Convergence Analysis of Global Optimization in Seismic Data Processing

Rosana Veroneze, Everton Z. Nadalin, Rafael Granzotti, Tainá Souza, Renato Lopes

This paper was prepared for presentation during the 15th International Congress of the Brazilian Geophysical Society held in Rio de Janeiro, Brazil, 31 July to 3 August, 2017.

Abstract

In the last years, global optimization has been more used in seismic data processing. These methods are not so sensitive to initial starting points as the local optimization methods, and usually provide accurate estimations.

In this work, we analyze the convergence of a global optimization algorithm called *differential evolution* (DE) for estimating the parameters of the common reflection surface (CRS) method for a 2D real dataset from Brazil.

We propose a new stopping criterion, based on an intrinsic DE characteristic, that is much more robust than the commonly used fixed number of iterations. Our results indicate that we can obtain good images without wasting computational resources with the usage of this new criterion.

Introduction

One of the major goals of geophysics is to find earth models that explain geophysical observations. Thus, a common task in geophysics is to infer model parameters, which is usually done iteratively by fitting geophysical observations with theoretical predictions. Both local and global optimization methods are used in the estimation of these model parameters. Therefore, the branch of mathematics known as optimization has great importance in many geophysical applications (Sen & Stoffa, 1995).

For many geophysical applications, the function to be optimized is very complicated, presenting several local minima and maxima, also known as valleys and hills. The minimum (maximum) of all the local minima (maxima) is called the global minimum (maximum). Note that the optimal points are minima, if the goal is to minimize a function, otherwise if the goal is to maximize a function, the optima are the maxima. In many situations, there are very poor local optima, i.e., solutions that have much worse values than the value of a global optimum.

Typically, local optimization algorithms attempt to find a local optimum in the close neighborhood of a starting point. So, these methods are quite sensitive to the starting point. It is usually difficult to provide a good starting point, leading the optimization to a poor local optima. This problem has troubled geophysicists for many years (Sen & Stoffa, 1995).

With the increasing computational capacity, global optimization methods have been applied to several geophysical problems (Sen & Stoffa, 1995). These methods attempt to find the global optimum by updating their current position using more global information about the function. Their convergence to the global optimum is not guaranteed, however user experience indicates that it is possible to find good solutions even with only poor starting points (Sen & Stoffa, 1995). Moreover, these methods can also be used to obtain additional information about the nature of the solution (Sen & Stoffa, 1995).

In this work, we analyze the convergence of a global optimization algorithm called *differential evolution* (DE) (Storn & Price, 1997) for estimating the parameters of the common reflection surface (CRS) (Mann at al., 1999) method. The DE algorithm is a metaheuristic that presents a good convergence rate, simple parametrization and fast convergence, when compared to other heuristics / metaheuristics (Barros et al., 2015).

Seismic stacking methods aim to provide a simulated zero offset (ZO) image of the subsurface in time. The CRS method is a powerful alternative to the classical and well-known common-midpoint (CMP) stack. CRS produces ZO sections with high signal-to-noise ratios, improved resolution, more continuous reflectors, and enhanced images of dipping reflectors (Garabito et al., 2012). This improvement is made possible by the fact that CRS uses more traces for stacking than CMP, due to the CRS traveltimes depend not only on the midpoints, but also on the offsets (Barros et al., 2013). Therefore, traces on neighboring CMP gathers are used for stacking.

The CRS stacking operator is a second-order hyperbolic traveltime approximation. It depends on three parameters in 2D data, and eight parameters in 3D data. The simultaneous determination of the CRS parameters by global optimization can be computationally expensive (Garabito et al., 2012). Therefore, a traditional strategy is to estimate the CRS parameters in a sequence of single-parameter searches. However, the CRS parameters that are determined simultaneously by global optimization are more accurate, and they can be confidently applied to other seismic processes (Garabito et al., 2012).

Barros et al. (2013, 2014) show a comparison between the stacking result using the sequential and the global search approach in both synthetic and real data. As expected, the sequential search provided poor images when compared to the global search of all parameters simultaneously. We can say that the main reason for that is that the sequential search does not find the optimal estimates of the parameters, in the sense of maximizing the coherence for the CRS traveltime. It has bad consequences not only in the values of the parameters themselves, but also in the image quality.

Although global optimization metaheuristics have been used in recent years to estimate CRS parameters, as far as we know, most works in the literature use a very simple stopping criterion: a fixed number of iterations. This can be problematic because convergence may not be reached if an optimization run is terminated too early. To avoid a convergence problem, it is common to use a high number of iterations, thus, wasting computational resources.

