Building starting models for full-waveform inversion using global optimization methods: A comparison between Particle swarm optimization and genetic algorithm

Oscar F. Mojica[1], Peterson Nogueira[*,1], Rodrigo Santana[1] and Daniel Revelo[2]

[1]SENAI CIMATEC Supercomputing Center for Industrial Innovation and [2]CPGG/IGEO/UFBA & INCT-GP

This paper was prepared for presentation during the 16th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 19-22 August 2019.

Abstract

The performance of two global optimization methods (GOMs) with embedded elitism strategies for solving the 2D acoustic seismic inverse problem is here compared using synthetic acoustic data of the Marmousi model. A real-coded elitist genetic algorithm (GA) that had been used successfully in the past for this end is compared with a newly developed elitist-mutated particle swarm optimization (EMPSO) technique to estimate acoustic macro models of the P-wave velocity field (V_p). We find that EMPSO seems to have higher performance than GA with respect to the final attained value of the cost function when the completion of specified number of iterations is chosen as stopping criterion. The results further show that while both EMPSO and the GA obtain high quality solutions, the computational effort required by PSO to arrive to such high quality solutions is less than the effort required to arrive at the same high quality solutions by the GA. Finally, multiple runs of descent-based full waveform inversion (FWI) started from either final GA or EMPSO models produce final high-resolution models.

Introduction

Global optimization methods (GOMs) have presented themselves as an alternative to estimate a starting model for Full Waveform Inversion (FWI). GOMs are an interesting choice since a proper parameterization technique coupled with sufficient computing power allow for a reduction of the geophysicist effort and time into building an initial estimate of the velocity model. Simulated annealing (SA) and genetic algorithm (GA) are the most commonly used GOMs in geophysics, they (or a variant) have already been used to estimate a starting model for FWI (Sajeva et al., 2016; Datta and Sen, 2016). In recent years, however, particle swarm optimization (PSO) algorithm has rapidly become an attractive alternative for solving geophysical inverse problems and seems to enjoy an ever increasing popularity (Shaw and Srivastava, 2007; Fernandez-Martinez et al., 2008). According to this trend, we propose to compare the effectiveness of the elitism-based GA and PSO to solve a 2D seismic optimization problem. The motivation is to validate or refute the widely speculated hypothesis that PSO has the same effectiveness as the GA (same rate of success in finding true global optimal solutions) but with better computational efficiency. The results of this test could prove to be significant for the future development of seismic inversion approaches using GOMs.

Theory

Real-coded GA with elitism: The GA begins its search from a randomly generated population that evolve over successive generations (iterations). To perform its optimization-like process, the GA employs three operators to propagate its population from one generation to another. The first operator is the “Selection” operator that mimics the principle of “Survival of the Fittest”. The second operator is the “Crossover” operator, which mimics mating in biological populations. The crossover operator propagates features of good surviving models from the current population into the future population, which will have better fitness value on average. The last operator is “Mutation”, which promotes diversity in population characteristics. The mutation operator allows for global search of the design space and prevents the algorithm from getting trapped in local minima. Once selection, crossover and mutation are complete, there will be two populations: the old and the offspring. Reinsertion is concerned with the means of combining them to produce the new population. Among the various reinsertion approaches, the elitist strategy is widely adopted. In such strategy, the best parents are reinserted in the next population. The encoding schemes in GA are either binary coding or real coding. The latter overcomes several issues regarding the former and it has also been adopted to address problems in a wide range of areas. Therefore, we used a real-coded elitist GA similar to that used by Sajeva et al. (2017). This version of GA is based on stochastic universal sampling type selection, intermediate recombination crossover, and mutation according to a defined probability P_m (A selected variable of an individual is mutated with probability P_n) on a population of fixed number individuals.

Elitist-mutated PSO: PSO is an evolutionary computation technique based on the social behavior metaphor. The PSO algorithm is initialized with a population of random candidate solutions, conceptualized as particles. Each particle $\mathbf{x}_i = x_{i1}, x_{i2}, \ldots, x_{iD}$ is assigned a randomized velocity $\mathbf{v}_i = v_{i1}, v_{i2}, \ldots, v_{iD}$ and is iteratively moved through the D-dimensional problem space. It is attracted towards the location of the best fitness achieved so far by the particle itself $\mathbf{p}_i = p_{i1}, p_{i2}, \ldots, p_{iD}$ and by the location of the best fitness achieved so far across the whole population $\mathbf{g}_i = g_1, g_2, \ldots, g_D$ ($gbest$-global version of the algorithm). At iteration k, the basic PSO algorithm (Clerc, 1999) can be described in vector notation as follows:

