

We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

6,900

Open access books available

186,000

International authors and editors

200M

Downloads

Our authors are among the

154

Countries delivered to

TOP 1%

most cited scientists

12.2%

Contributors from top 500 universities



WEB OF SCIENCE™

Selection of our books indexed in the Book Citation Index
in Web of Science™ Core Collection (BKCI)

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected.
For more information visit www.intechopen.com



Pareto-Based Multiobjective Particle Swarm Optimization: Examples in Geophysical Modeling

Ersin Büyük

Abstract

It has been recently revealed that particle swarm optimization (PSO) is a modern global optimization method and it has been used in many real world engineering problems to estimate model parameters. PSO has also led as tremendous alternative method to conventional geophysical modeling techniques which suffer from dependence to initial model, linearization problems and being trapped at a local minimum. An area neglected in using PSO is joint modeling of geophysical data sets having different sensitivities, whereas this kind of modeling with multiobjective optimization techniques has become an important issue to increase the uniqueness of the model parameters. However, using of subjective and unpredictable weighting to objective functions may cause a misleading solution in multiobjective optimization. Multiobjective PSO (MOPSO) with Pareto approach allows obtaining set of solutions including a joint optimal solution without weighting requirements. This chapter begins with an overview of PSO and Pareto-based MOPSO presented their mathematical formulation, algorithms and alternate approaches used in these methods. The chapter goes on to present a series synthetic modeled of seismological data that is one kind of geophysical data by using of Pareto-based multiobjective PSO. According to results matched perfectly, we believe that multiobjective PSO is an innovative approach to joint modeling of such data.

Keywords: particle swarm optimization, Pareto-based optimization, multiobjective, geophysical modeling

1. Introduction

As a conventional approach, least squares and linear programming optimization methods have been used modeling of geophysical data with a general form without requiring any special case. However, due to some disadvantages of these methods such as computational time and linearization problems, it has become inevitable to tend to new approaches for the researchers. Unlike traditional optimization methods, optimization of nonlinear models has been improved in two ways which are derivative based and non-derivative search methods. Unfortunately, one of the major disadvantages of derivative based methods is that solutions potentially trap at a local minimum because of depending on initial model. On the other hand,

non-derivative search methods partly provide a global solution, however, significantly increase computing time.

In recent years, two approaches that are artificial intelligence and meta-heuristic optimization algorithms have been effectively put forward in geophysical modeling studies having complex and nonlinear models. Meta-heuristic optimization algorithms called as modern global optimization methods are based on systematic characteristics of biological, molecular, neurobiological and animal swarms [1]. PSO as one of the modern global optimization algorithms, has increased its popularity with rapid convergence compared to various optimization algorithms [2, 3], especially when real model parameters are used [4].

PSO for multiobjective optimization has also been used in many studies in order to solve real world engineering problems having conflicted solutions between objective functions [5]. Despite this interest, very few researchers have studied MOPSO for joint modeling of geophysical data such as electromagnetic and gravity [6, 7]. In fact, simultaneous optimization of multiobjective functions is also favored to increase uniqueness of model parameters in joint modeling of geophysical data that are generally sensitive to different physical phenomena. Multiobjective functions can be transferred into single-objective by combination of objective functions by using weighted-sum approach. However, it is very difficult to find reasonable and optimum weighting coefficients [5]. In engineering problems, subjective and unpredictable weightings used to objective functions are the primary cause of a misleading solution, because different sensitivities and unpredictable noise of different data sets lead to uncertainty in weighting. Pareto optimality approach is a good way to obtain set of possible solutions including an optimum solution in objective function space overcoming the use of weighting and combining.

The purpose of this chapter is to review the literature on Pareto-based MOPSO. This chapter first gives a brief overview of the methods and approaches used in PSO and Pareto-based MOPSO and to look at how mathematical formulations and general algorithms of these optimization techniques work. In order to show the superiority of Pareto-based MOPSO over weighting-sum approaches, the chapter proceeds as joint modeled of two synthetic seismological data using Pareto-based MOPSO and analyses the obtained results. The results demonstrate that Pareto-based MOPSO is a useful approach to joint modeling of seismological data as explain in detail in our previous paper [8], of which TÜBİTAK is the publisher. Findings validate the usefulness of MOPSO as a technique to optimize objective functions simultaneously without weighting requirements. Finally, conclusion section gives a brief summary of MOPSO and critique of findings in modeling.

2. Particle swarm optimization

PSO method, inspired by social behavior of the bird or fish flock to reach to target in a shortest route was originally introduced by [9]. It was noticed that members of flocks suddenly change their movements as scattering and regrouping, when trajectory of swarms was observed. A striking feature of that was an effort of members to reduce their distance from both flock and surrounding members. It was found that knowledge within a flock was continuously shared by all members. PSO method was developed by defining each member in a flock as a particle. According to PSO, particles bearing an information of decision variable or model parameters take a position in an objective function space. Each particle is in communication and learning with other particles as schematically illustrated in **Figure 1a**. If a minimization problem is considered, each particle changes its position with a velocity vector and converges a global minimum as shown in **Figure 1b**.