We analyze the convergence of the DE algorithm in estimating the CRS parameters for a 2D real dataset from Brazil, called *Tacutu*. As the CRS parameters are estimated for each point-in-time of a ZO trace, and each CMP of the pre-stack data corresponds to a ZO trace, we performed an analysis of convergence on both time sample and CMP. Our experimental results show that the convergence of the optimization for each pair (*time sample*, *CMP*) can be very different from each other. Thus, a fixed number of iterations is an inappropriate stopping criterion for this problem.

Hence, we also explore a smarter stopping criterion based on an intrinsic DE characteristic: its selection scheme does not allow deterioration with regard to the objective function value, i.e., the DE selection scheme is *greedy*. In this way, the individuals of the population usually converge to one point in the search space. So, the distribution of individuals can be used to derive conclusions about the state of an optimization run (Zielinski & Laur, 2008).

We also show that a stopping criterion based on the distribution of the population individuals is much more robust than using only a fixed number of iterations. Moreover, we show that this new stopping criterion enable us to choose the resolution of the stack image.

Theory

Zero Offset Common Reflection Surface

The method zero offset common reflection surface (ZO-CRS) (Mann et al., 1999) produces a simulated ZO section from multicoverage data. This method can be used for 2D and 3D data, having a specific traveltime formulation for each case. Here, we will focus on the 2D case.

Let m_0 be the central point, i.e., the point where the ZO trace is being constructed. As usual, we associate each trace with a source-receiver pair with coordinates s and r, respectively. Alternatively, a trace may be identified by the midpoint m and the half-offset h of the source-receiver pair. The CRS traveltime relates the traveltime of a reflection that originates at a source in s and is received by a receiver in r with the two-way ZO traveltime t_0 of the same reflection event. It is written as

$$t_{CRS}(h, m_d)^2 = (t_0 + a m_d)^2 + b m_d^2 + c h^2, \qquad (1)$$

where $m_d = m - m_0$ is the trace midpoint displacement, the parameters a and b are related, respectively, to the dip and the curvature of the reflector image in the stacked section, and the parameter c is related to the normal moveout (NMO) velocity. As with traditional velocity analysis (Taner & Koehler, 1969), the CRS parameters a, b and c are estimated from the data by coherence analysis. The idea is that, for the right parameters, all the traces at time $t_{\mathrm{CRS}}(h, m_d)$ refer to the same reflection event, so that these samples should be coherent, or aligned.

Differential Evolution

Differential Evolution (DE) (Storn & Price, 1997) is a parallel direct search method for continuous space variables which utilizes NP D-dimensional parameter vectors $\mathbf{x}_{i,G}$, $i = 1, \ldots, NP$ as population, on each generation G. Each parameter vector constitutes a candidate solution of the optimization problem. The DE algorithm is divided in three stages: *mutation*, *crossover* and *selection* (see Algorithm 1).

The mutation operation generates a new vector for each individual by the following expression:

$$\mathbf{v}_{i,G+1} = \mathbf{x}_{r_1,G} + F\left(\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}\right). \qquad (2)$$

The indexes $r_1, r_2, r_3 \in \{1, \ldots, NP\}$ are mutually distinct, chosen randomly and different from the index i. F is a real and constant factor in the range of [0, 2], which controls the length of the step given in the direction defined by $\mathbf{x}_{r_2,G} - \mathbf{x}_{r_3,G}$.

The crossover operation is employed with the goal of enhancing the diversity of the mutated parameter vectors. Let $\mathbf{x}_{i,G}$ be the vector under analysis and $\mathbf{v}_{i,G+1}$ the mutated vector obtained by Eq. 2. The crossover resultant vector $\mathbf{u}_{i,G+1}$ is obtained by

$$\mathbf{u}_{ji,G+1} \begin{cases} \mathbf{v}_{ji,G+1}, \text{if } r_j \leq CR \text{ and } j \neq l_i \\ \mathbf{x}_{ji,G}, \text{if } r_j > CR \text{ and } j \neq l_i \end{cases}, \qquad (3)$$

where $j = 1, \ldots, D$, $r_j \sim \mathrm{U}(0,1)$, $CR \in [0,1]$is the crossover constant factor defined by the user and $l_i \in 1, \ldots, D$ is a random index, which ensures that $\mathbf{u}_{i,G+1}$ receives at least one component from $\mathbf{v}_{i,G+1}$.

