A Non-linear Geophysical Inversion Algorithm for the MT Data Based on Improved Differential Evolution

Jie Xiong, Member, IAENG, Caiyun Liu, *Yuantao Chen, and Song Zhang

***Abstract*—The magnetotelluric (MT) method has been widely employed in the exploration of hydrocarbon and mineral resources.Traditional linear iterative inversion method can determine the electrical resistivity of the Earths subsurface from MT data rapidly, but it relies on the gradient of the forward operator and its result dependents on the initial model extremely. In order to avoid the disadvantages of traditional linear inversion, a novel non-linear geophysical inversion algorithm is proposed for the MT data based on improved differential evolution. The proposed algorithm is applied to invert the synthetic MT data of 1D layered geo-electrical models. The consistent results are obtained in the noise-free cases. When Gauss noises of 10% and 20% are added to the synthetic data, the results of inversions remain fairly good. Numerical experiment results demonstrate that the improved inversion algorithm has the advantages of independent of initial model, capable of global exploration, and anti-noise capability. It makes MT data inversion more effective.**

***Index Terms*—Non-linear inversion, geophysical inversion, Magnetotelluric (MT) data, improved differential evolution.**

I. INTRODUCTION

THE magnetotelluric (MT) method [1] is a kind of geophysical methods which uses the natural electromagnetic field as the source to explore the electrical structure of the Earth's interior (illustrated in Figure 1, 2). It has some advantages such as low cost, large probing depth, not affected by high-resistivity shielding and high resolution to the low-resistivity layer [2]. This technique has been widely employed in the research of the structure in Earth's interior, exploration of hydrocarbon resources and survey for solid mineral resources [3].

Data inversion is one of the core issues of the MT exploration [3], which can be classified into linear and nonlinear categories. The linear inversion has a well-established theory, fast convergence and lots of applications, but its result depends on the initial model extremely, and easily falling into local minimum. The nonlinear inversion can search in the whole solution space and avoid local minimum because it explores the whold search space directly, not relying on the information of gradients at all. At present the most representative nonlinear inversion methods are Monte Carlo (MC)[7][8], simulated annealing (SA) [9] [10], genetic algorithm (GA)[11][12], and particle swarm optimization (PSO)[13][14][15], which can overcome the flaws of the linear inversion and obtain the global convergence. They have, however, some disadvantages, for instance the MC and SA algorithms require long computation time and converge slowly, the GA is subject to gene loss and early-maturing, and the PSO is subject to early-maturing too. Therefore, it is necessary to develop new and more effective algorithms for MT data inversion.

The differential evolution (DE) is a relatively new member in the family of evolutionary algorithms. It is a population-based stochastic parallel search evolutionary algorithm which is very simple yet powerful[16][17]. The main advantage of DE is its capability of solving optimization problems which require minimization process with nonlinear, non-differentiable and multi-modal objective functions[18]. At present DE has been successfully applied to many fields of optimization[19][20][21][22][23]. In recent years it has also been introduced into the geophysical inversion[24][25][26][27].

Previous work indicates that the performance of DE mainly depends on trial vector generation strategies including mutation and crossover operators, and control parameters such as size N_p, scale factor F and crossover factor CR[28]. Inspired by the ideal of Basu[18], we propose a non-linear improved differential evolution inversion algorithm for the MT data. We benchmark the performance of improved DE first. Then, we apply it to the inversion of synthetic MT data and analyze its accuracy and anti-noise capacity.

II. PROBLEM FORMULATION

A. Forward Modeling

Assume an 1D layered model through a subspace section of L layers, of which from up to down the resistivities are $\rho_1, \rho_2, ..., \rho_L$, respectively, and the thickness are $h_1, h_2, ..., h_{L-1}$, respectively, where $h_n = \infty$. The n-layer model (see Figure 3) can be described by the 2L-1 dimension vector:

$$\mathbf{m} = (\rho_1, \rho_2, ..., \rho_L, h_1, h_2, ..., h_{L-1})^T \tag{1}$$

MT data are generated by measuring the horizontal components of the electric and magnetic fields at the surface (see Figure1,2) to obtain the impendance $Z_{xy}(\omega)$ as a function of

Manuscript received June 25, 2017; revised October 29, 2017. This work was supported in part by the National Natural Science Fundation of China (No. 61273179, 61673006), the key project of Educational Commission of Hubei Province of China (No.D20131206, B2016034, 20141304).

J. Xiong is with the School of Electrics and Information, Yangtze Unversity, Jingzhou, Hubei, 434023 China.

C. Liu is with the School of Information and Mathematics, Yangtze Unversity, Jingzhou, Hubei, 434023 China

*Y. Chen, corresponding author, is with the School of Computer & Communication Engineering, Changsha Unversity of Science and Technology, Changsha, Hunan, 410114, China e-mail: chenyt@cscst.edu.cn

S. Zhang, is with the Department of Computer Science and Engineering, Mississippi State Unversity, Starkville, MS, 39759 USA.

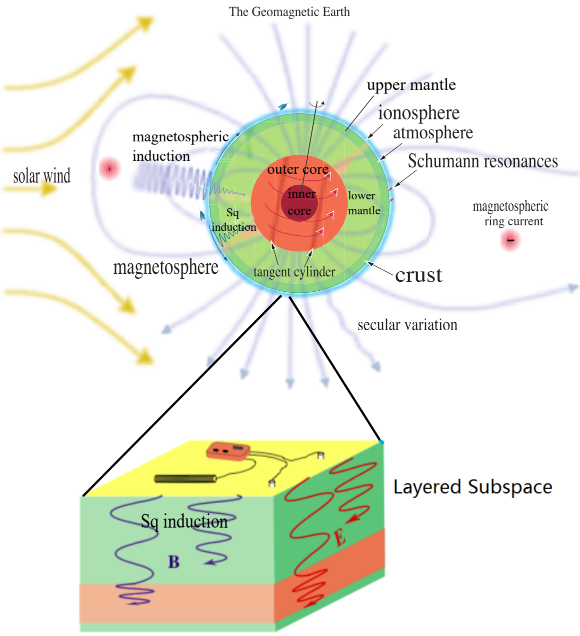


Fig. 1: The Earth's electromagnetic environment (upper) and the schematic layouts of MT experiment (lower) (Modified from Constable and Constable, 2004 [4] and Constable, 2007 [5]).

Fig. 2: Typical MT equipment (from Nieuwenhuis, 2011[6]).

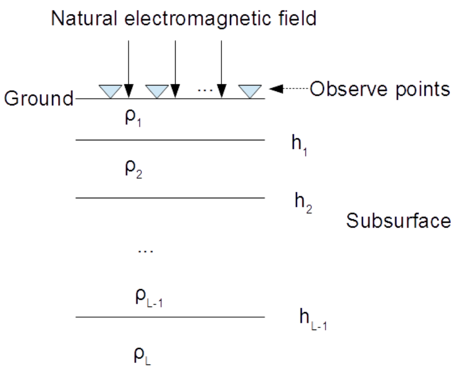


Fig. 3: The L layers geo-electrical model.

the frequency of the signal ω, which can be calculated for 1D case as:

$$Z_{xy}(\omega) = \frac{E_x(\omega)}{H_y(\omega)} \tag{2}$$

where $E_x(\omega)$ and $H_y(\omega)$ are mutually perpendicular components of the electric and magnetic fields, respectively. MT data are expressed by the apparent resistiveity $\rho_{xy}(\omega)$ and phase $\phi_{xy}(\omega)$, which can be calculated as follows:

$$\rho_{xy}(\omega) = \frac{||Z_{xy}(\omega)||^2}{\omega\mu}, \tag{3}$$

and

$$\phi(\omega) = \arctan\frac{Im[Z_{xy}(\omega)]}{Re[Z_{xy}(\omega)]}, \tag{4}$$

where $\omega = 2\pi/T$ is angular frequency, μ is the magnetic permeability of the medium, $Re(Z)$ and $Im(Z)$ are the real and imaginary parts of Z, respectively. MT data in the form of apparent resistivity and phase, as functions of frequency are used to model the subsurface resistivity.

