amcs

AN AGENT–ORIENTED HIERARCHIC STRATEGY FOR SOLVING INVERSE PROBLEMS

MACIEJ SMOŁKA [a,*], ROBERT SCHAEFER [a], MACIEJ PASZYŃSKI [a], DAVID PARDO [b,c,d],
JULEN ÁLVAREZ-ARAMBERRI [b,e]

[a] Department of Computer Science
AGH University of Science and Technology, Al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: {smolka,schaefer,paszynsk}@agh.edu.pl

[b] Department of Applied Mathematics, Statistics, and Operational Research
University of the Basque Country (UPV/EHU), Barrio Sarriena S/N, 48940 Leioa (Bizkaia), Spain
e-mail: {dzubiaur,julen.alvarez.aramberri}@gmail.com

[c] Basque Center for Applied Mathematics (BCAM), Alameda de Mazarredo 14, 48009 Bilbao, Spain

[d] Ikerbasque (Basque Foundation for Sciences), Maria Diaz de Haro 3, 48013 Bilbao, Spain

[e] Laboratory of Mathematics and Their Applications
University of Pau (UPPA), Avenue de l'Université BP 576, 64012 Pau cedex, France

The paper discusses the complex, agent-oriented hierarchic memetic strategy (HMS) dedicated to solving inverse parametric problems. The strategy goes beyond the idea of two-phase global optimization algorithms. The global search performed by a tree of dependent demes is dynamically alternated with local, steepest descent searches. The strategy offers exceptionally low computational costs, mainly because the direct solver accuracy (performed by the hp-adaptive finite element method) is dynamically adjusted for each inverse search step. The computational cost is further decreased by the strategy employed for solution inter-processing and fitness deterioration. The HMS efficiency is compared with the results of a standard evolutionary technique, as well as with the multi-start strategy on benchmarks that exhibit typical inverse problems' difficulties. Finally, an HMS application to a real-life engineering problem leading to the identification of oil deposits by inverting magnetotelluric measurements is presented. The HMS applicability to the inversion of magnetotelluric data is also mathematically verified.

Keywords: inverse problems, hybrid optimization methods, memetic algorithms, multi-agent systems, magnetotelluric data inversion.

1. Introduction

Inverse problems form an important area of contemporary research related to fundamental problems in science and engineering. Among its numerous applications one can find non-invasive geophysical exploration, tumor characterization, and analysis of unknown materials.

Parametric inverse problems are usually formulated as optimization ones, where the objective is to minimize the misfit between the simulated and measured forward solutions. When solving such problems, one usually faces some significant obstacles such as ill-conditioning, existence of multiple local minima (multi-modality), and possibly low regularity of the misfit functional. All of them significantly reduce the usefulness of convex optimization methods (such as gradient-based ones), as well as simple stochastic mechanisms (Monte Carlo and simple evolutionary schemes) because of

- the lack of guarantee of finding all solutions,

- enormous computational cost.

The main goal of this work is to obtain a global inverse solver capable of finding many global and/or local

*Corresponding author

solutions (misfit minima) with a satisfactory accuracy and an acceptable computational cost. In order to achieve this goal, the paper combines several different ideas: the *hierarchic genetic strategy* (HGS) (see, e.g., Schaefer and Kołodziej, 2003; Wierzba *et al.*, 2003) to decrease the cost of the main part of a global search (which consists of finding the basins of attraction), *cluster-based fitness deterioration* (see, e.g., Beasley *et al.*, 1993; Obuchowicz, 1997; Wolny and Schaefer, 2011), *memetic algorithms* (see, e.g., Neri *et al.*, 2012) composing various techniques into a single population-based stochastic strategy in order to gain efficiency and flexibility, and the *evolutionary multi-agent system* (EMAS) (see, e.g., Cetnarowicz *et al.*, 1996), allowing more flexible, asynchronous processing and easy embedding of local convex searches. The work presented in this paper is an extension of the hybrid model involving the global genetic search phase followed by the local gradient phase, as applied for the inverse resistivity logging measurement simulations with direct current electrodes (Gajda-Zagórska *et al.*, 2015), as well as for the identification of the Young modulus for imprint nanolithography (Barabasz *et al.*, 2014).

The proposed strategy develops dynamically a tree of dependent populations (demes) searching with various levels of accuracy that grow from the root to the leaves. Two types of individuals are utilized: passive individuals, containing candidate solutions only, and active individuals, consisting of computational software agents. Both active and passive individuals are gathered in demes governed by structural agents assigned to the nodes of the population tree. Demes of passive individuals are traditionally evolved by using common selection and genetic operators. Active individuals (agents) compete inside a deme one with another to perform their actions producing offspring agents with genotypes determined by traditional genetic operators or by a convex optimization process. The search accuracy is that associated to the solution of direct problems using an hp-adaptive finite element method (hp-FEM), where h (height) refers to the element size, and p to the polynomial order of approximation, which can be adapted/modified throughout the computational grid (Demkowicz, 2006).

The low computational cost is obtained primarily by an economical global search performed by the tree of demes, because the more detailed local searches are activated mainly in the promising regions found by the parental populations. The total number of individuals is significantly lower than in the case of the traditional, single population evolutionary search. The main computational cost decrement is achieved via the common inverse and forward error scaling, i.e., the rough global search is performed with a low misfit accuracy whereas the accuracy increases in the refined searches performed by branches and leaves. Such a policy minimizes the number of expensive hp-FEM solver

calls, which are necessary for accurate misfit evaluation. Finally, the agent-oriented architecture based on the EMAS idea allows economic and flexible invocation of local gradient methods at least once for a single solution and only if they outperform the stochastic search. Moreover, the agent-based architecture facilitates parallel evolution of deme populations.

We have verified our strategy by a series of benchmark problems that mimic the optimization landscape obtained in real-life inverse problems. The first phase of benchmark tests leads to establishing the set of the strategy's parameters that assure its most economic operation. In the second phase, we compare results of our proposed strategy with those of other global optimization methods executed using a comparable budget (computer resources). In this paper, we compare the HMS with two standard global optimization methods, i.e., the simple evolutionary algorithm (SEA) and multi-start. We have already performed a benchmark-based comparison with a version of the HGS (Smołka and Schaefer, 2014).

We conclude the paper with the application of our agent-based system to solve an inverse problem consisting in identifying the resistivity of the underground formation by using magnetotelluric measurements. The magnetotelluric (MT) method is a passive electromagnetic (EM) exploration technique which allows us to determine the resistivity distribution in the subsurface of the area of interest on scales varying from a few meters to hundreds of kilometers. Commercial applications include hydrocarbon, geothermal, underground water monitoring, and, more recently, monitoring CO_2 sequestration in the subsurface. The method does not require artificial power sources. Instead, natural power sources are used to induce the phenomena. These natural sources are nothing but electric currents within the ionosphere created by deformations of the magnetosphere. The motivation for the choice of the MT problem was the detection of several local minima in an MT inversion performed by means of a classical gradient method (Álvarez-Aramberri *et al.*, 2013).

The results of all comparative tests show a much higher quality and number of extremes found by the proposed agent-based system than by other reference methods. Single population evolutionary algorithms are able to find only a single solution within the assumed budget. A multi-start method, which consists in parallel execution of local gradient-type methods from a number of random starting points, cannot in general find satisfactory solutions, because within the assumed budget only a small number of local processes can be invoked.

The paper concentrates on the algorithmic aspects of the HMS, so we do not discuss implementation-related issues here. Some notes on a sample implementation can be found in the work of Smołka and Schaefer (2014) and we refer the interested reader there.

2. HMS architecture

The main idea of the HMS is to provide a global optimization tool especially suited for solving difficult inverse problems. The considered problems are difficult because of their inherent multi-modality accompanied by the nontrivial computational cost of direct problem solution, which is necessary for evaluating the objective that is the *misfit* between the observed and computed values of a *quantity of interest*. Nevertheless, inverse problems have also some advantageous features. First, their theoretical global minimum value is well known (and equal to zero). Although in practice this theoretical value is never attained because of noise, modeling errors, etc., we know at least that the objective function is bounded from below by zero and that its takes values close to this bound. This knowledge can be used, e.g., in the construction of stopping conditions for stochastic evolution. Second, in some important cases, the cost of the direct problem solution can be modulated by an assumed accuracy of the solution: it is the case of hp-FEM direct solvers (Demkowicz *et al.*, 2007).