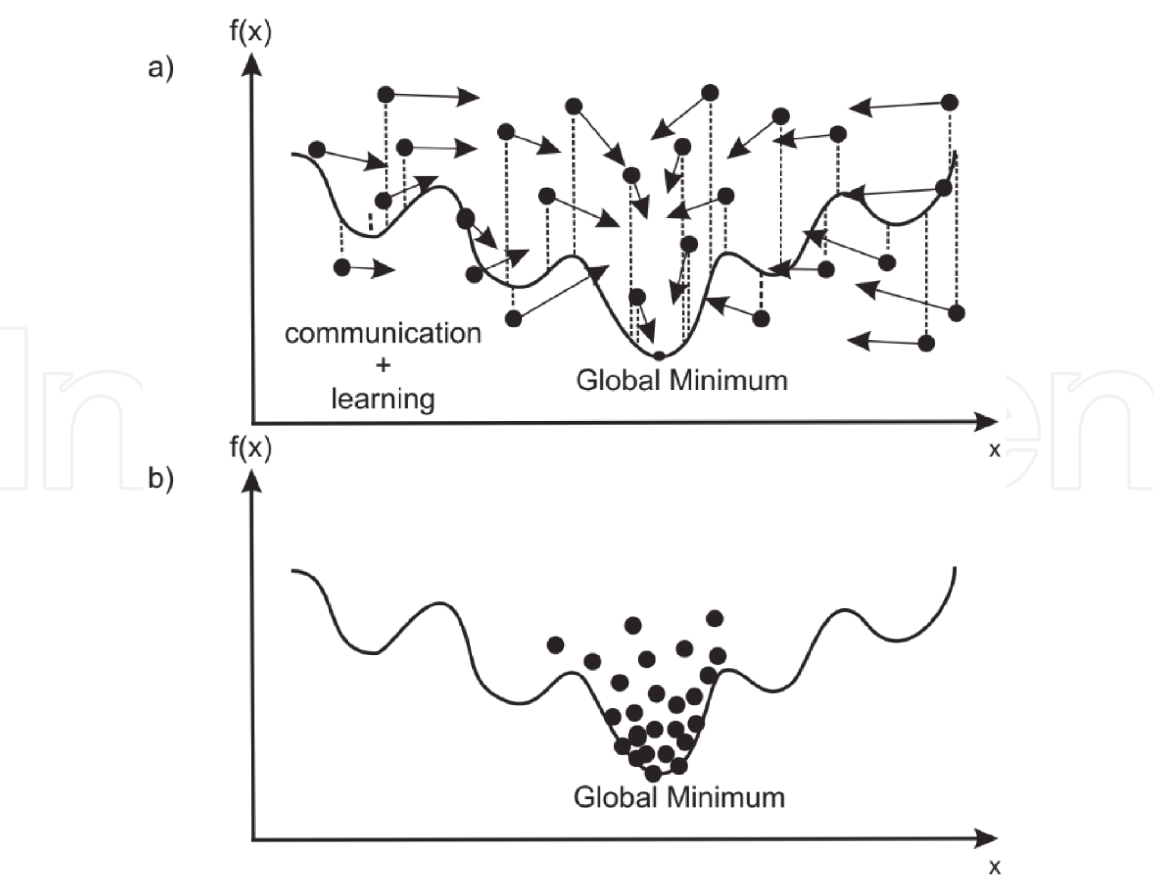


Figure 1. For a minimization problem, schematic illustration of a swarm trajectory. Randomly distributed particles in an objective function space (a); trajectory of a swarm towards a global best solution (b), where x and $f(x)$ denote a decision variable and an objective function, respectively.

The basic algorithm of PSO is outlined in **Figure 2**. According to this scheme, particles which velocities are initially assigned as zero are initiated by randomly selection between the minimum and maximum value of decision variables. After

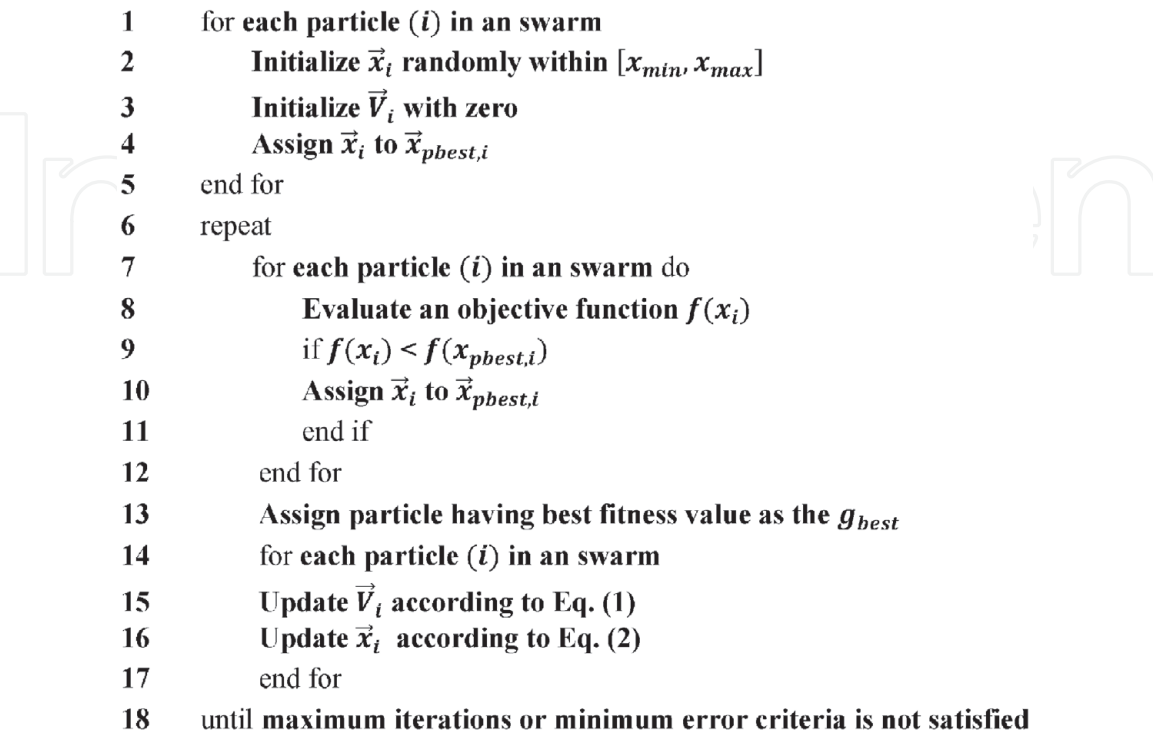


Figure 2. General PSO algorithm.

