A comparative study on using metaheuristics for the seismic-ray-tracing problem

Article *in* Earth Science Informatics · March 2021

DOI: 10.1007/s12145-020-00549-3

4 authors:

Mario Alberto Aguirre López
Universidad Autónoma de Chiapas (UNACH)
16 PUBLICATIONS **11** CITATIONS
SEE PROFILE

F-Javier Almaguer
Autonomous University of Nuevo León
27 PUBLICATIONS **102** CITATIONS
SEE PROFILE

Martha-Selene Casas-Ramírez
Centro de Investigación en Matemáticas (CIMAT)
9 PUBLICATIONS **77** CITATIONS
SEE PROFILE

Roberto Soto-Villalobos
Autonomous University of Nuevo León
6 PUBLICATIONS **2** CITATIONS
SEE PROFILE

Earth Science Informatics manuscript No.
(will be inserted by the editor)

A comparative study on using metaheuristics for the seismic-ray-tracing problem

Mario A. Aguirre-López · Roberto Soto-Villalobos · Martha-Selene Casas-Ramírez · F-Javier Almaguer

Abstract In this paper we deal the multi-layer case in the seismic-ray-tracing problem. Each ray is defined by its departure angle, and it spreads according to Snell's Law; on the other hand, the medium of propagation is characterized by its density, the number of seismic layers, and the deep of these reflectors. Consider the above in the building the models, it allows that the travel time of a ray depends only on one variable but generates an excessive number of degrees of freedom in the system, which restricts the search space and makes it difficult to obtain an optimal solution. The foregoing motivates to solve the problem through a metaheuristic. We propose a solving methodology based

Mario A. Aguirre-López
Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico-Matemáticas, Ciudad Universitaria, Pedro de Alba s/n San Nicolás de los Garza, Nuevo León, Mexico
ORCID: 0000-0002-5191-3462
E-mail: marioal1906@gmail.com
Present address: National Technological Institute of Mexico, Technological Institute of Nuevo León, Department of Basic Sciences, Av. Eloy Cavazos 2001, Tolteca, Guadalupe, Nuevo León, Mexico of Mario A. Aguirre-López

Roberto Soto-Villalobos
Universidad Autónoma de Nuevo León, Facultad de Ciencias de la Tierra, Carretera a Cerro Prieto km 8, Ex. Hacienda de Guadalupe, Linares, Nuevo León, Mexico
ORCID: 0000-0002-3172-8673
E-mail: robsotov@gmail.com

Martha-Selene Casas-Ramírez
Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico-Matemáticas, Ciudad Universitaria, Pedro de Alba s/n San Nicolás de los Garza, Nuevo León, Mexico
ORCID: 0000-0001-5618-5719
E-mail: martha.casasrm@uanl.edu.mx

F-Javier Almaguer
Universidad Autónoma de Nuevo León, Facultad de Ciencias Físico-Matemáticas, Ciudad Universitaria, Pedro de Alba s/n San Nicolás de los Garza, Nuevo León, Mexico
ORCID: 0000-0001-9422-5259
E-mail: francisco.almaguermrt@uanl.edu.mx

on the shooting method immersed in the ray tracing methodology to find a solution to the initial value problem by using some metaheuristics, namely, Spiral Dynamics Inspired Optimization, Gravitational Search Algorithm and Genetic Algorithm. To our knowledge, this methodology has not been reported to solve such problem. There are not analytical solutions for models with two or more layers. A comparative study about the performance of the metaheuristics implemented is presented. The simulation results shows the competitiveness of the proposed algorithms, but in terms of solution quality and consumed time the Spiral Dynamics Inspired Optimization is better, followed by the Genetic Algorithm. Furthermore, the one-layer model was solved with the proposed algorithms and the results agree with the analytical solution reported in literature. In turn, our methodology provides better solutions than the Dix's equation and a metaheuristic-bending method for all the simulations we present.

Keywords Seismic ray tracing · Snell's law · Spiral Optimization · Gravitational Search · Genetic Algorithm · Computational Seismology

1 Introduction

Seismic reflection is one of the geophysical methods most used in hydrocarbons, watersheds and building studies. This is because such method seismic studies bring high precision, resolution and depth of penetration in characterizing lithologies, detecting density anomalies and relating them to subsoil structures, which are known as seismic layers ([Stein & Wysession 2003, Kumar & Sharma 2013]).

The operation of the seismic methods is based on detecting by mean of sensors (geophones) the arrival times at which an emerging elastic wave reaches the surface at a point of interest. Such wave is originated by a controlled source on the surface, so that the wave spreads into the subsoil throughout one or several seismic layers until it bounces (occurring the reflection) with one of the layers and then, it returns to the surface. Each one of the seismic layers are commonly modelled as homogeneous and isotropic layers and they are characterized by different densities, so that the wave propagates with different velocity in each layer and in consequence, it arrives the surface at a time that depends on its path, i.e., the arrival time of the wave carries information of the seismic layers ([Telford 1990,Ikelle & Amundsen 2005, Chopra & Castagna 2014]). With a set of points of interest, the time and magnitude of the detected waves are recorded and arranged in seismograms and finally, a large process is carried out to eliminate noise and errors of the method, see [Yilmaz 2008]. The problem of determining when and what wave arrives to the geophone is complicated if the number of layers is increased and if there is an inclining (or dipping) of the layers. Due to the complexity of the problem under these assumptions, an optimal solution can not be obtained and numerical methods have to be applied, for example, the ray tracing methodology. This applies especially for the planning, in which non destructive testing methods, models and simulations of low computational cost should be used for saving material, economic and time resources, see [Chopra & Castagna 2014] for more description of the common seismic methodology.

The ray tracing methodology is based on considering the front wave as a square wave, so that it can be represented by a linear ray with perpendicular direction to the front wave. This assumption is justified when the wave, which travels through the homogeneous and isotropic medium, is far enough from the source so that the front wave approximates to a plane wave regarding the size of the layers, see [Aki & Richards 2002]. In this context, the seismic ray travels throughout the subsurface without deviation until reach a seismic layer, where the refraction of the path occurs according to the Snell's law:

$$\frac{\sin\theta_{\text{re}}}{v_{\text{re}}} = \frac{\sin\theta_{\text{in}}}{v_{\text{in}}} \tag{1}$$

where θ_* and v_* are the angle between layers and velocity associated to the incident ($*$ =in) and refracted ($*$ =re) ray ([Constain & Coruh 2004, Červený 2001]). This applies for all the interfaces between reflectors. In this way, the ray tracing methodology avoids to solve the energy equation or some approximations of it, in such way that it is a fast and common technique to performing the preliminary tests of seismic study ([Telford 1990, Červený 2001]).

In turn, there are two different schemes to found the path of rays in the ray tracing technique: bending and shooting methods. Bending methods are based on proposing an arbitrary path linking the seismic source and the geophone, regardless of whether it is possible that it is a real trajectory or not, and then adjust the geometry of that path iteratively until it becomes a true trajectory, that is, until it satisfies Fermat's principle of least time (or equivalently Snell's law). Shooting methods are, in turn, more conceptually simple than bending methods; they propose an initial trajectory and then formulate the initial value problem to complete the total ray path by calculating the intersections between the straight ray and the reflectors and applying the Snell's law in such intersections. In both of them, diverse ways have been used in literature to solve the problem, reporting satisfactory and unsatisfactory results under different seismic models, see [Rawlinson et al. 2008, Prothero et al. 1988, Zhao et al. 2004, Nowers et al. 2014] for an extended literature review of the implementation of bending and shooting methods with ray tracing. This motivates to study how sensitive are the methods of solution with respect to the number and characteristics of the seismic layers (density, dip, discontinuities, anisotropy, among others).

To our knowledge, metaheuristics have only been used in bending methods. [Contreras et al. 2008] reported good results (errors of less than 0.02% in the reflection position) by applying a Genetic Algorithm (GA) to seismic reflection for 2D-models considering up to five reflectors; the number of population and iterations varied when increasing the number of reflectors, using the values 100 and ~ 8000, respectively, for the largest instances. Similar results with precision of less than 1 millimeter were reported by [Dávila-Torres 2012], who used Evolutionary Programming with a mean of 500 generations and

50-100 population for seismic reflection and refraction with instances considering up to seven reflectors; the computational time was 3-10 minutes in a typical desktop machine. In turn, [Marroquín Navarro 2012] used the results, model and methodology of Davila Torres to deal with seismic inversion (the inverse problem in seismic reflection) for instances considering two reflectors obtaining satisfactory results in seconds. Meanwhile, [Poormirzaee et al. 2015] carried out a performance analysis of Particle Swarm Optimization (PSO) algorithm and its parameters in seismic (refraction) inversion for synthetic models and a real seismic data set (also composed of two reflectors); they conclude about the reliability of the PSO inversion method in seismic data interpretation with an acceptable misfit and convergence speed. In all these investigations, both the synthetic models and the true data considered allow only a positive (in-depth) gradient of velocities in the seismic reflectors, while the dip is arbitrary. Other applications of metaheuristics in ray tracing problems and seismic data can be found in [Benamou 1996,Conti et al. 2013, Vargas Contreras 2017,Poormirzaee et al. 2019].

