Philippine Journal of Science
149 (3-a): 747-772, October 2020
ISSN 0031 - 7683
Date Received: 20 Apr 2020

Comparative Study of Heuristic Algorithms for Electrical Impedance Tomography

Arrianne Crystal Velasco[1,2]*, **Marion Darbas**[1], **Renier Mendoza**[2,3], **Monica Bacon**[2], and **John Cedrick de Leon**[2]

[1]CNRS UMR 7352, LAMFA
Université de Picardie Jules Verne, 80000 Amiens, France
[2]Institute of Mathematics; [3]Natural Sciences Research Institute
University of the Philippines Diliman, Quezon City 1101 Philippines

Based on electrical measurements from electrodes placed around the boundary of a body, electrical impedance tomography (EIT) is an imaging procedure that recovers the spatial distribution of the conductivities in the interior of a body. Recent studies have shown promising results in reconstructing EIT images using heuristic algorithms. This work presents a study of the applicability of six heuristic algorithms – firefly algorithm (FA), novel bat algorithm (NBA), genetic algorithm with new multi-parent crossover (GA-MPC), success history-based adaptive differential evolution with linear population size reduction with semi-parameter adaptation hybrid with covariance matrix adaptation evolutionary strategy (LSHADE-SPACMA), ensemble sinusoidal differential covariance matrix adaptation (LSHADE-cnEpSin), and effective butterfly optimizer with covariance matrix adapted retreat phase (EBOwithCMAR) – for the EIT image reconstruction problem. These algorithms have never been employed to solve the EIT inverse problem. Series of numerical tests were carried out to compare the performance of the selected algorithms.

Keywords: electrical impedance tomography, heuristic algorithms, inverse problem

INTRODUCTION

EIT is a non-invasive imaging technique in which the conductivity distribution within a body is reconstructed given measurements of electrical current and voltage around its boundary. Most imaging modalities, such as computerized tomography scan and magnetic resonance imaging scan, make use of ionizing radiation which can adversely affect human health if not properly used or contained (Mettler 2012). This concern led to efforts to develop radiation-free tomographic procedures, one of which is the EIT.

Image quality is a constant subject in EIT research, as EIT images are known to have poor spatial resolution due to the technique's soft-field property and a limited number of independent measurements (Miao *et al*. 2014). Image reconstruction *via* EIT is also an ill-posed problem (Holder 2000). Nevertheless, due to its low cost, portability, and non-invasive property, EIT is safe for long-term continuous monitoring – a trait that is lacking in most of the currently

*Corresponding Author: acvelasco@math.upd.edu.ph

available imaging technologies. In addition, different regularization techniques have been developed over the past years to weaken its ill-posedness, resulting to more accurate and stable reconstructions.

EIT has wide-ranging applications in biomedical monitoring, geophysics, and industrial processes. For instance, it is a potential tool for the detection of breast cancer (Zuo and Guo 2003); imaging of gastric emptying (Mangnall *et al*. 2003); and monitoring of pulmonary, brain, and cardiovascular functions (Isaacson *et al*. 2003). It can also be used to obtain information about rock porosity and fracture formation (Parker 1984), detect leaks from buried pipes (Jordana *et al*. 2001), and nondestructively test for material defects (Eggleston *et al.* 1990).

A typical EIT system consists of a set of electrodes attached around the boundary of a body, through which low-frequency alternating currents are injected. The entire EIT operation can be divided into two parts: the forward problem and the inverse problem. The forward model involves the calculation of the resulting voltages at the boundary of a body after electric currents are injected. In this paper, the forward problem is solved using the finite element (FE) method, considering the complete electrode model, which is currently the most accurate for impedance tomography (Somersalo *et al.* 1992). On the other hand, the inverse problem, also known as the reconstruction problem, involves the recovery of the conductivity distribution within a body using boundary voltage data. Yorkey *et al.* (1987) compared several deterministic algorithms in solving the inverse EIT problem. They have shown that the modified Newton-Raphson method can be an effective reconstruction algorithm. Recent studies have proposed the use of heuristic approaches to solve the inverse problem. In the study of Rashid *et al.* (2010), differential evolution (DE) algorithm has been shown to recover the geometry of the inclusions of a circular domain with improved results. In the study of Ribeiro *et al.* (2014), a non-blind search is used to generate the first set of the population for GA and it obtained reasonable reconstruction of one object in a circular domain after a hundred iterations. Heuristic algorithms are capable of converging towards the global minimum, given sufficient computation time and an appropriate choice of parameters. They are also very flexible and, therefore, are less restricted to certain forms of constraints (Maringer 2005). Furthermore, these methods are not dependent on the initial guess and the gradient of the cost functional. There are several works on the use of heuristic algorithms in solving the inverse EIT problem. Popular heuristic optimization methods like genetic algorithms (Mendoza and Lope 2012; Feitosa *et al.* 2014b; Barbosa *et al.* 2017), DE (Li *et al.* 2003; Barbosa *et al.* 2018), particle swarm optimization (Feitosa *et al.* 2014a), and simulated annealing (Martins *et al.* 2012; Tavares *et al.* 2012) were used to solve the image reconstruction problem in EIT. A book chapter discussing the implementation of several evolutionary and bioinspired algorithms on the EIT image reconstruction problem can be found in the study of dos Santos *et al.* (2018).

The goal of this study is to present a comparative analysis of six heuristic algorithms – FA, NBA, GA-MPC, LSHADE-SPACMA, LSHADE-cnEpSin, and EBOwithCMAR – for the EIT image reconstruction problem. Comparative analysis of the above-mentioned algorithms is carried out in terms of the accuracy and precision of the produced estimates, and the average cost of the said estimates for the case where the conductivity distribution of the body is piecewise constant. Such an assumption arises, for instance, in medical imaging since various tissues in the body have contrasting conductivities with discontinuities at the boundaries of organs or masses (Mendoza and Keeling 2016). The algorithms are chosen in a way that a comparison between a standard heuristic algorithm (FA), improvements of some popular heuristic algorithms (NBA, GA-MPC), and combinations of two or more algorithms and the more recent ones (LSHADE-SPACMA, LSHADE-cnEpSin, EBOwithCMAR) is done.

This paper is organized as follows. Section 2 includes a brief overview of the forward and inverse problems. In Section 3, the proposed heuristic algorithms for the solution of the inverse problem are explained. The methodology used to conduct numerical simulations and their results are then discussed in Section 4. Finally, Section 5 draws conclusions about the results obtained and includes possible future works.

MATHEMATICAL FORMULATION OF THE EIT PROBLEM

Forward Problem

Consider a bounded simply connected domain $\Omega \subset \mathbb{R}^2$ with a smooth boundary $\partial\Omega$. In the low-frequency regime under consideration in EIT experiments, the electromagnetic field satisfies the quasi-static Maxwell equations where the time derivative is neglected (Cheney *et al.* 1999; Borcea 2011). The governing (elliptic) equation of EIT is given in Ω by:

$$\nabla \bullet (\sigma \nabla u) = 0 \tag{1}$$

where $\sigma : \Omega \to \mathbb{R}$ is the conductivity distribution and $u : \Omega \to \mathbb{R}$ is the potential function in the body (Calderón 1980). Moreover, assume that σ is piecewise constant, that is, $\sigma(x) = \sum_{i=o}^{n} \sigma_i \chi_i(\boldsymbol{x})$, $\boldsymbol{x} \in \Omega$, where σ_0 is the background conductivity, $\chi_0(\boldsymbol{x})$ is the characteristic function of the background domain $\Omega_0 = \Omega \setminus \cup_{i=1}^{n} \Omega_i$, n corresponds to the number of (possible) inclusions $\Omega_i, (i = 1, \ldots, n)$ in Ω, $\chi_i(\boldsymbol{x}) = \begin{cases} 1, & \boldsymbol{x} \in \Omega_i, \\ 0, & \boldsymbol{x} \notin \Omega_i, \end{cases}$ and σ_i is the conductivity of the i^{th} inclusion Ω_i with $\sigma_{min} \leq \sigma_i \leq \sigma_{max}, \boldsymbol{x} \in \Omega, \forall\, i = 1, \ldots, n$ for some constants $0 < \sigma_{min}, \sigma_{max} < +\infty$. This is the case, for example, in geophysics and medical imaging where the body under investigation could be divided into different regions.

There are three known mathematical models for the EIT forward problem – namely. the continuum model, the point electrode model, and the complete electrode model (CEM) (Borcea 2011). This work considers the CEM because it is accordingly the most accurate mathematical forward model for real-life EIT (Somersalo *et al.* 1992). In CEM, a finite number of electrodes denoted by $\{e_\ell\}_{\ell=1}^{L} \subset \partial\Omega$ is attached to the boundary on which the current patterns $I = (I_1, \ldots, I_L)^T \in \mathbb{R}^L$ are injected. Resulting boundary potentials $U = (U_1, \ldots, U_L)^T \in \mathbb{R}^L$ are measured. The electrode contact impedance, denoted by $\mathrm{Z} = (z_1, \ldots, z_L)^T \in \mathbb{R}^L$, is the effect of a thin and highly resistive layer formed at the electrode-object interface during electrode measurements (Somersalo *et al.* 1992) and it is assumed to satisfy $z_\ell > Z_{min}, \ \forall \ell = 1, 2, \ldots, L$, where Z_{min} is a positive constant. Using the CEM to model the electrodes, they are assumed to be of finite length and are perfect conductors, which results in voltage measurements on these electrodes to be constant. This contact impedance tends to be high for the frequencies used in EIT; thus, the voltage drop across the impedance layer is large. Ignoring this voltage drop introduces a large modeling error, which results in reconstruction errors. Accounting the voltage drop caused by the contact impedance, we have a Robin-type boundary condition.

