Regional Seismic Waveform Inversion Using Swarm Intelligence Algorithms

Ke Ding, Yanyang Chen, Yanbin Wang and Ying Tan

***Abstract*—Inversion is a critical and challenging task in geophysical research. Geophysical inversion can be formulated as an optimization problem to find the best parameters whose forward synthesis data most fit the observed data. The inverse problems are usually highly non-linear, multi-modal as well as ill-posed, so conventional optimization algorithms cannot handle it very efficiently. In the past decades, genetic algorithm (GA) and its many variants are widely applied to inverse problems and achieve great success. Swarm intelligence algorithms are a family of global optimizers inspired by swarm phenomena in nature, and have shown better performance than GA for diverse optimization problems. However, swarm intelligence algorithms are not utilized for geophysical inversion problems until recently and only limited number of works are reported. In this paper, we try to apply two swarm intelligence algorithms, Particle Swarm Optimization (PSO) and Fireworks Algorithm (FWA), to the regional seismic waveform inversion. To explore the advantages and disadvantages of swarm intelligence algorithms over GA, synthetic experiments are conducted by using these two swarm intelligence algorithm and several GA variants as well as Differential Evolution (DE). The experimental results show that, both swarm intelligence algorithms outperform the widely used GA, DE, and the models estimated by swarm intelligence algorithms are closer to the true solution. The promising results imply that swarm intelligence algorithms are a potentially more powerful tool for inversion problems.**

I. Introduction

An integral part of geophysics is to infer the inferior of the earth based on the observational data. Most often than not, the collected data are not directly related to the subsurface characterization, so an inverse problem must be solved to estimate these properties of interest. In general, the aim of an inverse problem is to find an parameterized model that is consistent with the observed data. It is usually assumed that the forward problem is well understood so that reasonably accurate simulated data can be calculated for an arbitrary model. The objective function (alternatively called misfit or fitness function) is typically some measure of the difference between observational data and synthetic data calculated for a trail model.

Ke Ding and Ying Tan (corresponding author) are with the Department of Machine Intelligence, School of Electronics Engineering and Computer Science, Peking University and Key Laboratory of Machine Perception (Ministry of Education), Peking University, Beijing, 100871, P.R. China. (Email: {keding,ytan}@pku.edu.cn).

Yanyang Chen and Yanbing Wang are with the Department of Geophysics, School of Earth and Space Sciences, Peking University, Beijing, 100871, P.R. China (E-mail: {chenyanyang,ybwang}@pku.edu.cn).

This work was supported by the Natural Science Foundation of China (NSFC) under grant no. 61375119, 61170057, 60875080 and 41174034, and partially supported by National Key Basic Research Development Plan (973 Plan) Project of China with grant no. 2015CB352300.

Most practical geophysical inverse problems are highly nonlinear and multi-modal. Conventional local optimizers such as gradient descent method, quasi-Newton method are sensitive to the initial model and prone to get stuck to local minima.

Compared to the conventional derivative-based methods, Genetic Algorithm (GA) works better with the ability to overcome the locally optimal solutions and have the advantages of being fit for parallel computation, independent of an initial model, stable to noise, strongly robust, easily constrained, and so on [1]. Thus, GA and its variants have been widely applied to tackle inverse problems [2], [3].

Swarm intelligence algorithms are a cluster of population-based meta-heuristic stochastic algorithms for optimization. They are based on the study of collective behaviors in decentralized, self-organized systems [4]. The collective system is capable of performing complex tasks in a dynamic environment without external guidance and central coordination. Swarm intelligence algorithms have been proven very effective for solving complicatedly non-linear and non-differentiable optimization problems.

Though swarm intelligence algorithms are similar to the GA in the sense that these two heuristics are population-based search methods, swarm intelligence algorithms emphasize the cooperation among agents instead of competition. These two school of algorithms show very different trajectory with respect to optimization. Swarm intelligence algorithms turn out to be more computationally efficient (consume less function evolutions to achieve solutions of the comparable quality) for many real world applications [5], [6]. In spite of the their efficiency, swarm intelligence algorithms are far less well-known to the geophysical inverse problems and only few trials are reported [7], [8], [9].

In this work, we apply two swarm intelligence algorithms, Particle Swarm Optimization (PSO) and Fireworks Algorithm (FWA), to the regional seismic waveform inversion problem. To our best knowledge, it is the first time that swarm intelligence algorithms are used for this type of geophysical inversion. To verify the applicability and efficiency of swarm intelligence algorithms, simulation experiments are conducted and the inverted results are compared to those achieved by evolution-based algorithms such as GA, Niche GA (NGA) and Differential Evolution (DE). Experimental results show that both swarm intelligence algorithms outperform GA et al., thus make them completive tools for geophysical inversion.