In this paper we implement the shooting method of the ray tracing problem in seismic reflection from a metaheuristic context. The algorithm developed solves the initial value problem by using the metaheuristics to approach the optimal ray in each step of the shooting method. A comparative study between the metaheuristics used is carried out and discussed for a seismic model with different configurations. The seismic model consists of homogeneous reflectors with an arbitrary associated velocity and dip, without discontinuities. The paper is structured as follows: The mathematical model and constraints of the problem are introduced in Section 2. In Section 3, we present the metaheuristics and their implementation in the algorithm. Next, the results and comparisons between metaheuristics are shown in Section 4. In addition, the comparison between the results obtained and the solutions in the literature is presented. Finally, Section 5 is dedicated to the concluding remarks and future work.

2 Mathematical model

In this section the problem and a mathematical formulation are described. The physical model raises the assumption of the ray tracing methodology. Each ray is defined by its departure angle θ_0. All the rays are originated by the seismic source at the initial position (x_s, y_s) and each one of them are a possible solution for reaching the geophone at the position (x_g, y_g). On the other hand, there are a set I of n homogeneous-isotropic reflectors in the subsoil, which could be almost inclining; this means that foldings, faults or other geological events are not considered. Thus, the reflectors numbered by the $i-$index, $i \in I$, such that the $i-$th reflector is defined by their depth h_i, dip β_i and velocity v_i, while the parameters (h_0, β_0, v_0) are associated to the surface, see Figure 1.

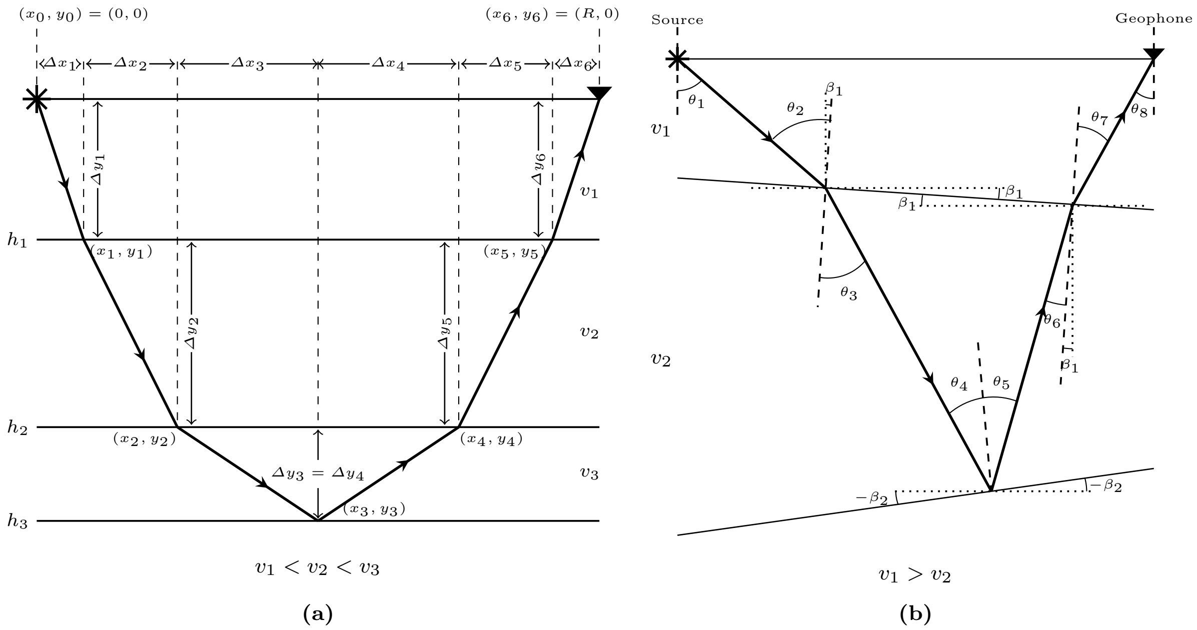

Fig. 1: A seismic ray path in a layered medium. **(a)** Layered medium with horizontal reflectors. Coordinates of the intersections between ray and seismic reflectors. **(b)** Layered medium with dipping reflectors. Angles of incidence and angles of refraction at each interface through which passes the ray.

The ray travels in a straight line until it intersects the first reflector with an incident angle θ_1; then, if there is more than one reflector, the ray continues its course refracting at an angle θ_2 from the first reflector, see Figure 2 (a) for visualization of the refraction; the process continues until the ray intersects the last reflector n, where it occurs a total reflection. Including the departure angle, there is a total of $2n$ angles before the angle produced by the total reflection. Then, the ray goes up by a similar process until it reaches the surface.

In this manner, let us properly define the sets, parameters and decision variables involved in the problem.

Sets

$I = \{1, \ldots, n\}$ Set of seismic reflectors

Parameters

h_i: Depth of reflector $i \in I$
β_i: Dip of reflector $i \in I$
v_i: Velocity of reflector $i \in I$

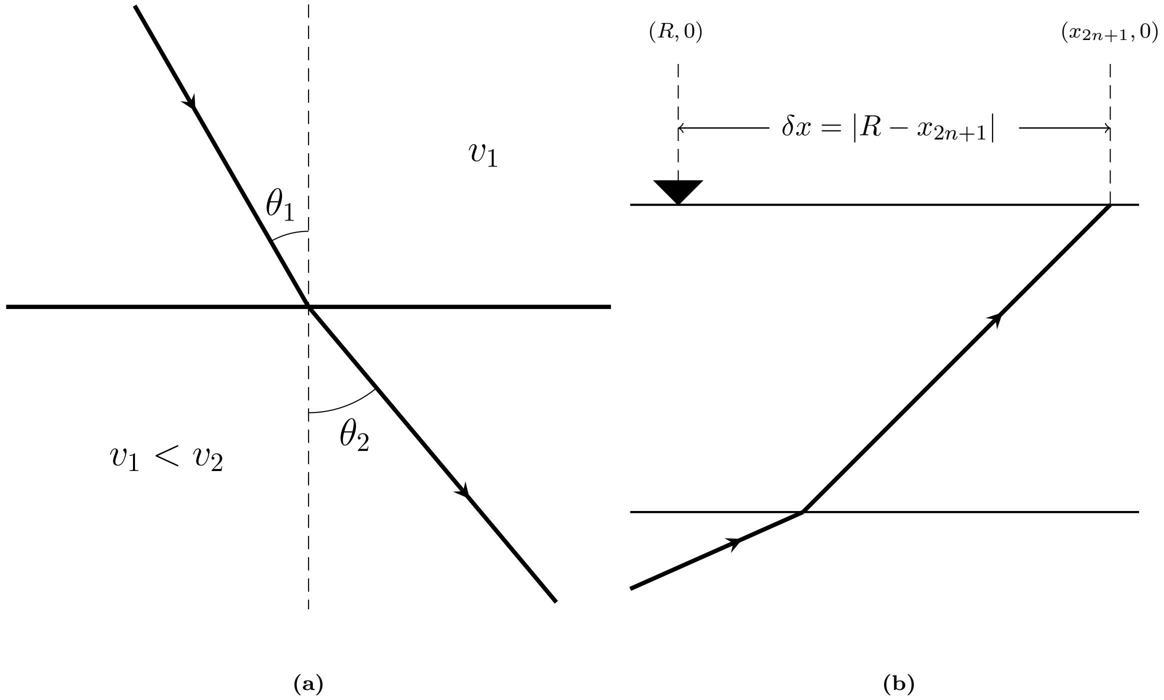

Fig. 2: **(a)** Scheme of the Snell's law for elastic waves in a medium with different densities. V_1 is the velocity associated with the first medium, and V_2 is the velocity associated with the second one. **(b)** End of a ray that reaches the surface: Visualization of the objective function of the problem.

R: Distance between the seismic source and the geophone or wave receiver.
(x_s, y_s): Coordinate where the seismic source is located.
(x_g, y_g): Coordinate where the geophone is located.