$$\mathbf{v}_i^{k+1} = \chi \left[\mathbf{v}_i^k + c_1 \mathbf{u}_1^k \otimes (\mathbf{p}_i^k - \mathbf{x}_i^k) + c_2 \mathbf{u}_2^k \otimes (\mathbf{g}^k - \mathbf{x}_i^k) \right] \quad (1)$$

$$\mathbf{x}_i^{k+1} = \mathbf{x}_i^k + \mathbf{v}_i^{k+1}, \quad (2)$$

In Eq. 1, χ, c_1, and c_2 are the control parameters called the constriction factor, cognitive parameter, and social parameter, respectively. The former is a function of c_1 and c_2 as reflected in Eq. 3.

$$\chi = \frac{2}{|\,2 - \varphi - \sqrt{\varphi^2 - 4\varphi}\,|}, \qquad \text{where} \qquad \varphi = c_1 + c_2 \geq 4. \quad (3)$$

On the other hand, vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ are D-dimensional vectors of uniformly distributed and independent random numbers in the $[0,1]$ range used to maintain the population diversity ($\otimes$ denotes element-by-element vector multiplication). We used an improved PSO, named EMPSO (Nagesh Kumar and Janga Reddy, 2007), which introduces an elitist-mutation strategy into the PSO to improve its performance. Pseudocode of the EMPSO algorithm is presented in Fig. 1. In the EMPSO, the elitist-mutation step is computed as follows: first, all particles are sorted in ascending order based on their fitness function and the index numbers for the respective particles are obtained; second, the elitist-mutation (EM) is performed on NM worst particles and the respective particle position vectors are replaced with the new mutated position vectors, whereas the velocity vectors of these particles are unvaried.

```
for i ← 1 to NM do
  l ← ASF[i]
  for d ← 1 to D do
    if rand < p_em then
      x_ld = g_d + 0.1 × VR_d × randn
    else
      x_ld = g_d
    end if
  end for
end for
```

Figure 1 Pseudo-code of the EMPSO algorithm. NM=number of particles to be elitist-mutated; p_{em}=probability of mutation; g_d = d-th component of global best particle; ASF=index of sorted population; `rand`=uniformly distributed random number $U(0,1)$; `randn`=Gaussian random number $N(0,1)$; and VR_d=range of decision variable d.

Testing approach (GA/EMPSO + local FWI)

In this section, we show the results obtained when the complete workflow (global + local search methods) was applied for retrieving a cropped Marmousi model (285×369 samples, with vertical and horizontal space sampling of 8 m and 25 m, respectively). For the forward modeling of the GA and EMPSO algorithms, we use the finite-differences (FD) method, with an accuracy of 2nd order in time and 8th order in space. The acquisition geometry consisted of 50 sources and 369 receivers, one receiver for each sample on the horizontal axis. We generated synthetic data using a Ricker wavelet with a maximum frequency around 17 Hz, and set the sampling and recording times to 0.8 ms and 3 s, respectively. This dataset was then filtered (below 6 Hz), and a Ricker wavelet with a maximum frequency around 6 Hz was used to compute the modeled data. To evaluate the misfit, we use the l_2 norm. Proceeding as in Sajeva et al. (2017), we use a simple 1D V_p model (which together with the water bottom depth constitute the prior information) with velocities linearly increasing with depth from 1500 to 3500 m/s. This model is used to centre the GA/EMPSO inversion ranges and also to build the irregular GA/EMPSO grid by following predefined resolution criteria (for full details, see Sajeva et al. (2017)). The resulting grid (black dots) and the linear 1D model are shown in Figure 2-a. This grid has 176 nodes. These nodes are bilinearly interpolated to the finite-difference grid for the forward-modeling following what has come to be called a "two grid strategy". The ranges for the V_p values during the GA/EMPSO inversion are shown in Figure 2-b. We defined the minimum and maximum limits for the first and last level of depth as a percentage of the velocity value of the grid nodes at these levels. The limits for the intermediate levels of depth are defined by the lines passing through the maximum and minimum points of the shallower and deeper levels.