The remainder of this paper is organized as follows. Section II formulates the inverse problems and give a brief overview of optimization for these problems. Section III discusses swarm

978-1-4799-7492-4/15/$31.00

intelligence algorithms, and special attention is given to PSO and FWA. The experimental setting and result analysis are presented in detail in section IV. Section V summarizes and concludes this paper.

II. Background

A. Formulation of Inverse Problems

To solve a geophysical inverse problem, the problem should be formulated properly to a optimization problem thus can be tackled by various optimizers [10].

With collected data and a proper forward model at hand, an objective function can be defined which is some measure of distance between the observational data and the simulated data.

$$f(\mathbf{d},\mathbf{m})=0. \tag{1}$$

The usual situation is that the observational data $\mathbf{d}$ represents the solution of the theoretical problem, while the model $\mathbf{m}$ represents the model witch typically a vector of parameters. In many cases, this becomes more explicit because the theory can be expressed in the reduced form.

$$\mathbf{d}=\mathbf{a}(\mathbf{m}). \tag{2}$$

By solving the forward problem, a solution $\mathbf{d}$ that represents the data can be obtained. A inverse problem arises when the data $\mathbf{d}$ is given and the task is to find a model $\mathbf{m}$ that is compatible with these data.

For most nontrivial problems, the inverse problem cannot be solved analytically to obtain a solution of the inverse problem in the form

$$\mathbf{m}=\mathbf{a^{-1}}(\mathbf{d}). \tag{3}$$

The usual situation is that only the forward problem can be solved analytically or semi-analytically [11]. The solution of the inverse problem then proceeds by solving the forward problem employing a candidate model $\mathbf{m}$ in order to obtain theoretical simulated data $\mathbf{a}(\mathbf{m})$. A comparison between the simulated data and the observed data can then be used to make improvements to the candidate model, as depicted by Fig. 1. Some measure of distance in the data space is needed which can be presented as

$$\mathbf{N}(\mathbf{d},\mathbf{m})=||\mathbf{d}-\mathbf{a}(\mathbf{m})||. \tag{4}$$

To minimizing the deviation from the model or fluctuation in the model, some regularization condition should be considered besides the distance measurement.

$$\mathbf{\Omega}(\tilde{\mathbf{d}},\mathbf{m})=\mathbf{N}(\tilde{\mathbf{d}},\mathbf{m})+\beta\cdot\mathbf{S}(\mathbf{m}). \tag{5}$$

where $\mathbf{S}$ is regularization term, β is a parameter weighting the relative importance of fitting the data and satisfying the regularization condition and $\tilde{\mathbf{d}}$ the observational data.

With the definitions above, the geophysical inverse problem can be formulated as a typical numerical optimization problem.

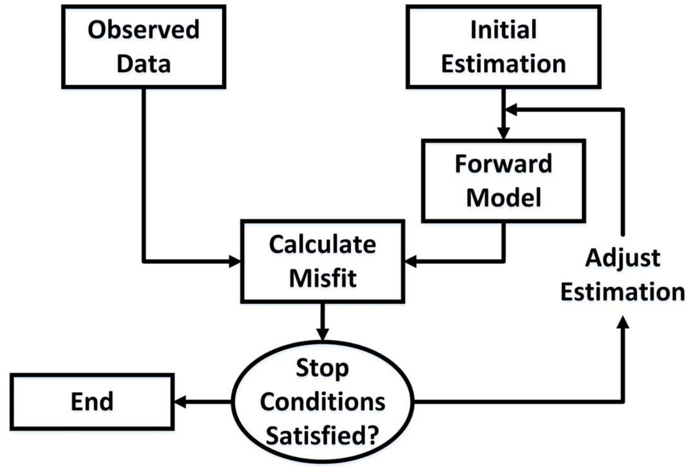

Fig. 1: In general, the target of inversion is to find a parameterized model that can fit the observed data as best

Given observational data $\tilde{\mathbf{d}}$ of finite dimension, determine a model $\mathbf{m}^*$ as the solution of

$$\begin{aligned} min \quad & \mathbf{\Omega}(\tilde{\mathbf{d}},\mathbf{m}),\\ s.t. \quad & \mathbf{c_e}(\mathbf{m})=0,\\ & \mathbf{c_i}(\mathbf{m})\geq 0. \end{aligned} \tag{6}$$

where $\mathbf{c_e}(\mathbf{m})$ and $\mathbf{c_i}(\mathbf{m})$ are constraints the model $\mathbf{m}$ should meet.

B. Global Optimization for Inverse Problems

Formulated as optimization problems, inverse problems can be tackled with standard optimizers, among which the most well studied and widely used are a family of local optimization algorithms such as gradient descent.