Decision variables
θ_0: Departure angle

The incident and refracted angles are in function of the departure angle by means of the geometry of the reflectors and the Snell's law, respectively. Incident and refracted angles are defined as elements of the vector $\boldsymbol{\theta} \equiv (\theta_0, \cdots, \theta_j, \cdots, \theta_{4n-1})$, where the angle θ_j is incident if its $j-$index is odd and it is refracted if its $j-$index is even.

The indexes of the incident angles are connected to the indexes of the reflectors i by

$$j_{\text{odd}} = \begin{cases} 2i - 1, & \text{if } j_{\text{odd}} \leq (2n-1) \\ 4n - (2i+1), & \text{otherwise} \end{cases}, \tag{2}$$

where the first case corresponds to incident angles before the total reflection at the deepest reflector, where the second case corresponds to angles after the total reflection. In a similar way, the indexes of the refracted angles are connected by

$$j_{\text{even}} = \begin{cases} 2i, & \text{if } j_{\text{even}} < (2n) \\ 4n - (2i), & \text{if } j_{\text{even}} > (2n) \end{cases} . \tag{3}$$

The indexation of the intersections are numbered by the sequence $(x_0, y_0), (x_1, y_1), \ldots, (x_{2n}, y_{2n})$, as illustrated in Figure 1 (a). With this indexing, there is not a refracted angle for the last deepest reflector since there occurs the reflection, therefore the corresponding reflected angle is the element θ_{2n}. In addition, the indexation ends at θ_{4n-1}, this is because neither refraction occurs when the ray reaches the surface (at (x_{2n}, y_{2n})). For visualization, consider a model with only one reflector, there are not refracted angles for such instance.

Thus, the first intersection of the ray is calculated by using the departure angle θ_0 (which denotes the slope of the ray) and the dips of both the first reflector β_1 and the surface β_0. The ray is then refracted, and its new direction is defined by the refracted angle θ_2, which is calculated from the Snell's law (1) by using the previous incident angle θ_1 and the velocities v_1 and v_2. The process for determining the total ray path continues until to reach the intersection of the ray and the deepest reflector; the reflection occurs and then, a similar process begins until to reach the surface.

However, this task is complicated since we look for a ray without backwards throughout its trajectory (because we are interested in the first arrivals), except for the change of direction in the deepest reflector. Moreover, the increasing of the number of seismic reflectors causes problems with the constraint of the Snell's Law, in order to calculate the refracted angle at each interface. In fact, this leads to physically non-existent paths because it can happen that the Snell's law is not satisfied for all refracted angles throughout the trajectory. In view of these facts, an algorithm is needed to calculate the final position (intersection) of the ray. Such an algorithm is presented in Figure 2.1.

Now, let us describe the constraints of the problem. Both seismic source and geophone are located on the horizontal surface, so $y_s = y_g = 0$. The reflectors cannot intersect between them in all the domain. h_i should be seen as the depth at which a vertical line intersects both the surface and the $i-$th reflector, so the $0 < h_1 < h_i < h_n$ vertical reflectors cannot be considered in the present model. In turn, β_i is the angle inclination defined between $(-90^o, 90^o)$ and in clockwise direction. At last, v_i refers to the constant velocity with which the ray travels through the $i-$th reflector. Figure 1 (b) outlines the numeration of the parameters.

Finally, the objective function consists in minimizing the distance between the last position of the ray (the intersection point between the ray and the surface) and the geophone's position, so that we want to find

$$\text{Min } \delta_x = |R - x_{2n}|, \tag{4}$$

where the variable of the model θ_0 takes part of the objective function as described in the algorithm shown in Figure 2.1. See Figure 2(b) for visualization.

Algorithm 2.1 Algorithm to obtain the final intersection of the ray with the surface (x_{2n}, y_{2n}) by using the variable θ_0.

```
Require: $\theta_0$                                                    ▷ Departure angle
  for all $i = 1 : n$ do                                               ▷ Before the reflection
     $j_{\text{odd}}$ and $j_{\text{even}}$ defined as in the *first* case of (2) and (3), respectively
     $m_i = \tan\left(\frac{3\pi}{2} + \theta_{i-1} - \beta_{i-1}\right)$                    ▷ Slope of the ray
     $x_i = \frac{(h_i - y_{i-1}) + m_i x_{i-1}}{m_i - \tan(\beta_i)}$          ▷ $(x, y)$ intersection with the $i$−th reflector
     $y_i = y_{i-1} + m_i (x_i - x_{i-1})$
     if $(x_i > x_{i-1}$ & $y_i < y_{i-1})$ then                         ▷ Not backward condition
        $\theta_{j_{\text{odd}}} = \theta_{j_{\text{even}}} + (\beta_i - \beta_{i-1})$                     ▷ Incident angle
        $\theta_{j_{\text{even}}} = \arcsin\left(\frac{v_{i+1}}{v_i} \cdot \sin\left(\theta_{j_{\text{odd}}}\right)\right)$             ▷ Refracted angle
        if $\theta_{j_{\text{even}}} \notin \mathbb{R}$ then                                ▷ Refraction condition
           Stop                                         ▷ Failure: the ray is unsuccessful
        end if
     else
        Stop                                            ▷ Failure: the ray is unsuccessful
     end if
  end for

  $\theta_{j_{\text{even}}} = \theta_{j_{\text{odd}}}$                                              ▷ Reflected angle

  for all $i = n - 1 : 0$ do                                           ▷ After the reflection
     $j_{\text{odd}}$ and $j_{\text{even}}$ defined as in the *second* case of (2) and (3), respectively
     $m_{2n-i} = \tan\left(\frac{\pi}{2} - \theta_{i-1} - \beta_{i-1}\right)$                  ▷ Slope of the ray
     $x_{2n-i} = \frac{(h_i - y_{2n-i-1}) + m_i x_{2n-i-1}}{m_{2n-i} - \tan(\beta_i)}$    ▷ $(x, y)$ intersection with the $i$−th reflector
     $y_{2n-i} = y_{2n-i-1} + m_{2n-i} (x_{2n-i} - x_{2n-i-1})$
     if $(x_i > x_{i-1}$ & $y_i > y_{i-1})$ then                         ▷ Not backward condition
        $\theta_{j_{\text{odd}}} = \theta_{j_{\text{even}}} + (\beta_i - \beta_{i+1})$                     ▷ Incident angle
        $\theta_{j_{\text{even}}} = \arcsin\left(\frac{v_i}{v_{i+1}} \cdot \sin\left(\theta_{j_{\text{odd}}}\right)\right)$             ▷ Refracted angle
        if $\theta_{j_{\text{even}}} \notin \mathbb{R}$ then                                ▷ Refraction condition
           Stop                                         ▷ Failure: the ray is unsuccessful
        end if
     else
        Stop                                            ▷ Failure: the ray is unsuccessful
     end if
  end for
  return $(x_{2n}, y_{2n})$                                              ▷ Final position
  ▷ Reminder: The refraction does not occur at the deepest reflector and at the surface, so
  the *Refraction condition* does not apply for the last iteration of each loop.
```

3 Methodology

Metaheuristics are iterative stochastic search algorithms based on looking for the optimal solution by means of a set of search agents or population that evaluate the objective function of the problem in many points in the solution space. They are used to solve complex optimization problems, providing good solutions and convergence in a reasonable computation time. As we mentioned above, there is an excessive number of degrees of freedom in the model, which restricts the search space and makes it difficult to obtain an optimal solution, specially for models with more than one layer, for which there is no analytical solution. Therefore, the use of metaheuristics is suitable for the present problem.

Since this paper intends to perform a comparative analysis, we have implemented three metaheuristics based on populations but that differ in their operation: Genetic Algorithm (GA), Gravitational Search Algorithm (GSA) and Spiral Optimization (SO). Namely, an initial population is created randomly in all algorithms, then that population evolves until a stop criteria is reached, in this case, a maximum number of iterations. However, both conception and operation of each metaheuristic differ in a large manner. In brief, SO is considered a deterministic algorithm because stochasticity is only present at the creation of the initial population, then its operation depends on the parameters of the spiral; in turn, GSA is based on the law of gravity but including random values through the operation; finally, GA is a full stochastic metaheuristic based on the theory of evolution, and it is one of the first metaheurisitcs reported in the literature. In this way, the comparison we make below considers a variety of methodologies within the heuristic context.

The above observations and personal interests related to understand the behavior of such metaheuristics in seismic problems are the only reasons taken into account for their use. Each metaheuristic is described in detail below.