Furthermore, assuming that the current flowing on each electrode is equal to the current injected and that there is no current flow on the parts of the boundary where there is no electrode, we have:

$$u + z_\ell \sigma \frac{\partial u}{\partial \vec{n}} = U_\ell, \qquad on\ e_\ell\ , \ell = 1, 2, \ldots, L. \tag{2}$$

$$\int_{e_\ell} \sigma \frac{\partial u}{\partial \vec{n}}\, ds = I_\ell, \qquad \ell = 1, 2, \ldots, L, \tag{3}$$

$$\sigma \frac{\partial u}{\partial \vec{n}} = 0, \qquad on\ \ \partial\Omega \setminus \bigcup_{\ell=1}^{L} e_\ell. \tag{4}$$

Equations 1−4 constitute the CEM for EIT. Let us introduce the space $\mathbb{R}^L_\diamond := \{W \in \mathbb{R}^L \,|\, \sum_{\ell=1}^{L} W_\ell = 0\}$. The CEM forward problem is formulated as follows: find potentials $(u, U) \in H := H^1(\Omega) \oplus \mathbb{R}^L_\diamond$ upon injecting current pattern $I \in \mathbb{R}^L$ on the boundary of a body Ω with known conductivity distribution σ in Ω and electrode contact impedance $Z \in \mathbb{R}^L$. The variational formulation of the CEM is given by (Somersalo *et al.* 1992)

$$\int_{\Omega} \sigma \nabla u \cdot \nabla v\, d\boldsymbol{x} + \sum_{\ell=1}^{L} \int_{e_\ell} \frac{1}{z_\ell} (u - U_\ell)(v - V_\ell)\, ds = \sum_{\ell=1}^{L} I_\ell V_\ell, \tag{5}$$

for any test functions $(v, V) \in H$. The existence and uniqueness of a solution $(u, U) \in H$ have been proved using the Lax-Milgram Milgram theorem (Somersalo *et al.* 1992). The conservation of charge $\sum_{\ell=1}^{L} I_\ell = 0$ assures the existence of solutions while the arbitrary choice of ground $\sum_{\ell=1}^{L} U_\ell = 0$ ensures the uniqueness of the solution.

We propose a discretization of the forward problem by means of two-dimensional $\mathbb{P}_1$ Lagrange FE (Rao 2005). We use a triangulation of the domain Ω, which consists of N nodes. We consider the approach of Kaipio *et al.* (2000) and we solve the following linear system:

$$K := \begin{bmatrix} M + Z & C \\ C^T & G \end{bmatrix} \begin{bmatrix} u^h \\ \beta \end{bmatrix} = \begin{bmatrix} \mathbf{0}_{\mathbb{R}^N} \\ \mathcal{P}^T I \end{bmatrix}, \tag{6}$$

where the matrix $K \in \mathbb{R}^{(N+L-1)\times(N+L-1)}$ is sparse, symmetric, and positive-definite (Kaipio *et al.* 2000; Crabb 2017). The matrices M, Z, C, and G are defined, respectively, by:

$$M_{ij} = \int_\Omega \sigma \nabla \varphi_i \cdot \nabla \varphi_j \, dA \,, 1 \le i, j \le N, Z_{ij} = \sum_{\ell=1}^{L} \int_{e_\ell} \frac{1}{z_\ell} \varphi_i \varphi_j \, dS \,, 1 \le i, j \le N,$$

$$C_{i\ell} = -\frac{1}{z_1} \int_{e_1} \varphi_1 \, dS + \frac{1}{z_{\ell+1}} \int_{e_{\ell+1}} \varphi_i \, dS \,, 1 \le i \le N, 1 \le \ell \le L-1$$

and for $1 \le \ell, k \le L-1$:

$$G_{\ell k} = \begin{cases} \dfrac{|e_1|}{z_1} + \dfrac{|e_{k+1}|}{z_{k+1}}, & \text{if } \ell = k, \text{ and} \\ \dfrac{|e_1|}{z_1}, & \text{otherwise} \end{cases}.$$

Moreover, $u^h \in \mathbb{R}^N$ is the vector of nodal values of the unknown potential and I is the vector of currents through the electrodes. The columns of the matrix $\mathcal{P} \in \mathbb{R}^{L\times L-1}$ are the set of vectors $\phi_\ell \in \mathbb{R}^L$ defined as $\phi_1 = (1, -1, 0, \ldots, 0)^T$,$\phi_2 = (1, 0, -1, \ldots, 0)^T$, . . . , $\phi_{L-1} = (1, 0, \ldots, -1)^T$. The above system uses a dummy variable $\beta \in \mathbb{R}^{L-1}$ to force the uniqueness condition. To transform back to an approximation U^h of the boundary potential U, we have the relation:

$$U^h = \mathcal{P}\beta = \sum_{k=1}^{L-1} \beta_k \phi_k. \tag{7}$$

Inverse Problem

The inverse problem (or the reconstruction problem) is the main part of the EIT problem where the conductivity distribution of a body Ω is recovered using voltage measurements at the boundary $\partial\Omega$. However, whereas the forward problem is well-posed, the inverse problem of EIT is nonlinear and highly ill-posed. In the mathematical literature, this is also known as Calderón's problem (Calderón 1980). A strictly positive conductivity in the elliptic Equation 1 is uniquely determined in a bounded domain by the entire corresponding Dirichlet-to-Neumann (DtN) map on the whole boundary of the domain. The main uniqueness, stability, and reconstruction results have been formulated using the so-called continuum model (Astala and Päivärinta 2006). However, in several applications in EIT, one can only measure currents and voltages on part of the boundary. Real-life data consist, essentially, of a finite-dimensional linear electrode current-to-electrode voltage operator. Results have also been obtained on the problem of whether one can determine the conductivity in the interior from only partial information on the DtN map (Uhlmann 2009; Hyvönen *et al.* 2012; Borcea 2011). Furthermore, since voltage measurements are known to be noisy in nature, the solution can be dominated by noise unless additional conditions are imposed. As such, EIT is a particularly difficult example of attempting to recover a signal from noise (Holder 2000).

The challenging issues are thus to provide numerical methods for reconstructing the conductivity of a medium from a finite number of boundary measurements. There are two primary types of algorithms in EIT*:* static imaging and

difference imaging. Static imaging attempts to recover the absolute conductivity distribution of a body, whereas difference imaging aims to recover an image of the change in conductivity distribution between the acquisition times of two data. In this work, we will focus on static imaging, which is suitable for the case when the electrical properties of the body under study do not vary significantly during the time necessary for data collection (Herrera *et al.* 2007). The inverse problem for EIT reconstructs the conductivity distribution σ inside the body Ω from voltage measurements $\{U_\ell\}_{\ell=1}^{L}$ on the electrodes $\{e_\ell\}_{\ell=1}^{L}$. The aim is to retrieve a finite number n of inclusions of different conductivities in Ω. More precisely, the goal is to estimate by means of an iterative procedure a vector $P \in \mathbb{R}^m$ of unknown parameters and the vector $S := (\sigma_i)_{i=1}^{n}$ of conductivities, for which the error between the measured voltages and that predicted by the CEM forward problem is minimum. The vector P contains geometric attributes (*e.g.* center, radius, side length) of the inclusions $\Omega_i, i = 1, \ldots, n,$ (of respective conductivities σ_i). The objective function reads:

$$C(P,S) = \|U(P,S) - U_{obs}\|_2^2, \tag{8}$$

where the voltages $U(P,S) \in \mathbb{R}^L$ are computed by solving the forward problem (Equations 1−4) at a fixed conductivity σ (described by the vectors P and S) and $U_{obs} \in \mathbb{R}^L$ is the measured voltage at the electrodes.$\|\cdot\|_2$ is the Euclidean norm.

PROPOSED HEURISTIC ALGORITHMS FOR THE INVERSE PROBLEM

The development of heuristic algorithms has experienced significant growth over the past two decades (Hussain *et al.* 2019). New algorithms, including improved variants of known methods, are continuously being proposed and applied to various real-world problems. This is in part due to efforts directed at encouraging the creation of more advanced methods, including those of the IEEE Congress on Evolutionary Computation, and Black-Box Optimization Competition (Molina *et al.* 2018). Inspirations behind the methods are wide-ranging – from evolution and the behavior of animals to physical processes. As such, the selection process of algorithms included in this study is an attempt to balance the diverse inspirations involved in developing the methods and the recency of such methods.

As pointed out, estimation of the conductivity distribution based on boundary voltages and electric currents is an ill-conditioned inverse problem. Minimizing the voltage error may then produce unsatisfactory results. Hence, reconstruction requires some methods of improving the conditioning so that the wild variations causing the instability are ruled out. The most common method is regularization, which involves applying further assumptions and constraints based on *a priori* information. Typically, this means that the inverse problem is augmented with a side constraint such as the minimum length solution, the minimum error with respect to *a priori* solution, or the smoothness of the solution (Holder 2000).

The following heuristic approaches allow restrictions to the solution space and introduction of prior information without using the classical regularization techniques described above. Moreover, no evaluation of objective function derivatives is needed and no assumption on function continuity needs to be made. However, heuristic algorithms are relatively expensive in terms of computing time and this limits their applicability to the field of difference imaging at present. Nevertheless, the continuous and rapid advancement of computing technology makes the development of real-time dynamic imaging applications based on heuristic methods conceivable in the near future. These heuristic algorithms belong to a class of algorithms called metaheuristics (Siarry 2016).

We will use the following heuristic algorithms: FA, NBA, GA-MPC, LSHADE-SPACMA, LSHADE-cnEpSin, and EBOwithCMAR. The detailed discussion of each algorithm and their pseudo-codes are presented in the appendices.

RESULTS AND DISCUSSION

Series of numerical tests were carried out to investigate and compare the performance of the proposed algorithms in solving the inverse EIT problem.

Presentation of the Test Configurations

Three different cases were considered (see Table 1). In Case 1, we study a CT (computer tomography) scan of a thorax domain obtained from the study of Venkatratnam and Nagi (2017), as shown in Figure 1a. Solving the forward problem using the FE method requires the parametrization of the boundaries of the lungs, the heart, and the whole body. These parametric curves are approximated using the Fourier series. The coefficients of the Fourier series are estimated by finding the parametric curve that fits the data points on the boundary curve. With this, any practical domain or object may be studied for real-life applications of EIT. Also, Nissinen *et al.* (2011) discussed that the errors due to the approximations of domain boundary affect the reconstruction of the conductivity.

Table 1. Different cases studied for the numerical test.

Case	Unknown variables
1 – The body Ω is a region representing the thorax with the lungs and the heart.	σ_l, σ_h (geometry is fixed)
2 – The body Ω is a unit circle region with one elliptical inclusion**.**	$\sigma_e, h, k, a, b, \theta$
3 – The body Ω is a unit circle region with two rotated elliptical inclusions.	$\sigma_1, h_1, k_1, a_1, b_1, \theta_1, \sigma_2, h_2, k_2, a_2, b_2, \theta_2$

Now, for the numerical simulations of Case 1, only the conductivity values inside the inclusions, σ_l and σ_h, are unknown. Indeed, the CT results already give the location and the geometry of the organs. We are only interested in determining the conductivity values of the inclusions (*i.e.* lungs and heart). This method is particularly applicable for lung or heart function monitoring to check if there is a deviation of the estimated conductivity values from the normal values. For Case 2, six parameters are unknown, *i.e.* σ_e is the conductivity of the inclusion; (h, k) is the center of the ellipse; a and b are the lengths of the major and minor axes, respectively; and θ is the angle of rotation. In Case 3, we aim in reconstructing two disjoint elliptical inclusions and the respective conductivity inside, *i.e.* 12 unknown parameters. The conductivity σ_0 of the background medium is assumed known in all configurations. In the first case, it is equal to $6.7\ mS.cm^{-1}$ while in both the second and third case, the conductivity is $1\ mS.cm^{-1}$. Both cases may represent the domains used in the application of EIT for brain or breast tumor detection.