As a global optimization tool, the HMS tries to combine the high-level exploratory ability with the accuracy and efficiency of a local optimization method. In contrast to two-phase methods in which local searches are executed right after the completion of the global phase, the HMS follows the overall idea of memetic algorithms, i.e., it intermixes local-optimization-oriented mechanisms into a global stochastic search machinery. The global part follows the multi-population evolutionary approach introduced by the HGS (Schaefer and Kołodziej, 2003). Namely, the global search is performed by a collection of genetic populations. The populations can evolve in parallel, but they are not mutually independent. The structure of the dependency relation is hierarchical (i.e., tree-like; see Fig. 1) with a restricted number of levels. The HGS proved to have
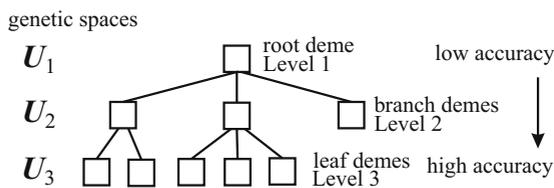

Fig. 1. HGS-like evolutionary population tree.

considerable exploratory capabilities together with a good search accuracy, especially with floating-point phenotype encoding (Wierzba *et al.*, 2003). The HMS naturally tries to retain these abilities while going beyond the HGS in some aspects. First of all, it adds local optimization to the set of operations applied to the genetic individuals. But this is performed carefully in order

to avoid the premature population convergence on the one hand and high cost of running instances of a local method from inappropriate points on the other. Namely, some genetic individuals (but not necessarily all of them) receive an identity and some intelligence, hence becoming independent agents in a multi-agent system (MAS). The decision of performing the local search becomes their own responsibility. Moreover, the demes are managed by special controller agents. The idea of turning a passive genetic individual into an intelligent agent has some further consequences. We have to redefine the genetic operations in such a way that they can be applied to agents. While it is straightforward for the mutation and the crossover (although in this case a new agent is activated), the selection cannot be performed in the simple genetic (or evolutionary) way. Instead, we follow the lines of the EMAS (Cetnarowicz *et al.*, 1996; Byrski *et al.*, 2013), thus performing an operation analogous to the tournament selection but realized as a two-agent rendezvous.

In the sequel, we shall present the structure of the HMS starting with a description of HMS agent types.

2.1. HMS agent types. The main HMS agent types along with their interrelations are shown in Fig. 2. The Master Agent (MA) is a global system coordinator. Deme Agents (DAs) manage evolutionary populations at various levels of the deme tree. DA specializations differ primarily in the type of population they can own. Evolutionary Agents (EAs) hold simple passive collections of chromosomes, whereas Local Agents (LAs) coordinate groups of Computational Agents (CAs). Any CA, apart from holding an immutable genotype, can perform one of the available actions, such as local optimization method execution. The primary responsibility of the Objective Agent (OA) is objective computation, which typically involves calling an external direct problem solver. In the following, we provide detailed descriptions of HMS agent types.
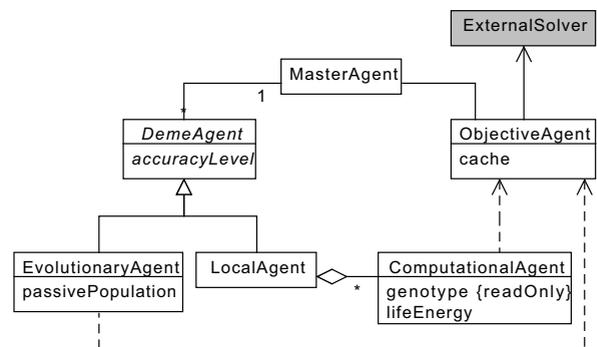

Fig. 2. HMS agent types (UML class diagram).

Master Agent (MA). As a global system coordinator, it is started as a first agent in the HMS MAS. Its responsibilities include performing system initialization such as the activation of other essential agents, i.e., the Objective Agent and a Deme Agent of the deme-tree root. After the initialization, the Master Agent starts the global loop of deme coordination and checks if the global stopping condition is satisfied. The deme coordination follows the lines of FIPA Contract-Net (FIPA, 2002). It begins by sending a call for proposals (CFP) to all active Deme Agents. Then, the MA waits for DA proposals and accepts those that are not in conflict. This is shown in Algorithm 1. DA proposals are in conflict when their

---

Algorithm 1. Master Agent (MA) algorithm.

1: create OA
2: create root location DA
3: **repeat**
4:     send CFP to all active DAs
5:     receive proposals from DAs
6:     accept all non conflicting proposals
7:     request fitness deterioration from OA
8: **until** global stop condition is satisfied.

---

corresponding activities cannot be executed in parallel. Using the terminology of Byrski *et al.* (2013), they are not *local*. In the current HMS realization, all actions are local (note that, in contrast to Byrski *et al.* (2013), we do not consider the migration), so all DA proposals can be accepted by the MA.

Deme Agent (DA). It is a deme-tree node coordinator. Each deme has an associated level of computational accuracy stored as a property of the corresponding Deme Agent. In fact, the Deme Agent is an abstract class with two different specializations: the Evolutionary Agent and the Local Agent.

Evolutionary Agent (EA). This is a simple (passive) evolutionary population owner. Periodically, after receiving the permission from the Master Agent, it evolves its population for a fixed number of generations (this sequence of genetic epochs is called a *metaepoch*), and then sprouts a new deme from the current best individual unless the sprout condition is not satisfied (see Algorithm 2). Note that similar agents form the structure of the globally balanced HGS (Jojczyk and Schaefer, 2009). Creating the initial population in Line 2 has two different meanings. If an EA contains the tree root deme, then the initial population is sampled using the uniform probability distribution. Otherwise, the EA is itself sprouted using its parent's best individual as a seed. In such a case, the initial population is sampled using the

---

Algorithm 2. Evolutionary Agent (EA) algorithm.

1: set accuracy level
2: create initial deme population
3: **repeat**
4:     send proposal to MA
5:     **if** MA has accepted proposal **then**
6:         **for all** epochs in metaepoch **do**
7:             perform selection
8:             perform crossover and mutation
9:             **for all** created individual **do**
10:                request objective computation from OA with stored accuracy
11:            **end for**
12:        **end for**
13:        **if** best individual satisfies sprout condition **then**
14:            sprout new child DA from best individual
15:        **end if**
16:    **end if**
17: **until** local stop condition is satisfied

---

normal distribution centered in the seed individual with the standard deviation depending upon the tree level.

Local Agent (LA). The Local Agent owns a population of Computational Agents and acts as a local scheduler of their actions. Namely, it receives action proposals from Computational Agents, selects one of them according to a probability distribution, sends a proposal to the Master Agent and, if the proposal is accepted, lets the selected Computational Agent perform its action (see Algorithm 3). The Local Agent's responsibilities include also some action coordination, such as checking if a selected sprout action is allowed. Creating the initial population is performed analogously as in the case of an Evolutionary Agent, i.e., by sampling using a proper probability distribution (different for root and non-root demes). The only difference is the type created individuals: passive chromosomes in the case of an Evolutionary Agent and active Computational Agents in the case of Local Agents.

Computational Agent (CA). It is an *active* individual of the HMS genetic population. It owns an immutable genotype consisting of an encoded domain point (a chromosome) and a level of computational precision. The precision level must be consistent with the owning Local Agent's level. The mutable part of a Computational Agent's state includes a nonnegative memetic parameter called *life energy*. It is exchanged during the Computational Agent's action execution such that the total energy remains constant within each deme. Only agents with positive life energy are considered active (alive) and take part in system evolution. There exists a set of actions

Algorithm 3. Local Agent (LA) algorithm.

1: set accuracy level
2: create initial deme population
3: **repeat**
4:     send CFP to all active CAs
5:     receive action proposals from CAs and choose one
6:     send corresponding proposal to MA
7:     **if** MA has accepted the proposal **then**
8:         **if** CA action creates new individual **then**
9:             create new CA
10:         **else if** chosen action is SPROUT **then**
11:             **if** sprouting can be performed **then**
12:                 create new child DA
13:             **end if**
14:         **end if**
15:     **end if**
16: **until** local stop condition is satisfied

from which an active Computational Agent chooses one at a time to perform. Which actions are actually available depends on parameters such as life energy, the objective value, the precision level, and others. Finally, the action is performed only if permitted by the owning Local Agent (see Algorithm 4).

Algorithm 4. Computational Agent (CA) algorithm.

1: set accuracy level according to LA's level
2: request objective computation from OA with stored accuracy
3: **repeat**
4:     remain temporarily inactive
5: **until** objective value is obtained
6: **while** life energy > 0 **do**
7:     receive CFP from owning LA
8:     choose an available action
9:     send the corresponding proposal to LA
10:     **if** received permission from LA **then**
11:         perform chosen action
12:         update life energy
13:     **end if**
14: **end while** {CA permanently inactive}

There is an energy quantum related to each action, which is spent (during GET it can sometimes be gained) by a Computational Agent during action execution. Currently, the following actions are considered (cf. Byrski *et al.*, 2013): GET, MUTATE, CROSSOVER, LOCOPT and SPROUT. In all tests prepared for this paper, we set all action energies to 1 and the initial CA energy to 5.

The GET action is a two-agent stochastic duel during which proper quantum energy moves from the loser to the winner. A Computational Agent with a lower (i.e., closer to the global minimum) objective value has more chances to win. MUTATE and CROSSOVER are straightforward counterparts of the corresponding genetic (or evolutionary) operations such as, e.g., the normal mutation and the arithmetic crossover. The SPROUT action is inspired by the child branch sprouting operation, which is fundamental in the HGS (Schaefer and Kołodziej, 2003). In the HMS, it produces a new deme together with its Deme Agent and an initial population of Computational Agents. Obviously, SPROUT makes no sense at the leaf level, where it can be optionally replaced with LOCOPT. LOCOPT is local optimization method execution started from the agent's decoded chromosome. In the current realization, LOCOPT is allowed only at the leaves.