each particle is evaluated with an objective function, particle with best fitness value is assigned as a global best solution. Particles move to the next position with a velocity vector

$$\vec{V}_i^{k+1} = \omega \vec{V}_i^k + c_1 \gamma_1 \otimes (\vec{x}_{pbest,i} - \vec{x}_i^k) + c_2 \gamma_2 \otimes (\vec{x}_{gbest} - \vec{x}_i^k), \quad (1)$$

where,

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

subscript i is the number of particles and k is the number of iterations.

Position and velocity vector of a particle i at iteration k are shown as \vec{x}_i^k and \vec{V}_i^k , respectively. ω is the inertia weight term forced on velocity vector. c_1 and c_2 are local and global learning constants, γ_1 and γ_2 are uniformly random numbers in the range $[0,1]$. $\vec{x}_{pbest,i}$ is the best position of particle i in the past. \vec{x}_{gbest} called as leader of a swarm is the position of the best particle indicating the best fitness value. If one particle position updated by a new velocity vector has a fitness value better than its previous best, new position is assigned as \vec{x}_{pbest} . If one of the particles has a best fitness value than the others, it is assigned as \vec{x}_{gbest} . These processes are repeated in a balance between exploration and exploitation until maximum iterations or minimum error criteria is not satisfied.

2.1 Selection of the PSO parameters

Velocity vector in Eq. (1) is controlled in the following factors: velocity limitation, learning coefficients and inertia weight. These factors are significantly contribute to prevent explosion in a swarm and ensure convergence.

Velocity limitation: In the PSO method, each particle changes its velocity vector by stochastic process. However, this leads a tendency exploding of velocity vector and exceeding limit of one particle constrained in a search space, especially if the particle is far from the personal and global best position. Therefore velocity limit approach, introduced by [10], has been exploited to avoid such problems. By this limitation of movement, it is ensured that particle does not exceed a search space. Fan and Shi [11] suggested that maximum and minimum velocity limits as follows: $V_{max,min} = \pm U$, where $U = (x_{max} - x_{min})/N$. x_{max} and x_{min} are the upper and lower limit of a particle. N is an interval number defined by user considering the x_{max} and x_{min} .

Learning coefficients: Local and global learning coefficients illustrated in Eq. (1) control acceleration of a particle. While local learning coefficient (c_1) enables particle to approach its individual best position, global learning coefficient (c_2) enables to be pulled towards the global best solution. These parameters sometimes called as acceleration coefficients are important to control the convergence. Higher learning coefficients can provide a rapid convergence, but also probably prevents exploration in an objective function space. In [12], author suggested that dynamic learning coefficients for optimization with a large number of unknown parameters. In addition, Ozcan and Mohan [13] concluded that if $c_1 + c_2 > 4$, trajectory of a swarm goes to infinity.

Inertia weight or constriction factor: Inertia weight term (ω) introduced by Shi and Eberhart [14] maintains a balance between exploitation and exploration by acting upon the current velocity vector. They proposed that linearly decreasing

inertia weight from 0.9 to 0.4 in each iteration to increase performance of the algorithm. If $\omega > 1$ is selected, particles tend to move away through boundary of a search space as velocity vectors increase continuously [5]. In [15], the author observed that particles indicate a stable and reasonable searching in case of $c_1 + c_2 \leq 4$. Nevertheless, it has even suggested a velocity limit by [16] to prevent drastically growing of velocity vector. In a major advance, Clerc [17] proposed a constriction factor (χ) to the PSO velocity vector equation. PSO algorithm with a constriction factor allowed us to rapid and reliable convergence without velocity limit requirements in case that $\phi = \phi_1 + \phi_2 > 4$, where ϕ_1 and ϕ_2 are the learning coefficients substituted with c_1 and c_2 . The velocity vector is changed in the following way:

$$\vec{V}_i^{k+1} = \chi \left[\vec{V}_i^k + \phi_1 \gamma_1 \otimes (\vec{x}_{pbest,i} - \vec{x}_i^k) + \phi_2 \gamma_2 (\vec{x}_{gbest} - \vec{x}_i^k) \right]. \quad (3)$$

Eq. (3) reveals that χ influences not only current velocity, but also new velocity vector. Theoretical studies in [18] show that if ϕ_1 and ϕ_2 used as 2.05, therefore $\phi = 4.1$, and $\chi = 0.7298$, rapidly convergence is observed without using velocity limits. On the other hand, in [2] the authors analyzed that if ϕ_1 and ϕ_2 are used as randomly, swarm trajectory behaves as a combination of divergence and convergence.

2.2 Swarm topologies

Network of particles in a swarm is provided by kind of neighborhood topologies that regulate sharing of information between particles. Small-scale topologies are selected to use in solving complex problems, whereas large-scale topologies are selected for simpler problem [19]. Empty, local best, fully connected, star network and tree network are the list of the neighborhood network topologies which are generally used in PSO [20].

Empty graph: Each particle is connected to itself and compared with personal best position. In this topology, it is considered that c_2 as equal to zero.

Local best: Local best (\vec{x}_{lbest}) indicates that the best particle position between one particle and its nearest neighbors. In this topology, particles are affected by both their personal best and local best which is also defined as equivalent to a leader. Each particle is connected to k nearest neighbors and if $k = 2$, network structure of particles is called ring topology as shown in **Figure 3a**.