We work with synthetic data. We take $L = 32$ electrodes. The contact impedance is constant across the L electrodes and it is equal to 0.03. Sixteen (16) current patterns are applied on the electrodes and the first current has the form $I^1 = \{I_\ell^1\}_{\ell=0}^{L-1} = \sin\left(\frac{2\pi\ell}{L}\right)$. The remaining 15 current patterns are obtained by "rotating" the values of the first current pattern, *i.e.* to get the second current pattern I^2, we have $I^2(0) = I^1(L-1), I^2(1:L/2) = I^1(0:L/2-1)$, and $I^2(L/2+1:L-1) = I^1(L/2:L-2)$. This is repeated until we obtain the fifteen additional current patterns. The synthetic voltage data (u, U) are obtained by solving the forward problem (Equations 1–4) with the exact conductivity distribution (see Section 2.1). A FEM mesh structure with 30240 triangular elements, 15409 nodes, and a mesh size $h = 0.014$ was used for the resolution of the forward problem in Case 1; and with 25858 triangular elements, 13122 nodes, and $h = 0.03$ for both Cases 2 and 3. In order to avoid an inverse crime [in the sense of Colton and Kress (1998)], the inverse computations are done on a mesh with 17240 triangular elements, 8845 nodes, and $h = 0.019$ for Case 1; and on a mesh with 17882 triangular elements, 9102 nodes, and $h = 0.037$ mesh size for Cases 2 and 3, which are different from the meshes used to solve the forward problems. To model possible experimental errors, a 1% random (additive) noise is added to the voltage data as $U_{data} = \left(1 + 0.01 * rand(L)\right) * U$, where $rand(L)$ gives a vector of length L where each element is uniformly distributed random number in the interval $[-1,1]$ [see Hintermüller and Laurain (2008)]. We note that to consider higher levels of noise, the use of regularization methods or more assumptions are needed in the formulation of the inverse problem, especially for Cases 2 and 3. In our simulations, one noise seed is composed of 16 different noise vectors added to the corresponding 15 current-voltage measurements.

The study is based on 20 independent runs of each proposed algorithm, with the same noise seed for all the runs and a stopping criterion based on a pre-defined number of function evaluations. The only stopping criterion used for all heuristic methods is when the maximum number of function evaluations is reached. In particular, for Case 1, we set

the maximum number of function evaluations to 1000; in Cases 2 and 3, we have $1000 * D$ function evaluations with D as the number of unknown parameters. The search space was restricted differently for each case. In Case 1, the bounds are given by $x_{min} = [0,0]$ and $x_{max} = [10,10]$. Case 2 has bounds $x_{min} = [5,-1,-1,0,0,0]$ and $x_{max} = [9,1,1,2,2,\pi]$. For Case 3, we fix $x_{min} = [5,-1,-1,0,0,0,5,-1,-1,0,0,0]$ and $x_{max} = [9,1,1,4,4,\pi,9,1,1,4,4,\pi]$. We implemented the numerical solver for the forward problem with FreeFEM++ (Hecht 2012) and the numerical optimization algorithms with Matlab. All experiments were executed in Matlab R2018a. The parameter settings used for each algorithm can be found in Appendix VII.

Numerical Results

The comparative analysis of the heuristic methods is done by measuring the accuracy and precision of the solutions generated, together with their respective average costs. This is because the inverse EIT problem is ill-conditioned, which means that solutions tend to be extremely sensitive to perturbations, potentially making them inaccurate or unstable. To quantitatively analyze the accuracy of recovered images, the average of the reconstruction errors for 20 runs of each algorithm was calculated. For the ith run, the relative reconstruction error is given by $RE = \frac{\|\hat{y}_i - y_{true}\|_2}{\|y_{true}\|_2}$, where $\hat{y}_i$ is a vector containing the values estimated by the algorithm at the i^{th} run, whereas y_{true} contains true values. Based on Table 1, $\hat{y}_i$ and y_{true} are vectors in $\mathbb{R}^2$ for Case 1, in $\mathbb{R}^6$ for Case 2, and in $\mathbb{R}^{12}$ for Case 3.

In order to measure the repeatability (reconstruction accuracy) of each algorithm, *i.e.* the degree to which the algorithm produced similar results for 20 runs, the standard deviation (std. dev.) of the reconstruction errors was also determined. Table 2 compares the accuracy, repeatability, and average costs of each algorithm's reconstructions for the three cases considered. From now on, the "average reconstruction error" will be referred to as the "mean error" for simplicity. The mean and standard deviation of the reconstruction errors of the generated solutions in the 20 independent runs is computed using the *mean* and *std* functions in Matlab.

In all the cases, the final solution or final reconstructed parameter values considered for each algorithm is the average solution of the 20 runs. Figures 1, 2, and 3 show the final solution of each algorithm for Cases 1, 2, and 3, respectively. The difference in color and the shape of the inclusion between the original image the images generated by the proposed heuristic algorithms is the difference between the true solution and the approximate from the true solution is. Table 3 shows the relative reconstruction error in % of the final solution of each algorithm for the three cases, while Tables 4 and 5 display the solution. Note that we fixed the range of the conductivities in the plots and so the difference in the color and the shape of the inclusion between the original image generated by the proposed heuristic algorithms is the difference between the true solution and the approximate solution.

Table 2. Comparison of accuracy, repeatability, and average cost of the solutions generated by the proposed heuristic algorithms for all cases.

Algorithm	Case 1 – thorax			Case 2 – one elliptical inclusion			Case 3 – two elliptical inclusions		
	Mean error	Std. dev.	Ave. cost	Mean error	Std. dev.	Ave. cost	Mean error	Std. dev.	Ave. cost
FA	5.54E-02	7.77E-02	1.43E-03	0.2193	0.1046	0.6115	0.3071	0.0717	2.8463
NBA	7.74E-03	4.64E-03	1.43E-03	0.1742	0.0983	0.2562	0.3435	0.0926	1.9620
GA-MPC	2.23E-03	1.94E-03	1.43E-03	0.1527	0.0554	0.2422	0.1790	**0.0530**	0.2558
LSHADE-SPACMA	**6.08E-04**	**7.26E-08**	1.43E-03	0.1143	**0.0385**	**0.2407**	**0.1546**	0.0674	**0.2165**
LSHADE-cnEpSin	6.15E-04	4.32E-05	1.43E-03	0.1251	0.0762	0.2423	0.2033	0.0675	0.2219
EBOiwthCMAR	6.10E-04	8.24E-05	1.43E-03	**0.0827**	0.0693	0.2426	0.1988	0.0685	0.2917

Table 3. Reconstruction errors in % of the final solution generated by the proposed heuristic algorithms for all cases.

Algorithm	Case 1	Case 2	Case 3
FA	5.28	6.82	16.56
NBA	0.152	**2.28**	21.91
GA-MPC	**0.054**	15.11	13.07
LSHADE-SPACMA	0.060	5.84	**7.90**
LSHADE-cnEpSin	0.061	9.63	14.95
EBOwithCMAR	0.058	6.43	12.00

Table 4. Reconstruction errors in % of each parameter in the final solution generated by the proposed heuristic algorithms for Cases 1 and 2.

Algorithm	Case 1 – thorax		Case 2 – one elliptical inclusion					
	σ_l	σ_h	σ_e	h	k	a	b	θ
FA	1.19E-02	5.34E-02	5.28	2.68	2.06	17.61	41.40	6.37
NBA	3.69E-03	1.42E-03	**1.94**	**0.68**	0.41	7.34	**2.15**	6.92
GA-MPC	**3.09E-03**	**2.51E-04**	15.50	1.26	0.43	4.76	2.43	4.07
LSHADE-SPACMA	3.18E-03	3.43E-04	5.92	1.46	**0.07**	4.00	8.65	**0.34**
LSHADE-cnEpSin	3.18E-03	3.62E-04	9.86	1.20	0.22	2.03	3.72	3.95
EBOiwthCMAR	3.19E-03	3.06E-04	6.55	1.13	0.14	**0.09**	4.56	4.21

Table 5. Reconstruction errors in % of each parameter in the final solution generated by the proposed heuristic algorithms for Case 3.

Algorithm	Case 3 – two elliptical inclusions											
	σ_1	h_1	k_1	a_1	b_1	θ_1	σ_2	h_2	k_2	a_2	b_2	θ_2
FA	21.41	11.99	26.56	22.77	70.79	3.81	**1.23**	86.77	112.4	3.50	23.06	1.63
NBA	24.73	6.25	19.51	28.82	114.9	5.93	3.66	81.20	88.59	27.28	45.07	2.54
GA-MPC	19.18	2.18	1.37	13.65	13.32	0.45	7.90	2.59	4.56	8.40	**0.51**	**0.05**
LSHADE-SPACMA	**12.59**	2.05	**0.49**	**5.32**	**6.05**	**0.25**	2.52	**0.57**	1.19	2.11	1.96	0.77
LSHADE-cnEpSin	23.51	**0.62**	0.76	9.28	8.18	0.27	5.82	1.66	**0.44**	**1.34**	1.52	8.08
EBOiwthCMAR	18.14	7.45	4.32	10.57	16.59	0.37	1.27	12.57	11.70	8.86	16.47	5.82

The results shown in black bold characters indicate the best values. For Case 1, all the heuristic algorithms studied were very successful in the recovery of the conductivity values of the inclusions in Ω. For Case 2, all the heuristic algorithms studied well retrieved the center of the ellipse (see Table 4). The difference in performance between them lies in the estimation of both the conductivity and the geometric parameters. EBOwithCMAR performed the best while getting the least mean error and relatively small average cost. LSHADE-SPACMA got the smallest standard deviation and average cost. As expected, because of the low-resolution property of EIT, the estimate for the conductivity value inside the inclusion is not as accurate as of the approximation of the geometry. Nevertheless, NBA provided an excellent approximation of the conductivity (error 1.94%) while FA, LSHADE-SPACMA, and EBOwithCMAR were still able to obtain good conductivity value estimates (see Table 4). GA-MPC yielded the least accurate conductivity estimate which justifies the relative error given in Table 3. This means that GA-MPC finds it hard to balance its exploration and exploitation when approximating both the geometry of the inclusion and the conductivity inside it. NBA and the three most recent algorithms – namely, LSHADE-SPACMA, EBOwithCMAR, and LSHADE-cnEpSin – presented impressive reconstructions (see Figure 2); FA is less efficient. Lastly, NBA offered the best relative error of the final solution.