Action selection is determined by a given probability distribution. The probability of LOCOPT can be computed using a formula like the following:

$$p_{\text{LOCOPT}} = p_0 + (1 - p_0)\frac{s}{1 + \text{objective}}, \qquad (1)$$

where $0 \leq p_0 < 1$ is the guaranteed probability and $s > 0$ is close but not equal to 1 to allow selecting other actions. Thus, the better the objective value, the more strongly LOCOPT is preferred. Other actions available according to the current life energy receive equal remaining probability. The same formula can also be used for computing SPROUT probability at non-leaf levels.

Objective Agent (OA). In a real HMS application (i.e., in solving inverse problems), the objective value is computed externally by a specialized direct solver. The responsibility of an Objective Agent (typically, one in the whole system) is to provide a proper solver gateway, i.e., to execute the solver process (or several parallel processes) properly and to transfer the input data to the solver and the output back to the HMS. Of special interest for us is the case of computing the objective by means of a direct hp-FEM solver when the direct problem solution is Lipschitz continuous with respect to the parameters. This property is not straightforward and has to be proved in any particular case. In Section 4.4 this is shown for the magnetotelluric problem (see Remark 1), which is our real-world test case. Then, we can adapt the solver accuracy to the assumed accuracy of HMS tree demes; see Algorithm 5. Finally, we know the dependency between the solver accuracy and the computational cost of the direct problem solution (for details, see Barabasz *et al.*, 2014; Gajda-Zagórska *et al.*, 2015) which in turn is the main unit component of the overall HMS cost. Therefore, we can optimize the overall cost by modulating the deme accuracy.

Additional Objective Agent activities may include caching solver results, solver instance pooling (in the case of parallel execution) and scheduling objective computations according to a sophisticated optimizing

Algorithm 5. Objective computation with hp-FEM direct solver (see Section 4.4).

1: **input:** tree level j, parameter value
2: compute relative FEM error e_{rel}
3: **while** $e_{\text{rel}} < \text{Ratio}(j)$ **do**
4:     execute 1 step of hp adaptation
5:     solve problem on new fine and coarse meshes
6:     compute new value of e_{rel}
7: **end while**
8: **return** objective computed by means of final mesh

policy (e.g., a diffusion-based one (Grochowski *et al.*, 2006)).

Quite a special kind of additional OA activity is the deterioration of the objective function. The latter is a proper objective modification that leads to the leveling of central regions of attraction basins of already found local and global minima (Beasley *et al.*, 1993; Obuchowicz, 1997). The idea is to discourage the evolutionary individuals from exploring already well-recognized areas. In this paper, we adopt the cluster-based fitness deterioration technique described by Wolny and Schaefer (2011). Namely, after a request from the MA, all gathered objective data (in this case, the points at which the objective has been computed) at selected accuracy levels are clustered (currently, using the DBSCAN algorithm (Ester *et al.*, 1996)). Afterwards, for each recognized cluster, we construct the ellipsoidal *cluster extension*

$$\text{CE} = \{x \in \mathbb{R}^N : (x - \overline{x})^T \Sigma^{-1} (x - \overline{x}) \le 1\}$$

determined by its center $\overline{x}$ and a symmetric matrix Σ, which in our case is the cluster unbiased sample covariance matrix (cf. Wolny and Schaefer, 2011). Then, each cluster extension contributes to the deterioration by the following formula:

$$f(x) := f(x) + f_{\max} \Psi(x), \quad (2)$$

where Ψ is a well-known bump function

$$\Psi(x)$$
$$= \begin{cases} \exp\left(1 - \dfrac{1}{1 - (x - \overline{x})^T \Sigma^{-1}(x - \overline{x})}\right) & \text{for } x \in \text{CE}, \\ 0 & \text{otherwise.} \end{cases}$$
$$(3)$$

and $f_{\max}$ is the maximal objective value over the cluster. Ψ is an infinitely smooth function and its support is exactly the considered ellipsoid. Note that, in contrast to the standard (cf. Wolny and Schaefer, 2011), we do not make use of simple Gaussian functions because they are positive everywhere, which destroys the desired zero-value feature of inverse problem global minima.

2.2. Population structure. As stated before, the HMS genetic population is decomposed into dependent demes forming a dynamically changing tree of the fixed maximal depth m. Genetic individuals, i.e., computing agents, located at the tree levels close to the root, perform the chaotic and inaccurate search. When approaching the leaves, the search becomes more and more focused and the accuracy is increased (see Fig. 1). The variability of the search accuracy results from the diversity of the genotype encoding precision used at different tree levels. The latter depends on the encoding type. In the case of binary encoding (as in the simple genetic algorithm), it can be achieved by the binary genotype length variation, whereas in the case of real number encoding (as in the simple evolutionary algorithm), it can be realized by appropriate phenotype scaling. The latter case is used in the prototype implementation of the HMS, so here we present some details. The description follows those presented in existing papers (Wierzba *et al.*, 2003; Jojczyk and Schaefer, 2009).

In real number encoding, both phenotypes and genotypes are vectors from $\mathbb{R}^N$. We assume that the solution domain is a box

$$\mathcal{D} = [a_1, b_1] \times \cdots \times [a_N, b_N],$$

and we take a sequence of scaling factors $\eta_i \in \mathbb{R}$ such that $\eta_1 > \eta_2 > \ldots \eta_{m-1} > \eta_m = 1$. Then, the genetic universum at the tree level j is

$$U_j = \left[0, \frac{b_1 - a_1}{\eta_j}\right] \times \cdots \times \left[0, \frac{b_N - a_N}{\eta_j}\right], \quad (4)$$

and the encoding mapping at the level j is defined as

$$\mathcal{D} \ni x \longmapsto \left\{\frac{x_k - a_k}{\eta_j}\right\}_{k=1}^N \in U_j. \quad (5)$$

Moreover, we define the scaling mapping,

$$\text{scale}_{i,j} : U_i \ni x \mapsto \frac{\eta_i}{\eta_j} x \in U_j.$$

In such genetic universa, the search at lower levels is more chaotic because the mutation acts stronger, and less precise because of limitations in the real number representation. It is possible to use various genetic operators in such encoding. Among the most important ones, we select the normal mutation

$$y_i = x_i + \mathcal{N}(0, \sigma_j^{\text{mut}}), \quad i = 1, \ldots, N,$$

where $\mathcal{N}(0, \sigma_j^{\text{mut}})$ is a normally distributed random variable with standard deviation σ_j^{mut} set separately for each level j, and the arithmetic crossover

$$y_i = x_i^1 + \mathcal{U}([0, 1])(x_i^2 - x_i^1), \quad i = 1, \ldots, N,$$

where $\mathcal{U}([0,1])$ is a random variable distributed uniformly over the interval $[0,1]$. Both operators are used in our sample implementation. Furthermore, we exploit the classical fitness-proportional (roulette-wheel) selection in passive populations (on Evolutionary Agents), additionally preserving the best individual of each generation. A newly sprouted deme's population is sampled according to the N-dimensional Gaussian distribution centered at the properly encoded fittest individual of the parent process with the diagonal covariance matrix taking values $(\sigma_j^{\text{sprout}})^2$ on the diagonal. The sprout cannot be performed in population P at level j if there exists a population P' at level $j+1$ such that

$$|\overline{y} - \text{scale}_{i,i+1}(y)| < c_j, \qquad (6)$$

where y is the best individual in P, $\overline{y}$ is the average phenotype of P', and c_j is a branch comparison constant.

Finally, it should be mentioned that further utilization of the knowledge gathered during multi-level enhanced genetic evolution is possible by means of the clustering technique, in which better approximation of attraction basins of the local minima can be developed, allowing yet more precise application of local optimization methods.

3. Benchmark tests

In order to prove HMS abilities to find the global minimum in multi-modal cases, we performed two benchmark tests. The test setup in both the cases was as follows. The HMS tree had three levels with Evolutionary Agents at the root and middle levels, and Local Agents at the leaf level, which seems to be quite a standard layout for an overall optimization. The proportional (roulette-wheel) selection, the normal mutation and the arithmetic crossover were chosen as genetic operators. First, we ran the HMS against both functions 30 times till the global minimum was reached with the assumed accuracy of 10^{-4}. The choice of such a type of test influenced the setting of the HMS stopping conditions. Namely, the global stopping condition was satisfied if a leaf approached the global minimum with the given accuracy, whereas a leaf stopping condition was satisfied if the leaf approached the global minimum or if a fixed number of its consecutive metaepochs were ineffective, i.e., there was no significant variation in the leaf's population average fitness. To make this stopping condition applicable to active populations, we need to adapt the notion of the genetic epoch. Namely, in the case of an LA population, it is simply a CA action execution sequence of the length equal to the initial population size. After the HMS terminated, we counted its objective calls, which set the computational budget for two comparative classical stochastic optimization methods: the simple evolutionary algorithm (SEA) and the multi-start. The comparative methods were also run 30 times.