Local optimization algorithms typically attempt to find a local minimum in the close neighborhood of the starting solution [11]. They use local properties of the objective function to calculate an update to the current answer and search in the downhill direction. As most practical geophysical inverse problems are highly nonlinear, multi-modal and ill-posed, these methods are prone to get stuck to local minimum with poor initial models. Thus a good starting model is required to obtain an acceptable solution, which however is not piratical considering the fact that in most cases, little knowledge can be gotten about the landscape of the objective function.

A simple alternative for local algorithms is grid search, which conduct a point-to-point search in the parameter space. However, such grid search is very ineffective thus infeasible for high dimensional inverse problems.

Since 1980s, Genetic Algorithm and its many versions, as global optimization algorithms, have widely been used for various geographical inverse problems [11]. GA addresses the optimization problems as a search problem, thus derivative is needed.

The earliest application of GA may be proposed in [2], where GA with a binary string representation model parameters was utilized for inversion of plane-wave seismograms.

Detailed discussions, along with the comparison with Monte Carlo method, is presented by Sambridge [3].

To tackle inverse problems with multiple distinct solutions, Niche Genetic Algorithm (NGA) was utilized in [12] where real coding was adopted for the inversion of teleseismic body waves.

As an emerging research topic, multi-objective optimization has been proposed recently [13], which is critical for the robust estimates of some inverse problems where a single misfit function may insensitive to certain properties.

In comparison with GA, swarm intelligence algorithms are applied to geophysical inverse problem more recently [7]. For specific inversion problems, comparative results show that the time required to execute a swarm intelligence algorithm is comparable to that of GA [7] with higher convergence speed and accuracy [8]. A few inverse problems have been addressed by swarm intelligence algorithms, especially PSO [8], [9].

III. Swarm Intelligence Algorithms for Optimization

Swarm intelligence is the collective behavior of decentralized, self-organized systems. A typical swarm intelligence system consists of a population of simple agents which can communicate (either directly or indirectly) locally with each other by acting on their local environment. Though the agents in a swarm following very simple rules, the interactions between such agents can lead to the emergence of very complicated global behavior, far beyond the capability of individual agents [4]. Examples in natural systems of swarm intelligence include bird flocking, ant foraging, and fish schooling.

Inspired by swarm's such behavior, a class of algorithms are proposed for tackling optimization problems, usually under the title of swarm intelligence algorithms [14]. In swarm intelligence algorithms, a swarm is made up of multiple artificial agents. The agents can exchange heuristic information in the form of local interaction. Such interaction, in addition with certain stochastic elements, generates the behavior of adaptive search, and finally leads to global optimization.

The most respected and popular swarm intelligence algorithms are Particle Swarm Optimization (PSO) which is inspired by the social behavior of bird flocking or fish schooling [15], and Ant Colony Optimization (ACO) which simulates the foraging behavior of ant colony [16]. PSO is widely used for real-parameter optimization while ACO has been successfully applied to solve combinatorial optimization problems, the most well-known of such problems are the Traveling Salesman Problem (TSP) and Quadratic Assignment Problem (QAP). Novel swarm intelligence algorithms with particular search mechanisms have been proposed and achieved success on specific problems.

There exists a class of optimization algorithms of similar favor with swarm intelligence algorithms – evolutionary algorithms, which are inspired by nature evolution. The field mainly includes: Genetic Algorithm (GA) [17], Evolutionary Strategies (ES) [18], Differential Evolution (DE) [19] and their variants.

Algorithm 1 Particle Swarm Optimization

Initialize N particles
Calculate the fitness value of each particle
while Termination condition unsatisfied **do**
    Update global best $\hat{P}$
    Update personal best $\tilde{P}_i$
    for $i = 1$ to N **do**
        Update velocity V_i according to Eq. 7
        Update position X_i according to Eq. 8
    end for
    Calculate the fitness value of each particle
end while

Both swarm intelligence algorithms and evolutionary algorithms are population-based, iterative stochastic global optimization algorithms which can be applied for black-box problems. They are both branches of computational intelligence, and can be discussed in similar framework [14]. However, one critical factor makes swarm intelligence algorithms and evolution-based algorithms different from each other. Swarm intelligence algorithms simulate the collaboration behavior in nature while evolutionary algorithms mimic the competitive phenomenon in natural evolution. The difference in heuristics may lead to different trajectory when optimizing particular problems. As will be seen, this difference is obvious with respect to the waveform inversion.

In our work, two swarm intelligence algorithms which are under active research are applied to the inversion problem. The remainder of the section, we will give a brief description of these two algorithms.

A. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO), developed by Eberhart and Kennedy in 1995, is a stochastic global optimization technique inspired by social behavior of bird flocking or fish schooling [15]. In the PSO, each particle in the swarm adjusts its position in the search space based on the best position it has found so far as well as the position of the known best-fit particle of the entire swarm, and finally converges to the global best point of the whole search space.

The procedure of PSO is demonstrated by Algorithm 1. Each solution of the optimization problem is called a particle in the search space. The search of the problem space is done by a swarm with a specific number of particles. Assume that the swarm size is N and the problem dimension is D. Each particle i ($i = 1, 2, ...N$) in the swarm has the following properties: a current position X_i, a current velocity V_i, a personal best position $\tilde{P}_i$. There is a global best position $\hat{P}$, which has been found in the search space since the start of the evolution. During each of the iteration, the position and velocity of every particle are updated according to $\tilde{P}_i$ and $\hat{P}$. This process in the PSO can be formulated as follows:

$$V_{id}(t+1) = wV_{id}(t)+c_1r_1(\tilde{P}_{id}(t)-X_{id}(t))+c_2r_2(\hat{P}_d(t)-X_{id}(t)) \tag{7}$$

$$X_{id}(t+1) = X_{id}(t) + V_{id}(t) \tag{8}$$

Algorithm 2 Attract-Repulse Fireworks Algorithm

Initialize n fireworks
Calculate the fitness value of each firework
Calculate A_i according to Eq.9
while Termination condition unsatisfied **do**
  for $i = 1$ to n **do**
    Search according to Algorithm 3
  end for
  for $i = 1$ to n **do**
    if $rand(0, 1) < cr$ **then**
      Mutate according to Algorithm 4
    end if
  end for
  Calculate the fitness values of the new fireworks
  Update A_i according to Eq.9
end while

Algorithm 3 FWA Search

Generate m sparks
Evaluate the fitnesses of each sparks
Find the best spark with best fitness value, replace it with the current firework if better.

where $i = 1, 2, ...N$, $d = 1, 2, ...D$. In (7) and (8), the learning factors c_1 and c_2 are nonnegative constants, r_1 and r_2 are random numbers uniformly distributed in the interval $[0, 1]$,. The parameter w is the inertia weight, which is a constant in the interval $[0, 1]$ used to balance the global and local search abilities.

B. Attract-Repulse Fireworks Algorithm (AR-FWA)

In this work, a FWA variant proposed by Ding et al. [20] is used. Though targeting the GPU platform, experiment shows that the proposed algorithm is competitive to the state-of-the-art FWA. Based on this proposal, a variant with minor modification is used here. As this FWA variant core components are FWA search and attract-repulse mutation, we name it AR-FWA hereafter.

The pseudo-code of AR-FWA is depicted by Algorithm. 2. Here, we just present the operations different from [20], other details about can be found in the original paper.

1) FWA Search: FWA Search is illustrated by Algorithm. 3. A fixed number of sparks are generated to exploit the neighborhood solution space. Instead of a global selection procedure in [21], each firework is updated by its current best spark.

2) Attract-Repulse Mutation: Attract-Repulse Mutation is taken to keep the diversity of the swarm, as illustrated by Algorithm 4, where $\mathbf{x}_i$ depicts the i-th firework, while $\mathbf{x}_{best}$ depicts the firework with the best fitness. Different from [20], a new parameter cr is introduced to control the mutation frequency, thus no inner iteration is necessary for Algorithm 3.

3) Amplitude Update: The explosion amplitude of firework i is calculated as follows.

$$A_i = \hat{A} \cdot \left(\frac{f(\mathbf{x}_i) - y_{min} + \xi}{\sum_{i=1}^{n} (f(\mathbf{x}_i) - y_{min}) + n \cdot \xi} + \delta \right), \qquad (9)$$

Algorithm 4 Attract-Repulse Mutation

Initialize the new location: $\hat{\mathbf{x}}_i = \mathbf{x}_i$;
$s = U(1 - \delta, 1 + \delta)$;
for $d = 1$ to D **do**
  $r = rand(0, 1)$;
  if $r < \frac{1}{2}$ **then**
    $\hat{\mathbf{x}}_{i,d} = \hat{\mathbf{x}}_{i,d} + (\hat{\mathbf{x}}_{i,d} - \mathbf{x}_{best,d}) \cdot s$;
  end if
  if $\hat{\mathbf{x}}_{j,d} > \mathbf{ub}_d$ or $\hat{\mathbf{x}}_{j,d} < \mathbf{lb}_d$ **then**
    $\hat{\mathbf{x}}_{j,d} = \mathbf{lb}_d + |\hat{\mathbf{x}}_{j,d} - \mathbf{lb}_d| \% (\mathbf{ub}_d - \mathbf{lb}_d)$;
  end if
end for