The forword calculation scheme for obtaining the 1-D MT response is implemented by calculating the complex Z value, uing the recurrence relation:

$$Z_i = Z_{0i}\frac{Z_{0i}(1-e^{-2k_ih_i}) + Z_{i+1}(1+e^{-2k_ih_i})}{Z_{0i}(1+e^{-2k_ih_i}) + Z_{i+1}(1-e^{-2k_ih_i})}, \tag{5}$$

$$Z_N = \frac{\omega\mu}{K_N} = Z_{0N}, \tag{6}$$

where $k_i = \sqrt{i\omega\mu/\rho_i}$ is the complex wave numbers of the $i-th$ layer, Z_{0i} is the characteristic impedance of the $i-th$ layer, and Z_i is the wave impedance of the top of the $i-th$ layer.

If we measure on the K frequence $\omega_1, \omega_2, ..., \omega_K$, we can obtain the observation data with 2K dimension:

$$\mathbf{d} = [\rho_{xy}(\omega_1), ..., \rho_{xy}(\omega_K), \phi(\omega_1), ..., \phi(\omega_K)]^T \tag{7}$$

The forward model can be written as

$$\mathbf{d} = A(\mathbf{m}), \tag{8}$$

where $\mathbf{m}$ is a model parameter, A is forward functional corresponding to the formula (2)-(6), and $\mathbf{d}$ is the theoretical data corresponding to the model $\mathbf{m}$.

B. Inversion Objective Function

Inversion is to solve the model parameter $\mathbf{m}$ from the observed data $\mathbf{d}^{obs}$, which makes the fitness error between the theoretical data $\mathbf{d} = A(\mathbf{m})$ and the observed data $\mathbf{d}^{obs}$ least. The objective function of inversion is defined as the norm L_2 of the difference between the observed and theoretical data to describe the fitness degree, which is expressed as

$$F(\mathbf{m}) = ||\mathbf{d}^{obs} - A(\mathbf{m})||^2 \tag{9}$$

III. Methodology

A. Differential Evolution Algorithm

Differential Evolution (DE)[16] is a parallel direct search method which utilizes N_p n-dimension vectors

$$x_i(k) = [x_{i1}(k), x_{i2}(k), ..., x_{in}(k)]^T \tag{10}$$

as a population for each generation k, where $i = 1, 2, ..., N_p$.

The basic DE algorithm can be described as follows:

Step 1. Initialization.

The initial population is chosen randomly and should cover the entire parameter space. As a rule, we assume a uniform probability distribution for all random decisions unless otherwise stated.

Step 2. Mutation.

DE generates new parameter vectors by adding the weighted difference between two population vectors to a third vector, named mutation. For each target vector $x_i(k)$, a mutant vector is generated according to

$$v_i(k+1) = x_{r1}(k) + F \cdot [x_{r2}(k) - x_{r3}(k)] \quad (11)$$

with random indexes $r1, r2, r3 \in (1, 2, ..., N_p)$, integer, mutually different and $F > 0$. F is a real and constant factor named scale factor.

Step 3. Crossover.

In order to increase the diversity of the perturbed parameter vectors, crossover is introduced and the trial vector:

$$u_i(k+1) = [u_{i1}(k+1), u_{i2}(k+1), ..., u_{in}(k+1)]^T \quad (12)$$

is formed, where:

$$u_{ij}(k+1) = \begin{cases} v_{ij}(k+1), & if\ rand(b_j) \leq CR \\ & or\ j = rang(i); \\ x_{ij}(k), & if\ rand(b_j) > CR \\ & or\ j \neq rang(i) \end{cases} \quad (13)$$

where $j = 1, 2, ..., n, rand(b_j)$ is the $j-th$ evaluation of a uniform random number generator with outcome ∈ [0,1]. CR is the crossover constant ∈ [0,1] which has to be determined by the user. $rang(i)$ is a randomly chosen index ∈ 1,2,...,n which ensure that $u_{ij}(k+1)$ gets at least one parameter from $v_{ij}(k+1)$.

step 4. Selection.

To decide whether or not it should become a member of generation $k+1$, the trial vector $u_{ij}(k+1)$ is compared to the target vector $x_{ij}(k)$ using the greedy criterion as follow:

$$x_i(k+1) = \begin{cases} u_i(k+1), & F[u_i(k+1)] > F[x_i(k)]; \\ x_i(k), & F[u_i(k+1)] \leq F[x_i(k)] \end{cases} \quad (14)$$

step 5. Stop Determination.

If the population convergent or reach the maximum iteration k_{max}, stop; otherwise go to step 2, beginning the next iteration.

B. Improvement Strategy

As pointed out in [28], the performance of DE mainly depends on trial vector generation stategies including mutation and crossover operators, and control parameters such as population size N_p, scale factor F, crossover factor CR and so on. In order to improve the exploration and exploitation of single mutation strategy, a Gaussian random variable scaling factor is introduced by Basu[18].

The new mutation strategy is described as following:

For each target vector $x_i(k)$, a mutant vector is generated according to

$$v_i(k+1) = x_{r1}(k) + N(0, \sigma_i^2) \cdot [x_{r2}(k) - x_{r3}(k)] \quad (15)$$

with random indexes $r1, r2, r3 \in (1, 2, ..., N_p)$, integer, mutually different. $N(0, \sigma_i^2)$ represents a Gaussian random variable with mean zero and standard deviation σ_i.

The standard deviation σ_i is given by expression

$$\sigma_i = f(X_i)/f_{min} \quad (16)$$

where f_{min} is the minimum cost value amoung N_p vectors. $f(X_i)$ is the value of the objective function associated with $i-th$ vector.

This Gaussin random variable controls the amount of perturbation added to the parent vector and aids the algorithm to escape from local optima. This maintains the diversity of the population through out iterative process which guarantees a high probability of achieving the global optimum.

C. Improved Differential Evolution Inversion Algorithm

One-dimensional inversion of layered models is a common method in MT study, which can yield reliable results in a survey area with little lateral variation in a fast and accurate manner. In an area with relatively large lateral variations, this inversion can describe the whole geo-electrical distribution qualitatively and provide initial models for 2D and 3D inversions. An improved differential evolution inversion algorithm (IDE-I) is proposed as following:

Step 1: Define the population size N_p, model layer L, crossover factor CR;
Step 2: Initialize the N_p individuals randomly in the entire parameter space, each individual correspond to a model vector **m**;
Step 3: Using formulas (2) - (8), conduct forward calculation to yield theoretical data $\mathbf{d} = A(\mathbf{m})$;
Step 4: In terms of formula (9), calculate fitness of each individual, respectively;
Step 5: Excute the mutation operation following the formula (15)(16);
Step 6:Excute the crossover operation following the formula (12)(13);
Step 7: Excute the selection operation following the formula (14);
Step 8: if global convergence achieved or maximum number of iteration is met, go to Step 9; Other -wise return to Step 3 to perform next iteration;
Step 9: output result of inversion, finish.

Fig. 4: Improved Differential Inversion (IDE-I) algorithm.

IV. Evaluation IDE algorithm

A. Benchmark functions and experimental setting

There are 27 benchmark functions[29] are used in the following evaluation. The details of these benchmark functions are described in Table I. These functions could be classified into two categories. The benchmark functions $f_1 \sim f_{13}$,which introduced by [30], are commonly used in the evolutionary computation community. The remaining 14 benchmark functions $f_{14} \sim f_{27}$ are the first 14 functions proposed for the CEC 2005 by Suganthan et al. [31], which are shifted and exceedingly difficult.

The parameters used in classical DE is NP=100, F=0.5, CR=0.9. The parameters used in IDE is $NP = 100$, CR=0.3. In our experiments, each algorithm runs 30 times independently for each benchmark function and the maximum number of function evalution is set to 10000*D, where D is the number of variables. All experiments are carried out on a computer equipped Intel i5 420 processor and 4G RAM with Windows 7 operation system.

TABLE I: Testsuit with 27 benchmark functions.