After the stages of mutation and crossover, the selection of the vectors to be preserved in the next generation is made by the use of a greedy criterion. The vector $\mathbf{u}_{i,G+1}$ is compared to the vector $\mathbf{x}_{i,G}$. If vector $\mathbf{u}_{i,G+1}$ yields a bigger cost function value than $\mathbf{x}_{i,G}$, then $\mathbf{x}_{i,G+1}$ is set to $\mathbf{u}_{i,G+1}$; otherwise, the old value $\mathbf{x}_{i,G}$ is retained.

Algorithm 1: Differential Evolution

Input: population size NP, scale factor of mutation F, crossover rate CR
Output: final population $\boldsymbol{P_G}$
$G \leftarrow 0$
$\boldsymbol{P_G} \leftarrow$ random population of NP individuals // $\boldsymbol{P_G} = \{\mathbf{x}_{1,G}, \mathbf{x}_{2,G}, \ldots, \mathbf{x}_{NP,G}\}$
$\mathbf{fx} \leftarrow$ fitness of each individual $\mathbf{x}_{i,G}$ // $i = \{1, 2, \ldots, NP\}$
While stopping criteria are not satisfied
  For i = 1 to NP
    // Mutation
    Calculate Eq (2)
    // Crossover
    Calculate Eq (3)
  $\mathbf{fu} \leftarrow$ fitness of each individual $\mathbf{u}_{i,G}$ // $i = \{1, 2, \ldots, NP\}$
  // Selection
  For i = 1 to NP
    If $\mathbf{fu}_i > \mathbf{fx}_i$
      $\mathbf{x}_{i,G+1} \leftarrow \mathbf{u}_{i,G+1}$
    Else
      $\mathbf{x}_{i,G+1} \leftarrow \mathbf{x}_{i,G}$
  $G \leftarrow G + 1$

Method

One possibility to measure the distribution of individuals is the standard deviation of the fitness of all individuals. The optimization can be stopped when the standard deviation is less or equal a given threshold. As the standard deviation is a scalar relative to a mean, we chose to use the relation between the standard deviation and the mean of the population fitness to measure the distribution of the individuals. We refer to this stopping criterion as *sc_sm* in what follows.

In this work, the *semblance* (Neidell & Taner, 1971) was used as the cost function to measure the fitness of the population individuals. This function returns a value close to 1 when the hyperboloid defined by the CRS traveltime equation (Eq. 1) has a good fit to the traces, and returns a value close to 0 otherwise. Each individual of the population is a vector in R^3 that has estimations for the parameters *a*, *b*, *c* of Eq. 1. The user-defined parameters of the DE algorithm were set to $NP = 20$, $F = 0.5$, and $CR = 0.3$.

We ran the CRS stack for the Tacutu data with and without the usage of a velocity guide, and analyze the convergence of the DE algorithm for both cases.

In an experiment, we ran the CRS stack with a fixed number of iterations as the stopping criterion. We show the decay of the *sc_sm* over iterations, analyzing it by CMP and time sample.

We also ran the CRS stack considering the proposed stopping criterion: *sc_sm*. In this case, we use the *sc_sm* together with a maximum number of iterations (*sc_maxit*). It was done because, as it can be seen in the results section, there are some points with a very complicated fitness surface. So, we use *sc_maxit* to limit the computational cost in terms of runtime.

Results

Figure 1 shows the *sc_sm* over iterations for the Tacutu data without using a velocity guide. As the optimization runs for each pair (*time sample*, *CMP*), we plotted the mean values (and the standard deviations in the case of Figure 1b and 1d) of the *sc_sm* over both CMPs and time samples. Analyzing the colormap of Figure 1a, we notice that the converge of the DE algorithm is similar inside two large subgroups of CMPs. However, the convergence is very distinct between these two subgroups. In the colormap by time samples (Figure 1c), we also notice that the converge is very different between them. It reinforces the idea that to use a fixed number of iterations as the stopping criterion of the optimization run is not a good option. We also observe in the colormaps that the DE algorithm did not converge for some points even after 500 iterations. We also see in Figure 1, more specifically Figure 1b and 1d, that the standard deviation of the *sc_sm* also decays over iterations for both CMPs and time samples.