For Case 3, LSHADE-SPACMA showed the best performance with the least mean error and average cost while GA-MPC is the most consistent among the algorithm because it obtained the least standard deviation of the reconstruction error. The recovered images of LSHADE-SPACMA and LSHADE-cnEpSin are the closest ones to the original image. GA-MPC and EBOwithCMAR were also able to reconstruct the two inclusions. NBA only obtained a not too bad

location of the bigger inclusion but failed to reconstruct the smaller ellipse. FA gave a poor reconstruction of both inclusions. For the estimation of conductivity values inside the inclusions, LSHADE-SPACMA achieved a good balance in the estimation of the conductivities, EBOwithCMAR yielded the most accurate estimate for the conductivity in the smaller inclusion, and LSHADE-SPACMA got the best conductivity estimate for the bigger inclusion. NBA is the least accurate in the estimation of the conductivity inside the bigger inclusion, while GA-MPC has the least accurate conductivity value estimate for the smaller ellipse.

Moreover, GA-MPC gave more accurate approximations for both the geometry of the inclusion and its conductivity than the standard GA used in (Kim *et al.* 2006; Ribeiro *et al.* 2014). Improvements of DE – namely, LSHADE-SPACMA and LSHADE-cnEpSin – also performed better than the DE used in the study of Rashid *et al.* (2010). Lastly, to compare our results to that of the deterministic method, we applied a quasi-Newton iterative method [Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm] to Cases 1 and 2 with a random initial guess. We use the Matlab built-in command function *fminunc* to implement the quasi-Newton iterative algorithm. In Case 1, the reconstruction error is $6.17E-04$, the standard deviation is $2.21E-04$, and the average cost function is $1.4E-03$. This is almost the same result that we got from the six heuristic algorithms. But for Case 2, BFGS yielded a mean error of 151.6, standard deviation 361.8, and average cost function 5.97, which are all huge comparing to the results of the six heuristic algorithms.

Metaheuristic algorithms can be computationally costly since they require many cost function evaluations. Because we made the stopping criterion based on the maximum number of function evaluations alone, the run time of all the algorithms are approximately equal. For this reason, we only provide the time for each domain case. For Case 1, it takes approximately $3s$ for one function evaluation to be done and the number of function evaluations is set to 1000. Meanwhile, for Case 2, we have $3.7s$ for one function evaluation and the number of function evaluations is 6000. Lastly for Case 3, we get $4.3s$ for one function evaluation with 12000 total number of function evaluations. To expect driving 3D computational simulations is challenging. Metaheuristic algorithms are time-consuming and improvements are needed to get 3D results.

CONCLUSION

In this paper, we have presented a conclusive study of the applicability of several heuristic approaches for EIT image reconstruction. Up to our knowledge, this is the first time that such a comparative study (between FA, NBA, GA-MPC, LSHADE-SPACMA, LSHADE-cnEpSin, and EBOwithCMAR) is addressed for EIT. Numerical simulations, given a fixed number of cost function evaluations and default heuristic algorithm parameters, showed that the more recent algorithms – namely LSHADE-SPACMA, LSHADE-cnEpSin, and EBOwithCMAR – obtained the best results in terms of accuracy, repeatability, and average cost. This indicates the continuous improvement in metaheuristic techniques, reinforcing their potential to solve other similar problems. FA did not fare as well the other algorithms, especially in retrieving two disjoint inclusions (Case 3), because the maximum number of evaluations and the population size for each iteration might not be enough to have a balance between exploitation and exploration. This might be also the case for NBA and GA-MPC.

Although NBA was successful in the image reconstruction of one defect (Cases 1 and 2), it failed in the configuration with two defects. Modifications and improvement in these three algorithms can be further studied to obtain more competitive results. Since FA, NBA, and GA-MPC populations for each iteration is fixed, we can recommend adjusting the population size so that there will be a balance between exploitation and exploration. Also, the maximum number of evaluations can be increased. Because of their population size linear reduction and self-adapting parameters system, the most recent algorithms LSHADE-SPACMA, LSHADE-cnEpSin, and EBOwithCMAR are more consistent and accurate. The different numerical results attest to their efficiency.

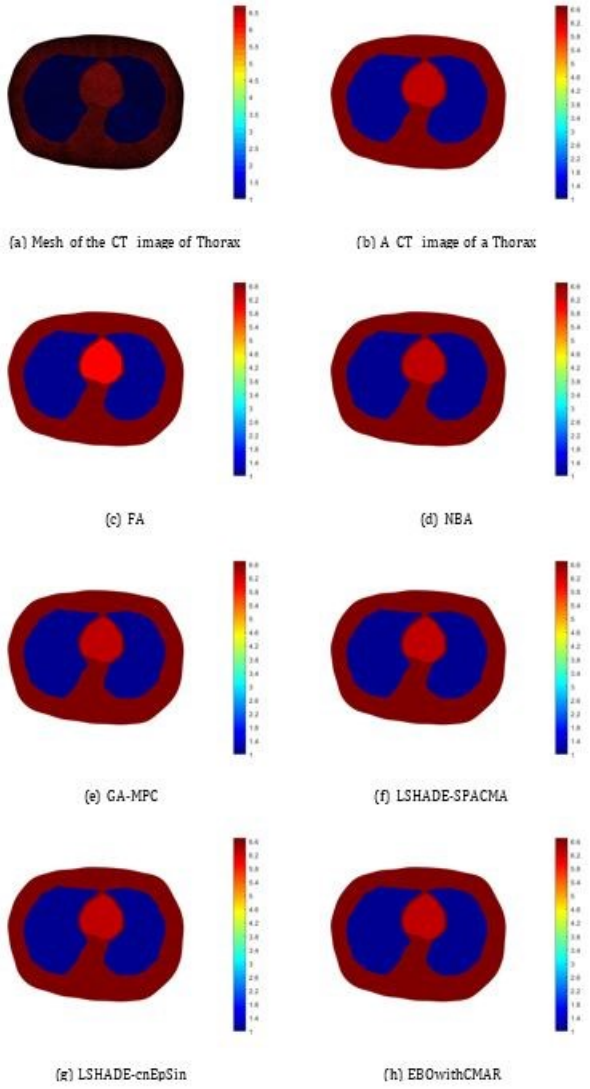

Figure 1. Estimation of conductivity values inside the fixed geometries of heart and lungs of all proposed heuristic algorithms for Case 1. The conductivity of the lungs and the heart are, respectively, $\boldsymbol{\sigma_l = 1\ mS.cm^{-1}}$ and $\boldsymbol{\sigma_h = 6.3\ mS.cm^{-1}}$ in the original image.

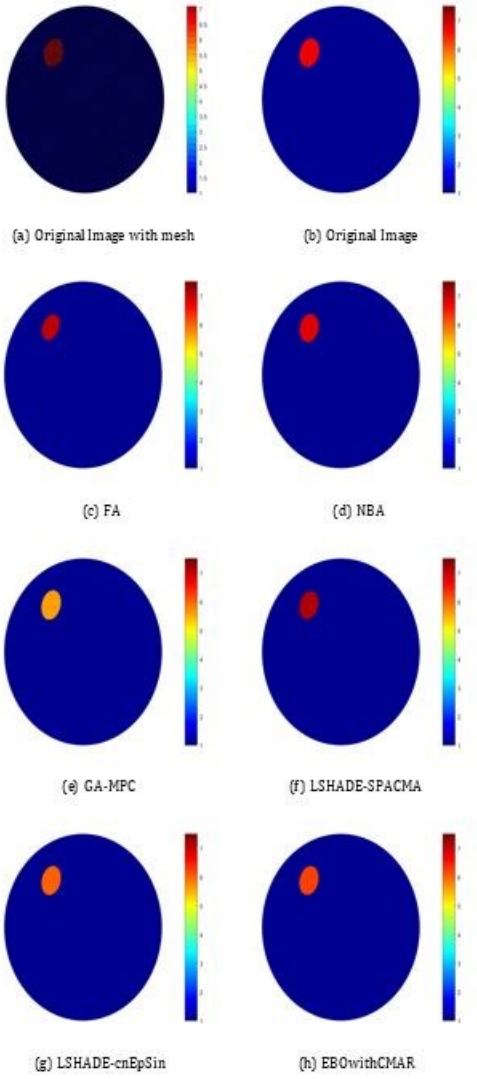

Figure 2. Image reconstructions for Case 2. The conductivity inside the ellipse of the original image is $\sigma_e = 6.7$ mS·cm^{-1}.

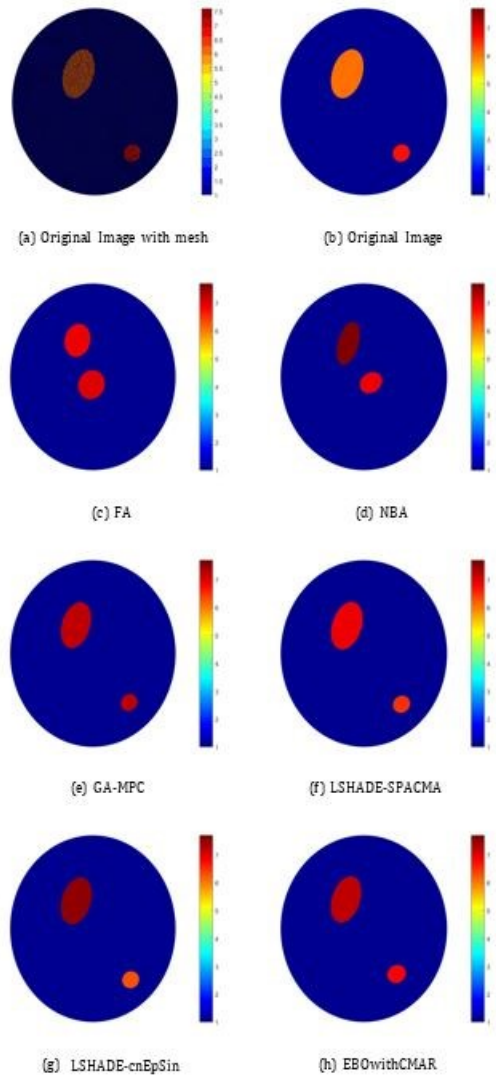

Figure 3. Image reconstructions for Case 3. The conductivities are $\sigma_1 = 6.1$ mS·cm^{-1} (big ellipse) and $\sigma_2 = 6.7$ mS·cm^{-1} (small ellipse) in the original image.