Type	Function	Name	Search range	Global optimum
Unimodal functions	f_1	Sphere	[-100,100]	0
	f_2	Schwefel2.22	[-10,10]	0
	f_3	Schwefel1.2	[-100,100]	0
	f_4	Schewefel2.21	[-100,100]	0
	f_5	Rosenbrock	[-30,30]	0
	f_6	Step	[-1.28,1.28]	0
	f_7	Quartic with Noise	[-100,100]	0
Multimodal functions	f_8	Schewefel2.26	[-500,500]	-418.98*D
	f_9	Rastrigin	[-5.12,5.12]	0
	f_{10}	Ackley	[-32,32]	0
	f_{11}	Griewank	[-600,600]	0
	f_{12}	Penalized1	[-50,50]	0
	f_{13}	Penalized2	[-50,50]	0
Shifted unimodal functions	f_{14}	Shifted Sphere Function	[-100,100]	-450
	f_{15}	Shifted Scwefels Problem 1.2	[-100,100]	-450
	f_{16}	Shifted Rotated High Conditioned Elliptic Function	[-100,100]	-450
	f_{17}	Shifted Schwefels Problem 1.2 with Noise	[-100,100]	-450
	f_{18}	Schwefels Problem 2.6 with Global Optimum on Bounds	[-100,100]	-310
Shifted unimodal functions	f_{19}	Shifted RosenBrocks Function	[-100,100]	390
	f_{20}	Shifted Rotated Griewanks Function without Bounds	[0, 600]	-180
	f_{21}	Shifted Rotated Ackleys Function with Global Optimum on Bounds	[-32,32]	-140
	f_{22}	Shifted Rastrigins Function	[-5,5]	-330
	f_{23}	Shifted Rotated Rastrigins Function	[-5,5]	-330
	f_{24}	Shifted Rotated Weierstrass Function	[-0.5,0.5]	90
	f_{25}	Schwefels Problem 2.13	[$-\pi$,π]	-460
	f_{26}	Shifted Expanded Griewanks Plus Rosenbrocks Function	[–3,1]	-130
	f_{27}	Shifted Rotated Expanded Scaffers F6 Function	[-100,100]	-300

TABLE II: Experimental results of DE/rand/1, DE/best/1, and IDE for all test functions at D = 30.

F	DE/rand/1	DE/best/1	IDE
	Ave Err ± Std Dev	Ave Err ± Std Dev	Ave Err ± Std Dev
f_1	2.68E-36±2.68E-36-	**5.40E-323±0.00E+00+**	6.47E-95±1.93E-95-
f_2	5.40E-18±3.59E-18-	**2.82E-84±1.37E-83≈**	**3.42E-84±2.76E-83≈**
f_3	3.33E-05±2.73E-05-	**9.48E-70±3.59E-69+**	5.29E-14±2.65E-14-
f_4	2.53E-01±7.73E-01-	6.58E-09±1.95E-08-	**3.22E-09±5.95E-09+**
f_5	2.29E-02±6.33E-02-	1.33E+00±1.88+00-	**2.16E-12±3.68E-14+**
f_6	**0.00E-00±0.00E-00≈**	7.53E+00±8.31E+00-	**0.00E-00±0.00E-00≈**
f_7	4.53E-03±1.47E-03-	7.36E-03±4.67E-03-	**2.71E-03±7.64E-04+**
f_8	6.57E+03±6.04E+02-	5.00E+03±6.27E+02-	**7.39E-04±8.47E-05+**
f_9	1.38E+E02±2.78E01-	5.26E+01±1.34E+01-	**0.00E+00±0.00E+00+**
f_{10}	4.14E-15±1.32E-15-	4.67E+00±1.29E+00-	**4.78E-15±0.00E+00+**
f_{11}	**0.00E+00±0.00E+00≈**	4.82E-02±4.94E-02-	**0.00E+00±0.00E+00≈**
f_{12}	**1.57E-32±5.47E-48≈**	2.36E+00±2.84E+00-	**1.58E-32±8.75E-48≈**
f_{13}	**1.35E-32±5.47E-48≈**	1.45E+00±1.74E+00-	**1.35E-32±5.27E-48≈**
f_{14}	**0.00E+00±0.00E+00≈**	3.64E-13±2.62E-13-	**0.00E+00±0.00E+00≈**
f_{15}	4.15E-05±3.95E-05-	1.18E-12±7.06E-13-	**3.55E-13±7.93E-14+**
f_{16}	3.54E+05±1.86E+05-	**1.39E+04±10.8E+04+**	1.65E+05±1.08E+05-
f_{17}	1.82E-02±2.54E-05-	2.06E+02±3.50E+02-	**2.66E-04±4.74E-04+**
f_{18}	**5.78E+01±6.46E+01≈**	2.09E+03±7.14E+02-	**5.61E+01±6.58E+01≈**
f_{19}	1.90E-01±8.27E-01-	1.06E+00±1.76E+00-	**1.26E-01±6.82E-01+**
f_{20}	**1.99E-14±1.30E-14+**	2.04E-02±1.92E-02-	2.79E-03±5.14E-03-
f_{21}	**2.09E+01±4.78E-02≈**	**2.09E+01±5.92E–02≈**	**2.09E+01±5.20E–02≈**
f_{22}	1.34E+02±2.23E+01-	1.06E+02±1.84E+01-	**2.73E-15±2.20E-15+**
f_{23}	1.83E+02±8.39E+00-	1.53E+02±3.95E+01-	**1.00E+02±1.30E+00+**
f_{24}	3.97E+01±1.13E+00-	**2.12E+01±3.14E+00≈**	**2.36E+01±2.94E+00≈**
f_{25}	**1.42E+03±2.98E+03≈**	**1.27E+03±1.32E+03≈**	5.58E+03±2.24E+04≈
f_{26}	1.53E+01±9.23E-01-	6.53E+00±2.24E+00-	**3.77E+00±2.76E-01+**
f_{27}	1.33E+01±1.21E-01-	**1.19E+01±5.42E-01≈**	**1.17E+01±5.61E-01≈**
-	18	19	4
+	1	3	12
≈	8	5	11

TABLE III: Mean errors of DE/rand/1, DE/best/1, and IDE for all functions at D=50 and D=100.

F	D=50			D=100		
	DE/rand/1	DE/best/1	IDE	DE/rand/1	DE/best/1	IDE
f_1	2.35E-39-	**1.46E-266+**	1.43E-143-	8.73E-44-	7.80E-100-	**3.21E-126+**
f_2	7.03E-20-	4.80E-05-	**2.68E-56+**	2.36E-24-	3.05E-01-	**2.72E-65+**
f_3	1.02E+00-	**5.29E-32+**	8.17E-06-	1.37E+03-	**1.27E-06+**	1.57E-01-
f_4	5.87E+00-	1.60E+01-	**1.25E-01+**	1.89E+01-	4.06E+01-	**1.18E+01+**
f_5	1.79E+01-	1.99E+00≈	**2.07E-01+**	1.22E+02-	**1.46E+00+**	5.52E+01-
f_6	**0.00E+00≈**	6.32E+02-	**0.00E+00+**	2.43E+00-	6.64E+03-	**0.00E+00+**
f_7	6.61E-03-	3.93E-02-	**2.87E-03+**	1.73E-02-	7.49E-01-	**6.99E-03+**
f_8	1.19E+04-	9.82E+03-	**5.89E-04+**	2.33E+04-	2.16E+04-	**1.82E-03+**
f_9	2.18E+2-	1.05E+02-	**4.44E+00+**	**8.80E+01+**	2.38E+02-	1.72E+02-
f_{10}	6.99E-15-	8.28E+00-	**5.62E-15+**	**1.67E-14≈**	1.33E+01-	**1.60E-14≈**
f_{11}	**0.00E+00≈**	8.87E-02-	**0.00E+00≈**	2.63E-03-	4.77E-01-	**1.28E-03+**
f_{12}	**9.45E-33≈**	1.86E+00-	**9.40E-33≈**	6.22E-03-	1.60E+00-	**2.00E-30+**
f_{13}	**1.37E-32+**	2.95E+00-	4.07E-32-	**7.32E-04≈**	3.14E+00-	**7.44E-04≈**
f_{14}	5.12E-14≈	6.25E-11-	**3.99E-14≈**	**8.53E-14≈**	7.84E-09-	**8.66E-14≈**
f_{15}	3.29E+00-	**1.29E-11+**	4.01E05-	4.47E+03-	**3.57E-08+**	1.53E+00-
f_{16}	2.58E+06-	**8.18E+04+**	2.82E+5-	6.36E+06-	**1.10E+06≈**	**1.08E+06≈**
f_{17}	3.39E+02-	6.84E+03-	**1.34E+02+**	3.61E+04-	6.83E+04-	**2.12E+04+**
f_{18}	**1.84E+03+**	6.86E+03-	2.74E+03-	4.79E+03-	2.11E+04-	**3.47E+03+**
f_{19}	3.69E+01-	1.59E+00-	**1.52E-01+**	1.18E+02-	**1.33E+00+**	5.68E+01-
f_{20}	**5.75E-04+**	1.23E-02-	3.52E-03-	2.96E-03-	7.80E-03-	**2.72E-03+**
f_{21}	**2.11E+01≈**	**2.11E+01≈**	**2.11E+01≈**	**2.13E+01≈**	**2.13E+01≈**	**2.13E+01≈**
f_{22}	2.02E+02-	2.49E+02-	**6.13E-01+**	**1.25E+02+**	6.14E+02-	**1.27E+02≈**
f_{23}	3.63E+02-	3.83E+02-	**2.84e+02+**	8.48E+02-	1.04E+03-	**6.28E+02+**
f_{24}	7.26E+01-	**4.71E+01+**	6.48E+01-	1.60E+02-	**1.17E+02≈**	**1.19E+02≈**
f_{25}	7.92E+03-	**4.77E+03+**	4.64E+04-	2.98E+04 -	**2.30E+04+**	8.33E+05-
f_{26}	3.00E+01-	2.20E+01-	**8.38E+00+**	6.54e+01-	1.03E+02-	**2.88E+01+**
f_{27}	2.30E+01-	**2.14E+01≈**	**2.18E+01≈**	4.76e+01-	**4.55E+01≈**	**4.58E+01≈**
-	19	18	9	21	18	6
+	3	6	13	2	5	13
≈	5	3	5	4	4	8

TABLE IV: Experimental results of DEGL/SAW, EPSDE, MGBED, and IDE for all test functions at D = 30.