where the predefined $\hat{A}$ denotes the maximum explosion amplitude, and $y_{min} = min(f(\mathbf{x}_i))\,(i = 1, 2, \ldots, n)$ i.e. the minimum (best) value of the objective function among the n fireworks, and ξ , which denotes the machine precision, is utilized to avoid zero division error. (Note that, in the original literature [21] and many following works, the ξ in the denominator is not multiplied by n which will cause the sum of all A surpass $\hat{A}$ when the finesses are very close.) δ is a small number to guarantee the amplitude is nonzero thus avoid the search process getting stalled. In [22], a minimum amplitude check is conducted instead of using δ.

IV. Experiments and Analysis

In order to test the feasibility and efficiency of each algorithms for the waveform inverse problem, simulation experiments are conducted here. In the simulation, swarm intelligence algorithms PSO and FWA, as well GA, NGA as well as DE are applied to find the structure parameters (density etc.).

A. Seismic waveform inversion

The goal of geophysics is to determine the properties of the Earth's interior from the surface and/or boreholes using measurements of physical phenomena. In seismology, the data consist of seismograms (seismic wave amplitude as a function of time and distance) from earthquakes or man-made explosions.

In this paper, we use reflectivity method developed by Fuchs and Muller [23] for calculating the waveform from source in layered medium (as illustrated by Fig. 2). This method with high-accuracy and speed is widely used in forward modeling [1] [24] .

A theoretical 4-layered isotropic medium model on top of a half-space is built based on a set of field data for forward modeling. The i-th layer is characterized by the P wave velocity α_i, S wave velocity β_i, the density ρ_i and the thickness h_i . The detail of model parameters settings are listed in Tab. I

We set a double couple source in the depth of 21 km, the value of the components of moment tensor come from a real earthquake source shown in the matrix below.

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} = \begin{bmatrix} 2.710 & 0.436 & -3.15 \\ 0.436 & -1.76 & -2.04 \\ -3.15 & -2.04 & -1.03 \end{bmatrix}$$

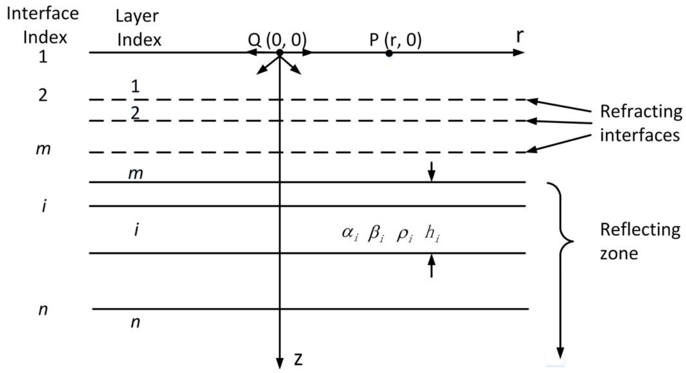

Fig. 2: The layered earth model (after [23])

TABLE I: Model Parameter

i-th layer	$h_i(km)$	$\alpha_i(km/s)$	$\beta_i(km/s)$	$\rho_i(g/cm^3)$
1	1.5	4	2.3	2.65
2	10	6	3.46	2.75
3	20	6.25	3.6	2.8
4	18	6.95	4.0	3.1
Half space	∞	8.1	4.67	3.37

11 receivers are settled in the free surface (i.e. $z = 0$), distances of receiver (r) away from epicenter are 77.8, 81.3, 197.6, 211.4, 282.7,292.3, 346.9, 352.7, 399.2, 406.3, 419.7 km.

As we assume the isotropic layered medium is Poisson solid, so the ratio of P wave velocity to S wave velocity α_i/β_i = 1.73, 9 parameters are inversed, including P wave velocity and thickness of layer 1 - 4, and P wave velocity of half space. The objective function in our inversion is weighted combination of root mean square residual (RMR) and waveform correlation of synthetic and observed waveform in time domain.

$$F = (1-\lambda) \times \frac{\sqrt{\sum_i \sum_j (O_{ij} - S_{ij})^2}}{N_w \times \sqrt{\sum_i \sum_j O_{ij}^2}} + \lambda \times [1 - \frac{1}{N_w} \times \sum_i \frac{max(O_i * S_i)}{\sqrt{O_i * O_i}\sqrt{S_i * S_i}}] \quad (10)$$

In Eq. 10, O_{ij} (S_{ij}) means the amplitude of i-th component of observed (synthetic) waveform at j-th time point. The first part of objective function is RMR of observed and synthetic waveform indicates the amplitude difference of two waveforms, the second part is correlation of two waveforms, indicates waveform similarity which contain phase information, and $(O * S)$ in equation is

$$(O * S)(\tau) = \int O(t)S(t-\tau)dt = \int O(t-\tau)S(t)dt$$

O_i and S_i is i-th observed and synthetic waveform, with the combined object function, we can evaluate the influence of amplitude, phase and arrival time on inversion, which is suitable for real data inversion. λ is weight parameter to balance RMR and waveform correlation error. N_w is number of waveform data.