Fully connected graph: All particles in a swarm are connected to one another as illustrated in **Figure 3b**. In fully connected topology also called as star topology [21], each particle is affected by both its personal best and a global best defined as equivalent to leader. Kennedy [22] analyzed that this structure tends to ensure that a rapidly, but also a premature convergence.

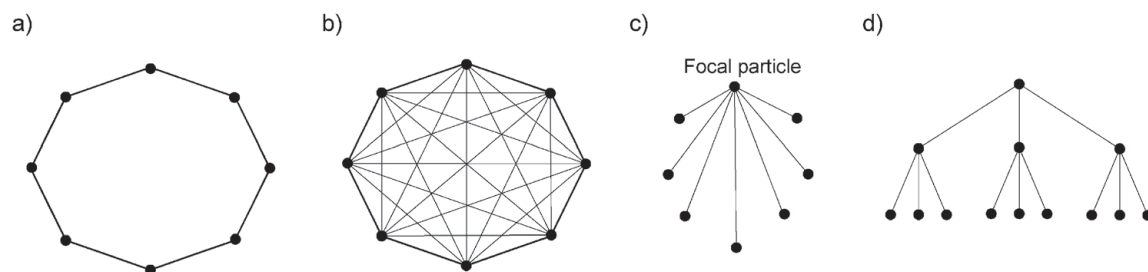


Figure 3. Neighborhood topologies generally used in PSO: Ring (a), fully connected (b), star network (c) and tree network topology (d).

Star network: In this topology, one particle called as focal particle [23] is in connection with others as shown in **Figure 3c**. Focal particle defined as equivalent to a leader adjusts trajectory of a swarm by comparing positions of others.

Tree network: In this network structure, all particles are taken shape of a tree and, each of nodes of tree consists of exactly one particle as illustrated in **Figure 3d** [24]. One particle at the node is connected to both child nodes bottom in a tree and a parent node directly above in a tree. If one of the child node particle has a solution better than its parent solution, position of both particles are replaced. In this topology, parent particle is defined as equivalent to a leader.

3. Pareto multiobjective optimization

In multiobjective optimization problems (MOOP) indicated as an optimization of more than one objective function, “trade-off” solutions that are conflicted to each other are obtained rather than single solution. MOOP is generally defined to obtain decision variables

$$x = (x_1, x_2, x_3, \dots \dots x_p) \in S \tag{4}$$

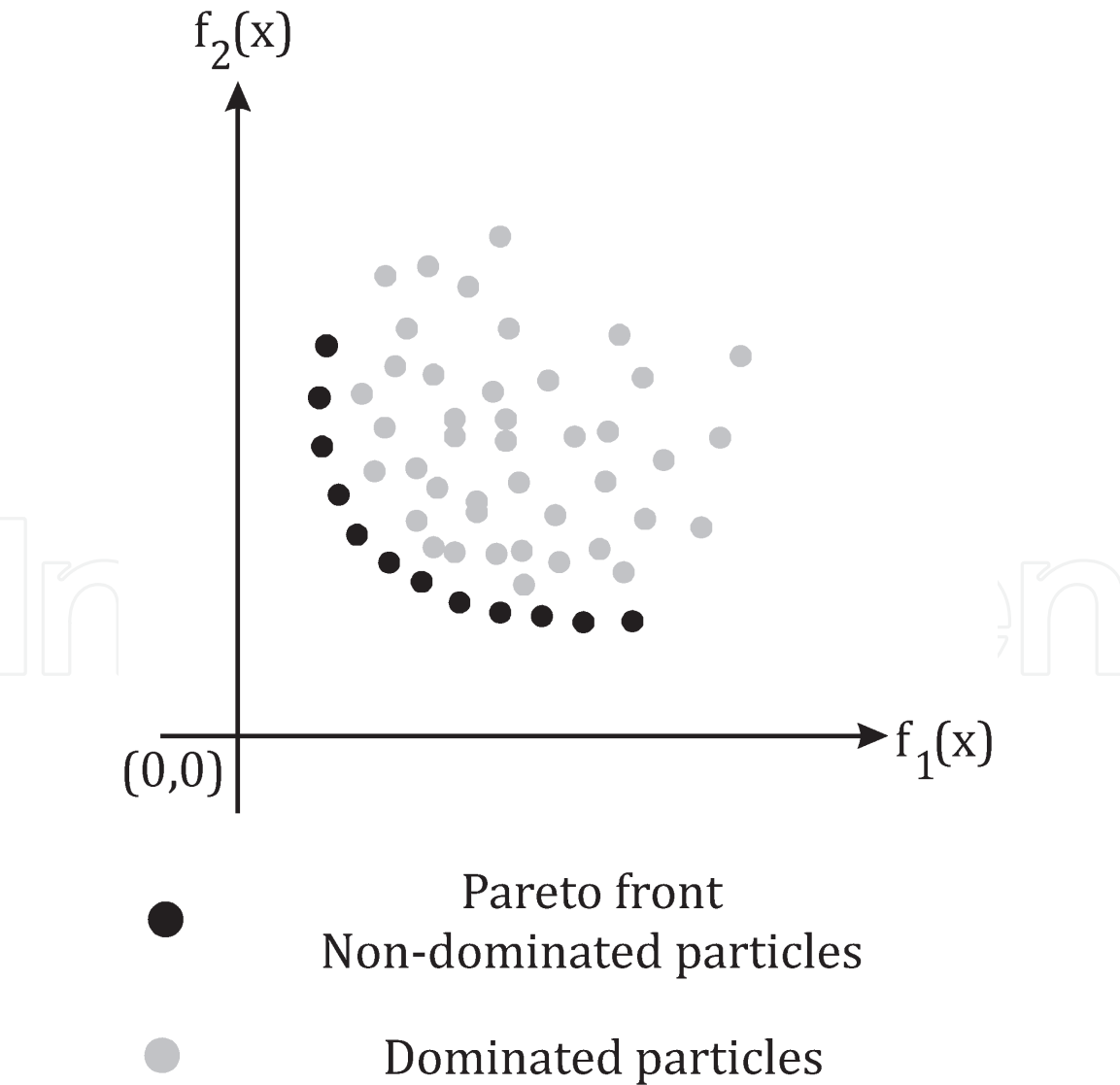


Figure 4. Conceptual representation of the non-dominated and dominated particles in objective function space. Here, $f_1(x)$ and $f_2(x)$ are objective functions. [8].