F	DEGL/SAW	EPSDE	MGBDE	IDE
	Ave Err ± Std Dev	Ave Err ± Std Dev	Ave Err ± Std Dev	Ave Err ± Std Dev
f_1	6.01E-101±2.10E-100-	**8.47E-174±0.00E+00+**	1.51E-91±6.81E-91-	6.47E-95±1.93E-95-
f_2	1.63E-49±1.53E-49-	8.69E-86±3.84E-85-	1.52E-53±8.15E-53-	**3.42E-84±2.76E-83+**
f_3	3.35E-24±6.82E-24-	**4.48E-36±2.40E-35+**	8.23E-05±4.36E-04-	5.29E-14±2.65E-14-
f_4	**5.18E-25±9.15E-25+**	2.68E+00±1.43E+00-	2.01E-08±3.08E-08-	3.22E-09±5.95E-09-
f_5	6.64E-01±1.49E+00-	3.99E-01±1.20E+00-	3.25E+00±1.25E+01-	**2.16E-12±3.68E-14+**
f_6	**0.00E-00±0.00E-00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E-00±0.00E-00≈**
f_7	1.20E-03±3.52E-04-	**8.88E-04±3.37E-04+**	2.37E-03±6.43E-04-	2.71E-03±7.64E-04-
f_8	7.30E+03±2.94E+02-	**3.82E-04±0.00E+00≈**	4.88E+02±2.89E+02-	**7.39E-04±8.47E-05≈**
f_9	1.01E+02±5.12E+01-	**0.00E+00±0.00E+00≈**	6.27E+00±2.33E+00-	**0.00E+00±0.00E+00≈**
f_{10}	**3.67E-15±6.38E-16≈**	**4.86E-15±1.71E-15≈**	7.22E-15±1.45E-15-	**4.78E-15±0.00E+00≈**
f_{11}	3.61E-03±5.46E-03-	7.40E-04±2.22E-03-	9.86E-04±2.51E-03-	**0.00E+00±0.00E+00+**
f_{12}	**1.57E-32±5.47E-48≈**	**1.57E-32±2.01E-35≈**	3.46E-03±1.86E-02-	**1.58E-32±8.75E-48≈**
f_{13}	3.66E-04±1.97E-03-	3.66E-04±1.97E-03-	**1.37E-32±8.85E-34≈**	**1.35E-32±5.27E-48≈**
f_{14}	3.79E-15±1.42E-14-	5.68E-15±1.71E-14-	5.31E-14±1.42E-14-	**0.00E+00±0.00E+00+**
f_{15}	**4.74E-14±2.1+**	1.74E-12±4.60E-12-	1.01E-04±3.78E-04-	3.55E-13±7.93E-14-
f_{16}	**5.80E+04±3.44E+04+**	1.84E+06±4.73E+06-	2.64E+05±1.67E+05-	1.65E+05±1.08E+05-
f_{17}	**6.44E-14±3.51E-14+**	2.52E+01±1.18E+02-	3.30E+01±3.12E+01-	2.66E-04±4.74E-04-
f_{18}	**1.06E-01±2.60E-01+**	1.98E+03±9.66E+02-	2.82E+03±7.10E+02-	5.61E+01±6.58E+01-
f_{19}	1.06E+00±1.76E+00-	7.97E-01±1.59E+00-	2.56E+00±3.87E+00-	**1.26E-01±6.82E-01+**
f_{20}	6.90E-03±8.65E-03-	1.32E-02±1.13E-02-	1.84E-02±1.40E-02-	**2.79E-03±5.14E-03+**
f_{21}	**2.09E+01±4.61E-02≈**	**2.09E+01±6.84E-02≈**	**2.10E+01±4.23E-02≈**	**2.09E+01±5.20E–02≈**
f_{22}	5.91E+01±5.03E+01-	**0.00E+00±0.00E+00+**	7.89E+00±3.03E+00-	2.73E-15±2.20E-15-
f_{23}	1.67E+02±9.65E+00-	**4.95E+01±1.06E+01≈**	**6.21E+01±1.36E+01≈**	1.00E+02±1.30E+00-
f_{24}	**3.97E+01±1.13E+00≈**	**2.73E+01±1.92E+00≈**	**2.52E+01±3.13E+00≈**	**2.36E+01±2.94E+00≈**
f_{25}	**1.98E+03±3.01E+03≈**	2.18E+04±6.19E+03-	**3.03E+03±3.84E+03≈**	**5.58E+03±2.24E+04≈**
f_{26}	1.29E+01±2.05E-01-	**1.92E+00±1.63E-01≈**	**2.39E+00±6.16E-01≈**	**3.77E+00±2.76E-01≈**
f_{27}	**1.31E+01±2.05E-01≈**	**1.28E+01±2.63E-01≈**	**1.28E+01±3.86E-01≈**	**1.17E+01±5.61E-01≈**
-	15	13	19	10
+	5	4	0	6
≈	7	10	8	11

TABLE V: Experimental results of SaDE, ODE, OXDE, and IDE for all test functions at D = 30.