3.1 Genetic Algorithm

Genetic algorithms (GA) are based on the evolution of biological systems by Darwin's theory ([Koziel & Yang 2011,Kalos & Whitlock 2008]). They have been widely used in studies of various issues ([Osman & Laporte 1996,Bergey et al. 2003, Janiaka & Portmann 1998,Dias & de-Vasconcelos 2002,Ooi & Tan 2003]), some of them related to problems of the area of Earth Sciences as [King 1995, Song 2014,Balkaya 2013,Ramillien 2001]. There are several types of genetic algorithms, however the main components of these metaheuristics are crossing, mutation and selection mechanism. The pseudocode for the GA is shown in Figure 3.1.

Initial population

Chromosome coding is very important in genetic algorithms. We consider a binary vector of size ν, where each element represents a possible departure angle. We generated a random number between 0 and 1, if the number was less than or equal to 0.5 then the gene on the chromosome takes the value of 1, and takes the value of 0 in the other case. Then, the hash function is used to assign a number to each chromosome (solution), that is, the departure angle. Afterwards, the feasibility is analyzed in the calculation of the ray trajectory, see the algorithm in Figure 2.1; if the solution is infeasible, it is discarded.

Selection mechanism
The chromosomes of the initial population are ordered according to the following formula:

$$p_i = \frac{\delta_{x_i}}{\sum_{j=1}^{N} \delta_{x_j}} \tag{5}$$

where p_i denotes the fitness (the fraction that represents the evaluation of the individual's objective function of individual with respect to the total sum of the evaluations) of the individual i. Then the roulette selection is carried out according to [Goldberg 1989], which consists in altering the order of the individuals in the following way: each individual i is associated with the cumulative sum of the fitness $S_i = \sum_{k=1}^{i} p_k$.

The selection begins by placing in the first position $j = 1$ the last individual that meets the constraint $s \leq S_i$, where s is a random value between 0 and 1, thus simulating a roulette. Individuals are reviewed in descending order of fitness. The selection process is repeated for each individual.

Finally, the fittest chromosomes according to the objective function are selected; some *GA*'s versions delete to the worst solutions to assure that the best solutions remain in the population.

Genetic operators
The objective of the genetic operators is to provide diversity which allows to search the solution space. Two different operators are considered: mutation and crossover.

The mutation allows to replicate almost all the genetic information of a father but with a small variation, in this way an offspring is obtained. We generated a random number between 1 and the chromosomal dimension, then we analyzed that position on the chromosome and if there was a 1 in the position, we changed it to 0; and vice versa. The feasibility of the mutated solution is analyzed in the same way described in the generation of initial solutions.

The crossover allows the genetic recombination of both parents, generating two offsprings. We implement the technique of a crossing point. We randomly generated a point, then we exchanged the genes of the first father that were left before the crossing point with the genes of the second father that were left after the crossing point; we do the same to generate the other offspring. The feasibility of offsprings is analyzed in the same way described above.

Algorithm 3.1 Genetic Algorithm

```
procedure GENETIC_ALGORITHM($NumberOfGenerations, size(Population)$)
    $IterCount \leftarrow 1$
    $Population \leftarrow 0$
    $Fitness \leftarrow 0$
    $Pool \leftarrow 0$
    $Parent_1 \leftarrow 0$
    $Parent_2 \leftarrow 0$
    $Offsprings_c \leftarrow 0$
    $Offsprings_m \leftarrow 0$
    $Flag_c \leftarrow 0$
    $Flag_m \leftarrow 0$
    for all $i = 1...size(Population)$ do
        $Population(i) \leftarrow GenerateRandomSolution$
        $Fitness(i) \leftarrow EvaluateObjectiveFunction(Population(i))$
    end for
    while $(IterCount \leq NumberOfGenerations)$ do
        $Population \leftarrow SortRussianRoulette(Population)$
        for all $i = 1...size(Population)$ do
            if $(random_1 \leq prob_c)$ then
                $Parent_1 \leftarrow Population(i)$
                $Parent_2 \leftarrow Population(i+1)$
                $Offsprings_c \leftarrow Crossover(Parent_1, Parent_2)$
            else
                $Flag_c \leftarrow 1$
            end if
            if $(random_2 \leq prob_m)$ then
                if $(Flag_c = 0)$ then
                    $Offsprings_m \leftarrow Mutation(Offsprings_c)$
                else
                    $Offsprings_m \leftarrow Mutation(Parent_1, Parent_2)$
                end if
                $Pool \leftarrow Pool \cup Offsprings_m$
            else
                $Flag_m \leftarrow 1$
                if $(Flag_c = 0)$ then
                    $Pool \leftarrow Pool \cup Offsprings_c$
                else
                    $Pool \leftarrow Pool \cup Parent_1$
                    $Pool \leftarrow Pool \cup Parent_2$
                end if
            end if
            $Flag_c \leftarrow 0$
            $Flag_m \leftarrow 0$
        end for
        $Fitness \leftarrow 0$
        $Fitness \leftarrow EvaluateObjectiveFunction(Pool)$
        $Population \leftarrow 0$
        $Population \leftarrow Population \cup Pool$
        $Pool \leftarrow 0$
        $IterCount \leftarrow IterCount + 1$
    end while
    return $Best$
end procedure
```

3.2 Gravitational Search Algorithm

Gravitational Search Algorithm (*GSA*) is based in Newton's Law of Universal Gravitation

$$F = G\frac{m_1 m_2}{r^2}, \tag{6}$$

which express that the force F exerted by the mass m_2 and acting on the mass m_1 is inversely proportional to the square of the distance r between both masses. The relation is closed by the gravitational constant G. In this sense, the operation of the *GSA* is described in three steps as follows.

Initial population
Each search agent (departure angle) consists of a generated random number between 0 and 90[1]. Similarly to the procedure in the *GA*, the feasibility is analyzed in the calculation of the ray trajectory, according to the algorithm shown in Figure 2.1; if the solution is infeasible, it is discarded.

Selection mechanism
We connect the $i-$th search agent to a mass m_i by the formula

$$m_i = \frac{\delta_{x_i} - \underset{j}{\max}\delta_{x_j}}{\underset{j}{\min}\delta_{x_j} - \underset{j}{\max}\delta_{x_j}}, \tag{7}$$

where the term $\underset{j}{\max}\delta_{x_j}$ indicates the maximum of the objective function over all the $j-$agents. So, the mass m_i is expressed in relative terms by $M_i = \frac{m_i}{\sum_{j=1}^{N} m_i}$. In this way, the agents that provide the best solutions are the heaviest masses, and the solutions are closed into the interval $[0, 1]$. The solutions are sorted in descending order.

The value of G is calculated and updated in each k iteration in accordance with the formula shown by [Nezamabadi et al. 2009], such that $G_k = G_0 e^{-\alpha\frac{k}{K}}$, where G_0, α are parameters and K is the total number of iterations. In this way, the agents are evaluated by a composed elitist-random selection.

Gravitational search operators
As mentioned above, the mechanism of operation in the *GSA* methodology is based on the law of gravity (6). The force F_{ij} exerted on the mass M_i by the mass M_j is $G_k\frac{M_i M_j}{r_{ij}^2}$, where r_{ij} is the distance between both masses, see Figure 3. For a fast convergence of the algorithm, it is introduced a parameter p, which indicates the percent of agents (normalized to 1) that are able of exert force in the last iteration; this means that the number of agents (masses) exerting force on the others is decreasing linearly throughout the iterations, so that only the $round(pN)$ heaviest masses exert force in the last iteration. In consequence the total force exerted on the mass M_i is the sum of all the forces exerted

[1] It is important to mention that no binary code is used for *GSA* and *SO* metaheuristics. It is only implemented by *GA*.

on it, so that $\sum_{j=1,j\neq i}^{N_k} rand_j F_{ij}$, where N_k is the number of masses exerting force in the $k-$iteration. The random number $rand_j$ is introduced to give a stochastic characteristic to the heuristic. In turn, the acceleration of the mass M_i is

$$A_i = \frac{\sum_{j=1,j\neq i}^{N_k} rand_j F_{ij}}{M_i},$$

whereas the velocity U_i of the mass M_i is obtained by the sum of the initial velocity V_0 in each iteration ($V_0 = 0$ in the first iteration) and the acceleration:

$$U_i = V_0 * rand_i + A_i, \tag{8}$$

where the velocity random number $rand_i$ increases the stochasticity to the operation.

Finally, the $i-$th solution is updated by adding equation (8) to the $i-$th solution in the last iteration. The pseudocode for the *GSA* is shown in Figure 3.2.