In the thorax domain, where only the conductivities of the inclusions are unknown, the six heuristic algorithms provided excellent results. In other cases, the recovery of conductivity needs more improvement. Possible future works include using a regularization term in the cost functional to deal with the ill-posedness of the problem or applying parameter tuning techniques to better fit the algorithm to the EIT inverse problem. A sensitivity analysis for the CEM forward problem could also be used to get *a priori* information on the dependence of the measured boundary voltages on the conductivity and the characteristics of inclusions. This is part of ongoing work.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support of the Natural Sciences Research Institute, University of the Philippines Diliman under the research grant MAT-16-1-01.

NOTES ON APPENDICES

The complete appendices section of the study is accessible at http://philjournsci.dost.gov.ph

REFERENCES

ASTALA K, PÄIVÄRINTA L. 2006. Calderon's inverse conductivity problem in the plane. Annals of Mathematics 163: 265–299.

AWAD N, ALI M, LIANG J, QU B, SUGANTHAN P. 2016a. Problem definitions and evaluation criteria for the CEC 2017 special session and competition on single objective bound constrained real-parameter numerical optimization [Technical Report]. Nanyang Technological University, Singapore.

AWAD N, ALI M, SUGANTHAN P, REYNOLDS R. 2016b. An ensemble sinusoidal parameter adaptation incorporated with L-SHADE for solving CEC 2014 benchmark problems. In: Proceeding of the 2016 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 2958–2965.

AWAD N, ALI M, SUGANTHAN P. 2017. Ensemble sinusoidal differential covariance matrix adaptation with Euclidean neighborhood for solving CEC 2017 benchmark problems. In: Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 372–379.

BARBOSA V *et al*. 2018. Image reconstruction of electrical impedance tomography using fish school search and differential evolution. Critical Developments and Applications of Swarm Intelligence. p. 301–338.

BARBOSA V, RIBEIRO R, SILVA A, ROCHA V, FREITAS R, FEITOSA AR, DE SOUZA RE, DOS SANTOS W. 2017. Reconstruction of electrical impedance tomography images using chaotic ring-topology particle swarm optimization and non-blind search. International Journal of Swarm Intelligence Research 8(2): 17–33.

BORCEA L. 2011. Electrical impedance tomography. Inverse Problems 18: R99–R136.

CAJAYON RC, LUCILO JA, PILAR-ARCEO CPC, MENDOZA ER. 2020. Comparison of two nature-inspired algorithms for parameter estimation of S-system models. Philippine Journal of Science 149(1): 63–78.

CALDERÓN A. 1980. On an inverse boundary value problem. In: Seminar on Numerical Analysis and its Applications to Continuum Physics; Río de Janeiro, Brazil. p. 65–73.

CHENEY M, ISAACSON D, NEWELL J. 1999. Electrical impedance tomography. SIAM Review 41: 85–101.

COLTON D, KRESS R. 1998. Inverse acoustic and electromagnetic scattering theory. Berlin: Springer-Verlag.

CRABB M. 2017. Convergence study of 2D forward problem of electrical impedance tomography with high order finite elements. Inverse Problems in Science and Engineering. p. 1–26.

DAS S, MULLICK SS, SUGANTHAN P. 2016. Recent advances in differential evolution – an updated survey. Swarm and Evolutionary Computation 27: 1–30.

DAS S, SUGANTHAN P. 2010. Problem definitions and evaluation criteria for CEC 2011 competition on testing evolutionary algorithms on real world optimization problems [Technical Report]. Jadavpur University, India.

DOS SANTOS WP *et al*. 2018. Hybrid metaheuristics applied to image reconstruction for an electrical impedance tomography prototype. In: Proceedings of the Hybrid Metaheuristics for Image Analysis; Springer. p. 209–251.

EGGLESTON M, SCHWABE R, ISAACSON D, COFFIN L. 1990. The application of electric current computed tomography to defect imaging in metals. Review of Progress in Quantitative Nondestructive Evaluation 24: 455–462.

ELSAYED S, SARKER R, ESSAM D. 2011. GA with a new multi-parent crossover for solving IEEE-CEC 2011 competition problems. In: Proceedings of the 2011 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 1034–1040.

FEITOSA AR, RIBEIRO R, BARBOSA V, DE SOUZA RE, DOS SANTOS W. 2014a. Reconstruction of electrical impedance tomography images using chaotic ring-topology particle swarm optimization and non-blind search. In: Proceedings of the 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC); IEEE. p. 2618–2623.

FEITOSA AR, RIBEIRO R, BARBOSA V, DE SOUZA RE, DOS SANTOS W. 2014b. Reconstruction of electrical impedance tomography images using particle swarm optimization, genetic algorithms and non-blind search. In: Proceedings of Biosignals and Robotics for Better and Safer Living (BRC) at the 5th ISSNIP-IEEE Biosignals and Biorobotics Conference (2014); IEEE. p. 1–6.

FISTER I, FISTER JR. I, YANG XS, BREST J. 2013. A comprehensive review of firefly algorithms. Swarm and Evolutionary Computation 13: 34–46.

HANSEN N. 2006. The CMA evolution strategy: a comparing review. In: Towards a new evolutionary computation. Springer. p. 75–102.

HECHT F. 2012. New development in FreeFem++. Journal of Numerical Mathematics 20(3–4): 251–265. Retrieved from https://freefem.org/

HERRERA CN, VALLEJO MF, DE MOURA F, AYA JC, LIMA R. 2007. Electrical impedance tomography algorithm using simulated annealing as a search method. In: Proceedings of COBEM 2007 at the 19th International Congress of Mechanical Engineering, Brazil.

HINTERMÜLLER M, LAURAIN A. 2008. Electrical impedance tomography: from topology to shape. Control and Cybernetics 37(4): 913–933.

HOLDER D. 2000. Electrical Impedance Tomography: Methods, History and Applications. London: IOP Publishing.

HOLLAND J. 1975. Adaptation in Natural and Artificial Systems. Ann Arbor, MI: University of Michigan Press.

HUSSAIN K, SALLEH MM, CHENG S, SHI Y. 2019. Metaheuristic research: a comprehensive survey. Artificial Intelligence Review 52(4): 2191–2233.

HYVÖNEN N, PIIROINEN P, SEISKARI O. 2012. Point measurements for a Neumann-to-Dirichlet map and the Calderón problem in the plane. SIAM Journal on Mathematical Analysis 44: 3526–3536.

ISAACSON D, MUELLER J, SILTANEN S. 2003. Biomedical applications of electrical impedance tomography. Physiological Measurement 24(2).

JORDANA J, GASULA M, PALLAS-ARENY R. 2001. Electrical resistance tomography to detect leaks from buried pipes. Measurement Science and Technology 12(8): 1061–1068.

KAIPIO J, KOLEHMAINEN V, SOMERSALO E, VAUHKONEN M. 2000. Statistical inversion and Monte Carlo sampling methods in electrical impedance tomography. Inverse Problems 16: 1487–1522.

KIM HC, BOO CJ, KANG MJ. 2006. Image reconstruction using genetic algorithm in electrical impedance tomography. In: Neural Information Processing. King I, Wang J, Chan LW, Wang D eds. Lecture Notes in Computer Science, Vol. 4234. Springer.

KUMAR A, MISRA R, SINGH D. 2015. Butterfly optimizer. In: Proceedings of the 2015 IEEE Workshop on Computational Intelligence: Theories, Applications and Future Directions (WCI); IEEE. p. 1–6.

KUMAR A, MISRA R, SINGH D. 2017. Improving the local search capability of effective butterfly optimizer using covariance matrix adapted retreat phase. In: Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 1835–1842.

LI Y, RAO L, HE R, XU G, WU Q, GE M, YAN W. 2003. Image reconstruction of EIT using differential evolution algorithm. In: Proceedings of the 25th Annual International Conference of the IEEE EMBS; IEEE. p. 1011–1014.

MANGNALL Y, BAXTER A, AVILL R, BIRD N, BROWN B, BARBER D, SEAGAR A, JOHNSON A, READ N. 2003. Applied potential tomography: a new noninvasive technique for assessing gastric function. Clinical Physics and Physiological Measurement 8: 119–129.

MARINGER DG. 2005. Portfolio management with heuristic optimization, 1st ed. Springer.

MARTINS T, CAMARGO E, LIMA R, AMATO M, TSUZUKI M. 2012. Image reconstruction using interval simulated annealing in electrical impedance tomography. IEEE Transactions on Bio-medical Engineering 59: 1861–1870.

MENDOZA R, KEELING S. 2016. A two-phase segmentation approach to the impedance tomography problem. Inverse Problems 33: 015001.

MENDOZA R, LOPE JE. 2012. Reconstructing images in electrical impedance tomography using hybrid genetic algorithms. Science Diliman 24(2): 50–66.

MENG XB, GAO X, LIU Y, ZHANG H. 2015. A novel bat algorithm with habitat selection and Doppler Effect in echoes for optimization. Expert Systems with Applications 42: 6350–6364.

METTLER F. 2012. Medical effects and risks of exposure to ionising radiation. Journal of Radiological Protection 32: N9–N13.

MIAO L, MA Y, WANG J. 2014. ROI-based image reconstruction of electrical impedance tomography used to detect regional conductivity variation. IEEE Transactions on Instrumentation and Measurement 63: 2903–2910.

MOHAMED A, HADI A, FATTOUH A, JAMBI K. 2017. LSHADE with semi-parameter adaptation hybrid with CMA-ES for solving CEC 2017 benchmark problems. In: Proceedings of the 2017 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 145–152.

MOLINA D, LATORRE A, HERRERA F. 2018. An insight into bio-inspired and evolutionary algorithms for global optimization: Review, analysis, and lessons learnt over a decade of competitions. Cognitive Computation 10: 517–544.

NISSINEN A, KOLEHMAINEN V, KAIPIO J. 2011. Compensation of modelling errors due to unknown domain boundary in electrical impedance tomography. IEEE Transactions on Medical Imaging 30: 231–242.

PARKER R. 1984. The inverse problem of resistivity sounding. Geophysics 49: 2143–2158.

PRICE K, STORN R, LAMPINEN J. 2005. Differential Evolution: A Practical Approach to Global Optimization. Berlin: Springer.

RAO S. 2005. The Finite Element Method in Engineering, 4th ed. Oxford: Elsevier Butterworth-Heinemann.

RASHID A, KHAMBAMPATI AK, KIM BS, LIU D, KIM S, KIM KY. 2010. A differential evolution based approach to estimate the shape and size of complex shaped anomalies using EIT measurements. In: Proceedings of the Grid and Distributed Computing, Control and Automation, Vol. 121. Berlin: Springer.

RIBEIRO R, FEITOSA A, SOUZA R, DOS SANTOS W. 2014. Reconstruction of electrical impedance tomography images using genetic algorithms and non-blind search. In: Proceedings of the 11th International Symposium on Biomedical Imaging. p. 153–156.