F	SaDE Ave Err ± Std Dev	ODE Ave Err ± Std Dev	OXDE Ave Err ± Std Dev	IDE Ave Err ± Std Dev
f_1	**1.22E-130±4.14E-130+**	9.94E-58±3.19E-57-	2.66E-59±5.94E-59-	6.47E-95±1.93E-95-
f_2	2.97E-79±5.46E-79-	5.86E-18±4.68E-18-	2.96E-33±2.33E-33-	**3.42E-84±2.76E-83+**
f_3	1.14E-06±2.93E-06-	3.06E-05±3.35E-05-	2.38E-05±2.33E-05-	**5.29E-14±2.65E-14+**
f_4	5.59E-07±3.01E-06-	1.98E-03 ±1.07E-02-	7.44E+00±3.32E+00-	**3.22E-09±5.95E-09+**
f_5	2.89E+01±2.34E+01-	2.55E+01±8.29E-01-	1.20E+00±1.83E+00-	**2.16E-12±3.68E-14+**
f_6	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E-00±0.00E-00≈**
f_7	**2.77-03±1.19E-03≈**	9.20E-04±3.21E-04-	4.08E-03±1.94E-03-	**2.71E-03±7.64E-04≈**
f_8	**3.82E-04±0.00E+00≈**	6.94E+03±3.85E+02-	**3.82E-04±0.00E+00≈**	7.39E-04±8.47E-05-
f_9	3.32E-02±1.79E-01-	3.36E+01±2.25E+01-	9.32E+00±3.02E+00-	**0.00E+00±0.00E+00+**
f_{10}	**9.31E-02±2.79E-01≈**	**3.55E-15±0.00E+00≈**	3.10E-02±1.67E-01-	**4.78E-15±0.00E+00≈**
f_{11}	3.36E-03±9.03E-03-	7.39E-04±2.78E-03-	2.46E-03±4.17E-03-	**0.00E+00±0.00E+00+**
f_{12}	1.04E-02±3.11E-02-	**1.58E-32±2.52E-34≈**	**1.57E-32±2.32E-34≈**	**1.58E-32±8.75E-48≈**
f_{13}	1.83E-03±8.07E-03-	**1.35E-32±5.47E-48≈**	**1.56E-32±5.93E-33≈**	**1.35E-32±5.27E-48≈**
f_{14}	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**	**0.00E+00±0.00E+00≈**
f_{15}	8.07E-06-±1.69E-05-	3.33E-04±3.31E-04-	5.66E-05±5.70E-05-	**3.55E-13±7.93E-14+**
f_{16}	4.86E+05±1.85E+05-	5.98E+05±3.67E+05-	4.78E+05±2.21E+05-	**1.65E+05±1.08E+05+**
f_{17}	1.14E+02±1.49E+02-	2.08E-01±2.55E-01-	1.26E+00±1.10E+00-	**2.66E-04±4.74E-04+**
f_{18}	3.30E+03 ±5.49e+02-	1.45E+02±8.04E+01-	**2.14E+01±5.84E+01+**	5.61E+01±6.58E+01-
f_{19}	4.64E+01±3.23E+01-	5.38E+01±3.22E+01-	6.64E-01±1.49+00-	**1.26E-01±6.82E-01+**
f_{20}	2.61E-02±2.83E-02-	6.57E-03±9.24E-03-	1.34E-02±9.54E-03-	**2.79E-03±5.14E-03+**
f_{21}	**2.09E+01±4.08E-02≈**	**2.10E+01±5.75-02≈**	**2.09E+01±4.17E-02≈**	**2.09E+01±5.20E–02≈**
f_{22}	1.66E-01±3.71E-01-	7.54E+01±2.89E+01-	1.39E+01±3.84E+00-	**2.73E-15±2.20E-15+**
f_{23}	4.89E+01±1.02E+01-	**6.57E-03±9.39E+00+**	3.73E+01±3.24E+01-	1.00E+02±1.30E+00-
f_{24}	1.70E+01±3.14E+00-	**8.99E+00±9.39E+00+**	3.73E+01±8.18E+00-	2.36E+01±2.94E+00-
f_{25}	**3.92E+03±2.81E+03≈**	**2.02E+03±2.35E+03≈**	3.34E+03±5.14E+03-	5.58E+03±2.24E+04-
f_{26}	3.92E+00±3.95E-01-	7.58E+00±2.15E+00-	**2.08E+00±6.28E-01+**	3.77E+00±2.76E-01-
f_{27}	**1.26E+01±2.69E-01≈**	**1.31E+01±2.57E-01≈**	**1.33E+01±1.72E-01≈**	**1.17E+01±5.61E-01≈**
-	18	17	18	7
+	1	2	2	12
≈	8	8	7	8

B. Benchmark results

The results of DE/rand/1, DE/best/1, and IDE at D=30 for the test suite are listed in Table II, where "Ave Err" and "Std Dev" indicated the mean and standard deviation of the function error values obtained in 30 runs, respectively. The best results are shown in boldface. The Wilcoxon's runk sum test results amoung IDE and others are summaried at the bootom of the table, in which "-", "+", and "≈" indicate that the performance of the compared algorithm is worse than, better than, and similar to that of IDE, respectively.

Based on the results, IDE achieves better results than DE/rand/1 and DE/best/1 on the majority of test functions. Compared with DE/rand/1, IDE is significantly better than it on 18 out of 27 test functions, and similar to it on 8 test functions. DE/rand/1 beat IDE on only 1 test function. Compared with DE/best/1, IDE is also significantly better DE/best/1 on 19 out of 27 test functions, and similiar to it on 3 test functions. DE/best/1 beat IDE only on 3 test functions. Comparing the three algorithms together, IDE beat both of two algorithms at the same time on 12 out of 27 test functions and similiar with one of other two algorithms on 11 test functions.

We alse present scalable tests of DE/rand/1, DE/best/1, and IDE on the test suites for D=50 and D=100. The mean errors and comparison results between IDE and the others based on Wilcoxon's rank sum test are listed in Table III. IDE outperforms DE/rand/1 on 19 and 21 out of 27 test functions for D=50 and 100, while IDE beats DE/best/1 on 18 and 18 out of 27 test functions for D=50 and 100. When the problem dimension increases from 50 to 100, the performance of IDE has not affected by the increasing of dimension.

We also compare the IDE with six other state-of-art DE variants, including DEGL/SAW [32], DPSDE [33], MGBDE [34], OXDE [35], SaDE [36], and ODE [37]. In EPSDE and SaDE, the parameter F and CR are free to set and self-adaptive. To have a fair comarison, we set the maximum number of functions in all of these algorithms to 10,000 $\times$ D, and use the same parameter settings as described in their original literatures for these six cometitors as following:

(1) MGBDE: NP =100, F =0.5, CR =0.9

(2) OXDE: NP =D, F =0.9, CR =0.9

(3) SaDE: $NP = 50, LP = 50$

(4) ODE: $NP = 100, F = 0.5, CR = 0.9, Jr = 0.3$

(5) EPSDE: $NP = 50$

(6) DEGL/SAW: $NP = 10 \times D$, $\alpha = \beta = F$=0.8, $CR =$ 0.9

The results of the seven DE varians on each test function on D=30 are summarized in Table IV, and V. The results indicates that IDE is significantly better than DEGL/SAW, EPSDE, MGBDE, SaDE, ODE, and OXDE on 15,13, 19, 18, 17, 18 respectively, but they beat IDE on only 5, 4, 0, 1, 2, 2, respectively. It is obvious that the IDE is the best one among the seven methods on the test suite.

V. EVALUATION IDE-I FOR MT DATA INVERSION

To test the effectiveness of the IDE-I algorithm presented above, we have implemented it using the C++ programming language. We conducted theoretical models computation on a PC for two-layer (type G), three-layer (type K), four-layer (type HK) and five-layer (type HKH) models, respectively, and compared the results with previous works.

The platform for numerical experiment is a PC equipped Intel i5 420 processor and 4G memory with the Windows 7 operation system.

TABLE VI: Comparison of inversion results by IDE-I and other algorithms (noise free) on the two-layer model.

algorithm	$\rho_1(\Omega m)$	$\rho_2(\Omega m)$	$h_1(m)$	CPU time(s)
Real model	10	100	600	–
MC[13]	10.5177	100.2022	632.4447	37.4375
SA[13]	10.7115	100.4541	643.3216	36.3297
DPSO[13]	10.0000	99.9999	600.0000	8.5087
APSO-I[14]	10.0000	99.9999	600.0000	7.3945
IDE-I	10.0000	100.0000	600.0000	5.421

TABLE VII: Comparison of anti-noise capabilities between IDE-I, DPSO and APSO-I algorithms on the two-layer geo-electrical model.

		$\rho_1(\Omega m)$	$\rho_2(\Omega m)$	$h_1(m)$	NRE (%)
	Real model	10	100	600	–
	DPSO[13]	9.79	99.65	559.69	7.05
10% noise	APSO-I[14]	9.93	103.09	592.28	3.34
	IDE-I	9.95	102.21	595.47	2.35
	DPSO[13]	10.54	108.79	728.01	20.02
20% noise	APSO-I[14]	9.83	102.76	587.87	3.82
	IDE-I	9.88	101.54	590.36	2.53

A. Two-layer (type G) geo-electrical model

1) Comparison of various algorithms: In the noise-free case, we uses the IDE-I algorithm to invert the MT data of a two-layer geo-electrical model, and compared its result with those of Monte Carlo(MC), simulated annealing(SA), genetic algorithm(GA), damping PSO(DPSO)[13], and Adaptive PSO Inversion(APSO-I)[14] algorithm, in order to evaluate the performance of IDE-I in the issues of accuracy and computation time.

We took 50 individuals for inversion with a maximum iteration number 1000 and crossover factor $CR = 0.3$. In inversion, the value ranges taken for ρ_1, ρ_2 and h_1 are $1\sim50\Omega m$, $10\sim500\Omega m$ and 100∼1000m, respectively. The comparison is shown in Table VI.

Table VI indicates that the inversion accuracy of the IDE-I is much better than those of MC and SA algorithms, and slightly superior to the DPSO and APSO-I algorithm. On the issue of computation time, the IDE-I is remarkably faster than the MC and SA, and a little faster than those of the DPSO and APSO-I algorithm.