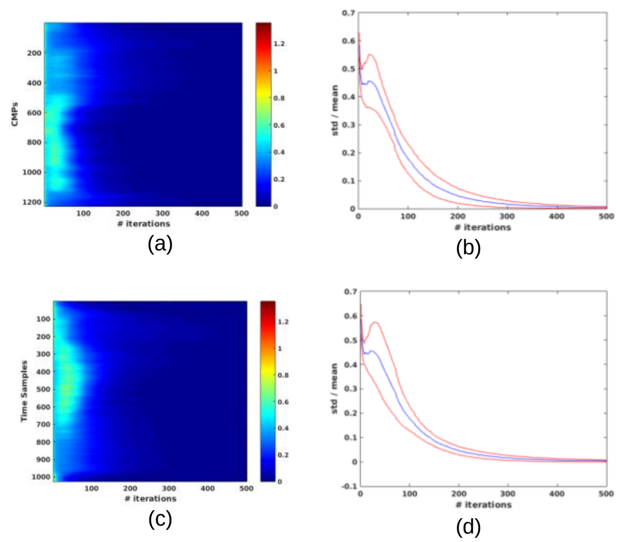


Figure 1 - sc_sm over iterations for Tacutu data without using velocity guide. Plot of the colormap of the sc_sm over iterations (a) by CMP, and (c) by Time Sample. Plot of the mean (in blue) and standard deviation (in red) of the sc_sm over iterations (b) by CMP, and (d) by Time Sample.

Figure 2 is similar to Figure 1, but now using a velocity guide. The usage of a velocity guide decreases the size of the search space, making the problem easier to solve. For this reason, we can observe that the DE algorithm converges using less iterations. Another interesting aspect to observe is that the most difficult regions, for both CMPs and time samples, is the same with or without using a velocity guide. Comparing Figure 1b and 1d with Figure 2b and 2d, we also notice that the standard deviation of the *sc_sm* over the iterations is smaller with the usage of a velocity guide.

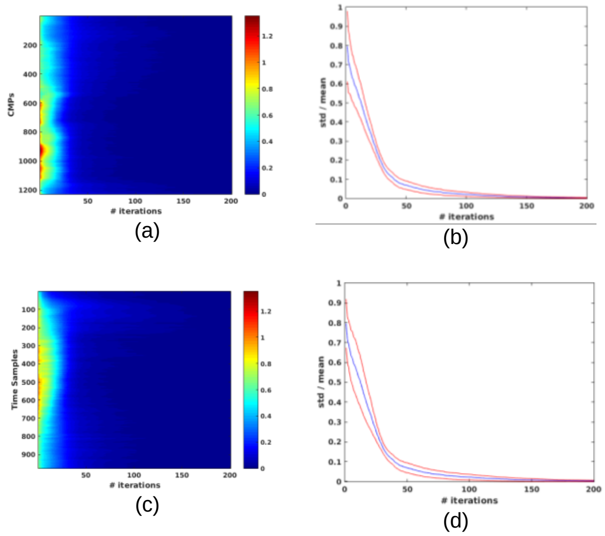


Figure 2 - sc_sm over iterations for Tacutu data using velocity guide. Plot of the colormap of the sc_sm over iterations (a) by CMP, and (c) by Time Sample. Plot of the mean (in blue) and standard deviation (in red) of the sc_sm over iterations (b) by CMP, and (d) by Time Sample.

Considering all the optimization runs necessary to stack the Tacutu data, we can observe in Figure 3 (without the usage of a velocity guide) and Figure 4 (with the usage of a velocity guide) in which iteration the optimization runs would stop for different thresholds for *sc_sm*. As expected, the higher the threshold value, the less iterations are required for convergence.

An interesting fact is that the images produced for these different thresholds for *sc_sm* and for the fixed number of iterations (500 and 200 without and with the usage of a velocity guide, respectively) are very similar to each other (see Figure 5 and 6). An important information about the stack images of Figure 5 and 6 is that we use *sc_sm* together with *sc_maxit* to limit the runtime of the stacking process. We set *sc_maxit* to 500 and 200 in the case of not using and using a velocity guide, respectively.

These stack images of Figures 5 and 6 show us that even using a rough threshold for *sc_dm*, we can obtain a stack image with good resolution and quality. So, we can save computational resources without giving up quality. The small differences among the images is only on the continuity of some stretches.

Conclusions

In this work, we analyze the convergence of a global optimization algorithm commonly used in the CRS stacking method, called differential evolution. As expected, the number of iterations necessary for each optimization run depends on the CMP and the time sample being processed. Thus, the choice of a fixed number of iterations as stopping criterion can compromise significantly the result.