For more details see [Nezamabadi et al. 2009]. Some applications and versions of *GSA* can be found in [Hosseinabadi et al. 2014,Hosseinabadi et al. 2012, Eldos & Al-Qasim 2013,Xu & Zhang 2014,Rashedi et al. 2011].

Algorithm 3.2 Gravitational Search Algorithm

```
procedure GRAVITATIONAL_SEARCH_ALGORITHM($NumberOfIterations, size(Population), \alpha, p, G_0$)
    $IterCount \leftarrow 1$
    $Population \leftarrow 0$
    $Fitness \leftarrow 0$
    $Mass \leftarrow 0$
    $Acceleration \leftarrow 0$
    $V_0 \leftarrow 0$
    $Velocity \leftarrow 0$
    for all $i = 1...size(Population)$ do
        $Population(i) \leftarrow GenerateRandomSolution$
        $Fitness(i) \leftarrow EvaluateObjectiveFunction(Population(i))$
        $Mass(i) \leftarrow CalculateMassValue(Population(i))$
    end for
    while ($IterCount \leq NumberOfIterations$) do
        $Population \leftarrow SortDescendingOrderMass(Population)$
        $Fitness \leftarrow SortDescendingOrderMass(Population)$
        for all $i = 1...size(Population)$ do
            $Acceleration(i) \leftarrow CalculateAcceleration(Population(i), Mass(i), \alpha, p, G_0)$
            $Velocity(i) \leftarrow CalculateVelocity(V_0, Acceleration(i))$
            $Population(i) \leftarrow Population(i) + Velocity(i)$
            $Fitness(i) \leftarrow EvaluateObjectiveFunction(Population(i))$
            $Mass(i) \leftarrow CalculateMassValue(Population(i))$
        end for
        $V_0 \leftarrow Velocity$
        $IterCount \leftarrow IterCount + 1$
    end while
    return $Best$
end procedure
```

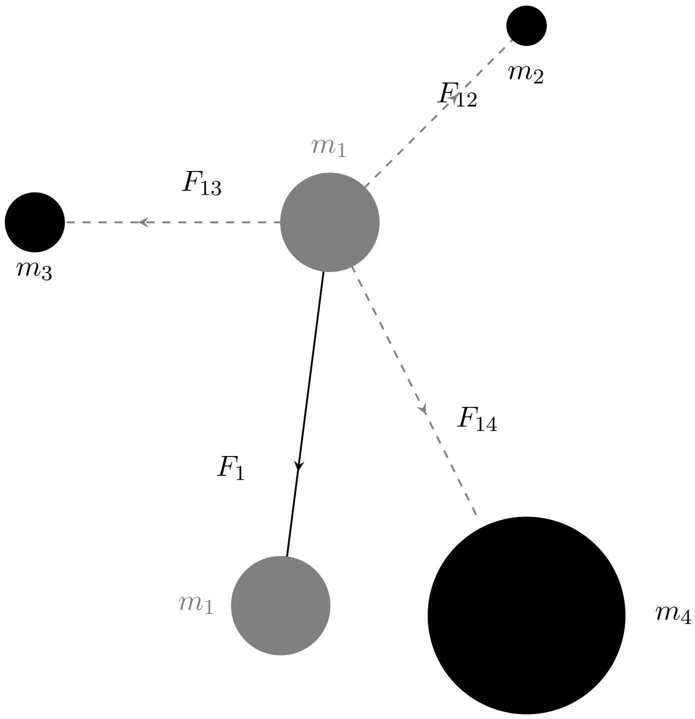

Fig. 3: Visualization of *GSA*. Acceleration of the mass m_1 in the direction of the resultant force F_1 acting on it by the masses m_2, m_3, and m_4.

3.3 Spiral Optimization

Spiral Optimization (*SO*) is a deterministic metaheuristic inspired by the phenomena occurring in nature in a spiral shape. This means that the diversification and intensification of the solutions are achieved by means of a spiral approaching to the best solution. Namely, the primary stage of *SO* is focused in diversifying the solutions whereas the second stage consists in the intensification by simple construction of a spiral, see [Tamura & Yasuda 2011].

Initial population

Similarly to *GSA*, each search agent (departure angle) consists of a generated random number between 0 and 90. However, due to the performance of the *SO*, the departure angle is obtained from calculating the slope of the straight line that joins a two-dimensional point and the origin of the system. So, the solution space is a two-dimensional search space. In addition, the feasibility is analyzed in as in *GA* and *GSA*.

Selection mechanism

An elitist selection is implemented. All the search agents are evaluated according to the objective function (4) and the best (that that minimizes (4) in each iteration) is tagged as x^*.

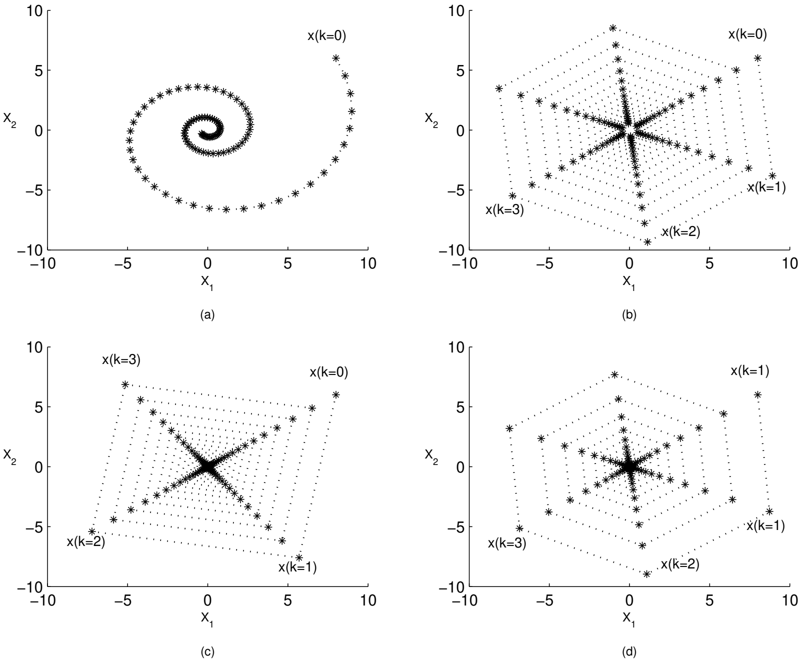

Fig. 4: Search of an agent x with initial position at coordinate $(8, 6)$ to a fixed solution x^* located at coordinate $(0, 0)$ using SO with a $K = 100$ iterations. Case with different search parameters r and ϕ. **Up-left:** $r = 0.97, \phi = \pi/20$. **Up-right:** $r = 0.97, \phi = \pi/3$. **Bottom-left:** $r = 0.95, \phi = \pi/2$. **Bottom-right:** $r = 0.95, \phi = \pi/3$.

Spiral optimization operators

As mentioned above, the search of the solution space is performed in an spiral shape. To achieve this, the best agent x^* acts as an attractor for all the others $N - 1$ agents. The agents x_i, $i = 1, \ldots, N$ s.t. $x_i \neq x^*$, rotate around x^* in each iteration with an constant rotation angle ϕ, $0 \leq \phi < 2\pi$, by means of a rotation matrix $R(\phi)$, which is defined as

$$R(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi \\ \sin\phi & \cos\phi \end{bmatrix}.$$

In turn, the x_i agents approaches in each iteration to the best solution by a convergence rate r, $0 < r < 1$, of the distance between each agent x_i and

x^*. For visualization of the performing of SO, Figure 4 illustrates four special cases supposing a fixed best solution (at the center of the spiral) for all the $K-$iterations, and varying the parameters ϕ and r.

According to [Tamura & Yasuda 2011], the complete formula for the search of the x_i agent in the k-th iteration is written by:

$$x_i(k) = (rR(\phi)x_i(k-1)) - (rR(\phi) - I)x^*(k-1) \tag{9}$$

where $I \in M_{2\times 2}$ is the identity matrix of dimension two. It can be noted that SO is a deterministic metaheuristic.

The pseudocode for the SO is shown in Figure 3.3. Other applications and variants of the SO algorithm can be seen in [Siddique & Adeli 2014, Tsai et al 2014,Benaslaa et al. 2014,Nasir & Tokhi 2015,Kania & Sidarto 2016].