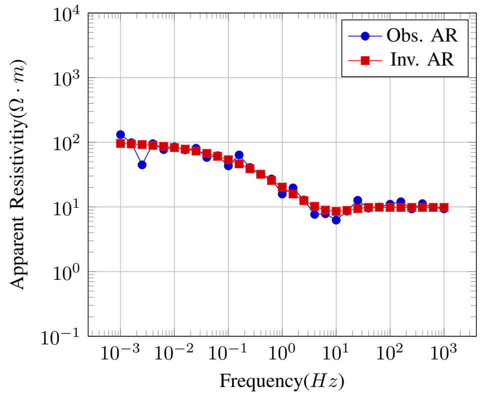
2) Analysis of anti-noise capability: To simulate real MT data, we invert the data added 10% and 20% Gauss random noise, respectively.

For the two vectors, $\mathbf{x} = [x_1, x_2, ..., x_n]$ and $\mathbf{y} = [y_1, y_2, ..., y_n]$, the Norm of Reletive Error (NRE) is used to evaluate the inversion accuracy, which is defined as following:

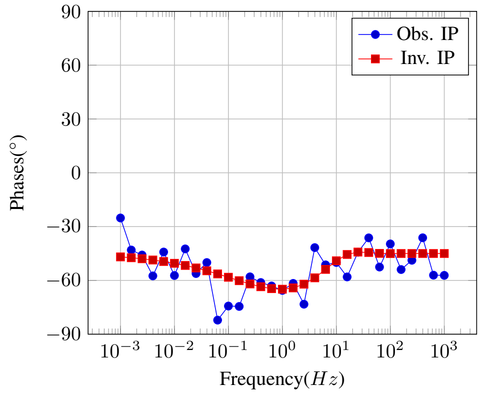
$$NRE(\mathbf{x},\mathbf{y}) = (x_1 - y_1)/x_1)^2 + (x_2 - y_2)/x_2)^2 + \\ ... + (x_n - y_n)/x_n)^2 \quad (17)$$

We analyze the anti-noise capability of the IDE-I algorithm with same model parameters, and compare it with that of the DPSO, and APSO-I (Table VII), where the norm of relative error(NRE) is that of inversion model parameters and real model parameters.

As shown in Table VII, after the Gauss noise of 10% and 20% levels is added to the data, the inversion result of IDE-I is better than the DPSO and APSO-I, implying its stronger anti-noise capability than the others.



(a) Apparent Resistivity inversion result



(b) Phase inversion result

Fig. 5: Two-layer-model synthetic observation data with 20% Gause noise (blue) and its inversion result (red) using IDE-I.

Fig.5 displays the two-layer-model synthetic observation data with 20% Gause noise (apparent resistivity and impedance phase) and its inversion result. The figure exhibits good fitting of synthetic data and inversion result.

B. Three-layer (type K) geo-electrical model

IDE-I inversion is employed to the noise-free MT data as well as the data added by 10% and 20% Gauss random noise, with 50 individuals, maximum iteration number 1000 and crossover factor $CR = 0.3$. In inversion, the value ranges taken are $\rho_1 = 1\sim 100\Omega m$, $\rho_2 = 10 \sim 500\Omega m$, $\rho_3 = 1 \sim 50\Omega m$, $h_1 = 100 \sim 1000m$, and $h_2 = 1000 \sim 10000m$, respectively. The inversion result is compared with that of APSO-I [14] shown in Table VIII.

The data listed in Table III demonstrate that in the case of no noise, the inversion result of IDE-I is as good as that of APSO-I. When after 10% and 20% Gauss noise is added to the data, The inversion result of IDE-I is better than that of APSO-I. It indicates the good anti-noise ability of IDE-I.

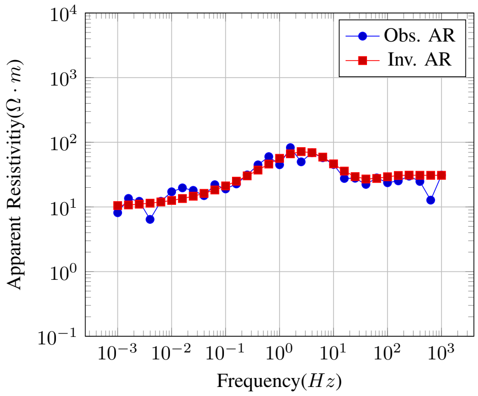
The Fig.6 shows apparent resistivity and phase curves from IDE-I inversion on the three-layer model after adding 20% Gauss noise, exhibiting a good fit between the inverted and synthetic observed data.

C. Four-layer (type HK) geo-electrical model

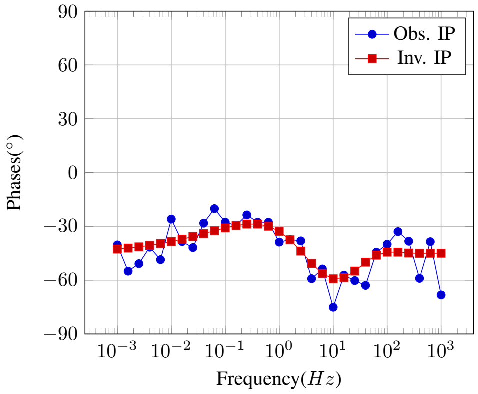
IDE-I algorithm is employed to the noise-free MT data of the four-layer (type KH) model, as well as the data added

TABLE VIII: Comparison of anti-noise performance between IDE-I, DPSO and APSO-I algorithms on the three-layer model.

		$\rho_1(\Omega m)$	$\rho_2(\Omega m)$	$\rho_3(\Omega m)$	$h_1(m)$	$h_2(m)$	NRE(%)
	Real model	30	200	10	500	2000	–
0% noise	APSO-I[14]	30.00	200.00	10.00	500.00	2000.00	0.00
	IDE-I	30.00	200.00	10.00	500.00	2000.00	0.00
10% noise	APSO-I[14]	28.94	170.59	10.29	476.72	2047.50	16.26
	IDE-I	30.45	181.72	9.86	481.17	2020.37	10.15
20% noise	APSO-I [14]	30.86	248.71	9.64	493.85	1860.07	25.78
	IDE-I	30.42	231.68	9.73	495.28	1898.35	16.94

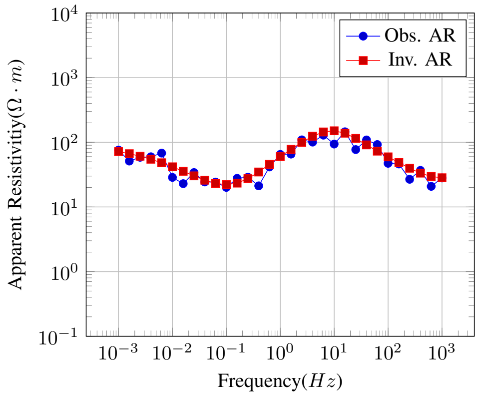


(a) Apparent Resistivity inversion result

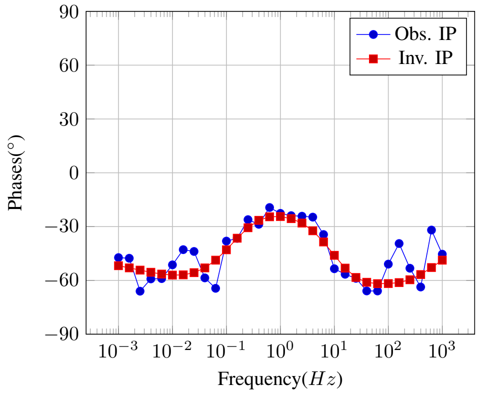


(b) Phase inversion result

Fig. 6: Three-layer-model synthetic observation data with 20% Gause noise (blue) and its inversion result (red) using IDE-I.



(a) Apparent Resistivity inversion result



(b) Phase inversion result

Fig. 7: Four-layer-model synthetic observation data with 20% Gause noise (blue) and its inversion result (red) using IDE-I.

by 10% and 20% Gauss random noise. The result of IDE-I, DPSO[13], and APSO-I [14] are listed in TableIX. In inversion, 50 individuals, maximum iteration number 1000 and crossover factor $CR = 0.3$ are adopted, and the value ranges are $\rho_1 = 1 \sim 50\Omega m$, $\rho_2 = 10 \sim 500\Omega m$, $\rho_3 = 1 \sim 50\Omega m$, $\rho_4 = 10 \sim 500\Omega m$, $h_1 = 10 \sim 50m$, $h_2 = 100 \sim 4000m$, $h_3 = 1000 \sim 10000m$, respectively. In these ranges, 50 initial individuals are generated randomly.