TABLE III: Feasible Search Range

	1	2	3	4	5	6	7	8	9
Real Value	4	6	6.25	6.95	8.1	1.5	10	20	18
Lower Bound	3.5	5.5	5.75	6.45	7.6	1.125	7.5	15.0	13.5
Upper Bound	4.5	6.5	6.75	7.45	8.6	1.875	12.5	25.0	22.5

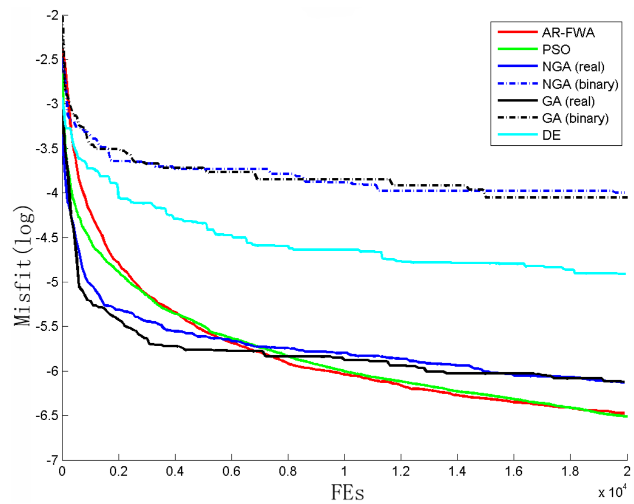

Fig. 3: Average misfit value on 35 trails along with function evaluations

B. Experimental Setting

GA and NGA have long been used for solving inverse problems [12]. In our experiments, the code of GA and NGA used in [1] is utilized. 8-bit binary encoding is added instead of only real encoding. DE is well studied and many variants are proposed, in our work, the classic versions is adopted [19].

For each algorithm, some parameters should be determined for acceptable performance. As the optimization process is very time-consuming, it is impossible to tune these parameters within limited time. In our experiment, parameters recommended in literature are adopted which perform well in general. The detailed parameters are listed in Tab. II

The feasible research ranges of the model parameters are listed by Tab. III. Each parameter is normalized such that the search range falls between 0 and 1 for optimization. All algorithms are uniformly initialized in $[0, 1]$. 35 independent trials are conducted for each algorithm and during each trail, the optimizer can conduct up to 2×10^4 function evaluations.

C. Results and Analysis

Fig. 3 presents how misfit value changes along the increasing of function evaluation times.

Overall, GA and NGA with binary string presentation show very poor performance compared to their real-encoded counterparts. Also, DE performs obviously poorly.

GA and NGA have similar convergence trajectory, so do AR-FWA and PSO. However, swarm intelligence algorithms

TABLE II: Parameter Settings for each Optimization Algorithm

Methods	Parameters
GA	POP = 100, cr = 0.9, m = 0.15
NGA	#demes = 5, #models = 20, #elitist = 1, others are as GA
DE	POP = 50, $F = 0.7$, $cr = 0.5$, rand/1/1
PSO	POP = 30, $\omega = 0.722984$, $c_1 = c_2 = 2.05$, global topology
AR-FWA	n = 10, m = 10, cr = 0.3, $\delta = 0.01$

TABLE IV: The true and inverted results using NGA, PSO, DE and AR-FWA for the given model

	1	2	3	4	5	6	7	8	9
Groundtruth	4	6	6.25	6.95	8.1	1.5	10	20	18
GA	4.090	6.017	6.257	6.956	8.102	1.628	10.43	**19.63**	**17.94**
GA (binary)	4.029	6.061	6.342	6.917	8.333	1.799	12.42	17.47	21.02
NGA	3.955	5.970	6.222	6.921	8.111	1.478	8.278	20.56	19.03
NGA (binary)	3.998	5.982	6.350	6.823	8.569	1.375	12.50	**19.71**	17.49
DE	3.830	6.102	6.282	7.022	8.066	1.725	11.055	**19.741**	18.612
PSO	3.907	5.983	6.237	6.969	8.099	1.347	**9.908**	**20.312**	**17.993**
AR-FWA	4.043	5.998	6.242	6.946	8.095	**1.521**	**9.945**	**19.849**	**18.079**

(PSO and FWA) and evolution-based algorithm (GA, NGA) show different trajectories. GA and NGA can find a solution with relative less misfit more quickly than swarm intelligence algorithms do. So the curves of GA and NGA drop more steeply than PSO and AR-FWA. But, this advantage in the early phase implies that GA and NGA are more prone to get premature, which is well-known for evolution-based algorithms [8]. As the optimization procedure goes on, the GA and NGA make progress much less than PSO and AR-FWA, and the advantage is reversed to the side of PSO and AR-FWA.