The first benchmark was the 10-dimensional Ackley path function with domain $[-5,5]^{10}$. It is a standard global optimization test with one hard-to-find global zero-valued minimum surrounded by numerous other local minima with greater values. This makes it similar to a few important inverse problem objectives, such as the magnetotelluric problem (see Section 4). A special feature of the Ackley function is its flatness outside the narrow attraction basin of the global minimum, which makes it still more similar to the MT problem. The execution parameters for the 10D Ackley function are summarized in Table 1. Note that the metaepoch length parameter is specially adapted to Local Agents (see above). Similarly, the population size in this case is not constant; in our simulations it varied between 4 and 8.

Table 1. HMS execution parameters (Ackley 10D).

	Root	Middle	Leaf
Population (initial)	50	10	4
Metaepoch length	2	2	2
Encoding scale	4.0	2.0	1.0
Mutation rate	0.2	0.05	0.01
Crossover rate	0.5	0.5	0.5
Mutation std. dev.	5.0	1.0	0.2
Sprout std. dev.	–	2.0	1.0
Sprout min. dist.	–	2.0	1.0

The obtained HMS objective call means are shown in Table 2. The cost of local method application is included in the leaf level cost. The averages from Table 2

Table 2. Average number of objective evaluations (Ackley 10D).

Root	Middle	Leaf	Total
1022.8	584.7	4493.6	6101.1

allowed us to compute the predicted cost for the SEA and multi-start. Taking the average cost of running the local method from HMS executions, we set the starting pool size for multi-start to 70. Similarly, estimating the average cost of running the evolutionary algorithm, also on the basis of HMS runs, we set the initial SEA population size to 100. The SEA genetic parameters were set according to the corresponding HMS parameters from the leaf level, i.e., the crossover rate was set to 0.5, the mutation rate to 0.01, and the mutation standard deviation to 0.2. Table 3 shows the average obtained minimum values for all the three methods. Neither the SEA nor the multi-start ever succeeded in reaching the actual global minimum, which the HMS managed to do in every run. Full statistics are shown in a concise way in Fig. 3. Horizontal bars indicate minimum, mean and maximum, respectively. The width of the 'violin' indicates the distribution of values.

The second benchmark was the product of three reflected and vertically translated Gaussian functions over

Table 3. Average values of computed global minimum (Ackley 10D).

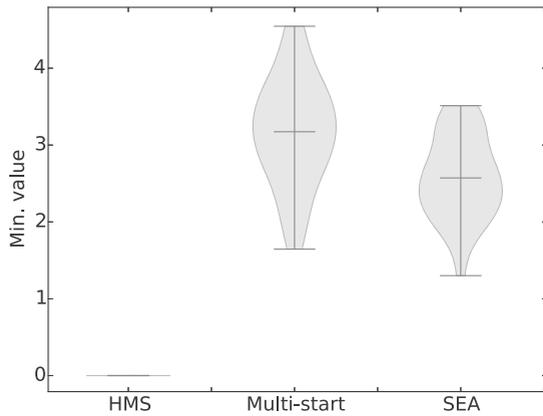|	HMS	Multi-start	SEA
Average	$5.27 \cdot 10^{-9}$	3.17	2.57
Best	$1.96 \cdot 10^{-10}$	1.65	1.3

Fig. 3. Global minima statistics (Ackley 10D).

the 4-dimensional box $[-5, 5]^4$,

$$f_2(x) = \prod_{i=1}^{3} \left[1 - \exp\left(-(x - m_i)^T A_i (x - m_i) \right) \right], \quad (7)$$

where $m_1 = (3, 2, 1, 0)$, $m_2 = (1, -3, -1, 3)$, $m_3 = (-2.5, 2, 2, 2)$ and

$$A_1 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$A_2 = \begin{bmatrix} 0.2 & 0 & 0 & 0 \\ 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0.2 \end{bmatrix},$$

$$A_3 = \begin{bmatrix} 1.5 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 0 & 1.5 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Benchmark f_2 has three separate zero-valued global minima m_1, m_2 and m_3 with no other local minima. The difficulty of finding the global minima is graded. Here m_2 has the broadest attraction basin, so it is fairly easily reachable. Then m_1 is steeper, hence more difficult to find. The main problem is reaching m_3. Outside their attraction basins, one can find quite large plateaus,

which can cause serious trouble for gradient optimization methods. Thus, the aim of this test is to check the compared methods' abilities of finding all global minima.

The HMS execution parameters for benchmark f_2 are gathered in Table 4. The obtained HMS objective call

Table 4. HMS execution parameters (three Gaussians).

	Root	Middle	Leaf
Population (initial)	50	10	4
Metaepoch length	2	2	2
Encoding scale	4.0	2.0	1.0
Mutation rate	0.4	0.1	0.01
Crossover rate	0.5	0.5	0.5
Mutation std. dev.	10.0	1.0	0.1
Sprout std. dev.	–	2.0	1.0
Sprout min. dist.	–	2.0	1.0

means are shown in Table 5. This time, to make things

Table 5. Average number of objective evaluations (three Gaussians).

Root	Middle	Leaf	Total	Weighted
6901.5	1135.1	1655.2	9691.8	3183.8

still more similar to MT computations, instead of a simple sum of different-level costs, we used a weighted cost

$$c = c_{\text{leaf}} + \frac{c_{\text{middle}}}{3} + \frac{c_{\text{root}}}{6}, \quad (8)$$

expressing the fact that in real problems the leaf computations are the most expensive ones, whereas the root calls are the cheapest. The denominators in (8) are taken from MT simulations, cf. see Section 5. As the comparative methods operated at the top level of accuracy, i.e., the leaf level, based on Table 5 we set the multi-start starter pool size to 50 and the SEA initial population to 70. As in the Ackley 10D case, we set the SEA genetic parameters accordingly to the corresponding HMS leaf-level parameters. We stopped the SEA right after exceeding 4000 calls. The main results of the test are shown in Table 6. The HMS successfully found all

Table 6. Statistics of finding all global minima (three Gaussians).

	HMS	Multi-start	SEA
Fully successful runs	30/30	16/30	0/30
Avg. number of mins	3	2.47	0

three minima in every execution. The multi-start failed in almost half of the runs. Moreover, in two runs it found only one global minimum. The SEA never succeeded, obtaining an average computed global minimum of 0.57 and a best found solution with the value of 0.18. As the results show, the three Gaussians benchmark is quite

a difficult optimization test, despite its relatively low dimensionality. The source of the difficulty is the small volume of global minima's attraction basins and the presence of significant plateaus.

4. Magnetotelluric inverse problem

The solar wind produces a radiation pressure that causes a compression on the day-side and a tail on the night-side onto the magnetosphere. Due to this interaction, hydromagnetic waves are created. When those waves reach the ionosphere, they induce an EM field that works as a power source in magnetotellurics. Depending upon the type of source we are dealing with, the geomagnetic fluctuations range between the frequencies of $10^{-3} - 10^5$ Hz, which allows us to make measurements with a resolution that ranges from a few meters to hundreds of kilometers (Vozoff, 1972)

The magnetotelluric (MT) technique is used to determine a resistivity map of the Earth's subsurface by performing electromagnetic (EM) measurements. The main difference of MT with respect to other geophysical measurement acquisition scenarios (e.g., marine controlled electromagnetic measurements) is that MT uses natural electromagnetic radiation sources generated within the ionosphere, instead of human powered antennas. Thus, acquisition of MT measurements is rather inexpensive, and can cover large areas. Applications of MT measurements include hydrocarbon (oil and gas) exploration and finding suitable regions for storage of CO_2.

4.1. Forward problem. MT measurements are governed by electromagnetic phenomena, which can be described by Maxwell's equations. When the electrical field depends only upon two spatial variables (x, z), then two independent and uncoupled modes are derived from these equations, namely, *transverse electric* (TE) and *transverse magnetic* (TM). The TE mode involves (E_y, H_x, H_z) field components while TM uses (H_y, E_x, E_z). We focus on the TE mode and we solve the equation for E_y.

We decouple Maxwell's equations by pre-multiplying both sides of Faraday's law by μ^{-1} and applying the curl operator. Then, after incorporating Ampere's law, we substitute component by component the result into the double curl operator on the electric field to obtain the equation for the y-component of the electric field:

$$-\Delta u - k^2 u = f \quad \text{in } \Omega \subset \mathbb{R}^2, \qquad (9)$$

where Ω is a simply connected bounded domain with a Lipschitz boundary and $u(x, z) = E_y(x, z)$, $(x, z) \in \Omega$. σ, $(\sigma > \sigma_0 > 0$ in $\Omega)$ is the electrical conductivity field, $k^2 = \omega^2 \mu \epsilon - j\mu\omega\sigma$, where $\omega \neq 0$ is the wave frequency,

$\epsilon, \mu \in \mathbb{R}, \epsilon, \mu > 0$ stand for the permittivity and permeability of the medium considered. $f = -j\omega\mu J_y^{\text{imp}}$, where J_y^{imp} is a prescribed, impressed electric current density radiating in the y-direction.