The data listed in table IX indicates that in the case of no noise, IDE-I is superior to DPSO and same as APSO-I. After 10% and 20% Gauss noise is added to the data, the NREs of IDE-I are all less than those of APSO-I, which indicate IDE-I has a better anti-noise ability than APSO-I.

Fig.7 shows apparent resistivity and phase curves from IDE-I inversion on four-layer-model after adding 20% Gauss noise, which demonstrates good fitting between inverted and synthetic observed data.

D. Five-layer (type HKH) geo-electrical model

Inversion using IDE-I is made on a five-layer (HKH) geo-electrical model and compared with the inversion results by other algorithms[14] (Table X). For this inversion, the following parameters are adopted: 80 individuals, maximum iteration number 2000, crossover factor $CR = 0.3$, $\rho_1 = 1 \sim 100\Omega m$, $\rho_2 = 1 \sim 10\Omega m$, $\rho_3 = 1 \sim 100\Omega m$, $\rho_4 = 1 \sim 10\Omega m$, $\rho_5 = 1 \sim 100\Omega m$, $h_1 = 100 \sim 4000m$, $h_2 = 100 \sim 4000m$, $h_3 = 1000 \sim 10000m$, $h_4 = 100 \sim 4000m$.

The data in table X indicate that the inversion result of IDE-I is considerably superior to that of the generalized inverse and Bostick algorithms, and slightly better than HGA and APSO-I.

In order to evalute the anty-noise performance between APSO-I and IDE-I, they are employed to invert the noise-free MT data of five-layer model, as wellas the data added by 10% and 20% Gauss random noise, respectively. The results

TABLE IX: Comparison of inversion results by IDE-I, DPSO and APSO-I algorithms on the four-layer model (type HK) with different Gauss noise levels.

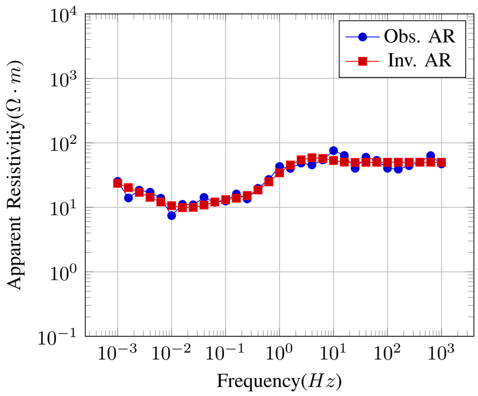
		$\rho_1(\Omega m)$	$\rho_2(\Omega m)$	$\rho_3(\Omega m)$	$\rho_4(\Omega m)$	$h_1(m)$	$h_2(m)$	$h_3(m)$	NRE (%)
	Real model	30	200	10	100	100	2000	3000	–
0% noise	DPSO-I[13]	30.53	198.07	8.38	103.70	106.80	2072.00	2580.10	2.58
	APSO-I[14]	30.00	200.01	9.99	99.99	100.01	1999.91	2999.64	0.10
	IDE-I	30.00	200.00	9.99	100.01	100.00	1999.98	3000.24	0.10
10% noise	APSO-I[14]	30.35	213.95	8.89	99.46	112.26	1992.73	2547.04	23.49
	IDE-I	30.24	210.28	8.92	99.67	109.83	1994.63	2667.13	19.07
20% noise	APSO-I[14]	31.87	216.53	11.64	96.80	103.92	1902.67	3393.70	24.45
	IDE-I	30.30	212.41	11.44	97.92	102.40	1930.17	3347.25	20.08

TABLE X: Comparison of inversion results by IDE-I and other algorithms (noise free) on the five-layer model

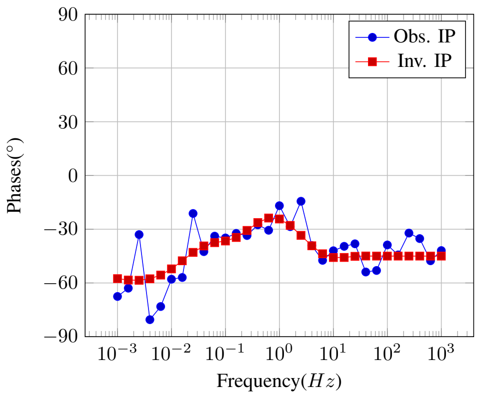
	$\rho_1(\Omega m)$	$\rho_2(\Omega m)$	$\rho_3(\Omega m)$	$\rho_4(\Omega m)$	$\rho_5(\Omega m)$	$h_1(m)$	$h_2(m)$	$h_3(m)$	$h_4(m)$	NRE (%)
Real model	50	3	50	3	50	2000	1000	4000	2000	–
Generalized inversion[14]	50.0	3.0	56.1	3.2	50.1	2000	1000	3900	2150	15.99
Bostick inversion[14]	52.0	7.0	10.0	9.0	59.0	1900	770	2830	6400	338.12
HGA[14]	50.00	2.99	47.80	2.94	49.99	2000.30	994.27	4033.04	1955.76	5.42
APSO-I[14]	50.00	3.00	49.77	2.98	49.99	2000.01	999.60	4006.40	1987.56	1.03
IDE-I	50.00	3.00	49.95	2.98	49.99	2000.00	999.98	4003.37	1992.61	0.77

TABLE XI: Comparison the anti-noise performance between APSO-I and IDE-I on the five-layer model

		$\rho_1(\Omega m)$	$\rho_2(\Omega m)$	$\rho_3(\Omega m)$	$\rho_4(\Omega m)$	$\rho_5(\Omega m)$	$h_1(m)$	$h_2(m)$	$h_3(m)$	$h_4(m)$	NRE (%)
	Real model	50	3	50	3	50	2000	1000	4000	2000	–
0% noise	APSO-I	50.00	3.00	49.77	2.98	49.99	2000.01	999.60	4006.40	1987.56	1.03
	IDE-I	50.00	3.00	49.95	2.98	49.99	2000.00	999.98	4003.37	1992.61	0.77
10% noise	APSO-I	45.82	2.66	44.28	2.44	45.13	1839.71	976.29	4489.4	1727.56	34.32
	IDE-I	47.18	2.74	47.33	2.68	45.62	1950.12	987.45	4242.79	1946.71	19.43
20% noise	APSO-I	43.39	2.51	42.99	2.38	44.77	1728.32	926.48	4845.32	1628.59	46.95
	IDE-I	46.48	2.69	46.77	2.64	46.92	1953.47	982.46	4248.58	1940.05	20.88



(a) Apparent Resistivity inversion result



(b) Phase inversion result

Fig. 8: Five-layer-model synthetic observation data with 20% Gause noise (blue) and its inversion result (red) using IDE-I.

are listed in Table XI.

The Table XI indicates that in the case of noise-free, IDE-I is superior to APSO-I slightly. However, after 10% and 20% Gauss random noised is added, the NREs of IDE-I is much better than those of APSO-I, which indicate that IDE-I has a better anti-noise ability than APSO-I.

Fig.8 shows apparent resistivity and phase curves from IDE-I inversion on five-layer-model after adding 20% Gauss noise, which demonstrates good fitting between inverted and synthetic observed data.

VI. Conclusions

We have described the reason that the previous differential evolution (DE) algorithm is prone to falling into local optimum. After introduce improvement strategy to improve the classical DE, we presented an improved differential evolution inversion (IDE-I) algorithm for MT data. We evaluate the performance of IDE, and apply IDE-I algorithm to MT data inversion on 1D layered models, and analyze its inversion accuracy and anti-noise capability.

Numerical experiments show that the IDE algorithm is better than classic DE (DE/rand/i, and DE/test/1) and state-of-art variant DEs. The IDE-I algorithm does not rely on any initial models, can reach global optimum very well, costs short time for computation, and can resist noise effectively. The comparison results demonstrate that it is superior to other algorithms in inversion accuracy and anti-noise capability.

In this work, numerical experiments are conducted only on 1D layered models with $2 \sim 5$ layers to inverse $3 \sim 9$ model parameters. When this algorithm is applied to MT data inversion on models of 2D or 3D, some problems will be encountered, such as large computation time, much more difficult to find a global optimum and so on. Further studies will focus on how to solve these issues.

REFERENCES

[1] M. J. Comeau, M. J. Unsworth, F. Ticona, et al., "Magnetotelluric images of magma distribution beneath Volcan Uturuncu, Bolivia: Implications for magma dynamics," *Geology*, vol. 43, no. 3, pp. 243-246, MAR. 2015.