From first glance, PSO and AR-FWA obtain misfit values only slightly better than GA and NGA. As less misfit value does not mean necessarily that the archived model is closer to the real model, we list the inverted models in Tab. IV along with the groundtruth values.

As can be seen, parameter 1 to 5 are relatively easy to estimate. All algorithms can achieve estimates very close to the real values. However, parameter 6 to 9 are more sensitive (in Tab. IV, the estimated values close to the real values are emphasized in bold type). Based on the observation above, it is obvious that the slightly better misfit contributes substantially to a more accurate model estimate. To see intuitively how good the estimate is, Fig. 5 illustrates the estimated model by AR-FWA compared to the real model.

It is interesting to see how each algorithm performs along different trails. Fig. 4 shows the boxplot for 35 trials for all algorithms. Once again, the binary-encoded GA and NGA show poor performance with respect to the robustness. Though enjoying different performance for misfit value, the variations are comparable for all the other algorithm.

In summary, the inverted parameters with small misfit value are closer to the real values and some parameters are more sensitive to the misfit than others. The better misfit achieved by PSO and AR-FWA contribute to the a better estimates of these sensitive parameters, thus PSO and AR-FWA get more accurate models compared to other methods.

V. Conclusions and Future Work

In this work, two swarm intelligence algorithms, PSO and AR-FWA, are applied to regional waveform inverse problem

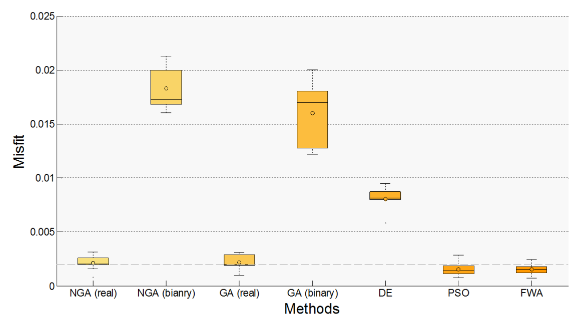

Fig. 4: Boxplot comparing the variations of misfit achieved by different algorithms

for the first time. Relying on the collaboration of the agents, PSO and AR-FWA avoid premature better in the experiment, compared to the widely used evolution-based algorithms, GA, NGA as well as DE. As some of the parameters in inverse problem are sensitive to misfit, the improvement of misfit achieved PSO and AR-FWA bring substantial improvement to the estimation of the model. Experimental results show that PSO and AR-FWA can find models very close to the real model and whose synthetic data can better fit the observed waveforms. As the simulation experiments have shown PSO and AR-FWA are potentially more power tools for waveform inversion, in the future we will use them to real world seismic data to invert the real structures.

References

[1] S. Li, Y. Wang, Z. Liang, S. He, and W. Zeng, "Crustal structure in southeastern gansu from regional seismic waveform inversion," *Chinese Journal of Geophysics*, vol. 55, no. 4, pp. 1186–1197, April 2012.

[2] P. L. Stoffa and M. K. Sen, "Nonlinear multiparameter optimization using genetic algorithms: nversion of plan-wave seismograms," *GEOPHYSICS*, vol. 56, no. 11, pp. 1794–1810, 1991.

[3] M. Sambridge and G. Drijkoningen, "Genetic algorithms in seismic waveform inversion," *Geophysical Journal International*, vol. 109, no. 2, pp. 323–342, 1992.

[4] R. C. Eberhart, Y. Shi, and J. Kennedy, *Swarm Intelligence*. Morgan Kaufmann, 2001.