To overcome this problem, we explore another type of stopping criterion based on the distribution of the population individuals. This new criterion speeds up the algorithm significantly without practically change the quality of the final stack image.

Since we can obtain good images with a rough threshold to this new stopping criterion, we can test user-defined CRS parameters quickly. Therefore, it helps the users in the data processing flow.

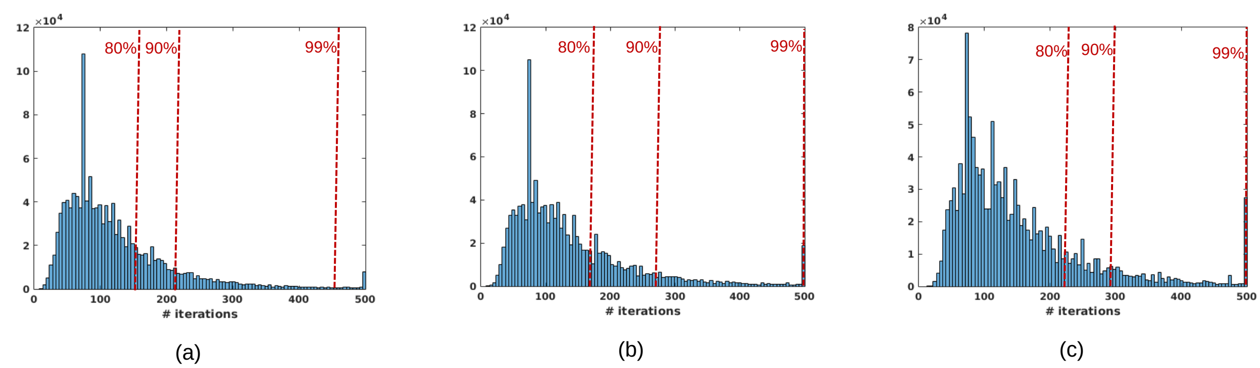
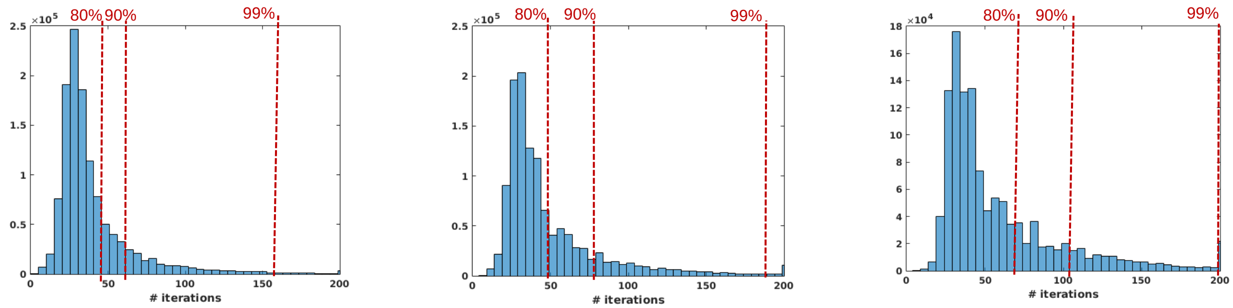
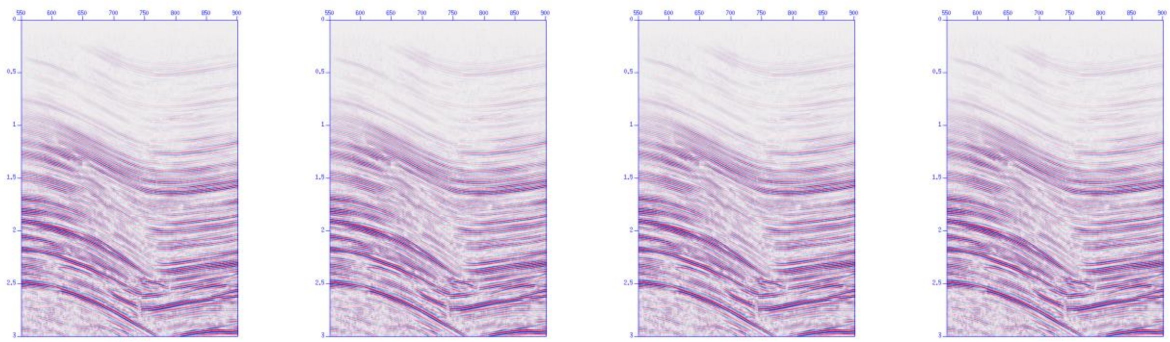


Figure 3 - Number of the iteration in which the user-defined threshold for sc_sm was reached (without using velocity guide) for threshold = (a) 0.15, (b)0.10, and (c) 0.05.