SIARRY P. 2016. Metaheuristics. Springer International Publishing.

SOMERSALO E, CHENEY M, ISAACSON D. 1992. Existence and uniqueness for electrode models of electric current computed tomography. SIAM Journal of Applied Mathematics 52: 1023–1040.

TANABE R, FUKUNAGA A. 2013. Success-history based parameter adaptation for differential evolution. In: Proceedings of the 2013 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 71–78.

TANABE R, FUKUNAGA A. 2014. Improving the search performance of SHADE using linear population size reduction. In: Proceedings of the 2014 IEEE Congress on Evolutionary Computation (CEC); IEEE. p. 1658–1665.

TAVARES RS, MARTINS TC, TSUZUKI MSG. 2012. Electrical impedance tomography reconstruction through simulated annealing using a new outside-in heuristic and GPU parallelization. In: Journal of Physics: Conference Series Vol. 407(012015); IOP Publishing. p. 1–15.

TIAN N, LAI CH, PERICLEOUS K, SUN J, XU W. 2012. Contraction-expansion coefficient learning in quantum-behaved particle swarm optimization. In: Proceedings of the 2011 Tenth International Symposium on Distributed Computing and Applications to Business, Engineering and Science (DCABES); IEEE.

UHLMANN G. 2009. Electrical impedance tomography and Calderón's problem. Inverse Problems 25: 123011.

VENKATRATNAM C, NAGI F. 2017. Spatial resolution in electrical impedance tomography: a topical review. Journal of Electrical Bioimpedance 8: 66.

YANG XS. 2008. Nature-Inspired Metaheuristic Algorithms. Luniver Press.

YANG XS. 2010. A new metaheuristic bat-inspired algorithm. In: Cruz C, González JR, Krasnogor N, Pelta DA, Terrazas G eds. Nature Inspired Cooperative Strategies for Optimization (NISCO 2010); Vol. 284 of Studies in Computational Intelligence; Berlin: Springer. p. 65–74.

YORKEY T, WEBSTER J, TOMPKINS W. 1987. Comparing reconstruction algorithms for electrical impedance tomography. IEEE Transactions on Biomedical Engineering 34: 843–852.

ZHANG J, SANDERSON A. 2009. JADE: adaptive differential evolution with optional external archive. IEEE Transactions on evolutionary computation 13(5): 945–958.

ZUO Y, GUO Z. 2003. A review of electrical impedance techniques for breast cancer detection. Medical Engineering & Physics 5: 79–90.

APPENDICES

Appendix I. Firefly algorithm.

The flashing light produced by fireflies in a tropical summer sky are known to have two fundamental functions – namely, to attract mating partners and to attract potential prey. In FA, these flashes can be formulated in such a way that they are linked to the objective function to be optimized. It is a metaheuristic algorithm designed by Xin-She Yang in 2007 (Yang 2008). One can find many applications of this algorithm in the literature [see, for example, Fister *et al.* (2013) and Cajayon *et al.* (2020)].

An initial population of virtual fireflies is randomly generated. In each time step, the light intensity of each firefly is compared pairwisely. In the standard FA, light intensity is determined by the objective function. If a firefly j has a greater light intensity than firefly i, the latter will fly towards the former. Note, however, that the movement of firefly i is determined by three terms: its current position, attraction to the brighter firefly j, and a random walk. While the light intensity is referred to as an absolute measure of emitted light by the firefly, the attractiveness is a relative measure of the light that should be seen in the eyes of the beholders and judged by other fireflies (Fister *et al.* 2013). Attractiveness is affected by the distance r between firefly i and firefly j, attractiveness at $r = 0$, and the degree of absorption of light in the air. The light intensities of the fireflies are then updated given the new positions. The fireflies' positions (solutions) are ranked and the current best solution is updated. Detailed discussion of this firefly-inspired algorithm can be found in the study of Fister *et al.* (2013), while the pseudo-code is given below.

Extensive simulations shown by Fister *et al.* (2013) were carried out to compare the performance of FA with particle swarm optimization and GA. Results showed that FA finds the global minima more efficiently and with higher success rate.

Firefly Algorithm Pseudo-code

Input: Objective function $f(\boldsymbol{x}), \boldsymbol{x} = (x_1, \ldots, x_D)$ for D dimensions, number of fireflies N,

$MaxGen, \alpha, \gamma, \beta$, light intensity I_i is determined by $f(\boldsymbol{x}_i)$

Output: cost function $f(\boldsymbol{x}^*)$ at optimal $\boldsymbol{x}^*$

1: Generate initial population of fireflies $\boldsymbol{x}_i, i = 1,2,\ldots,N$.

2: Initial evaluation of all N fireflies.

3: **while** $FuncEvals < MaxEval$ **do**

4: Increment k.

5: **for** $i = 1$ to N **do**

6: **for** $j = 1$ to N **do**

7: **if** $I_i < I_j$ **then**

8: Move firefly i towards j with $\boldsymbol{x}_i = \boldsymbol{x}_i + \beta e^{-\gamma ij}(\boldsymbol{x}_j - \boldsymbol{x}_i) + \alpha\epsilon_i$, where r_{ij} is the Cartesian distance between two fireflies $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ and ϵ_i is a vector of random numbers.

9: **end if**

10: Evaluate new solutions and update light intensity.

11: **end for**

12: **end for**

13: Reduce α.

14: Rank the fireflies and find the current global best solution.

15: **end while**

Appendix II. Novel bat algorithm.

NBA is a metaheuristic method proposed by Xian-Bing Meng *et al.* (Meng *et al.* 2015). NBA is one of the variants of the basic bat algorithm (BA) developed by Xin-She Yang in 2010 (Yang 2010) based on the echolocation behavior of bats. One of the bat species, known as microbats, are famous for using echolocation extensively. These bats emit a very loud sound pulse and listen for the echo that bounces back from the surrounding objects. They use this echolocation behavior to detect prey, avoid obstacles, and locate their roosting crevices in the dark (Yang 2010). The original BA, however, did not take into account the capacity for Doppler shift compensation of these bats. Their ability to locate surrounding objects or targets is attributed not only to their advanced capability of echolocation but also to their self-adaptive compensation for Doppler Effect in echoes. Moreover, the original BA did not consider the fact that bats hunt in a wide range of habitats. For these reasons, we employed NBA instead of the basic BA.

In NBA, all virtual bats, depicted by their positions and velocities, search for food in an $N-$dimensional space. Starting with a randomly generated population of bats, each bat is subjected to a selection of habitat/s where it will forage. This habitat selection is a stochastic decision such that if a uniform random number in $[0,1]$ is smaller than the selection threshold, bat i will forage in a wide range of habitats; otherwise, it would hunt in limited habitats. If a randomly generated number is bigger that bat i's pulse emission rate, a local search is performed by making the bat fly randomly around a certain neighborhood of the current best position (solution). If bat i's new position is closer to the food than the current best, the rate of its pulse emissions is increased while the loudness is decreased. Finally, after looping through all the bats, the bats are ranked according to their proximity to the food, which is represented by their objective function values. If the best solution does not improve after a certain time steps, the loudness and pulse emission rates of the bats are re-initialized. A detailed explanation of NBA can be found in the study of Meng *et al.* (2015), while the pseudo-code is given below.

In the study (Meng *et al.* 2015), the performance of NBA was tested under twenty optimization problems and four real-world engineering designs. Simulations showed that NBA is effective, efficient, stable, and superior over some well-known algorithms such as the original BA, particle swarm optimization, flower pollination algorithm, and even DE.

Novel Bat Algorithm Pseudo-code

Input: Objective function $f(\boldsymbol{x}), \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D)$ for D dimensions, number of bats N,

maximum number of iterations $\widetilde{M}$, $\alpha, \gamma, G, w, \theta, C_{min}, C_{max}, P_{min}, P_{max}, A_{min}$,

$A_{max}, r_{min}, r_{max}, f_{min}, f_{max}$

Output: cost function $f(\boldsymbol{x}^*)$ at optimal $\boldsymbol{x}^*$

1: Generate initial population of bats x_{ij}^k and velocities v_{ij}^k, $i = 1,2,\ldots,N, j = 1,2,\ldots,D$.

2: Initial evaluation of all N bats.

3: Rank the bats with g_j^k as the best global position and its velocity is v_{gj}^k.

4: **while** $FuncEvals < MaxEval$ **do**

5: **if** $rand(0,1) < P$, where $P \in [P_{min}, P_{max}]$ **then**

6: Generate new solutions with

$$\boldsymbol{x}_{ij}^{k+1} = \begin{cases} g_j^k + \theta\big|_{\text{mean}_j^t} - \boldsymbol{x}_{ij}^k\big| \ln\left(\frac{1}{\mu_{ij}}\right), & \text{if rand } j(0,1) < 0.5 \\ g_j^k - \theta\big|_{\text{mean}_j^t} - \boldsymbol{x}_{ij}^k\big| \ln\left(\frac{1}{\mu_{ij}}\right), & \text{otherwise.} \end{cases}$$

7: **else**

8: Generate new solutions with

$$f_{ij} = f_{min} + (f_{max} - f_{min})\, rand(0,1),$$

$$f_{ij} = \left(\frac{c + v_{ij}^k}{c + v_{gj}^k}\right) f_{ij} \left(1 + C_i \frac{g_j^k - \boldsymbol{x}_{ij}^k}{\left|g_j^k - \boldsymbol{x}_{ij}^k\right| + \epsilon}\right),$$

$$v_{ij}^{k+1} = w v_{ij}^k + (g_j^k - \boldsymbol{x}_{ij}^k) f_{ij},$$

$$\boldsymbol{x}_{ij}^{k+1} = \boldsymbol{x}_{ij}^k + v_{ij}^k,$$

where $w \in [0,1]$ is a uniform random vector, ϵ is the smallest constant in the computer, $C \in [C_{min}, C_{max}]$, and $c = 340 m/s$ is the speed in the air.

9: **end if**

10: **if** $rand(0,1) > r_i$ **then**

11: Generate a local solution around the selected best solution using

$$\boldsymbol{x}_{ij}^{k+1} = g_j^k\big(1 + rand\; n(0, \sigma^2)\big),$$

where $\sigma^2 = \left|A_i^k - A_{\text{mean}}^k\right| + \epsilon$, and rand $n(0, \sigma^2)$ is a Gaussian distribution with mean 0 and standard deviation σ^2, and A_{mean}^k is the average loudness of all bats at time step t. Note that ϵ is used to ensure that $\sigma^2 > 0$.

12: **end if**

13: Evaluate new solutions.

