold distance of 1.0. As result, the mean value is $62 \pm 21\%$ (with 1-sigma confidence), the minimum is 3.8% and the maximum, 80%.

Table 4 Errors for chosen test runs

Test no.	Method	L^2 error	H^1 error	Plateau error
7	L^2 projection	0.1737	0.7517	0.621
	H^1 projection	0.1605	0.4565	1.551
	Kriging	0.1185	0.3689	0.603
17	L^2 projection	0.1531	0.5701	1.017
	H^1 projection	0.1341	0.4072	0.830
	Kriging	0.1185	0.3689	0.917
Average	L^2 projection	0.1850 ± 0.02	0.7643 ± 0.11	0.948 ± 0.20
	H^1 projection	0.1601 ± 0.02	0.4524 ± 0.05	1.420 ± 0.49
	Kriging	0.1383 ± 0.02	0.4211 ± 0.05	1.140 ± 0.30

The approximation errors are presented in Table 5. In this test case the difference between L^2 and H^1 projections are minor. While the Kriging approximation yields L^2 errors greater than both projections, the quality of the plateau approximation is better on average. The similarity of the results obtained with different methods suggests that in this test the most significant factor corresponds to the plateau coverage by the population.

3.4 MT data inversion

This test consists of solving an engineering problem from the domain of Earth's subsurface exploration. The MT method [6] utilizes the solar wind, which induces Telluric currents in the crust. By placing antennas at the Earth's surface one can measure these currents indirectly. These measurements depend on the underground formation, thus enabling their inversion to obtain the unknown Earth's conductivity distribution. This method is typically employed for a variety of applications, including the CO₂ storage, hydrocarbon extraction, or earthquake prediction.

Figure 4 describes the selected Earth model for the MT problem. The computational domain consists of air and a 1D layered media where a 2D heterogeneity (grey rectangle) is embedded in one of the layers. The blue rectangle corresponds to the natural source located in the ionosphere, while the red triangles correspond to the receivers, located at the Earth's surface. The physical domain is truncated with a Perfectly Matched Layer complemented with a Dirichlet homogeneous BC imposed on its outer part, and the problem is solved by employing the hp -Finite Element Method (see [6] for further details).

A thorough introduction of the problem has been performed by Smolka et al. [12, Sec. 3]. The task is to identify the conductivity $\sigma = \{\sigma_i\}_{i=1,\dots,4}$ of the regions presented in Fig. 4. The domain of the problem $\mathcal{D} = [0, 6]^4$ is mapped to the actual conductivity by a function $\mathcal{D} \ni x = \{x_i\}_{i=1,\dots,4} \rightarrow \{10^{x_i-1}\}_{i=1,\dots,4} = \sigma$.

Fig. 3 Results of test no. 17. **a** L^2 projection. **b** H^1 projection. **c** Plateau approximations

We perform an analysis of the fitness values and individual positions in demes. By assessing how they behave throughout all the demes, we are able to determine what are the characteristics of the fitness landscape in that region. To this end, we compute the relative variance (relVar) for each LBA deme as:

Table 5 Errors for 3D X-shaped plateau

Method	L^2 error	H^1 error	Plateau error
L^2 projection	0.3318 ± 0.04	0.8611 ± 0.10	4.4619 ± 2.17
H^1 projection	0.2935 ± 0.04	0.6536 ± 0.07	4.8921 ± 2.20
Kriging	0.3661 ± 0.07	0.6821 ± 0.07	4.1807 ± 2.20

Fig. 4 The domain of the MT computation

Fig. 5 A combined sample from all the LBA runs for MT. Selected dimensions are plotted, including fitness and a projection of points with fitness below 10^{-10} in gray

$$\text{relVar} = s_{\log f} \left(s_{x_1}^2 + s_{x_2}^2 + s_{x_3}^2 + s_{x_4}^2 \right)^{-1/2}, \quad (12)$$

where s denotes taking standard deviation of a specified sample. The small values of relVar depict flat areas. In our case it varies between 0.19 and 3.02 with the median at 1.44 which means that the regions of different characteristics have been identified.

The best fitness values in most of the demes fluctuates around 10^{-11} and the median is around 10^{-9} . In Fig. 5, all points from the local phases are shown. An insensitive region in x_4 dimension is clearly visible. The most variation is recorded in the positional deviation, min 0.21 and max 2.55. The high-end outliers correspond to plateau regions. However, in the most extended deme-15/5 (15. run, 5. deme)

Table 6 Errors for MT

Method	L^2 error	H^1 error
L^2 projection	1.8465 ± 0.30	12.366 ± 3.78
H^1 projection	1.8376 ± 0.30	2.778 ± 0.47
Kriging	2.1901 ± 0.33	3.878 ± 0.81

Table 7 Coverage results for MT

Method	Nearest plateau (%)	Best solutions (%)
L^2 projection	69.20 ± 35	9.75 ± 13.9
H^1 projection	47.46 ± 39	8.37 ± 13.1
Kriging	80.97 ± 32	10.04 ± 13.3

the fitness varies from $6.53 \cdot 10^{-11}$ to $2.48 \cdot 10^{-8}$ with a median of $1.85 \cdot 10^{-9}$. The variance in the fitness is quite significant within the deme. Therefore, the strategy is not very selective to fitness.

Since it is impossible to use the exact MT computation as a reference solution for the local phase, we use instead an approximation based on evaluations gathered from multiple MT experiments (over 80,000 evaluations with different precision levels, 2,000 chosen to cover the domain with the highest available precision solutions). For this test, we have identified 30 individual plateau regions.