The GA parameters are set as follow: 360 individuals that evolve for 100 generations, mutation rates $P_m = 0.7$ and $P_n = 0.5$. As many offspring as parents are produced and 25% of the better-fitting parents are saved to enable the replacement of the least fit offspring in the new generations by the most fit parents. It is checked that the offspring are replaced by better parents. For the EMPSO we used 360 particles, 100 iterations, NM=25%, P_{em}=0.3 and the *gbest* topology. In EMPSO, the EM step begins from 10th iteration (10% of the maximum number of iterations) and the coefficients of cognitive (c_1) and social(c_2) acceleration were set to 1,2 and 2,9, respectively. In both GOMs, if the model variables violate their upper or lower bounds, they are artificially brought back into the search space. In the PSO/GA inversions, we performed 36.000 model evaluations and the final best-fitting model is used as a starting point in a local full-waveform inversion. Our implemented descent-based FWI algorithm uses the steepest-descent method and a multiscale approach (performing thirty iterations for each frequency band with maximum frequencies of 4.6, 11.5, 18.4, 25.3, 32.2 and 39 Hz). The line search along the gradient search directions uses the Barzilai-Borwein (BB) formula for an initial step length (Barzilai and Borwein, 1988). When required, it applies a backtracking line search method to update the step length. The forward problem in FWI is formulated in the time domain and solved using an FD method having an accuracy of 2nd order in time and 16th order in space, with a time step of 0.8 ms to ensure stability. The recording time and sampling grid (*dx* and *dy*) were equal to those used by EMPSO/GA.

Parallel implementation

Knowing in advance that the GOM's require the evaluation of many thousands of models, we implemented parallel versions that rely on running in parallel the evaluation of the fitness function of each particle/individual. The algorithms were implemented using an hybrid (MPI/OpenMP) master/worker programming paradigm where particles/individuals fitness evaluation is handled through dynamic scheduling. The experiments were run on the YEMOJA Supercomputer at SENAI CIMATEC, which uses an InfiniBand interconnection. Each compute node

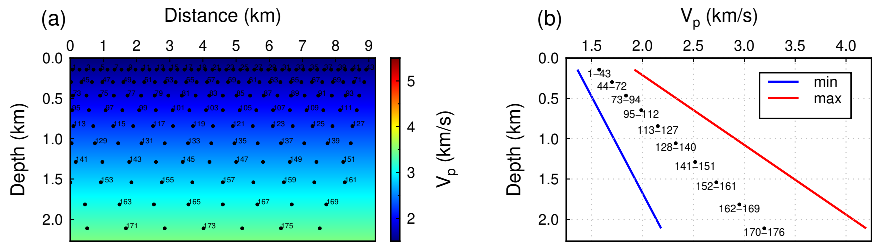

Figure 2 (a) 1D gradient model and the irregular grid nodes. (b) The search range used in the inversions.

used contains 128 GB of RAM and two sockets where each socket has an Intel Xeon E5-2690 v2 CPU at 3 GHz (https://www.top500.org/system/178420). Of the various possible scenarios for the distribution of processes and threads in each node of YEMOJA, the 10 MPI tasks $\times$ 2 OpenMP threads per node configuration gave the best performance in the testing approach described above (36.000 model evaluations). The overall run time of inversions was approximately 20 hours, using 18 nodes (180 MPI processes). Note that the runtime could have been even lower if a higher number of compute nodes had been used and if more code optimizations had been carried out. To put it more clearly, the time that inversion takes will depend on the complexity of the forward model, the parallelization strategy and the hardware. Yet, thanks to advances in high-performance computing, time is not viewed as a constraint today - at least in the simplest case (2D acoustic approximation).

Results

Figures 3 and 4 ilustrate both GA and EMPSO results for three random trials (both techniques used the same seed for random number generation at each experiment). The error vs iteration plots are shown in Figure 5. For both methods, the error gradually decreases along the time. It should be noted, however, that EMPSO gives best fitness values over different trials than GA. EMPSO does not seem to experience long periods of stagnation as GA (apparent from the staircase pattern in fitness curves). Figures 6 and 7 show the final models after descent-based FWI using as starting models the velocity estimates retrieved by GA and EMPSO, respectively. The correct Marmousi model is shown repeatedly in Figs. 3-a, 4-a, 6-a and 7-a for ease of comparison. In general, the starting velocity models obtained using GA seem to be smoother than those generated by EMPSO. However, the FWI results using either GA and EMPSO outputs are virtually equivalent. Putting it another way, the EMPSO and GA methods allow the recovery of the low wavenumber components in the background model to avoid the cycle-skipping problem and their results are comparable. As the chosen stopping criterion was the number of iterations/generations neither of the methods stands out over another in computational time. However, it was noted that the final attained value of the fitness in GA trials ($\approx 2 \times 10^6$) is reached in fewer iterations in EMPSO trials. This clearly shows that EMPSO could save computational time using another stopping criterion (i.e., the achievement of a predetermined data misfit value)