in p dimensional of model space S , while simultaneously optimizing of a vector

$$f(x) = [f_1(x), f_2(x), f_3(x) \dots \dots f_N(x)] \quad (5)$$

including all objective functions [25]. In MOOP, all objective functions in $f(x)$ vector may not be optimized simultaneously. Therefore Pareto optimality sometimes called as Pareto dominance approach is used to find set of possible solutions. According to Pareto approach, if x is better solution than y in minimization problem, $f(x)$ dominates $f(y)$, if and only if $f_k(x) \leq f_k(y)$, $k = 1, \dots, N$. If there is no other $\bar{x} \in S$ satisfies a condition such that $f(\bar{x}) < f(x)$ without deteriorating any other objective function, non-dominated solutions called Pareto optimal set $P^* \in S$ or Pareto front $PF^* = \{f(x) \in \mathbb{R} \mid x \in P^*\}$ exists in an objective function space as schematically illustrated in **Figure 4**.

All particles of Pareto front generally spreads in two ways which are concave and L-shaped curves. Concave shaped spreading indicates that one objective function does not be ameliorated without deteriorating any other objective function(s). On the other hand, L-shaped spreading indicates that complete optimal solution which means that all objective functions can be optimized simultaneously [26]. Other remarkable distribution of Pareto front is its deviation from symmetry referenced by utopia point $[0,0]$. Deviation is an indication that one objective function has many local minimums relatively the others (minimum one or more) and/or modeling has not properly accomplished with defined parameter search space and used methods [27].

4. Pareto-based multiobjective particle swarm optimization

Several considerations that are increasing the diversity, maintaining the non-dominated particles and selecting a leader should be taken into account when using MOPSO. [20] A general MOPSO algorithm modified from PSO algorithm by these considerations outlined below is shown in **Figure 5**.

Increasing of diversity: In MOPSO, it is expected to increase of diversity of all particles, while dominated particles are required to converge towards to Pareto-front. Mutation, niching and clustering techniques are generally used to increase diversity. Mutation operator is originally a stage of the genetic algorithm and it prevents to trap of particles a local minima by altering decision variables within the bounds of possibility [28, 29]. suggested that polynomial and dynamic mutation rate which should be changed from higher to lower degree to keep increasing diversity in an entire search space. This feature allows particles to be explored, even in far field on this space [30]. Niching method, which was also developed for evolutionary algorithms [31], prevents a premature convergence that is a leading cause of a movement of particles towards a solution [32]. In NichePSO presented by [33], alteration of each particle is kept track after each iteration. If some particles do not change after several iterations, niche formed as a sub-swarm including both these particles and their surroundings is generated with a radius called σ_{share} as illustrated in **Figure 6**. By comparing of two particles in niches, a control mechanism is provided to be adding particles to a sub-swarm or to be merging of sub-swarms [34]. In clustering technique introduced by [35], main swarm is divided into k sub-swarms and determined leader in each sub-swarm is moved to centroid of its sub-swarm. Diversity is provided by a migration of leaders to different sub-swarms [36].

Maintaining non-dominated particles: Maintaining non-dominated particles is key component of MOPSO. External archive technique allowing a storage of non-dominated particles is generally used to spread and maintain Pareto-front. Global


```

1   for each particle ( $i$ ) in an swarm
2       Initialize  $\vec{x}_i$  randomly within  $[x_{min}, x_{max}]$ 
3       Initialize  $\vec{V}_i$  with zero
4       Assign  $\vec{x}_i$  to  $\vec{x}_{pbest,i}$ 
5   end for
6   for each particle ( $i$ ) in an swarm do
7       Evaluate an objective function  $f(x_i)$ 
8       if  $f(x_i) < f(x_{pbest,i})$ 
9           Assign  $\vec{x}_i$  to  $x_{pbest,i}$ 
10      end if
11  end for
12  Initialize non-dominated particles in an external archive
13  Select a leader
14  repeat
15      for each particle ( $i$ ) in an swarm do
16          Update  $\vec{V}_i$  according to Eq. (1)
17          Update  $\vec{x}_i$  according to Eq. (2)
18          Increase diversity
19          Evaluate an objective function  $f(x_i)$ 
20          if  $f(x_i) < f(x_{pbest,i})$ 
21              Assign  $\vec{x}_i$  to  $x_{pbest,i}$ 
22          end if
23      end for
24      Maintain non-dominated particles in an external archive
25      Select a leader
26  until maximum iterations or minimum error criteria is not satisfied
    
```

Figure 5.
General MOPSO algorithm.

best, personal best or local best can be stored to an archive when extending PSO to MOPSO. However, as dominated particles tending to converge and join to Pareto-front, a rapid increase number of particles in an archive is a disadvantage [20]. A restriction is applied to prevent rapidly growing number of non-dominated particles. In [37], they proposed ϵ -dominance value defined as set of boxes in objective function space as shown in **Figure 7** controls adding of dominated particles to Pareto-front by sizing of an archive. In [38], their attempts to update of external archive is composed niching method used to add or remove particles from archive when it is full. In [30], geographically-based global repository approach is used to update of an archive based on removing of non-dominated solutions from cells which have fewer particles. Li [39] proposed non-dominated sorting method to determine Pareto-front in each iteration. In his study, rather than comparing of a new position with a personal best, temporary swarms have been generated by combining both of them. In fact, the fundamental condition maintaining non-dominated particles is that a new archive should be dominant over previous archives. ϵ -dominance approach is one of the most feasible method to provide this condition [20].