Algorithm 3.3 Spiral Optimization Algorithm

```
procedure SPIRAL_OPTIMIZATION_ALGORITHM($NumberOfIterations, size(Population), r, \phi$)
    $IterCount \leftarrow 1$
    $Population \leftarrow 0$
    $Fitness \leftarrow 0$
    $center \leftarrow 0$
    $Solutions \leftarrow 0$
    for all $i = 1...size(Population)$ do
        $Population(i) \leftarrow GenerateRandomSolution$
        $Fitness(i) \leftarrow EvaluateObjectiveFunction(Population(i))$
    end for
    $center \leftarrow SelectBest(Population)$
    while ($IterCount \leq NumberOfIterations$) do
        for all $i = 1...size(Population)$ do
            $Solutions(i) \leftarrow CalculateRotation(center, r, \phi)$
        end for
        $Fitness \leftarrow EvaluateObjectiveFunction(Solutions)$
        $center \leftarrow SelectBest(Solutions)$
        $IterCount \leftarrow IterCount + 1$
    end while
    return $Best$
end procedure
```

4 Computational experimentation

In order to evaluate the quality of the solutions obtained with the metaheuristics described in Section 3 (GA, GSA and SO), we prove the implementation of the methodology in our model with four different features according to Section 2. Specifically, the numerical proves were chosen in order to evaluate the behavior of the metaheuristics when increasing the complexity of the model and decreasing the solution space.

The computational environment consisted of Intel Core i3 8th generation, 8 GB memory, Windows 10 OS, and Matlab programming language. The results for all proves were obtained by 100 runs, with 100 search agents and 500 iterations (for each run). The computational time varies about 10, 30 and 60 seconds by using the SO, GSA and GA, respectively, for the shortest instances to 30, 120 and 500 seconds for the largest instances, considering all the 100 runs. The values of the parameters used for each metaheuristic are listed in Table 1. The parameters were setting according to those that provide a better performance of the corresponding metaheuristic.

Table 1: Metaheuritic parameters.

Metaheuristic	**Parameter(s)**		
Spiral Optimization	$r = 0.95$	$\theta = \pi/60$	
Gravitational Search	$p = 0.02$	$G_0 = 100$	$\alpha = 20$
Genetic Algorithm	$Mutation = 0.1$	$Cross = 0.9$	$\nu = 50$

4.1 Model 1

The first model consists of a medium with only one layer without dip; the reflector is located at a depth of 15 m and it has an associated wave velocity of 1400 m/s, see Table 2. Two geophones are located: i) at source location, and ii) to 5 m source.

Table 2: Parameters of Model 1.

Reflector	Velocity	Depth	Dip
1	1400 m/s	15 m	0°

The mean value of the objective function, the ray's travel time and their corresponding root-mean-square errors (RMS-E) are shown in Table 3 for the two geophones. Travel times are also compared with the analytical solution cited in [Burger & Burger 1992] and with the approximation obtained by implementing the methodology of the Dix's equation for a medium with horizontal layers [Burger & Burger 1992,Grenchka et al. 1999].

The calculation of the travel times is carried out by applying the formula time $= \frac{\text{Distance}}{\text{Velocity}}$ for all the parts of the ray throughout its trajectory. The trajectory is decomposed in $2n$ parts, so that the time t_i taken for a ray travel the distance d_i between the coordinates (x_i, y_i) and (x_{i+1}, y_{i+1}), with a constant velocity v_i, is $t_i = d_i/v_i$, see Figure 1 (a). In the notation $\Delta x_i = x_{i+1} - x_i$, $\Delta y_i = y_{i+1} - y_i$, $\forall_{1 \leq i \leq n-1}$, the time taken for propagate the seismic ray from source G to geophone F is

$$T = \sum_{i=1}^{n} t_i + \sum_{i=1}^{n} t_{n+i} \equiv \sum_{i=1}^{n} \frac{[(\Delta x_i)^2 + (\Delta y_i)^2]^{1/2}}{v_i} + \sum_{i=1}^{n} \frac{[(\Delta x_{i+n})^2 + (\Delta y_{i+n})^2]^{1/2}}{v_{n+1-i}} \tag{10}$$

In (10), the first term on right side of the equation corresponds to the travel time of seismic ray from the source to the last interface, while the second corresponds to the upward path until reaching the geophone.

Table 3: Results for Model 1, and analytical travel times.

Metaheuristic	Geophone	Measurement	Objective function (m)	Travel time (ms)
SO	i)	Mean	5.1212 $\times 10^{-11}$	21.4286
		RMS-E	1.2586 $\times 10^{-10}$	3.2600 $\times 10^{-12}$
	ii)	Mean	1.5621 $\times 10^{-11}$	21.7242
		RMS-E	0.2076 $\times 10^{-10}$	3.0678 $\times 10^{-12}$
GSA	i)	Mean	2.0679 $\times 10^{-1}$	21.4297
		RMS-E	2.2990 $\times 10^{-1}$	2.7462 $\times 10^{-3}$
	ii)	Mean	1.1877 $\times 10^{-1}$	21.7230
		RMS-E	1.5291 $\times 10^{-1}$	2.2716 $\times 10^{-2}$
GA	i)	Mean	1.0455 $\times 10^{-6}$	21.4286
		RMS-E	2.8069 $\times 10^{-7}$	3.0703 $\times 10^{-14}$
	ii)	Mean	3.5228 $\times 10^{-3}$	21.7242
		RMS-E	1.9862 $\times 10^{-2}$	2.3504 $\times 10^{-3}$
Dix's equation	i)	Direct method	—	21.4286
	ii)	Direct method	—	21.7241
Burger and Burger (1992)	i)	Analytical solution	—	21.4
	ii)	Analytical solution	—	21.7

4.2 Model 2

Model 2 consists of six horizontal layers (reflectors) with positive velocity gradient, as shown in Table 4. The last reflector is located at a depth of 1250 m. There is a geophone to 1250 m source. Results for Model 2 are shown in Table 5. They are compared with the Dix's equation, whereas no analytical solution is reported for this instance.

Table 4: Parameters of Model 2.

Reflector	Velocity	Depth	Dip
1	1500 m/s	125 m	0°
2	1700 m/s	235 m	0°
3	3500 m/s	600 m	0°
4	4500 m/s	730 m	0°
5	5000 m/s	1090 m	0°
6	6000 m/s	1250 m	0°

Table 5: Results for Model 2.

Metaheuristic	Layer	Measurement	Objective function(m)	Travel time (ms)
SO	i)	Mean	1.4231 $\times 10^{-11}$	849.8366
		RMS-E	1.4672 $\times 10^{-11}$	1.3333 $\times 10^{-11}$
	ii)	Mean	5.4957 $\times 10^{-1}$	830.6100
		RMS-E	4.8981	2.7825
	iii)	Mean	1.5566 $\times 10^{-11}$	686.9835
		RMS-E	3.5774 $\times 10^{-11}$	9.2250 $\times 10^{-12}$
	iv)	Mean	1.7815 $\times 10^{-11}$	705.4806
		RMS-E	5.7878 $\times 10^{-11}$	1.1485 $\times 10^{-11}$
	v)	Mean	1.1080 $\times 10^{-11}$	794.2233
		RMS-E	1.4011 $\times 10^{-11}$	2.3485 $\times 10^{-12}$
	vi)	Mean	2.5041 $\times 10^{-11}$	832.0921
		RMS-E	6.1642 $\times 10^{-11}$	7.1318 $\times 10^{-12}$
GSA	i)	Mean	43.5700	850.6764
		RMS-E	44.7998	40.9723
	ii)	Mean	71.6461	806.5927
		RMS-E	66.0822	49.6254
	iii)	Mean	47.8833	684.4790
		RMS-E	49.9916	16.1516
	iv)	Mean	52.2156	702.3221
		RMS-E	51.8051	13.4524
	v)	Mean	18.2245	794.5856
		RMS-E	18.4399	3.3045
	vi)	Mean	16.7144	831.9263
		RMS-E	21.2645	2.8715
GA	i)	Mean	1.3512 $\times 10^{-2}$	849.8373
		RMS-E	1.9406 $\times 10^{-2}$	1.5466 $\times 10^{-2}$
	ii)	Mean	4.4622 $\times 10^{-2}$	830.9174
		RMS-E	8.6401 $\times 10^{-2}$	5.5183 $\times 10^{-2}$
	iii)	Mean	3.8295 $\times 10^{-2}$	696.9823
		RMS-E	6.2780 $\times 10^{-2}$	1.7286 $\times 10^{-2}$
	iv)	Mean	2.6057 $\times 10^{-2}$	705.4810
		RMS-E	4.5363 $\times 10^{-2}$	1.0015 $\times 10^{-2}$
	v)	Mean	4.6316 $\times 10^{-2}$	794.2235
		RMS-E	1.3905 $\times 10^{-1}$	1.8798 $\times 10^{-2}$
	vi)	Mean	4.2442 $\times 10^{-2}$	832.0911
		RMS-E	9.9879 $\times 10^{-2}$	1.1624 $\times 10^{-2}$
Dix's equation	i)	Direct method	—	849.8366
	ii)	Direct method	—	839.6279
	iii)	Direct method	—	677.1994
	iv)	Direct method	—	681.4589
	v)	Direct method	—	743.3147
	vi)	Direct method	—	770.6618

4.3 Model 3

Model 3 is more complex than the above ones in terms of dipping layers, but less complex in terms of number of layers, see Table 6. The instance is taken from the work of [Dávila-Torres 2012]. This instance is also complex because the different sign in dip of the reflectors. The models posses a linear velocity gradient so that the maximum wave velocity (1400 m/s) belongs to the last layer. A geophone is located to 100 m. source so that the rate geophone/depth is more than one.