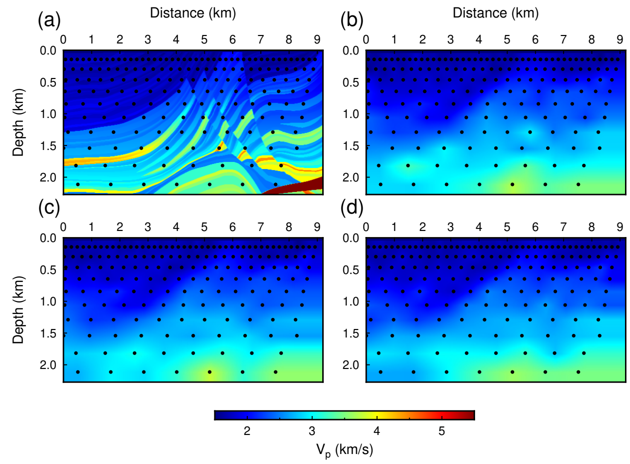

Figure 3 (a) Cropped Marmousi model. (b-d) The inversion estimates obtained with real-coded GA with elitism for three random trials.

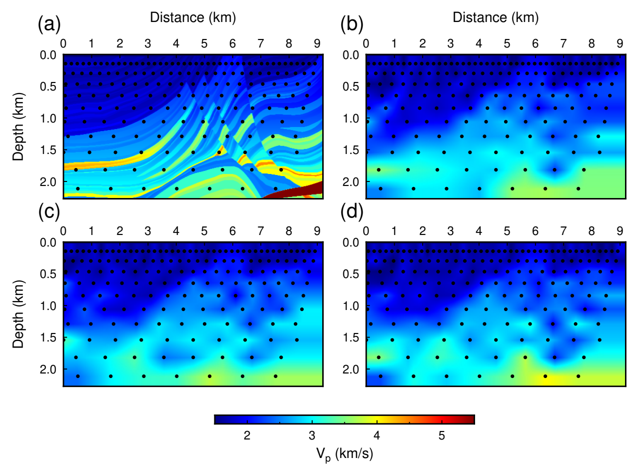

Figure 4 (a) Cropped Marmousi model. (b-d) The inversion estimates obtained with EMPSO for three random trials.

Conclusions

In this paper, we compare the performances of two of the most used GOMs in applied geophysics (GA and PSO) with implemented elitism strategies for estimating acoustic macro models of the V_p field using synthetic acoustic data of the Marmousi model. For the implemented versions, EMPSO yields V_p models that provide lower data misfits than those supplied by GA's outputs, although both sets of models reproduce the long-wavelength structures of

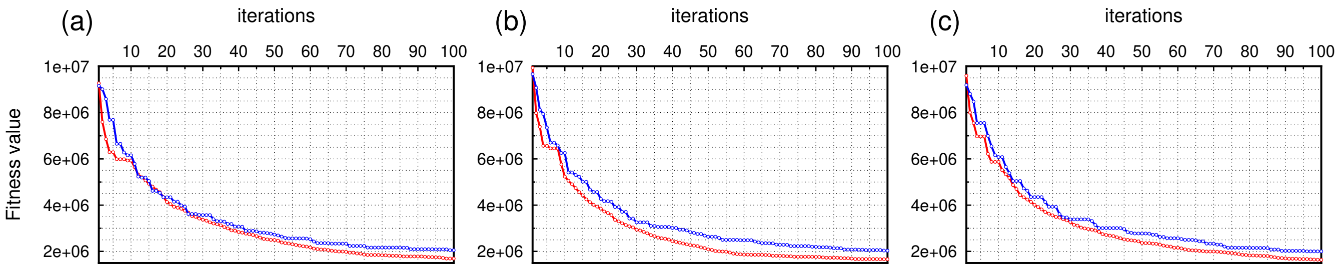

Figure 5 (a-c) The evolution of the energy function for EMPSO (red) and GA (blue) for the three diferent random trials in sequence (b-d) from Figs. 3 and 4, respectively.