Leader selection: Unlike one leader that guides a swarm in the PSO, non-dominated particles of Pareto front set indicate presence of multiple leaders in MOPSO. Neighbor density and kernel density estimator methods based on density measurement techniques are suggested to determine particles that are most likely to be selected as a leader. In [40], they studied and further supported the concept of crowding factor according to nearest neighbors of a given particle. The value of

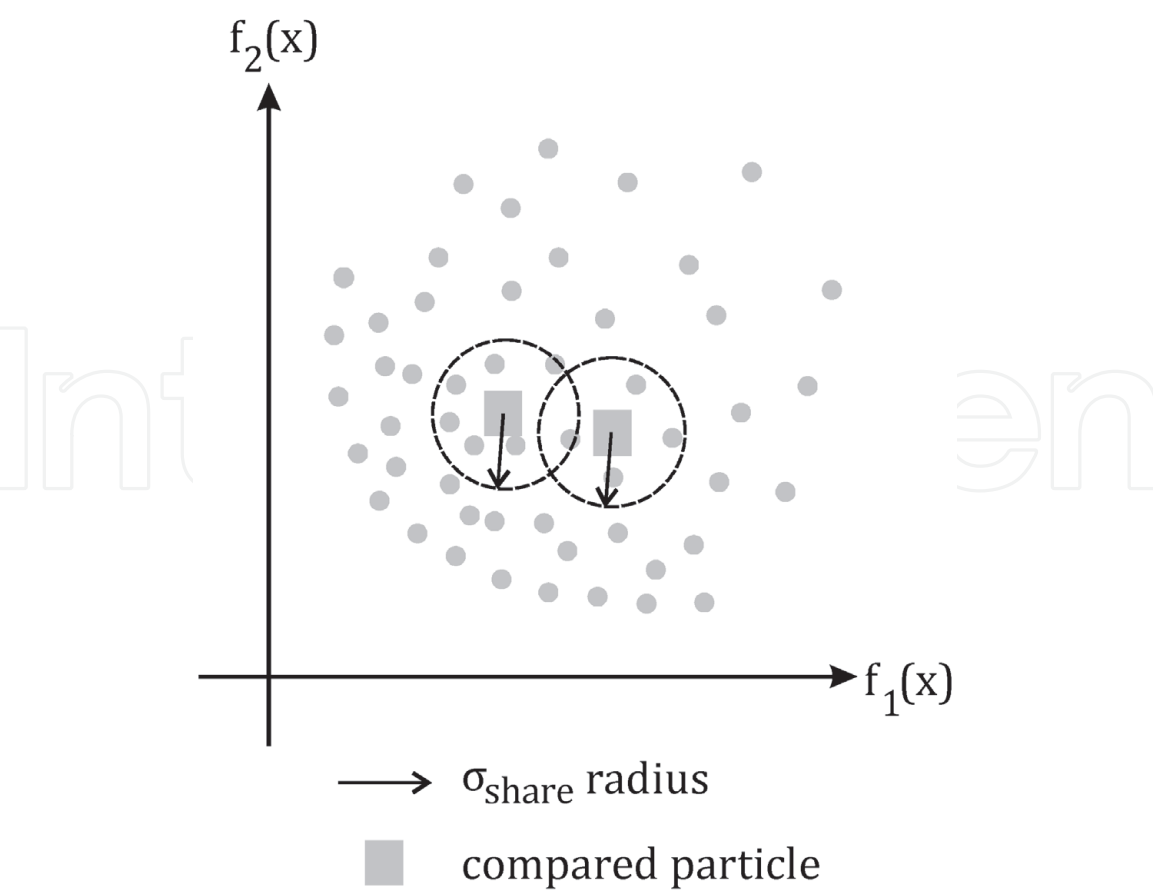


Figure 6.
Niches having radius indicated by σ_{share} are generated to compare of particles that have not change after several iterations.