[2] C. Li, D. Bai, S. Xue, et al., "A magnetotelluric study of the deep electric structure beneath the Ordos Block," *CHINESE JOURNAL OF GEOPHYSICS*, vol. 60, no. 5, pp. 1788-1799, MAY. 2017.

[3] M. M. Sarvandani, A. N. Kalateh, M. Unsworth, et al., "Interpretation of magnetotelluric data from the Gachsaran oil field using sharp boundary inversion," *Journal of Petroleum Science and Engineering*, vol. 149, pp. 25-39, JAN. 2017.

[4] S. Constable, S. Constable, "Satellite magnetic field measurements: Applications in studying the deep earth," *The State of the Planet: Frontiers and Challenges in Geophysics*, editor R. Sparks and C. Hawkesworth, pp.147-159, AGU, Washington D.C., 2004.

[5] S. Constable, "Geomagnetism," *Treatise on Geophysics*, editor G. Schubert, M. Kono, Vol. 5, pp.147-159, Elsevier, 2007.

[6] G. Nieuwenhuis, "Magnetotelluric Imaging of Precambrian Lithosphere Beneath Southern Alberta," *Master degree thesis*, Department of Physics, University of Alberta, Canada, 2011.

[7] Q. Tang, R. Hobbs, C. Zheng, et al., "Markov Chain Monte Carlo inversion of temperature and salinity structure of an internal solitary wave packet from marine seismic data," *Journal of Geophysical Research-Ocean*, vol. 121, no. 6, pp. 3692-3709, JUN. 2016.

[8] M. K. Sen, and R. Biswas, "Transdimensional seismic inversion using the reversible jump Hamiltonian Monte Carlo algorithm," *Geophysics* , vol. 82, no. 3, pp. R119-R134, MAY. 2017.

[9] L. T. Nguyen, T. Nestorovic, " Unscented hybrid simulated annealing for fast inversion of tunnel seismic waves," *Computer Methods in Applied Mechanics and Engineering*, vol. 30, pp. 281-299, APR. 2016.

[10] Y. Lu, S. Peng, W. Du, et al., "Rayleigh wave inversion using heat-bath simulated annealing algorithm," *Journal of Applied Geophysics*, vol. 134, pp. 267-280, NOV. 2016.

[11] C. Sun, L. Li, W. Huang, et al., "1D inversion of CSAMT data based on adaptive genetic algorithm," *Oil Geophysical Prospecting*, vol.52, no. 2, pp. 392-397, APR. 2017.

[12] P. Thakur, D. C. Srivastava, P. K. Gupta, et al., " The genetic algorithm: A robust method for stress inversion," *Journal of Structural Geology*, vol. 94, pp. 227-239, JUL. 2017.

[13] X. Shi, M. Xiao, J. Fan, et al., "The damped PSO algorithm and its application for magnetotelluric sounding data inversion," *Chinese Journal of Geophysics*, vol. 52, no. 4, pp. 1114-1120, APR. 2009.

[14] J. Xiong, C. Liu, T. Zhang, et al., "A non-linear inversion algorithm based on adaptive PSO for the MT oil-gas exploration data," *Energy Education Science and Technology Part A. Energy Science and Research*, vol. 32, no. 1, pp. 473-480, JAN. 2014.

[15] J. Xiong, T. Zhang, " Multiobjective particle swarm inversion algorithm for two-dimensional magnetic data," *Applied Geophysics*, vol. 12, no. 2, pp. 127-136, JUN. 2015.

[16] R. Storn, K. Price, " Differential Evolution- A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, DEC. 1997.

[17] J. Wang, J. Song, "Application and performance comparison of biogeography-based optimization algorithm on unconstrained function optimization problem," *IAENG International Journal of Applied Mathematics*, vol. 47, no. 1, pp. 82-88, 2017.

[18] M. Basu, "Improved differential evolution for short-term hydrothermal scheduling," *Electrical Power and Energy Systems*, vol. 58, pp. 91-100, JUN. 2014.

[19] N. Ferrante, V. Tirronen, " Recent advances in differential evolution: a survey and experimental analysis," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 61-106, FEB. 2010.

[20] S. Das, S. S. Mullick, " Recent advances in differential evolution- An updated survey," *Swarm and Evolutionary Computation*, vol. 27, pp. 1-30, APR. 2016.

[21] L. Chuang, Y. Chiang, C. Yang, " Differential evolution with application to operon prediction," *IAENG International Journal of Computer Science*, vol. 40, no.3, pp. 200-206, 2013.

[22] F. Nie, P. Zhang, " Fuzzy partition and correlation for image segmentation with differential evolution," *IAENG International Journal of Computer Science*, vol. 40, no.3, pp. 164-172, 2013.

[23] C. Balkaya, Y. L. Ekinci, G. Gokturkler, et al., "Recent advances in differential evolution- An updated survey," *Journal of Applied Geophysics*, vol. 136, pp. 372-386, JAN. 2017.

[24] C. Balkaya, " An implementation of differential evolution algorithm for inversion of geoelectrical data," *Journal of Applied Geophysics*, vol. 98, pp. 160-175, NOV. 2013.

[25] Z. Pan, ,J. Wu, Z. Gao, et al., "Adaptive Differential Evolution by Adjusting Subcomponent Crossover Rate for High-Dimensional Waveform Inversion," *IEEE Transaction on Geoscience and Remote Sensing Letters*, vol. 12, no. 6, pp. 1327-1331, JUN. 2015.

[26] Z. Gao,Z. Pan, J. Gao, "Multimutation Differential Evolution Algorithm and Its Application to Seismic Inversion," *IEEE Transaction on Geoscience and Remote Sensing*, vol. 54, no. 6, pp. 3626-3636, JUN. 2016.

[27] J Zhang, H. Zhang, Y. Zhang, et al., "Calibrating one-dimensional velocity model for downhole microseismic monitoring using station-pair differential arrival times based on the differential evolution method," *Physics of the earth and planetary interiors*, vol. 261, pp. 124-132, DEC. 2016.

[28] G. D. Tollo, F. Lardeux, J. Maturana, et al., "An experimental study of adapive control for evolutionary algorithm," *Applied Soft Computing*, vol. 35, pp. 359-372, OCT. 2015.

[29] H. Peng, Z. Guo, C. Deng, et al., "Enhancing differential evolution with random neighbors based strategy." *Journal of Computational Science.*,Available online 14 July 2017, In Press.

[30] X. Yao, Y. Liu, G. Lin, "Evolutionary programming made faster," *IEEE Trans, Evol. Comput.*, vol. 3, no. 2, pp. 82-102, 1999.

[31] P.N. Suganthan, N. Hansen, J.J. Liang, et al., "Problem Definitions and Evaluation Criteria for the CEC 2005 Special Session on Real-Parameter Optimization," *KanGAL Report 2005005*, 2005.

[32] S. Das, A. Abraham, U.K. Chakraborty, et al., "Differential evolution using a neighborhood-based mutation operator," *IEEE Trans. Evol. Comput.*, vol. 13, no. 3, pp. 526-553, 2009.

[33] R. Mallipeddi, P.N. Suganthan, Q.K. Pan, et al., "Differential evolution algorithm with ensemble of parameters and mutation strategies," *Appl. Soft Comput.* , vol. 11, no. 2, pp. 1679-1696, 2011.

[34] H. Wang, S.Rahnamayan, H. Sun, et al., "Gaussian bare-bones differential evolution," *IEEE Trans. Cybernet*, vol. 43, no. 2, pp.634-647, 2013.

[35] Y. Wang, Z. Cai, Q. Zhang, "Enhencing the search ability of differential evolution through orthogonal crossover," *Inform. Sci.*, vol. 185, no. 1, pp. 153-177, 2012.

[36] A.K. Qin, V.L. Huang, P.N. Suanthan, "Differential evolution algothrithm with strategy adaption for global numerical optimization," *IEEE Trans. Evol. Comput.*, vol. 13, no. 2, pp. 398-417, 2009.

[37] S. Rahnamayan, H.R. Tizhoosh, M.M. Salama, "Opposition-based differential evolution," *IEEE Trans. Evol. Comput.*, vol. 12, no. 1, pp. 64-79, 2008.

Jie Xiong received his PhD in Geophysics and Information Technology from the China University of Geosciences (Beijing) in 2012. He was a Visiting Professor at Department of Computer Science and Engineering, Mississippi State University on 2016. He is currently an Associate Professor at School of Electrical and Information, Yangtze University. His research interests include geophysical inversion theory, computer application, and cloud computing.