To obtain the corresponding variational formulation, we pre-multiply Eqn. (9) by a test function $v \in H_0^1(\Omega; \mathbb{C})$. After integrating by parts and incorporating the Dirichlet boundary conditions over $\Gamma_D = \partial\Omega$, the following abstract variational formulation (suitable for finite element computations) is obtained:

$$\begin{cases} \text{Find } u(\sigma) \in H_0^1(\Omega; \mathbb{C}) \text{ such that} \\ b(\sigma; u(\sigma), v) = F(v), \quad \forall v \in H_0^1(\Omega; \mathbb{C}), \end{cases} \qquad (10)$$

with

$$b(\sigma; u, v) = \int_\Omega \nabla\overline{v}\,\nabla u - \int_\Omega k^2\,\overline{v}\,u \qquad (11)$$

$$F(v) = \int_\Omega \overline{v}\,f, \qquad (12)$$

where now we assume that $\sigma \in L^\infty(\Omega)$ and $J_y^{\text{imp}} \in L^2(\Omega; \mathbb{C})$. The exact solution $u(\sigma) = E_y$ to the problem (10) is called the primal forward solution.

Let us define now the functionals $L^i : H_0^1(\Omega; \mathbb{C}) \to \mathbb{C}$ associated with the receiving antennas occupying Ω^i domains, respectively, $i = 1, \ldots, M$,

$$L^i(v) = \frac{1}{\text{meas}(\Omega^i)} \int_{\Omega^i} v. \qquad (13)$$

Now, we are able to define the set of dual forward problems,

$$\begin{cases} \text{Find } G(\sigma) \in H_0^1(\Omega; \mathbb{C}) \text{ such that} \\ b(\sigma; v, G(\sigma)) = L^i(v), \quad \forall v \in H_0^1(\Omega; \mathbb{C}), \end{cases} \qquad (14)$$

for each receiving antenna, $i = 1, \ldots, M$.

4.2. *hp*-FEM approximation. Using an *hp*-FEM finite dimensional internal approximation $V_{hp} \subset H_0^1(\Omega; \mathbb{C})$ (see Demkowicz, 2006), we obtain the discrete versions of both primal and dual forward problems (10), (14)

$$\begin{cases} \text{Find } u_{h,p}(\sigma) \in V_{hp} \text{ such that} \\ b(\sigma; u_{h,p}(\sigma), v) = F(v), \quad \forall v \in V_{hp}, \end{cases} \qquad (15)$$

$$\begin{cases} \text{Find } G_{h,p}^i(\sigma) \in V_{hp} \text{ such that} \\ b(\sigma; v, G_{h,p}^i(\sigma)) = L^i(v), \quad \forall v \in V_{hp}, \end{cases} \qquad (16)$$

for each receiving antenna, $i = 1, \ldots, M$.

To control the discretization error by performing grid refinements, we use the two-dimensional (2D) *hp*-adaptive algorithm described by Demkowicz (2006). It incorporates two basic components:

- Fine grid solution: giving an existing coarse mesh u_{hp}, we refine the mesh in both h and p to obtain a finer mesh. Its solution is denoted by $u_{h/2,p+1}$. The hp refined grid provides a reference solution that is used for comparison purposes and to decide what areas of the computational mesh contain larger errors.

- Optimal mesh selection: given the fine mesh solution, we use it to determine optimal mesh refinements on the coarse mesh by minimizing the projection based interpolation error (see Demkowicz, 2006).

The optimization algorithm contains the following steps:

1. For each element in the coarse mesh, we select between p and h refinements. Using the projection-based interpolation error, we select the refinement that provides the biggest error decrease rate by comparing the fine and coarse mesh solutions.

2. We determine which elements to refine by computing the maximum decrease rate over all elements, and selecting for refinements those that exhibit an error decrease rate larger than a fraction of the maximum. In our case, we select $1/3$, as suggested by Demkowicz (2006).

3. We determine an optimal order of approximation for the elements to be refined: if p-refined is selected, then we increase p by one. If h-refinement wins, then we determine the value of the new polynomial orders of approximation for those new elements that yield the biggest descent rate.

For goal-oriented adaptivity where the main objective is to accurately approximate $L(E_y)$, we employ the adjoint problem to represent the error in the quantity of interest. Thus, the corresponding refinement strategy is based on minimizing the error in the quantity of interest rather than an arbitrary energy norm (for details, see Pardo *et al.*, 2006).

4.3. Inverse problem. Our aim is to obtain the impedance, a suitable physical magnitude to perform the inversion. To do so, the magnetic field is obtained from Maxwell's equations as

$$H_x(\sigma) = \frac{1}{j\omega\mu} \frac{\partial E_y(\sigma)}{\partial z}, \tag{17}$$

and then impedance $\mathcal{Z}$ is computed according to

$$\mathcal{Z} = \mathcal{Z}_{yx} = \frac{E_y}{H_x}. \tag{18}$$

The impedance at each antenna is a nonlinear functional $g^i(\sigma)$ computed from two linear quantities:

$$
\begin{aligned}
g^i(\sigma) &= \frac{L^i(E_y(\sigma))}{L^i(H_x(\sigma))} \\
&= j\omega\mu\, L^i(u(\sigma)) \left(L^i\left(\frac{\partial u(\sigma)}{\partial z} \right) \right)^{-1}
\end{aligned}
\tag{19}
$$

for $i = 1, \ldots, M$, whose derivative with respect to an arbitrary variable s (in particular, the conductivity σ) can be obtained from the formula

$$
\begin{aligned}
\frac{\partial g^i(\sigma)}{\partial s} = \frac{1}{(L^i(H_x))^2} &\left(\frac{\partial L^i(E_y)}{\partial s} L^i(H_x) \right. \\
&\left. - \frac{\partial L^i(H_x)}{\partial s} L^i(E_y) \right),
\end{aligned}
\tag{20}
$$

where all the above partial derivatives are distributional ones. Let us denote by d^i_{obs} the measured value of impedances at the antennas, which correspond to the quantities $g^i(\overline{\sigma})$, where $\overline{\sigma} \in L^\infty(\Omega)$ is the real, unknown conductivity distribution.

We introduce the stepwise representation of the conductivity function σ as a member of a family

$$
\begin{aligned}
\mathcal{D} =\ & \{\xi \in L^\infty(\Omega); \xi(x) = \textstyle\sum_{i=1,\ldots,n} \xi_i \chi_i(x), \\
& 0 < \xi_i^{\min} \leq \xi_i \leq \xi_i^{\max} < +\infty\},
\end{aligned}
\tag{21}
$$

where $\{\chi_i\}_{i=1,\ldots,n}$ are the indicator functions of the disjoint covering $\{\Omega_i\}_{i=1,\ldots,n}$ of domain Ω ($\bigcup_{i=1,\ldots,n} \Omega_i = \Omega, \Omega_i \cap \Omega_j, i \neq j$). This representation is utilized by the hp-FEM computing of the primary and dual Galerkin problems (15), (16). Next, we define for $\sigma \in \mathcal{D}$ the approximate impedance functions

$$
g^i_{hp}(\sigma) = j\omega\mu\, L^i(u_{h,p}(\sigma)) \left(L^i\left(\frac{\partial u_{h,p}(\sigma)}{\partial z} \right) \right)^{-1}.
\tag{22}
$$

The inverse problem under consideration reads as follows: Find $\hat{\sigma} \in \mathcal{D}$ such that, for all $\sigma \in \mathcal{D}$,

$$
\begin{aligned}
&\lim_{\substack{h \to 0 \\ p \to +\infty}} \left\{ \frac{1}{2M} \sum_{i=1}^{M} \left| g^i_{h,p}(\hat{\sigma}) - d^i_{\text{obs}} \right|^2 \right\} \\
&\quad \leq \lim_{\substack{h \to 0 \\ p \to +\infty}} \left\{ \frac{1}{2M} \sum_{i=1}^{M} \left| g^i_{h,p}(\sigma) - d^i_{\text{obs}} \right|^2 \right\}.
\end{aligned}
\tag{23}
$$

4.4. Common error scaling. The hp-FEM adaptation is performed by solving forward primal and dual problems for quantities of interest $L^i, i = 1, \ldots, M$. The relative hp-FEM error scaling is selected based on the following result.