14: Update solutions, the loudness, and pulse emission rate using

$$f(\boldsymbol{x}) = f(\boldsymbol{x}_i)\; if\; rand\; (0,1) < A_i\; and\; f(\boldsymbol{x}_i) < f(\boldsymbol{x}),$$

$$A_i^{k+1} = \alpha A_i^k,$$

$$r_i^{k+1} = r_i^0 (1 - e^{-\gamma k}),$$

15: Rank the solutions and find the current best g^k.

16: **if** g^k does not improve in G time step **then**

Re-initialize the loudness A_i and set temporary pulse rates r_i which is a uniform

number between [0.85,0.9].

17: **end if**

18: Increment k.

19: **end while**

Appendix III. Genetic algorithm with multi-parent crossover.

First proposed by John Holland in 1975 (Holland 1975), GA is originally based on the Darwinian principle of evolution. There have been a number of developments in GA theory and it is still a growing area. GA is one of the most popular heuristic algorithms and has been applied to different problems in science and engineering. A typical design for a classical GA would be as follows. Starting with a randomly generated population, GA carries out a process of fitness-based selection and recombination to produce a successor population – the next generation. In the fitness-based selection, the more fit members – called parents – of the population are selected. The selected members are then recombined to form members of the successor population. Recombination has two components: crossover operator and mutation operator. The crossover operator represents the combination of vector entries of a pair of parents to produce children. The mutation operator, on the other hand, refers to making random changes to a single parent. The new population is then carried over to the next generation. A widely used evolution scheme is elitism, where the best one or two individuals from the current population are carried over to the next generation unaltered to guarantee that the solution quality obtained by the algorithm will not decrease from one generation to the next.

In this study, GA-MPC – a variant of GA that proposes a new crossover method – and randomized operation in lieu of mutation, is considered (Elsayed *et al.* 2011). Unlike GA's original formulation, GA-MPC creates an archive pool where the best m individuals are stored and a selection pool with size tc, chosen randomly, is reserved for successful individuals from the tournament selection. Individuals in the selection pool are used for performing crossover, where three parents generate three offsprings: two are designed for exploitation; the other for exploration. After which, a randomized operator is performed with probability p to escape any local minimum. Individuals in the archive pool are then merged with all of the offsprings, where the worst individuals are removed from the population. The surviving population are then carried over to the next generation. A detailed explanation of GA-MPC can be found in the study of Elsayed *et al.* (2011), while the pseudo-code is given below.

In the study (Elsayed *et al.* 2011), GA-MPC was tested using the real-world numerical optimization problems of the IEEE CEC 2011 Real-World Numerical Optimization Special Session (Das and Suganthan 2010), and ranked first among fourteen participating algorithms (Molina *et al.* 2018).

GA-MPC Pseudo-code

Input: Objective function $f(\boldsymbol{x}), \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D)$ for D dimensions, $PS, \boldsymbol{x}_{min}, \boldsymbol{x}_{max}, \beta, cr, m$

Output: cost function $f(\boldsymbol{x}^*)$ at optimal $\boldsymbol{x}^*$

1: Generate initial random population of size PS with

$$\boldsymbol{x}_i = \boldsymbol{x}_{min,i} + u\left(\boldsymbol{x}_{max,i} - \boldsymbol{x}_{min,i}\right),$$

where u is a random vector with values in $[0,1]$.

2: **while** $FuncEvals < MaxEval$ **do**

3: Rank all the individuals in the population by their cost function value and choose the best m individuals to form the archive pool $\tilde{A}$.

4: Apply a tournament selection with size TC and fill the selection pool.

5: Generate a random number $\bar{u}$ in $[0,1]$.

6: **for** each three consecutive individuals in the selection pool **do**

7: **if** one of the selected individual is the same to another **then**

8: Replace one by a randomly-selected individual in the selection pool.

9: **end if**

10: **if** $\bar{u} < cr$ **then**

11: Rank these three individuals $f(\boldsymbol{x}_i) \leq f(\boldsymbol{x}_{i+1}) \leq f(\boldsymbol{x}_{i+2})$.

12: Calculate $\beta = N(\tilde{\mu}, \tilde{\sigma})$.

13: Generate three offspring from the three parents with

$$o_1 = \boldsymbol{x}_1 + \beta(\boldsymbol{x}_2 - \boldsymbol{x}_3),$$

$$o_2 = \boldsymbol{x}_2 + \beta(\boldsymbol{x}_3 - \boldsymbol{x}_1),$$

$$o_3 = \boldsymbol{x}_3 + \beta(\boldsymbol{x}_1 - \boldsymbol{x}_2).$$

14: **end if**

15: **for** each offspring o_i **do**

16: Generate a random number $\bar{u}$ in $[0,1]$.

17: **if** $\bar{u} < p$ **then**

18: Mutate the offspring by $o_i^j = \boldsymbol{x}_{arch}^j$, where $\boldsymbol{x}_{arch}$ is an individual from the archive pool and $arch \in [1, m]$.

19: **end if**

20: **end for**

21: **end for**

22: **if** there is a duplicate individual **then**

23: Replace the duplicate with

$$\boldsymbol{x}_i^j = \boldsymbol{x}_i^j + N(0.5 * \bar{u}, 0.25 * \bar{u}),$$

where $\bar{u} \in [0,1]$.

24: **end if**

25: **end while**

Appendix IV. LSHADE-SPACMA algorithm.

DE is another popular population-based algorithm similar to GA. Currently, there is a huge progress in the study of improvements of DE and its diverse applications (Das *et al.* 2016). LSHADE, a DE-based algorithm (Price *et al.* 2005), is a proposed improvement to SHADE (Tanabe and Fukunaga 2013) that implements a linear population size reduction scheme to focus on exploitation as the optimization process progresses (Tanabe and Fukunaga 2014). Similar to SHADE, it uses the current-to-p best/1 mutation strategy initially proposed in JADE (Zhang and Sanderson 2009), and a binomial crossover method (Tanabe and Fukunaga 2013, 2014). LSHADE also retains the improvements proposed in SHADE for the adaptation of the scaling factor F and the crossover rate cr using function distributions (Tanabe and Fukunaga 2013, 2014).

Covariance Matrix Adaptation – Evolutionary Strategy (CMA-ES), on the other hand, is another adaptive algorithm that adapts the multi-variate normal distribution (Hansen 2006). As an evolution-inspired algorithm, CMA-ES steps are very similar to that of DE (Price *et al.* 2005) and GA (Holland 1975). It starts with an initial "population" of search points sampled from the initial multi-variate normal distribution followed by selection and recombination to update the mean, step size control to update the evolution path, and CMA (Hansen 2006).

In this study, LSHADE-SPACMA is considered. It is an improved version of LSHADE (Tanabe and Fukunaga 2014) that uses a semi-parameter adaptation method for the scaling factor F and crossover rate cr, and a hybridization framework with a modified version of CMA-ES (Hansen 2006; Mohamed *et al.* 2017). As opposed to complete- or self-adaptation, LSHADE-SPACMA uses semi-adaptation. The adaptation process depends on the number of function evaluations carried out so far, *i.e.* until the algorithm reaches half of the defined maximum number of function evaluations, it will only focus on adapting the crossover rate cr, while the scaling factor F is generated randomly using a uniform distribution within the range of $(0.45, 0.55)$. In the second half of the adaptation process, the scaling factor F is adapted. Furthermore, the algorithm uses a modified CMA-ES where a crossover operation is added after the offspring generation (sampling of new points) step. The hybridization is done by allocating subpopulations between LSHADE and modified CMA-ES to produce donor vectors, where the allocation throughout the optimization process varies depending on the performance of each algorithm. A detailed discussion of LSHADE-SPACMA can be found in the study of Mohamed *et al.* (2017), while a simplified pseudo-code is given below.

In the study (Mohamed *et al.* 2017), LSHADE-SPACMA was evaluated using the set of problems presented in IEEE CEC 2017 Real-Parameter Special Session bound constrained case (Awad *et al.* 2016a), where it ranked fourth out of the twelve participating algorithms (Molina *et al.* 2018).

Simplified LSHADE-SPACMA Pseudo-code

Input: Objective function $f(\boldsymbol{x}), \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D)$for D dimensions, memories M_{CR}, M_F, and

$M_{FCP}, Arc_{rate}, p_{best}, c, H, PS_{min}, PS_{max}$.

Output: cost function $f(\boldsymbol{x}^*)$ at optimal $\boldsymbol{x}^*$

1: Generate initial random population with size PS_{max}.

2: Set values of memories M_{CR}, M_F,and M_{FCP} to 0.5.

3: Initialize CMA parameters.

4: **while** $FuncEvals < MaxEval$ **do**

5: Semi-parameter adaptation (SPA) for scaling factor F and crossover rate CR.

During the first part of SPA, the adaptation is concentrated on CR and for the second part the focus of the adaptation is on F.

6: Split the population into two.

7: Generate donor vectors using LSHADE or modified CMA-ES.

8: Concatenate resulting vectors from LSHADE and modified CMA-ES.

9: Generate trial vectors and mutate with p_{best}.

10: Evaluate fitness of trial vectors.

11: Implement selection strategy.

12: Update population allocated to LSHADE and CMA-ES according to the relative performance of the two methods.

13: Store successful parameters to memory of size H.

14: Update archive with rate Arc_{rate}.

15: Update memory memory M_F during the first part of SPA and M_{FCP} with c during the second part. Update also M_{CR}.

16: Implement linear population size reduction with min pop size PS_{min}.

17: Sort individuals and retain them based on the new population size.

18: Update CMA-ES parameters.

19: **end while**

Appendix V. LSHADE-cnEpSin algorithm.

As the name suggests, LSHADE-cnEpSin is another DE-based algorithm that is similar to LSHADE (Tanabe and Fukunaga 2014) and its variants. While it shares certain characteristics with LSHADE (Tanabe and Fukunaga 2014) or LSHADE-SPACMA (Mohamed *et al.* 2017) such as linear population size reduction or its use of current-to-p best/1 mutation strategy, LSHADE-cnEpSin adds an ensemble sinusoidal approach to adapt the values of the scaling factor F non-adaptive sinusoidal decreasing adjustment and adaptive history-based sinusoidal increasing adjustment (Awad *et al.* 2016b, 2017). In the former, a decreasing sine-based formula is used where the wave-like configuration dampens as the optimization process progresses; in the latter, an increasing adaptive sine-based formula using Cauchy distributions with mean taken from an external memory (which stores successful mean frequencies) is considered. The choice of which sinusoidal approach to use is based on previous performance. In particular, a learning period is first implemented for a certain number of generations before the respective probabilities for each sine-based formula is updated (Awad *et al.* 2017). These sinusoidal approaches are only activated in the first half of the optimization process, while the usual formulation of SHADE (Tanabe and Fukunaga 2013) in adapting the scaling factor F, using Cauchy distributions, is used in the second half (Awad *et al.* 2017).