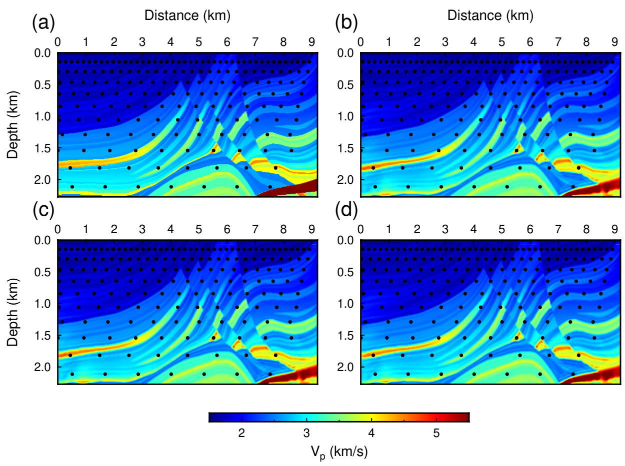

Figure 6 (a) Cropped Marmousi model. (b-d) Final models after descent-based FWI from the starting models (b-d) after GA of Figure 3.

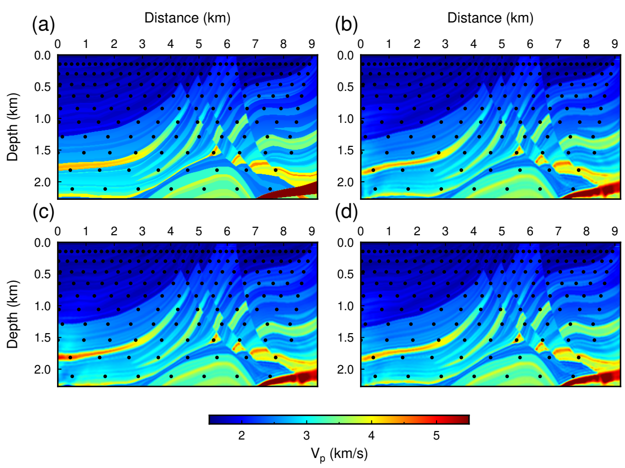

Figure 7 (a) Cropped Marmousi model.(b-d) Final models after descent-based FWI from the starting models (b-d) after EMPSO of Figure 4.

the Marmousi. Descent-based FWIs results using final GA and EMPSO models are close to the true Marmousi model, demonstrating the ability of both GOMs to yield velocity models suitable as input to descent-based FWI. From our experience, it is observed that EMPSO has faster convergence rate and the best robustness than GA with elitism. EMPSO requires fewer iterations than GA to find the same fitness value, which would have resulted in less CPU time with the proper stopping criterion (supporting the tested hypothesis). However, note that comparisons of GOMs are problem dependent and restricted to the implemented versions.

Acknowledgements

The authors gratefully acknowledge support from Shell Brasil through the PSO-FWI project at SENAI CIMATEC and the strategic importance of the support given by ANP through the R&D levy regulation.

References

Barzilai, J., and J. M. Borwein, 1988, Two-point step size gradient methods: IMA journal of numerical analysis, **8**, 141–148.

Clerc, M., 1999, The swarm and the queen: towards a deterministic and adaptive particle swarm optimization: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on, IEEE, 1951–1957.

Datta, D., and M. K. Sen, 2016, Estimating a starting model for full-waveform inversion using a global optimization method: Geophysics, **81**, R211–R223.

Fernandez-Martinez, J. L., J. P. Fernández-Alvarez, M. E. García-Gonzalo, C. M. Pérez, and H. A. Kuzma, 2008, Particle swarm optimization (pso): a simple and powerful algorithm family for geophysical inversion, *in* SEG Technical Program Expanded Abstracts 2008: Society of Exploration Geophysicists, 3568–3571.

Nagesh Kumar, D., and M. Janga Reddy, 2007, Multipurpose reservoir operation using particle swarm optimization: Journal of Water Resources Planning and Management, **133**, 192–201.

Sajeva, A., M. Aleardi, and A. Mazzotti, 2017, Genetic algorithm full-waveform inversion: uncertainty estimation and validation of the results.: Bollettino di Geofisica Teorica ed Applicata, **58**.

Sajeva, A., M. Aleardi, E. Stucchi, N. Bienati, and A. Mazzotti, 2016, Estimation of acoustic macro models using a genetic full-waveform inversion: Applications to the marmousi modelgenetic fwi for acoustic macro models: Geophysics, **81**, R173–R184.

Shaw, R., and S. Srivastava, 2007, Particle swarm optimization: A new tool to invert geophysical data: Geophysics, **72**, F75–F83.