Proposition 1. *Assuming* $g^i(\overline{\sigma}) = d^i_{obs}, i = 1, \ldots, M,$ *the misfit function might be evaluated as the product of*

the squared norms of the relative hp-FEM errors of primal and dual solutions added to the squared norm of the absolute hp-FEM errors of primal solutions obtained for all coordinates of the logging curve plus the square of the error associated with the inverse problem, i.e.,

$$
\frac{1}{2M} \sum_{i=1}^{M} \left| g_{h/2,p+1}^{i}(\sigma) - g^{i}(\overline{\sigma}) \right|^{2}
$$

$$
\leq A \left\| e_{\text{rel}}(\sigma) \right\|_{0,1}^{2} \left(\sum_{i=1}^{M} \left\| \epsilon_{\text{rel}}^{i}(\sigma) \right\|_{0,1}^{2} \right) \quad (24)
$$

$$
+ B \left\| u_{h,p}(\sigma) - u(\sigma) \right\|_{0,1}^{2}
$$

$$
+ C \left\| \sigma - \overline{\sigma} \right\|_{L^{\infty}(\Omega)}^{2},
$$

where $e_{\text{rel}}(\sigma) = u_{h/2,p+1}(\sigma) - u_{h,p}(\sigma)$ is the relative hp-FEM error in solving the forward primal problem (15), $\epsilon_{\text{rel}}^{i}(\sigma) = G_{h/2,p+1}^{i}(\sigma) - G_{h,p}^{i}(\sigma), i = 1, \ldots, M$ are the relative hp-FEM errors in solving the forward dual problems (16) and A, B, C are positive constants.

We start with a useful lemma which is a particular case of a more general result obtained for the AC logging data inversion (see Smołka *et al.*, 2015, Proposition 9).

Lemma 1. *The distance between the approximated and exact quantities of interest might be evaluated as the product of the squared norms of the relative hp-FEM errors of primal and dual solutions added to the squared norm of the absolute hp-FEM errors of primal solutions obtained for all coordinates of the logging curve plus the square of the error associated to the inverse problem, i.e.,*

$$
\sum_{i=1}^{M} \left| L^{i}(u_{h/2,p+1}(\sigma)) - L^{i}(u(\overline{\sigma})) \right|^{2}
$$

$$
\leq C' \left\| e_{\text{rel}}(\sigma) \right\|_{0,1}^{2} \left(\sum_{i=1}^{M} \left\| \epsilon_{\text{rel}}^{i}(\sigma) \right\|_{0,1}^{2} \right) \quad (25)
$$

$$
+ C'' \left\| u_{h,p}(\sigma) - u(\sigma) \right\|_{0,1}^{2}
$$

$$
+ C''' \left\| \sigma - \overline{\sigma} \right\|_{L^{\infty}(\Omega)}^{2}.
$$

Proof. (Proposition 1) For simplicity write $a = u_{h/2,p+1}(\sigma)$, $b = u(\overline{\sigma})$ and denote by $a_{,z}$, $b_{,z}$ their partial derivatives with respect to the variable z. Now, we can evaluate

$$
\left| g_{h/2,p+1}^{i}(\sigma) - g^{i}(\overline{\sigma}) \right| = j\omega\mu \left| \frac{L^{i}(a)}{L^{i}(a_{,z})} - \frac{L^{i}(b)}{L^{i}(b_{,z})} \right|. \quad (26)
$$

The right-hand-side of the above expression is less than or equal to the following:

$$
\left| \frac{L^{i}(a)}{L^{i}(a_{,z})} - \frac{L^{i}(b)}{L^{i}(b_{,z})} \right| + \left| L^{i}(b) \right| \left| \frac{1}{L^{i}(a_{,z})} - \frac{1}{L^{i}(b_{,z})} \right|
$$

$$
= \frac{1}{\left| L^{i}(a_{,z}) \right|} \left| L^{i}(a) - L^{i}(b) \right|
$$

$$
+ \frac{\left| L^{i}(b) \right|}{\left| L^{i}(a_{,z}) \right| \left| L^{i}(b_{,z}) \right|} \left| L^{i}(a_{,z} - b_{,z}) \right|. \quad (27)
$$

Because we consider positive frequencies of a cosmic propagation, it is possible to establish a universal constant $\alpha > 0$ that bounds from below $\left| L^{i}\left(\frac{\partial u}{\partial z} \right) \right|$ for all solutions u to (13) and for all $i = 1, \ldots, M$. Moreover, all solutions to (13) are bounded from above in $H_{0}^{1}(\Omega)$, so we have the net universal constant $\beta < +\infty$ that is greater than or equal to $L^{i}(u)$ for all solutions to (13) and for all $i = 1, \ldots, M$. Next, all functionals $L^{i}\left(\frac{\partial u}{\partial z} \right)$ are uniformly Lipschitz continuous on $H_{0}^{1}(\Omega)$ with the constant L_{1}. Finally,

$$
\left\| u_{h/2,p+1}(\sigma) - u(\overline{\sigma}) \right\|_{1,0}
$$

$$
\leq \left\| u_{h/2,p+1}(\sigma) - u(\sigma) \right\|_{1,0} + \left\| u(\sigma) - u(\overline{\sigma}) \right\|_{1,0}
$$

$$
\leq \left\| u_{h,p}(\sigma) - u(\overline{\sigma}) \right\|_{1,0} + L_{0} \left\| \sigma - \overline{\sigma} \right\|_{L^{\infty}(\Omega)}.
$$

Summing up (26) and (27), we have

$$
\left| g_{h/2,p+1}^{i}(\sigma) - g^{i}(\overline{\sigma}) \right|
$$

$$
\leq C_{1} \left| L^{i}(u_{h/2,p+1}(\sigma)) - L^{i}(u(\overline{\sigma})) \right|
$$

$$
+ C_{2} \left\| u_{h,p}(\sigma) - u(\overline{\sigma}) \right\|_{1,0} \quad (28)
$$

$$
+ C_{3} \left\| \sigma - \overline{\sigma} \right\|_{L^{\infty}(\Omega)},
$$

where $C_{1} = j\omega\mu/\alpha$, $C_{2} = C_{1}\beta L_{1}/\alpha$, $C_{3} = (C_{2} + 1)L_{0}$ and L_{0} is the Lipschitz constant of the function that maps the conductivity to the solution of the primal problem (13), i.e., $\sigma \to u(\sigma)$ in norms $\left\| \cdot \right\|_{1,0}$ and $\left\| \cdot \right\|_{L^{\infty}(\Omega)}$ (see Smołka *et al.*, 2015, Lemma 4). Using twice the simple inequality $(a + b)^{2} \leq 2a^{2} + 2b^{2}$, we obtain

$$
\left| g_{h/2,p+1}^{i}(\sigma) - g^{i}(\overline{\sigma}) \right|^{2}
$$

$$
\leq C_{1} \left| L^{i}(u_{h/2,p+1}(\sigma)) - L^{i}(u(\overline{\sigma})) \right|^{2}
$$

$$
+ C_{2} \left\| u_{h,p}(\sigma) - u(\overline{\sigma}) \right\|_{1,0}^{2} \quad (29)
$$

$$
+ C_{3} \left\| \sigma - \overline{\sigma} \right\|_{L^{\infty}(\Omega)}^{2},
$$

for other constants C_{1}, C_{2}, C_{3}.

Coupling (29) with the result of Lemma 1, we obtain

$$
\frac{1}{2M} \sum_{i=1}^{M} \left| g_{h/2,p+1}^{i}(\sigma) - g^{i}(\overline{\sigma}) \right|^{2}
$$

$$
\leq \frac{C_1 C'}{2M} \left\| e_{\mathrm{rel}}(\sigma) \right\|_{0,1}^{2} \left(\sum_{i=1}^{M} \left\| \epsilon_{\mathrm{rel}}^{i}(\sigma) \right\|_{0,1}^{2} \right) \qquad (30)
$$

$$
+ \frac{C_1 C'' + C_2}{2M} \left\| u_{h,p}(\sigma) - u(\sigma) \right\|_{0,1}^{2}
$$

$$
+ \frac{C_1 C''' + C_3}{2M} \left\| \sigma - \overline{\sigma} \right\|_{L^{\infty}(\Omega)}^{2}.
$$

It is sufficient to set

$$
A = \frac{C_1 C'}{2M},
$$
$$
B = \frac{C_1 C'' + C_2}{2M},
$$
$$
C = \frac{C_1 C''' + C_3}{2M}
$$

to finish the proof. ∎

Remark 1. The first right-hand side component of (25) expresses the influence of the relative forward errors (primal and dual ones) imposed on the hp-FEM refinement process. The second one is proportional to the square of the FEM error, which decreases to zero during hp refinements. The third component is evaluated from below by $C\delta_j^2$, where δ_j is the accuracy in the inverse search domain $\mathcal{D}$ utilized by the j-th level HMS branch. In order to make the hp-HMS inversion at the j-th level computationally economic, we should keep the first and third components comparable. In other words, decreasing $\left\| e_{\mathrm{rel}}(\sigma) \right\|_{0,1}$ and $\left\| \epsilon_{\mathrm{rel}}^{i}(\sigma) \right\|_{0,1}$ below the quantity

$$
\mathrm{Ratio}(j) = \sqrt{\frac{\delta_j}{M}} \sqrt[4]{\frac{C}{A}}
$$

does not improve the accuracy of fitness evaluation.

5. Sample magnetotelluric simulations

5.1. Domain description. We consider a 2D formation of a subsurface consisting of horizontally layered media with some 2D inhomogeneities. We model the source as an infinitely long (in the x and y directions) volumetric rectangular surface located at the ionosphere. This allows us to treat the electromagnetic fields as plane waves that propagate in the vertical direction towards the Earth's surface (Vozoff, 1972).