LSHADE-cnEpSin, although sharing the same adaptation process for the crossover rate cr with LSHADE (Tanabe and Fukunaga 2014), adds another crossover operator using covariance matrix learning with Euclidean neighborhood (Awad *et al.* 2017). This process is done by marking the best individual, and computing the Euclidean distance between the best and every other individual in the population. A number of best individuals in terms of Euclidean distance are then used to generate the covariance matrix, which are then used to update the target and trial vectors (Awad *et al.* 2017). A detailed discussion of LSHADE-cnEpSin can be found in the study of Awad *et al.* (2017), while a simplified pseudo-code is given below.

In the study (Awad *et al.* 2017), LSHADE-cnEpSin was evaluated using the set of problems presented in IEEE CEC 2017 Real-Parameter Special Session bound constrained case (Awad *et al.* 2016a), where it ranked third out of the twelve participating algorithms (Molina *et al.* 2018).

Simplified LSHADE-cnEpSin Pseudo-code

Input: Objective function $f(\boldsymbol{x}), \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D)$for D dimensions, memories M_{CR}, M_F, and

$M_{freq}, Arc_{rate}, p_{best}, freq, H, PS_{min}, PS_{max}, G_{max}$.

Output: cost function $f(\boldsymbol{x}^*)$ at optimal $\boldsymbol{x}^*$

1: Generate initial random population with size PS_{max}.

2: Initialize $G_{max} = 2163$.

3: Set values of memories M_{CR}, M_F,and M_{freq} to 0.5.

4: Initialize covariance matrix settings.

5: **while** $FuncEvals < MaxEval$ **do**

6: **if** number of generation is $< G_{max}/2$ **then**

7: Implement sinusoidal configuration to adapt F using $freq$.

8: **else**

9: Use Cauchy distribution to adapt F.

10: **end if**

11: Adapt CR using normal distribution.

12: **for** $i = 1$ to population size **do**

13: Generate mutant vectors with p_{best}.

14: Apply covariance matrix learning or binomial crossover to generate trial vectors.

15: Store successful F and CR.

16: **end for**

17: Update memory M_{CR}, M_F of size H, and archive with rate Arc_{rate}.

18: Implement linear population size reduction with min pop size PS_{min}.

19: Sort individuals and retain them based on the new population size.

20: **end while**

Appendix VI. EBOwithCMAR algorithm.

Effective butterfly optimizer (EBO), proposed in 2015, is inspired by the mate locating behavior of male butterflies (Kumar *et al.* 2015). Two mate locating behaviors are used for the modification of the population: perching for exploitation and patrolling for exploration. The algorithm starts with an initial population of male butterflies, which are then divided into two subpopulations: main and auxiliary butterflies. The main population of butterflies start perching and only change to patrolling if their positions are not updated by the former strategy.

In this study, an improvement to EBO named EBOwithCMAR is considered. Unlike EBO that uses two subpopulations, EBOwithCMAR uses three: two are for EBO (Kumar *et al.* 2015) and one for covariance matrix adapted retreat (Hansen 2006; Kumar *et al.* 2017). The additional subpopulation is meant to improve the exploitation capability of the algorithm. Moreover, EBOwithCMAR improves the perching and patrolling strategies by adjusting the crisscross and towards-best modification, and adding a crossover operator (Kumar *et al.* 2017). It also uses an adaptive strategy for the scaling factor F and the crossover rate cr and a linear population size reduction – the same schemes used in LSHADE (Tanabe and Fukunaga 2014). A data sharing scheme is also added where the better performing algorithm shares information about the solution to the other algorithm (Kumar *et al.* 2017).

Each cycle (iteration) begins with a "learning period" where both perching and patrolling schemes are given equal probability. Probabilities are held fixed until half a cycle is reached, which are then updated. Once a full cycle is reached, the algorithm implements a data sharing scheme where the better performing algorithm between EBO and CMAR is determined. If EBO is the better algorithm, the population dedicated to CMAR is replaced by a random element from the main population of EBO. On the other hand, if CMAR is the better algorithm, the worst individual in the main population of EBO is replaced by the best individual from the population dedicated to CMAR. After the data sharing scheme, parameter values are reset and the probabilities for perching and patrolling are returned to their initial values. EBOwithCMAR further enhances the exploitation capability of EBO by employing sequential quadratic programming at the later phases of the optimization process. A detailed discussion of EBOwithCMAR can be found in the study of Kumar *et al.* (2017), while a simplified pseudo-code is given below.

In the study (Kumar *et al.* 2017), EBOwithCMAR was evaluated using the set of problems presented in IEEE CEC 2017 Real-Parameter Special Session bound constrained case (Awad *et al.* 2016a), where it ranked first out of the twelve participating algorithms (Molina *et al.* 2018).

Simplified EBOwithCMAR Pseudo-code

Input: Objective function $f(\boldsymbol{x}), \boldsymbol{x} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_D)$for D dimensions, population sizes $PS_{1,max}$ and $PS_{1,min}$, $PS_{2,max}$ and $PS_{2,min}$, and PS_3, $prob_{ls}$, $prob_1$, $prob_2$, and cycle CS

Output: cost function $f(\boldsymbol{x}^*)$ at optimal $\boldsymbol{x}^*$

1: Generate initial random population with size PS_{max}.

2: Set $prob_1$ for EBO and $prob_2$ for CMAR to 1. Set other parameters.

3: Randomly assign members of the main population to three subpopulations X_1, and X_2 for EBO and X_3 for CMAR. The max and min population size for X_1 are $PS_{1,max}$ and $PS_{1,min}$, respectively, $PS_{2,max}$ and $PS_{1,min}$ for X_2, and PS_3 is the pop size for X_3.

4: **while** $FuncEvals < MaxEval$ **do**

5: **if** number of cycle is $== CS/2$ **then**

6: Update $prob_1$ and $prob_2$.

end if

if number of cycle is $== CS$ **then**

Implement data sharing.

Update parameters for EBO using H and CMAR with $\tilde{\sigma}$.

Reset $prob_1$, $prob_2$ and number of cycles.

end if

if $rand(0,1) < prob_1$ **then**

Apply EBO.

Implement linear population size reduction and reallocate subpopulations.

end if

if $_{\text{rand}}(0,1) < prob_2$ **then**

Apply CMAR.

end if

if $_{\text{rand}}(0,1) < prob_{ls}$ and $FuncEvals \geq 0.75 * MaxEval$ **then**

Apply SEQ.

if best solution is improved **then**

Set $prob_{ls}$ to 0.1.

Update X_1 and X_2.

else

Set $prob_{ls}$ to 0.0001.

end if

end if

Sort individuals and update allocations between subpopulations.

end while

Appendix VII. Parameter setting for heuristic methods.

The parameter settings used in all the algorithms are set to values in their original formulation. In particular, the default values in the main papers cited in Section 3 for each algorithm are chosen, unless certain configurations need to be made to address issues, such as algorithmic complexity. The lack of parameter tuning to fit the EIT problem in conducting the numerical simulations is intended to ensure that the study does not favor any algorithm, which allows for a thorough comparison of the original methods.

Parameters for FA were set as follows: $\alpha_0 = 0.2, \gamma = 1$, and $\beta = 0.2$, where α_0 is the initial randomization parameter, γ is the light absorption coefficient, and β is the base attraction coefficient. The population size in each

generation for FA varies for each case: Case 1 has 20 fireflies, Case 2 has 30, and Case 3 has 60. Each firefly would require the computation of the cost functional (18), which is expensive and accounts for the major running time of the algorithm.

r_{min}	0	α	0.9
r_{max}	1	P_{min}	0.6
f_{min}	0	P_{max}	0.9
f_{ma}	1.5	C_{min}	0.1
A_{min}	1	C_{max}	0.9
A_{max}	2	G	10
γ	0.9		

In the table above, the parameter settings for NBA are enumerated: r, f, and A denote the rate, frequency, and loudness of sound pulses emitted by bats, respectively. P denotes the probability of habitat selection, while C denotes the compensation rate for Doppler effect in echoes. The constants γ and α are used to update r and A. The parameters r and A are re-initialized if the best solution does not change after G time steps. Meanwhile, the contraction-expansion coefficient θ and the inertia weight w are adjusted according to the parameter control method described by Tian *et al.* (2012), *i.e.* using $a \cdot \cos\left(\frac{\pi t}{2\widetilde{M}}\right) + a$, where t is the current time step, and $a = 0.5$ for θ while $a = 0.4$ for w. While $\widetilde{M}$ is defined to be the maximum number of time steps in the study of Tian *et al.* (2012), in this work, we set its value to be the IEEE arithmetic representation for positive infinity.

Parameters for GA-MPC were set as follows: population size $PS = 90$, "crossover factor" $\beta \sim N(0.5,0.3)$, crossover rate $cr = 1$, tournament selection size generated randomly between 2 and 3, and archive pool size $|A| = 45$ (half the population size) (Elsayed *et al.* 2011). LSHADE-SPACMA and LSHADE-cnEpSin share several parameter values: initial population size $PS_{max} = 18 * D$ and minimum population size $PS_{min} = 4$ as both algorithms implement a linear population size reduction, initial values for M_{CR}, M_F at 0.5, factor that controls the greediness of the mutation strategy $p_{best} = 0.11$, archive rate $Arc_{rate} = 1.4$, and memory size $H = 5$ used in storing adapted parameters (Awad *et al.* 2017; Mohamed *et al.* 2017). Parameter settings specific to LSHADE-SPACMA are as follows: initial value for $M_{FCP} = 0.5$ used in the hybridization framework, and learning rate $c = 0.8$ used in updating the probability for hybridization (Mohamed *et al.* 2017); while the parameter specific to LSHADE-cnEpSin used for the non-adaptive sinusoidal decreasing adjustment $freq$ is set to 0.5 (Awad *et al.* 2017).

EBOwithCMAR uses the following parameter values: for EBO, it uses the same initial population size $PS_{1,max} = 18 * D$, minimum population size $PS_{1,min} = 4$ for PS_1 with LSHADE-SPACMA and LSHADE-cnEpSin (Awad *et al.* 2017; Mohamed *et al.* 2017), $PS_{1,max} = 46.8 * D, PS_{2,min} = 10$, and memory size $H = 6$; while for CMAR, $PS_3 = 4 + 3log(D), \tilde{\sigma} = 0.03$, number of evaluations constituting a cycle $CS = 50$, and local search update probability $prob_{ls} = 0.1$ (Kumar *et al.* 2017).