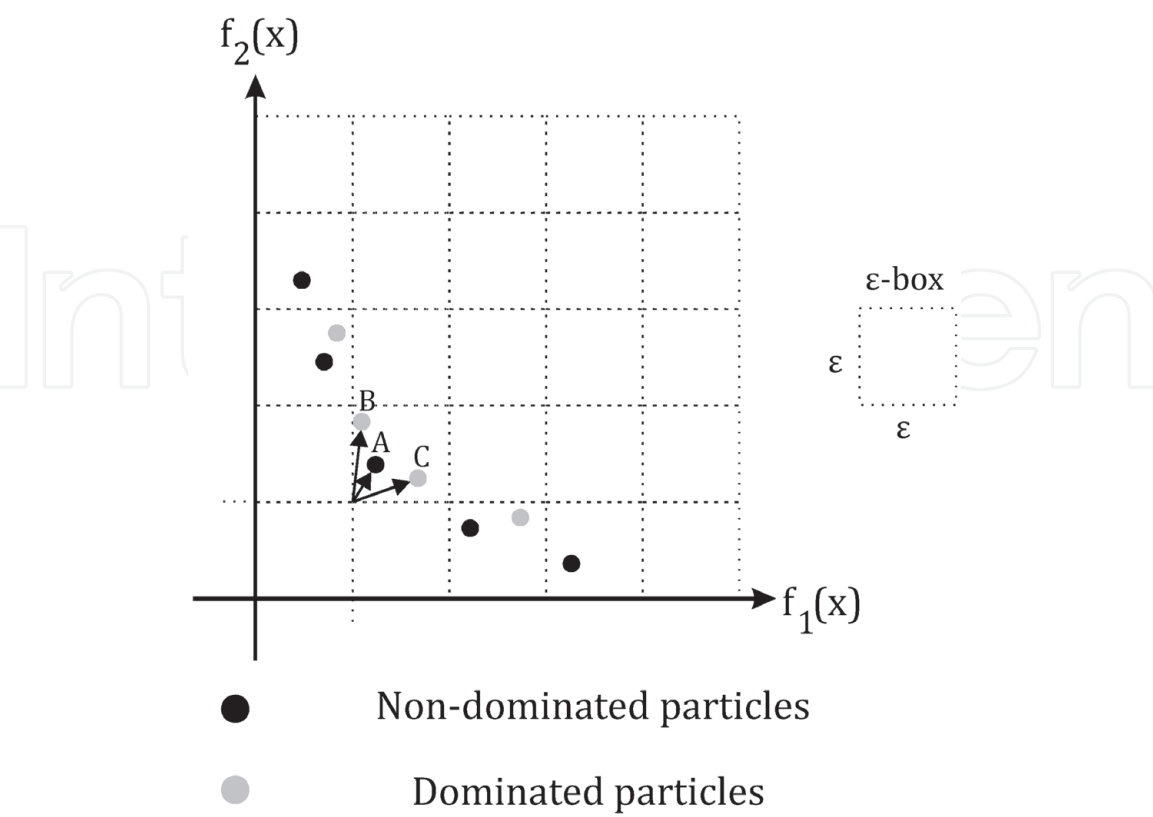


Figure 7.
 ϵ -dominance concept for minimizing two objective functions. Particle A, B and C are incommensurable particles, however, particle a is considered to be dominant to B and C due to more closer the lower left hand corner.

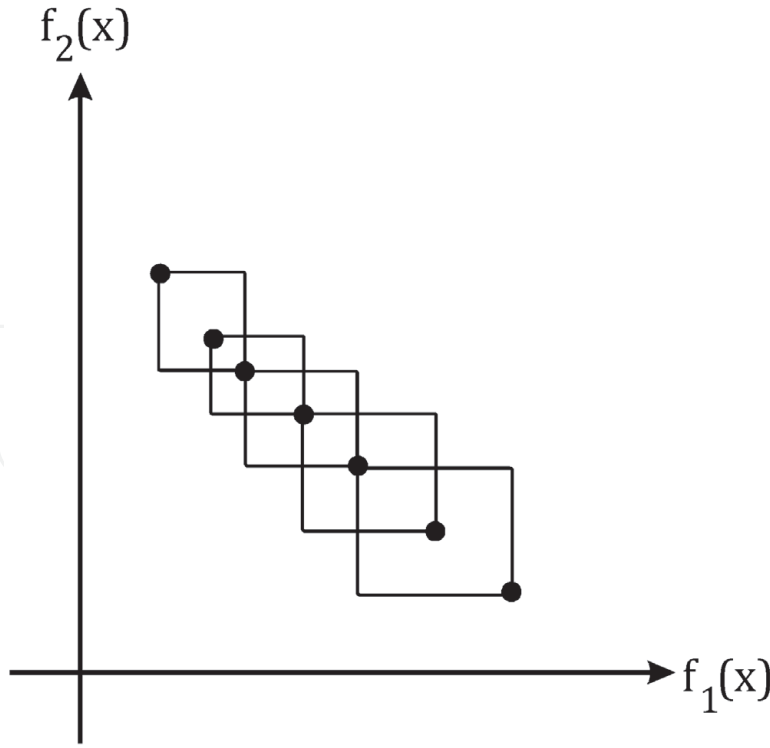


Figure 8.

Neighbor density estimator defined by perimeter of rectangular formed between one particle and its neighbors.

estimators are determined by perimeter of a rectangular formed between one particle and neighbors as illustrated in **Figure 8**. Particles which have larger value of estimator are preferred to select as a leader.

The kernel density estimator is based on the niching method. A niche which has radius as σ_{share} is generated for each of non-dominated particles. It is preferable selected a leader from the least crowded niche [41]. As for leader selection from preferred area, different ways are used. For instance, neighbor density estimator has been used by Raquel and Naval [42] to select a leader randomly from top a list ranking in descending order according to estimator values. If an external archive is full, particle has been removed by randomly selecting from bottom of the list. To select a leader, Coello Coello et al. [30] suggested to fitness values based on the number of particles in each hypercube formed by division of an objective function space. A leader is randomly selected from a hypercube determined by roulette-wheel selection scheme which its lengths of divided segments proportional to fitness values.

5. Selecting the methods and parameters for Pareto-based multiobjective particle swarm optimization

We joint modeled two synthetic seismological data obtained by response of five-layered models by using MOPSO. In one synthetic model, shear wave velocities increase smoothly with depth (SM-1), while the other has a noticeable velocity contrast at third layer interface (SM-2). In the modeling stage, we simultaneously optimized two objective functions related with Rayleigh wave dispersion (RWD) and horizontal to vertical spectral ratio (HVSr) methods, which have different sensitivities to physical phenomena. The estimated parameters were shear-wave velocities and depth obtained by cumulative sum of layer thicknesses in each layer. Parameter search space and more technical details were be given in [8].