The physical problem is illustrated in Fig. 4. The computational domain of $2500\,\mathrm{km} \times 130$ km, consists of air and layered media with 2D inhomogeneities located at the center of the domain (in the x direction). The horizontal length of the domain where the inhomogeneities are considered is 10 km.

The physical domain is surrounded by a perfectly matched layer (PML) (Berenger, 1994) used to truncate the computational domain, and we consider the same relative permittivity and permeability in all materials (equal to one). The dark gray, middle gray and light gray regions in the lower half of the figure are filled with soil (or rock) deposits characterized by different conductivities $\sigma_1 - \sigma_4$. Receivers $A_1 - A_7$ are located at the Earth's surface, and are represented with the inverted t-marks $\perp$. The source, located at the ionosphere, is represented with a thin dark gray rectangle at the top of the figure.

5.2. Results. In our simulations we used the goal-oriented hp-adaptive finite element method solver, which computed both the impedance and its partial derivatives at the receivers in one run. For the simulations, the solver operated at three accuracy levels: 70%, 20% and 3%, meaning the maximal relative FEM error percentage. The 'exact' values d_{obs}^{i} were computed using the same solver with the accuracy of 1.2%. The HMS was executed with the same deme-tree layout as in the case of Ackley and three Gaussians. However, this time the computations were much more expensive. Therefore, in this case we performed only five executions of all three comparative methods, i.e., HMS, multi-start and SEA. The HMS execution parameters are summarized in Table 7. Every HMS run lasted about 32 hours. The simulations were executed on a single node of a Linux cluster with 16 Intel® Xeon® CPU E5620 at 2.40 GHz and 16 GB of total memory available. Our simulation at this point utilizes one core only; however, it is possible to speed it up by linking to a parallel MUMPS solver, which delivers around 60% efficiency up to 16 cores. It

Table 7. HMS execution parameters (MT).

	Root	Middle	Leaf
Population (initial)	20	10	5
Metaepoch length	2	2	2
Encoding scale	16384.0	128.0	1.0
Mutation rate	0.2	0.05	0.01
Crossover rate	0.5	0.5	0.5
Mutation std. dev.	3.0	0.6	0.1
Sprout std. dev.	–	1.0	0.2
Sprout min. dist.	–	1.0	0.2

turned out that the average time of FEM computations was about $45\,\mathrm{s}$ for the accuracy level of 70%, $1.5\,\mathrm{min}$ for the accuracy level of 20% and $4.5\,\mathrm{min}$ for the accuracy level of 3%. This justifies the weighted cost formula (8) applied in the three Gaussian test and actually appropriate in this case. The obtained HMS objective call means are shown in Table 8. After appropriate estimations, we set the multi-start initial pool size to 100 and the initial SEA population to 40. The SEA genetic parameters were set as in the previous tests. During the computations,
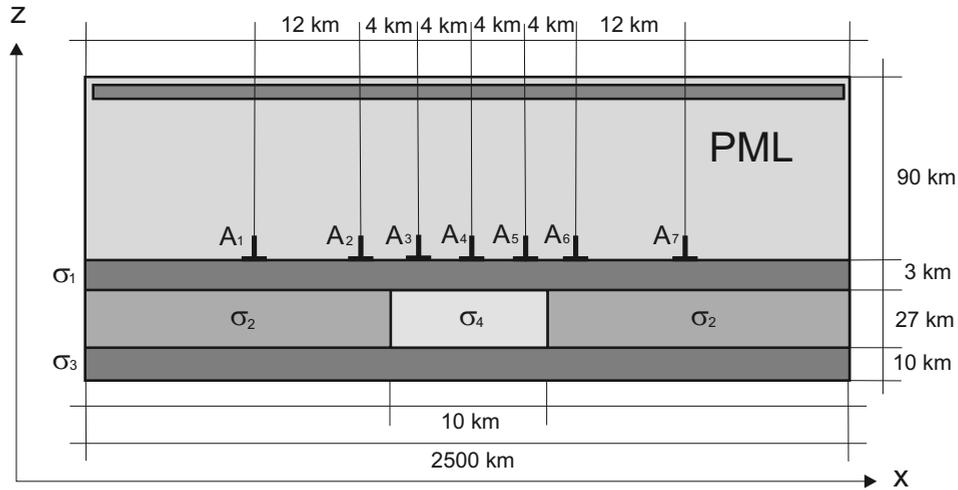
Fig. 4. Domain geometry.

Table 8. Average number of objective evaluations (MT).

Root	Middle	Leaf	Total	Weighted
398.2	629.9	148.6	1176.7	424.9

it appeared that the objective function has several flat regions where gradient methods got stuck, revealing many local minima of a mediocre or even poor value. Hence, in the comparisons we emphasize the good minima, i.e., the ones with the objective value less than a given threshold. In Table 9, we show the total number of such minima found in all runs for 2 threshold values. Figure 5 shows

Table 9. Total number of 'good' minima found (MT).

Upper bound	HMS	Multi-start	SEA
$1 \cdot 10^{-10}$	4	2	0
$5 \cdot 10^{-10}$	35	21	0

Fig. 5. All minima statistics (MT).

a comparison of all minimal values obtained by the three methods.

The results show that both the HMS and multi-start win by far to the SEA. The advantage of the HMS over multi-start is more apparent in the best minima area. Namely, the HMS found twice as many minima smaller than 10^{-10} as multi-start and 1.7 times as many minima smaller than $5 \cdot 10^{-10}$ as multi-start, cf. Table 9. Moreover, Fig. 5 shows that multi-start is more prone to getting stuck in poor local minima.

6. Conclusions

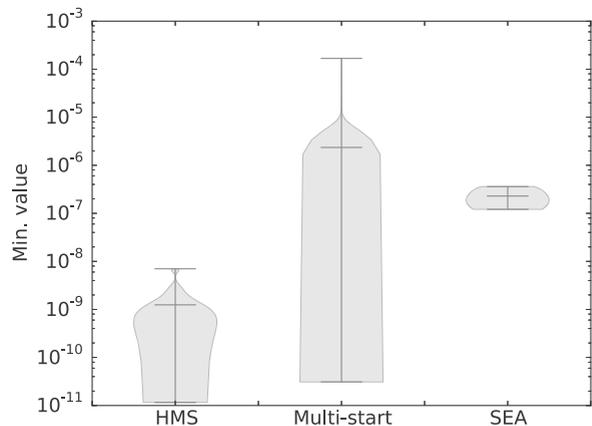The proposed strategy, HMS, constitutes a significant extension of the standard, two-phase global optimization

strategies composed of the global search phase followed by a couple of local convex searches. The essence of this extension is the application of the adaptive hierarchic genetic search alternating with the local searches with dynamically adapted accuracy. The genetic multi-deme component, together with the inter-processing of solutions (clustering of individuals) and fitness deterioration, allows a broad, global search, along with concentrating local searches in the basins of attraction of the deepest local minima. An additional novelty is introduction of the active individuals (computing agents) and agent-based distributed computing, governing the whole strategy. Contrarily to traditional inversion algorithms, our hybrid strategy delivers multiple solutions in the case of misfit multimodality, which enables an expert in the field to determine the best possible solution as well as the

uncertainty level.

The HMS offers an exceptionally low computational cost, which is crucial in solving inverse parametric problems involving partial differential equations. First, the hierarchic structure of demes decreases the number of individuals searching with a high accuracy. The number of fitness evaluations is additionally decreased by the conditional sprouting and redundancy reduction as well as the restrictive deme stopping condition. Next, the global search is boosted by concurrent identification of separate basins of attractions by separate well fitted leaf-demes. A huge cost reduction is caused by the dynamical adaptation of the accuracy of the self-adaptive hp goal-oriented finite element method applied as the forward solver (see Algorithm 5). This accuracy is adapted to the inverse error at the particular level of the population tree, which reduces the cost of the given fitness evaluation for the root and branch demes.

Simulations show the main HMS advantages over the standard global optimization methods (evolutionary algorithm and multi-start) for the Ackley and three Gaussian peaks benchmarks that exhibit typical difficulty characteristics of the inverse parametric problems (objective multimodality and ill conditioning). Another computational example shows the HMS behavior for the difficult, real-life engineering problem of magnetotelluric data inversion, originated by oil deposit investigation. Apart from the simulation study of the HMS efficiency, the theoretical verification of the common error scaling was performed, which is necessary for the computational cost reduction shown in Proposition 1 and Remark 1.

The tests we performed were of the "finite budget" type. We intended to provide the HMS with enough resources to obtain satisfactory solutions, and then check the results of standard global optimization methods running within the same computational budget. In our case, the budget is determined by the forward server calls, because in the case of parametric inverse problems their total CPU time exceeds the cost of all other operations by no less than two orders of magnitude. For instance, the least expensive, 70%-accuracy MT solver run lasted about 45 seconds, whereas the most expensive of the other operations, i.e., the objective deterioration (including the clustering), took a fraction of a second. Every MA global loop (see Section 2.1) featured much more than 10 solver calls at various accuracy levels and at most one objective deterioration request. Therefore, the cost of the other operations does not affect the comparison results of the method noticeably. On the other hand, all methods were implemented and run in a distributed environment, so the wall time of each computational run completion depended significantly on the background load of all nodes, scheduling policy, etc. Because the main goal of the paper was to analyze the algorithmic aspects of the proposed strategy, we have not refined the implementation and scheduling policies. It will be the subject of our future research based on our former result concerning agent-based "diffusive scheduling" in distributed environments (Grochowski *et al.*, 2006).