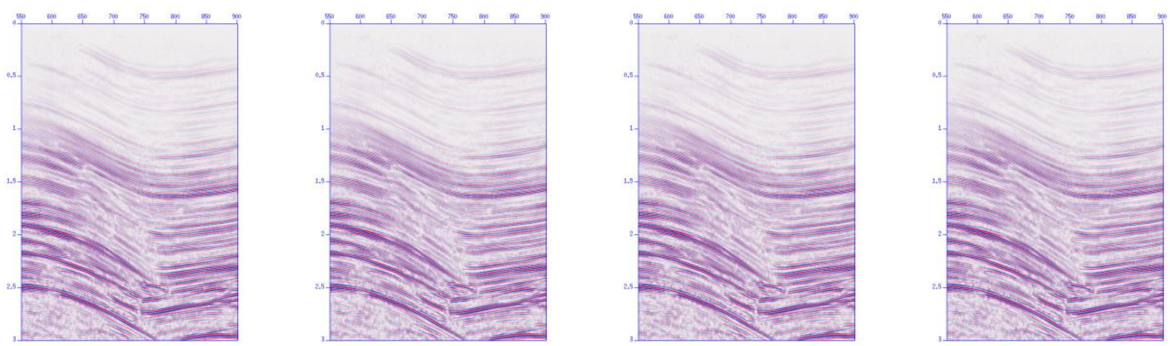
(a) (b) (c)

Figure 4 - Number of the iteration in which the user-defined threshold for sc_sm was reached (using velocity guide) for threshold = (a) 0.15, (b)0.10, and (c) 0.05.



(a) (b) (c) (d)

Figure 5 - Stack images of the Tacutu data using different thresholds for sc_sm and for a fixed number of iterations (without using velocity guide). For threshold = (a) 0.15, (b) = 0.10, (c) = 0.05, and for (d) 500 iterations.



(a) (b) (c) (d)

Figure 6 - Stack images of the Tacutu data using different thresholds for sc_sm and for a fixed number of iterations (using velocity guide). For threshold = (a) 0.15, (b) = 0.10, (c) = 0.05, and for (d) 200 iterations.

Acknowledgments

We would like to thank FAPESP for the financial support to the projects 2015/50475-0, 2016/04525-9, 2016/08866-5, 2016/12257-4, and 2016/21742-3. We also would like to thank professor Levy Boccato for the help in choosing DE stopping criteria.

References

Barros, T., Ferrari, R., Krummenauer, R., Lopes, R. R., Tygel, M. (2013). The Impact of the Parameter Estimation Strategy in the CRS Method. In: *13th International Congress of the Brazilian Geophysical Society*, Rio de Janeiro.

Barros, T., Ferrari, R., Krummenauer, R., Lopes, R. R., (2014). Global Optimization of the Parameters of the

Common Reflection Surface Traveltime Using Differential Evolution. In: *76th EAGE Conference and Exhibition*, Amsterdam.

Barros, T., Ferrari, R., Krummenauer, R., Lopes, R. R., (2015). Differential evolution-based optimization procedure for automatic estimation of the common-reflection surface traveltime parameters. In: *Geophysics*, 80(6), WD189-WD200.

Garabito, G., P. L. Stoffa, L. Lucena, J. C. Cruz. (2012). Part I - CRS stack: Global optimization of the 2D CRS-attributes. *Journal of Applied Geophysics*. 85: 92–101.

Mann, J., Jäger, R., Müller, T., Hubral, P. (1999). Common-reflection-surface-stack - a real data example. J. Appl. Geoph, 42(3,4):283–300.

Neidell, N. S., Taner, M. T. (1971). Semblance and other coherency measures for multichannel data. Geophysics, 36(3), 482-497.

Sen, M. K., Stoffa, P. L. (1995). *Global optimization methods in geophysical inversion*. Cambridge University Press.

Storn, R., K. Price. (1997). Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*. 11(4):341–359.

Taner, M. T., Koehler, F. (1969). Velocity spectra—digital computer derivation applications of velocity functions. *Geophysics*, *34*(6), 859-881

Zielinski, K., Laur, R. (2008). Stopping Criteria for Differential Evolution in Constrained Single-Objective Optimization. *Advances in Differential Evolution*. 143:111-138.