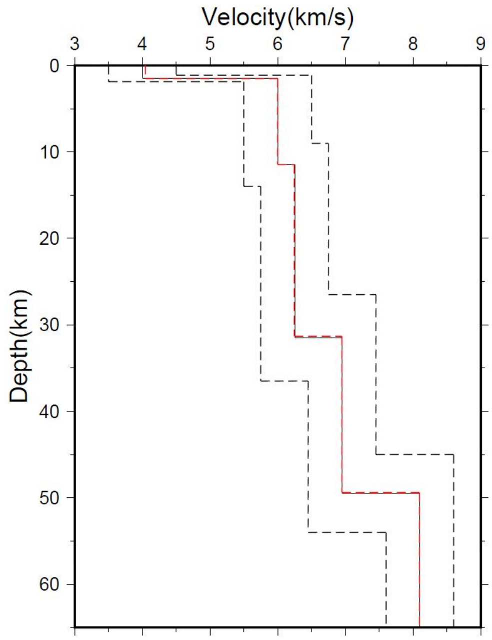

Fig. 5: Comparison between the given velocity model with the final inverted average model. Black solid line show the given model, red dashed line show the final inverted result of AR-FWA. The prescribed ranges for velocity and depth in which the model parameters are allowed to change in the inversion are shown by black dashed lines.

[5] S. Panda and N. P. Padhy, "Comparison of particle swarm optimization and genetic algorithm for facts-based controller design," *Applied Soft Computing*, vol. 8, no. 4, pp. 1418–1427, 2008.

[6] R. Hassan, B. Cohanim, O. de Weck, and G. Venter, "A comparison of particle swarm optimization and the genetic algorithm," in *46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*. Austin, Texas: American Institute of Aeronautics and Astronautics, April 2005, pp. 1–13.

[7] R. Shaw and S. Srivastava, "Particle swarm optimization: A new tool to invert geophysical data," *GEOPHYSICS*, vol. 72, no. 2, pp. F75–F83, 2007.

[8] S. Yuan, S. Wang, and N. Tian, "Swarm intelligence optimization and its application in geophysical data inversion," *Applied Geophysics*, vol. 6, no. 2, pp. 166–174, 2009.

[9] T. Mukerji, A. Suman, J. L. Fernández-Martínez, and E. García-Gonzalo, "A family of particle swarm optimizers for reservoir characterization and seismic history matching," in *9th Biennial International Conference & Explosition on Petrolelum Geophysics*, Hyderabad, India, February 2012.

[10] J. Barhen, "Optimization and geophysical inverse problems," San Jose, California, Tech. Rep., February 1999.

[11] M. K. Sen and P. L. Stoffa, *Global Optimization Methods in Geophysical Inversion*, 2nd ed. Cambridge, UK: Cambridge University Press, April 2013.

[12] K. D. Koper, M. E. Wysession, and D. A. Wiens, "Multimodal function optimization with a niching genetic algorithm: A seismological example," *Bulletin of the Seismological Society of America*, vol. 89, no. 4, pp. 978–988, 1999.

[13] A. Padhi and S. Mallick, "Multicomponent pre-stack seismic waveform inversion in transversely isotropic media using a non-dominated sorting genetic algorithm," *Geophysical Journal International*, vol. 196, no. 3, pp. 1600–1618, 2014.

[14] X.-S. Yang, "Swarm intelligence based algorithms: a critical analysis," *Evolutionary Intelligence*, vol. 7, no. 1, pp. 17–28, 2014.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Neural Networks, 1995. Proceedings., IEEE International Conference on*, vol. 4, nov/dec 1995, pp. 1942 –1948 vol.4.

[16] M. Dorigo and L. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *Evolutionary Computation, IEEE Transactions on*, vol. 1, no. 1, pp. 53–66, 1997.

[17] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Ppplications to Biology, Control, and Artificial Intelligence*. Oxford, England: Univ. Michigan Press, 1975.

[18] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies a comprehensive introduction," *Natural Computing*, vol. 1, no. 1, pp. 3–52, 2002.

[19] R. Storn and 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, 1997.

[20] K. Ding, S. Zheng, and Y. Tan, "A gpu-based parallel fireworks algorithm for optimization," in *Proceeding of the fifteenth annual conference on Genetic and evolutionary computation conference*, ser. GECCO '13. New York, NY, USA: ACM, 2013, pp. 9–16.

[21] Y. Tan and Y. Zhu, "Fireworks algorithm for optimization," in *Advances in Swarm Intelligence*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2010, vol. 6145, pp. 355–364.

[22] S. Zheng, A. Janecek, and Y. Tan, "Enhanced fireworks algorithm," in *Evolutionary Computation (CEC), 2013 IEEE Congress on*, 2013, pp. 2069–2077.

[23] K. Fuchs and G. Müller, "Computation of synthetic seismograms with the reflectivity method and comparison with observations," *Geophysical Journal of the Royal Astronomical Society*, vol. 23, no. 4, pp. 417–433, 1971.

[24] S. Jia and X. Zhang, "Study on the crust phases of deep seismic sounding experiments and fine crust structures in the northeast margin of tibetan plateau," *Chinese Journal of Geophysics*, vol. 51, no. 5, pp. 1431–1443, September 2008.