Table 6: Parameters of Model 3.

Reflector	Velocity	Depth	Dip
1	700 m/s	20 m	10^o
2	1000 m/s	60 m	-5^o
3	1400 m/s	70 m	7^o

Results of Model 3 are shown and compared with those of [Dávila-Torres 2012] in Table 7. He reported an accuracy of more than 1.0×10^{-4}m and a travel time of 193.3752 ms with a computational time of less than a second, [2]. No analytical solution is reported for this instance.

Table 7: Results for Model 3.

Metaheuristic	Measurement	Objective function (m)	Travel time (ms)
SO	Mean	8.0490×10^{-13}	193.3752
	RMS-E	1.2055×10^{-12}	9.2625×10^{-13}
GSA	Mean	1.5472	193.2292
	RMS-E	1.7689	1.4417
GA	Mean	3.2740×10^{-3}	193.3750
	RMS-E	7.7912×10^{-3}	5.2451×10^{-3}
Dávila-Torres	Mean	$< 1.0 \times 10^{-4}$	193.3752

4.4 Comparative analysis

SO presents the best results for all the instances, obtaining solutions in order of 10^{-11}-10^{-13} m for the mean and RMS-E values of the objective function. This is also written on the travel times. In general, the solutions provided by the *SO* algorithm are not affected by the increasing of layers, but it struggles when the best solution is on the limits of the solution space, as in the second layer

[2] Values measured at intersections between the path and the layers.

in Model 2. Results for Model 1 are acceptable because the approaches to the global minimum are relatively good and travel times obtained differ by 10^{-2} ms with respect to the analytical solution obtained by [Burger & Burger 1992].

GA is also very stable and it approaches to global minimum about 10^{-2}-10^{-3} m for all the models, in mean and RMS-E. This affects travel times by no more than 1μs compared to $S0$. Therefore, the results obtained by *GA* are also considered as acceptable. Indeed, *GA* maintains the same evaluation order in the results for the second layer in Model 2, which suggests that *GA* is more stable than *SO* when the solution space is reduced. Moreover, *GA*'s results do not vary according to complexity of the model.

GSA provides acceptable solutions for simple instances like Model 1, but with increasing the complexity of the model it gives bad solutions (10^2 m for Model 2 and 10^1 m for Model 3) compared to the other metaheuristics. This causes errors in order of 10^0-10^1 ms and 10^{-1} ms for the respective travel times.

The above observations suggest that the performance of the metaheuristics depends on how search agents approach to the best solution because the restricted solution space. Moreover, the restricted solution space causes the constantly generation of new agents because the restrictions in the ray trajectory, see Figure 2.1. Then, the complexity of the models increase with the layers' number and dip but mainly with the relation geophone/depth, when solving by the proposed methodology. In this context, the implementation of the rotation search in *SO* allows both converge to the best solution as rapidly integrate new solutions created by the restricted solution space to the set of good solutions, whereas *GSA* has not a good adaptation when increasing the size of the instances. In turn, the performance of *GA* is also acceptable and stable.

Finally, $S0$ and *GA* provide more precise results than the Dix's equation in Models 1 and 2, and guarantee the observance of the Snell's law. The Dix's equation brings better solutions when the relation geophone/depth diminishes. For Model 3 they provide solutions at least of the same order than the result obtained by [Dávila-Torres 2012], but in lower computational time (few minutes by using the bending methods). In effect, the computational time is good for both metaheuristics, namely, 0.1-0.3 s by using *SO* and 0.6-5 s by *GA* for a single run.

5 Concluding remarks

The initial value problem in seismic-reflection ray tracing was dealt with, defining each ray by its departure angle. The seismic model were constructed considering homogeneous-isotropic reflectors, which are defined by their depth, dip and velocity. The solution space is then limited depending of the instances, namely, of the velocity and dip of the seismic reflectors. The solution methodology is based on the shooting method and a metaheuristic for the search of the optimal in each step of the shooting method. The Spiral Dynamics Inspired

Optimization (*SO*), Gravitational Search Algorithm (*GSA*) and Genetic Algorithm (*GA*) metaheuristics were implemented as described in Section 3 to solve this problem. A comparative study between the metaheuristics used was carried out for diverse instances with different configurations. Also, there is shown a comparison between the results obtained and the solutions in the literature.

We conclude that the limited search space is an important factor in the search of solutions in this problem. As a consequence, the performance of the metaheuristics depend on how the search agents approach to the best solution. The proposed methodology was superior to others methodologies in terms of computational time; namely, our methodology provides solutions in a few seconds for all tests, whereas other methodologies provide inaccurate solutions or good solutions in no less than a few minutes.

Furthermore, *SO* and *GA* are the metaheuristics that provide acceptable solutions to the problem in terms of quality. *SO* is more stable and it brings better solutions than *GA* and *GSA*, in spite of instances where the best solution is on the limits of the solution space. As shown in arrival times shown in Section 4, *GSA* has not a good adaptation to the problem because the restriction of the solution space, causing results acceptable only for models with small instances. We recommend using *SO* and *GA* for future or similar works because, besides their good solutions, both methodologies converge to the best solution and rapidly integrate new solutions created by the restricted solution space to the set of good solutions. So for more complex models the search space is restricted more, so a very stable and accurate metaheuristic will be needed.

As future works we propose to implement the method in models containing anisotropies, discontinuities, folding reflectors, as well as three-dimension and inverse problems. It is also proposed to apply more metaheuristics for a greater view of their performance to the problem of ray tracing. Finally, the stochasticity of metaheuristics could be analyzed because it would allow the model to be more real.

Author contributions

Conceptualization: Roberto Soto-Villalobos, F-Javier Almaguer; *Mathematical model:* Roberto Soto-Villalobos, Mario A. Aguirre-López; *Methodology:* Roberto Soto-Villalobos, Mario A. Aguirre-López, Martha-Selene Casas-Ramírez; *Formal analysis and investigation:* Roberto Soto-Villalobos, F-Javier Almaguer, Mario A. Aguirre-López; *Writing - original draft preparation:* Mario A. Aguirre-López; *Writing - review and editing:* Mario A. Aguirre-López, Martha-Selene Casas-Ramírez, F-Javier Almaguer, Roberto Soto-Villalobos; *Funding acquisition:* F-Javier Almaguer; *Computational experimentation:* Mario A. Aguirre-López; *Supervision:* F-Javier Almaguer.

Acknowledgements We thank to CONACyT and UANL by the financially supporting. This study was partially funded by the PAICyT-UANL program (grant number CE855-19).

Conflict of interest

The authors declare no conflict of interest.

References

Aki & Richards 2002. Aki K, Richards PG (2002) Quantitative Seismology, second ed. University Science Books, California.

Balkaya 2013. Balkaya Č (2013) An implementation of differential evolution algorithm for inversion of geoelectrical data. Journal of Applied Geophysics 98:160-175.

Benamou 1996. Benamou J-D (1996) Big Ray Tracing: Multivalued Travel Time Field Computation Using Viscosity Solutions of the Eikonal Equation. Journal of Computational Physics 128:463-474.

Benaslaa et al. 2014. Benaslaa L, Belmadanib A, Rahlia M (2014) Spiral Optimization Algorithm for solving Combined Economic and Emission Dispatch. International Journal of Electrical Power and Energy Systems 62:163-174.

Bergey et al. 2003. Bergey PK, Ragsdale CT, Kote MH (2003) A Simulated Annealing Genetic Algorithm for the Electrical Power Districting Problem. Annals of Operations Research 121:33–55.

Burger & Burger 1992. Burger HR, Burger DC (1992) Exploration Geophysics of the Shallow Subsurface. Prentice Hall, N.J.