Acknowledgment

References

Álvarez-Aramberri, J., Pardo, D. and Barucq, H. (2013). Inversion of magnetotelluric measurements using multigoal oriented hp-adaptivity, *Procedia Computer Science* **18**: 1564–1573.

Barabasz, B., Gajda-Zagórska, E., Migórski, S., Paszyński, M., Schaefer, R. and Smołka, M. (2014). A hybrid algorithm for solving inverse problems in elasticity, *International Journal of Applied Mathematics and Computer Science* **24**(4): 865–886, DOI: 10.2478/amcs-2014-0064.

Beasley, D., Bull, D.R. and Martin, R.R. (1993). A sequential niche technique for multimodal function optimization, *Evolutionary Computation* **1**(2): 101–125.

Berenger, J.-P. (1994). A perfectly matched layer for the absortion of electromagnetic waves, *Journal of Computational Physics* **114**(2): 185–200.

Byrski, A., Schaefer, R., Smołka, M. and Cotta, C. (2013). Asymptotic guarantee of success for multi-agent memetic systems, *Bulletin of the Polish Academy of Sciences: Technical Sciences* **61**(1): 257–278.

Cetnarowicz, K., Kisiel-Dorohinicki, M. and Nawarecki, E. (1996). The application of evolution process in multi-agent world (MAW) to the prediction system, *in* M. Tokoro (Ed.), *Proceedings of the 2nd International Conference on Multiagent Systems (ICMAS-96), Menlo Park, CA, USA*, pp. 26–32.

Demkowicz, L. (2006). *Computing with hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems*, Chapman & Hall/CRC, Boca Raton, FL.

Demkowicz, L., Kurtz, J., Pardo, D., Paszyński, M., Rachowicz, W. and Zdunek, A. (2007). *Computing with hp-Adaptive Finite Elements, Vol. 2. Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*, Chapman & Hall/CRC, Boca Raton, FL.

Ester, M., Kriegel, H.-P., Sander, J. and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining KDD-96, Portland, OR, USA*, pp. 226–231.

FIPA (2002). *Foundation for Intelligent Physical Agents (FIPA) Specifications*, www.fipa.org.

Gajda-Zagórska, E., Schaefer, R., Smołka, M., Paszyński, M. and Pardo, D. (2015). A hybrid method for inversion of 3D DC resistivity logging measurements, *Natural Computing* **14**(3): 355–374. DOI: 10.1007/s11047-014-9440-y.

Grochowski, M., Smołka, M. and Schaefer, R. (2006). Architectural principles and scheduling strategies for computing agent systems, *Fundamenta Informaticae* **71**(1): 15–26.

Jojczyk, P. and Schaefer, R. (2009). Global impact balancing in the hierarchic genetic search, *Computing and Informatics* **28**(2): 181–193.

Neri, F., Cotta, C. and Moscato, P. (Eds.) (2012). *Handbook of Memetic Algorithms*, Studies in Computational Intelligence, Vol. 379, Springer, Heidelberg.

Obuchowicz, A. (1997). The evolutionary search with soft selection and deterioration of the objective function, *Proceedings of the 6th International Conference on Intelligent Information Systems IIS'97, Zakopane, Poland*, pp. 288–295.

Pardo, D., Demkowicz, L., Torres-Verdín, C. and Tabarovsky, L. (2006). A goal-oriented hp-adaptive finite element method with electromagnetic applications, Part I: Electrostatics, *International Journal for Numerical Methods in Engineering* **65**(8): 1269–1309.

Schaefer, R. and Kołodziej, J. (2003). Genetic search reinforced by the population hierarchy, *in* K.A. De Jong, R. Poli and J. Rowe (Eds.), *Foundations of Genetic Algorithms 7*, Morgan Kaufman, San Francisco, CA, pp. 383–399.

Smołka, M., Gajda-Zagórska, E., Schaefer, R., Paszyński, M. and Pardo, D. (2015). A hybrid method for inversion of 3D AC logging measurements, *Applied Soft Computing* **36**: 442–456.

Smołka, M. and Schaefer, R. (2014). A memetic framework for solving difficult inverse problems, *in* A.I. Esparcia-Alcázar and A.M. Mora (Eds.), *EvoApplications 2014*, Lecture Notes in Computer Science, Vol. 8602, Springer, Berlin/Heidelberg, pp. 138–149.

Vozoff, K. (1972). The magnetotelluric method in the exploration of sedimentary basins, *Geophysics* **37**(1): 98–141.

Wierzba, B., Semczuk, A., Kołodziej, J. and Schaefer, R. (2003). Hierarchical genetic strategy with real number encoding, *Proceedings of the 6th Conference on Evolutionary Algorithms and Global Optimization, Łagów, Poland*, pp. 231–237.

Wolny, A. and Schaefer, R. (2011). Improving population-based algorithms with fitness deterioration, *Journal of Telecommunications and Information Technology* **4**: 31–44.

Maciej Smołka received his Ph.D. (2000) in the area of shape optimization at Jagiellonian University in Kraków. Until 2012 he worked in the Chair of Optimization and Control Theory at Jagiellonian University. Currently he is a member of the Adaptive Algorithms and Systems Group in the Department of Computer Science, AGH University of Science and Technology, Kraków. His present research interests include stochastic modeling of computational systems, application of evolutionary and hybrid algorithms in solving inverse problems and computational multi-agent systems.

Robert Schaefer is a full professor at the Department of Computer Science, Faculty of Computer Science Electronics and Telecommunication, AGH University of Science and Technology. An author and co-author of about 180 books, papers and conference contributions, in particular of the book *Foundations of Genetic Global Optimization* (Springer, 2007). The general chair of the *PPSN 2010* conference and a steering committee member of the PPSN series. A PC member and a co-chair of more than 100 scientific conferences in computational sciences and artificial intelligence. Recent research areas: algorithms of oil and gas surveying, genetic and memetic algorithms, computing multi-agent systems, adaptive algorithms of solving forward and inverse problems for PDEs. Former research: modeling of the blood flow in arteries, modeling of nonlinear flow in porous media.

Maciej Paszyński received his Ph.D. (2003) in mathematics with applications to computer science from Jagiellonian University, Kraków, Poland, and his habilitation (2010) in computer science from the AGH University of Science and Technology, Kraków. In 2003–2005, he worked as a postdoctoral fellow at the Institute for Computational Engineering and Sciences (ICES) at the University of Texas at Austin. In the summer of 2006 and 2007, he worked as a postdoctoral fellow at the Department of Petroleum and Geosystems Engineering at the University of Texas at Austin. Since 2007 he has been a frequent visitor at the ICES at the University of Texas at Austin, USA, at the King Abdullah University of Science and Technology (KAUST), Saudi Arabia, at the Basque Center for Applied Mathematics (BCAM) and at the Department of Mathematics, Universidad del Pais Vasco, Bilbao, Spain. Since 2013 he has been the vice-director of the Department of Computer Science, AGH University, and holds a university professorial position there. His research interests include parallel self-adaptive hp finite element method, parallel forward solvers, models of concurrency and isogeometric methods.

David Pardo is a research professor at Ikerbasque, the University of the Basque Country UPV/EHU, and the Basque Center for Applied Mathematics (BCAM). He has published over 120 research articles and given over 200 presentations. In 2011, he was awarded as the best Spanish young researcher in applied mathematics by the Spanish Society of Applied Mathematics (SEMA). He leads several national and international research projects, as well as research contracts with national and international companies. He is now the PI of the research group on mathematical modeling, simulation, and industrial applications (M2SI). His research interests include computational electromagnetics, petroleum-engineering applications (borehole simula-

tions), adaptive finite-element and discontinuous Petrov–Galerkin methods, multigrid solvers, image restoration algorithms, and multiphysics and inverse problems.

Julen Álvarez-Aramberri completed his degree in physics at the University of the Basque Country in 2007. After that, he studied for an M.Sc. in quantitative finance and an M.Sc. in mathematical modeling, statistics and computation at the University of the Basque Country. Since 2011 he has been a Ph.D. student within the Department of Applied Mathematics, Statistics and Operational Research at the University of the Basque Country under the supervision of David Pardo and within the Team Project *MAGIQUE-3D* at the University of Pau under the supervision of Hélène Barucq. He currently is in the last year of his Ph.D. working at the Basque Center for Applied Mathematics (BCAM). His Ph.D. is expected to be finished in 2015. His research interest include efficient implementation of numerical schemes, methods, and tools. In particular, he is focused on solving direct and inverse problems arising in applications of the magnetotelluric technique, which is used to retrieve information about the resistivity distribution of the Earth's subsurface.