The statistics of all the LBAs are presented in Table 6. For each LBA we have also determined coverage of the reference plateaus by the plateau approximations obtained with L^2 , H^1 projections and the Kriging method. The average coverage percents of the best covered reference plateau are presented in Table 7. In the same manner, we have computed the coverage of the set of best known solutions. Again, the L^2 projection-based method performs slightly better than H^1 projections when it comes to plateau coverage despite higher L^2 and H^1 norm errors. The Kriging method still seems superior to both projection-based approaches. In particular, in this test, it exhibits a relatively high L^2 error.

4 Conclusions

In the present work, we have proposed a strategy to analyze the insensitivity of the solutions to some global optimization problems inspired by ill-conditioned inverse problems. This is a two-phase strategy with a global search phase followed by a local evolution phase. The aim of the global search phase is the detection of global minima attraction basins. In the local evolution phase, the flat parts of the basins (plateaus) are uniformly filled with the individuals, which in turn forms a basis for building local approximations of the objective function. Finally, appropriate level sets of these approximations

serve as estimates of plateau shapes. The main novelty of the proposed methodology consists in precise approximating the areas of misfit insensitivity. It is possible because HMS can effectively discovered the basins of attraction of misfit minima.

The performed experiments show that the results obtained with the projection-based and Kriging methods are comparable. However, the Kriging method provides a faster approximation evaluation. The H^1 projection yields smaller norm errors, similar to those obtained with the Kriging. In general, the L^2 projection delivers better plateau shape approximations in terms of Hausdorff distance to the exact plateau. The H^1 projection yields smoother and more regular boundaries, but seems more prone to missing large regions of the plateau.

The computational cost of the projections grows exponentially with the dimension of the problem. This fact limits the applicability of the strategy to problems of dimension close to ten.

We plan to construct a simpler implementation of the plateau representation and to improve the strategy performance for anisotropic sensitivity.

Acknowledgements Julen Alvarez-Aramberri was partially funded by the RISE Horizon 2020 European Project GEAGAM (644602), the Projects of the Spanish Ministry of Economy and Competitiveness with reference MTM2016-76329-R (AEI/FEDER, EU), and MTM2016-81697-ERC/AEI, the Consolidated Research Group Grant IT649-13 on “Mathematical Modeling, Simulation, and Industrial Applications (M2SI)”, the Basque Government through the BERC 2014-2017 program and by the Spanish Ministry of Economy and Competitiveness BCAM Severo Ochoa excellence accreditation SEV-2013-0323.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Beilina L, Klivanov MV (2012) Approximate global convergence and adaptivity for coefficient inverse problems. Springer, Berlin
2. Spain E, DeJong KA, Sheu A (2016) From optimization to mapping: an evolutionary algorithm for protein energy landscapes. IEEE/ACM Trans Comput Biol Bioinform PP(99):1
3. Faliszewski P, Sawicki J, Schaefer R, Smolka M (2017) Multiwinner voting in genetic algorithms. IEEE Intell Syst 32(1):40–48
4. Duan Q, Sorooshian S, Gupta V (1992) Effective and efficient global optimization for conceptual rainfall-runoff models. Water Resour Res 28(4):1015–1031
5. Paruch M, Majchrzak E (2007) Identification of tumor region parameters using evolutionary algorithm and multiple reciprocity boundary element method. Eng Appl Artif Intell 20(5):647–655
6. Alvarez-Aramberri J, Pardo D (2017) Dimensionally adaptive hp-finite element simulation and inversion of 2d magnetotelluric measurements. J Comput Sci 18:95–105

7. Preuss M (2015) Multimodal optimization by means of evolutionary algorithms. Natural computing. Springer, Berlin
8. Wolny A, Schaefer R (2011) Improving population-based algorithms with fitness deterioration. *J Telecommun Inf Technol* 4:31–44
9. Gupta D, Ghafir S (2012) An overview of methods maintaining diversity in genetic algorithms. *Int J Emerg Technol Adv Eng* 2(5):56–60
10. Łoś M, Schaefer R, Sawicki J, Smółka M (2017) Local misfit approximation in memetic solving of ill-posed inverse problems. *Lecture notes in computer science*, vol 10199. Springer, Berlin, pp 297–309
11. Łoś M, Sawicki J, Smółka M, Schaefer R (2017) Memetic approach for irremediable ill-conditioned parametric inverse problems. *Procedia computer science*, vol 108C. Elsevier, Amsterdam, pp 867–876
12. Smółka M, Schaefer R, Paszyński M, Pardo D, Álvarez-Aramberri J (2015) An agent-oriented hierarchic strategy for solving inverse problems. *Int J Appl Math Comput Sci* 25(3):483–498
13. Smółka M, Gajda-Zagórska E, Schaefer R, Paszyński M, Pardo D (2015) A hybrid method for inversion of 3D AC logging measurements. *Appl Soft Comput* 36:422–456
14. Ankerst M, Breunig MM, Kriegel HP, Sander J (1999) Optics: ordering points to identify the clustering structure. *SIGMOD Rec* 28(2):49–60
15. Schubert E, Koos A, Emrich T, Züfle A, Schmid KA, Zimek A (2015) A framework for clustering uncertain data. *PVLDB* 8(12):1976–1979
16. Ursem RK (1999) Multinational evolutionary algorithms. In: *Proceedings of the 1999 congress on evolutionary computation CEC'99*, vol 3. IEEE
17. Jin Y (2005) A comprehensive survey of fitness approximation in evolutionary computation. *Soft Comput* 9(1):53–59
18. Ciarlet PG (1978) *The finite element method for elliptic problems*. North-Holland, Amsterdam
19. Folland GB (1999) *Real analysis. Pure and applied mathematics. Modern techniques and their applications*, 2nd edn. Wiley, New York
20. Gao L, Calo VM (2014) Fast isogeometric solvers for explicit dynamics. *Comput Methods Appl Mech Eng* 274:19–41