Červený 2001. Červený V (2001) Seismic Ray Theory. Cambridge University Press, USA.

Constain & Coruh 2004. Constain JK, Coruh C (2004) Basic Theory of Exploration Seismology. Elsevier, USA.

Conti et al. 2013. Conti CR, Roisenberg M, Schwedersky NG, Porsani MJ (2013) Fast Seismic Inversion Methods Using Ant Colony Optimization Algorithm. IEEE Geoscience and Remote Sensing Letters 10:1119-1123. https://doi.org/10.1109/LGRS.2012.2231397

Contreras et al. 2008. Contreras OA, Pacheco JR, Larrazábal G (2008) Trazado de rayos sísmicos usando un algoritmo genético. Ingeniería UC 15(1):50-58.

Chopra & Castagna 2014. Chopra S, Castagna JP (2014) AVO. Society of Exploration Geophysicists, USA.

Dávila-Torres 2012. Dávila-Torres RF (2012) Trazado de rayos sísmicos mediante la optimización de la ecuación de tiempos de arribo utilizando Programación Evolutiva. Degree Thesis, Universidad Autónoma de Nuevo León.

Dias & de-Vasconcelos 2002. Dias AHF, de-Vasconcelos JA (2002) Multiobjective genetic algorithms applied to solve optimization problems. Transactions on magnetics 38:2.

Eldos & Al-Qasim 2013. Eldos T, Al-Qasim R (2013) On The Performance of the Gravitational Search Algorithm. International Journal of Advanced Computer Science and Applications 4(8):74-78.

Goldberg 1989. Goldberg DE (1989) Genetic Algorithms in Search, Optimization and Machine Learning. Addison-Wesley, Boston.

Grenchka et al. 1999. Grenchka V, Tsvankin I, Cohen JK (1999) Generalized Dix equation and analytic treatment of normal-moveout velocity for anisotropic media. Geophysical Prospecting 47,117-148.

Hosseinabadi et al. 2012. Hosseinabadi AR, Yazdanpanah M, Rostami AS (2012) A New Search Algorithm for Solving Symmetric Traveling Salesman Problem Based on Gravity. World Applied Sciences Journal 16(10):1387-1392.

Hosseinabadi et al. 2014. Hosseinabadi AR, Siar H, Shamshirband S, Shojafar M, Nasir M (2014) Using the gravitational emulation local search algorithm to solve the multi-objective flexible dynamic job shop scheduling problem in Small and Medium Enterprises. Annals of Operations Research. https://doi.org/10.1007/s10479-014-1770-8

Ikelle & Amundsen 2005. Ikelle LT, Amundsen L (2005) Introduction to Petroleum Seismology. Society of Exploration Geophysicists, USA.

Janiaka & Portmann 1998. Janiaka A, Portmann MC (1998) Genetic algorithm for the permutation flow-shop scheduling problem with linear models of operations. Annals of Operations Research 83:95-114.

Kalos & Whitlock 2008. Kalos MH, Whitlock PA (2008) Monte Carlo Methods. WILEY-VCH, Germany.

Kania & Sidarto 2016. Kania A, Sidarto KA (2016) Solving mixed integer nonlinear programming problems using spiral dynamics optimization algorithm. AIP Conference Proceedings 1716:020004. https://doi.org/10.1063/1.4942987

King 1995. King SD (1995) Radial models of mantle viscosity: results from a genetic algorithm. Geophysical Journal International 122:725-734.

Koziel & Yang 2011. Koziel S, Yang X (2011) Computational Optimization, Methods and Algorithms. Studies in Computational Intelligence 356.

Kumar & Sharma 2013. Kumar M, Sharma MD (2013) Reflection and transmission of attenuated waves at the boundar between two dissimilar poroelastic solids satured with two immiscible viscous fluids. Geophysical Prospecting 61:1035-1055.

Marroquín Navarro 2012. Marroquín Navarro LM (2012) Inversión sísmica de reflexión 2D, mediante la técnica de optimización Programación Evolutiva. Degree Thesis, Universidad Autónoma de Nuevo León.

Nasir & Tokhi 2015. Nasir ANK, Tokhi MO (2015) An improved spiral dynamic optimization algorithm with engineering application. IEEE Transactions on Systems, Man, and Cybernetics: Systems. http://www.ieee.org/publications_standards/publications/rights/index.html. Accessed 30 September 2019.

Nezamabadi et al. 2009. Nezamabadi-Pour H, Rashedi E, Saryazdi S (2009) GSA: A Gravitational Search Algorithm. Information Sciences 179:2232-2248.

Nowers et al. 2014. Nowers O, Duxbury DJ, Zhang J, Drinkwater BW (2014) Novel ray-tracing algorithms in NDE: Application of Dijkstra and A* algorithms to the inspection of an anisotropic weld. NDT&E International 61:58-66.

Ooi & Tan 2003. Ooi CH, Tan P (2003) Genetic algorithms applied to multi-class prediction for the analysis of gene expression data. Bioinformatics 19:37-44.

Osman & Laporte 1996. Osman IH, Laporte G (1996) Metaheuristics: A bibliography. Annals of Operations Research 63:513-623.

Poormirzaee et al. 2015. Poormirzaee R, Moghadam RH, Zarean A (2015) Inversion seismic refraction data using particle swarm optimization: a case study of Tabriz, Iran. Arabian Journal of Geosciences 8:5981-5989. https://doi.org/10.1007/s12517-014-1662-x

Poormirzaee et al. 2019. Poormirzaee R, Sarmady S, Sharghi Y (2019) A new inversion method using a modified bat algorithm for analysis of seismic refraction data in dam site investigation. Journal of Environmental and Engineering Geophysics. https://doi.org/10.2113/JEEG24.2.201

Prothero et al. 1988. Prothero WA, Taylor WJ, Eickemeyer JA (1988) A fast two-point, three-dimensional ray tracing algorithm using a simple step search method. Bulletin of the Seismological Society of America 78:1190-1198.

Ramillien 2001. Ramillien G (2001) Genetic algorithms for geophysical parameter inversion from altimeter data. Geophysical Journal International 147:393-402.

Rashedi et al. 2011. Rashedi E, Nezamabadi-Pour H, Saryazdi S (2011) Filter modeling using gravitational search algorithm. Engineering Applications of Artificial Intelligence 24:117-122.

Rawlinson et al. 2008. Rawlinson N, Hauser J, Sambridge M (2008) Seismic ray tracing and wavefront tracking in laterally heterogeneous media. Advances in Geophysics 49:203-273. https://doi.org/10.1016/S0065-2687(07)49003-3

Siddique & Adeli 2014. Siddique N, Adeli H (2014) Spiral dynamics algorithm. International Journal on Artificial Intelligence Tools. https://doi.org/10.1142/S0218213014300014

Song 2014. Song X, Li L, Zhang X, Huang J, Shi X, Jin S, Bai Y (2014) Differential evolution algorithm for nonlinear inversion of high-frequency Rayleigh wave dispersion curves. Journal of Applied Geophysics 109:47-61.

Stein & Wysession 2003. Stein S, Wysession M (2003) An introductionto seismology, earthquakes, and earth structure. Blackwell, Oxford.

Tamura & Yasuda 2011. Tamura K, Yasuda K (2011) Spiral Dynamics Inspired Optimization. Journal of Advanced Computational Intelligence and Intelligent Informatics 15(8):1116-1122.

Telford 1990. Telford WM, Geldart LP, Sheriff RE (1990) Applied Geophysics. Cambridge University Press, New York.

Tsai et al 2014. Tsai C, Huang B, Chiang M (2014) A Novel Spiral Optimization for Clustering. Lecture Notes in Electrical Engineering 274:621-628.

Vargas Contreras 2017. Vargas Contreras GA (2017) Estimación de parámetros para sísmica multicapa mediante aproximaciones a las ecuaciones de Zoeppritz utilizando algoritmos evolutivos. Master Thesis, Universidad Autónoma de Nuevo León.

Xu & Zhang 2014. Xu B, Zhang Y (2014) An Improved Gravitational Search Algorithm for Dynamic Neural Network Identification. International Journal of Automation and Computing 11(4):434-440.

Yilmaz 2008. Yilmaz Ö (2008) Seismic Data Analysis: processing, inversion, and interpretation of seismic data. Society of Exploration Geophysicists, USA.

Zhao et al. 2004. Zhao A, Zhang Z, Teng J (2004) Minimum travel time tree algorithm for seismic ray tracing: improvement in efficiency. Journal of Geophysics and Engineering 1:245-251.