Minimization was carried out between observed data obtained by response of synthetic models (y_i^{obs}) and calculated data (y_i^{cal}) obtained by response of models

which their parameters changing with particle movements. Fitness value (α) minimized at each of the objective function was calculated using,

$$\alpha = \left[\frac{1}{n} \sum_{i=1}^n (y_i^{obs} - y_i^{cal})^2 \right]^{\frac{1}{2}}, \tag{6}$$

where n is the number of observations. Number of iterations and particles was selected as 200 and 100, respectively. Number of particles was selected as 100 referenced by 10-fold the number of model parameters (thicknesses and velocities for 5 layers). We used PSO parameters defined in [43] with velocity equation in Eq. (3) which provides an optimization without a velocity limitation. A mutation operator was used to increase diversity and mutation rate was selected as two percent. ϵ -dominance approach was selected to maintain of non-dominated particles and the value of ϵ was used as 0.01 for each of the objective functions. Number of hypercubes was selected as 30 for 100 particles recommended by [30].

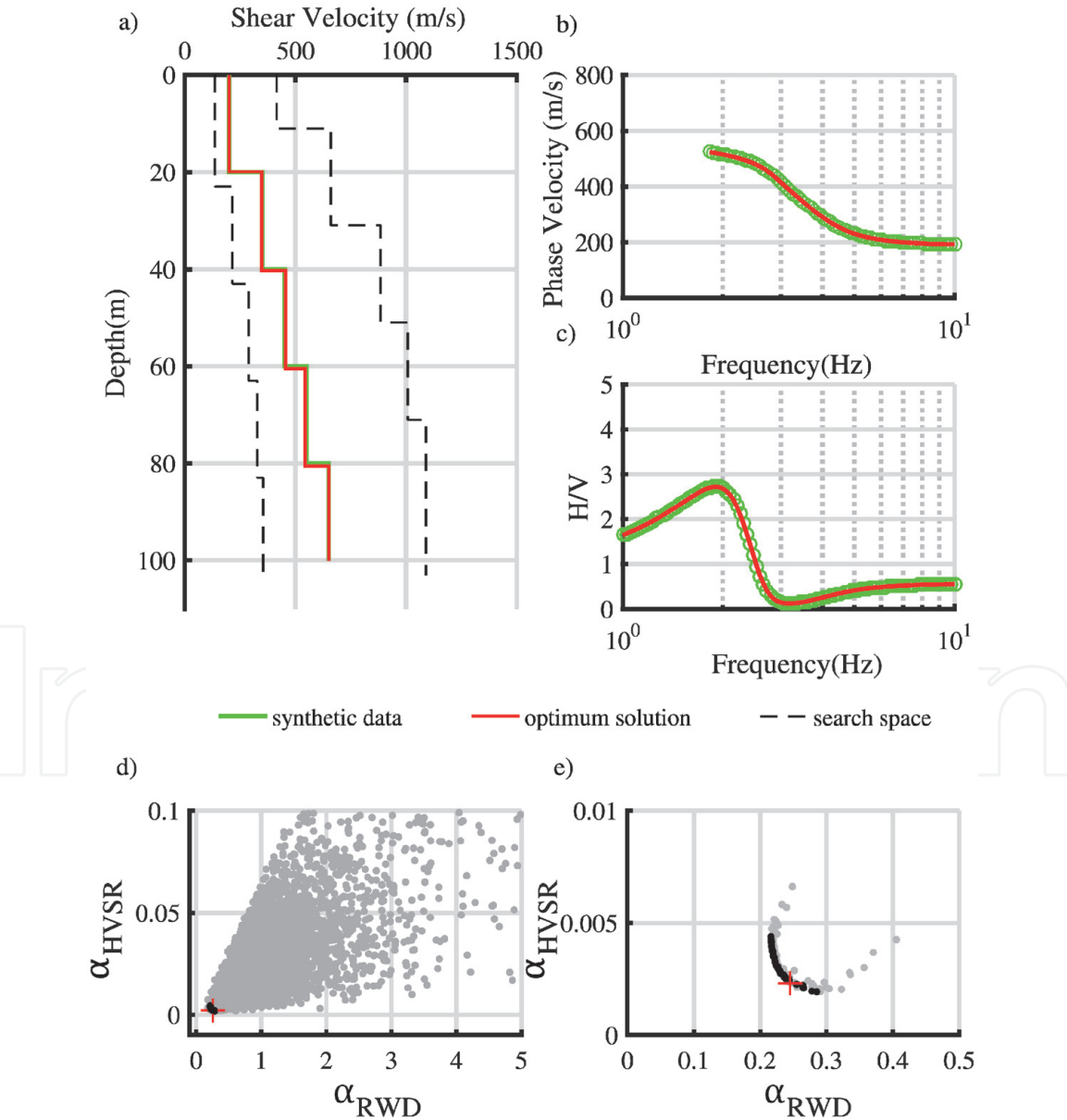


Figure 9. Results for SM-1. Synthetic model and a model obtained from the Pareto optimum particle and the parameter search space (a); the fit between the observed and calculated RWD (b), and HVS (c); the Pareto optimum solution marked as + and Pareto front (dark dots) with the dominated particles (light dots) for all iterations (d); and the Pareto optimum solution (+) with Pareto front and dominated particles at the last iteration (e) [8].

Hypercubes having more particles were preferred for leader selection. After a fitness hypercube was determined by using Roulette-wheel approach, a leader was selected randomly from this hypercube.

6. Results and discussions

The results of synthetic models (SM-1 and SM-2) are shown in **Figure 9** and **Figure 10**, respectively. As can be seen in **Figure 9a** and **Figure 10a**, MOPSO was successful in proving to obtain the real models with defined methods and parameters. These tests showing matched perfectly between synthetic data and model

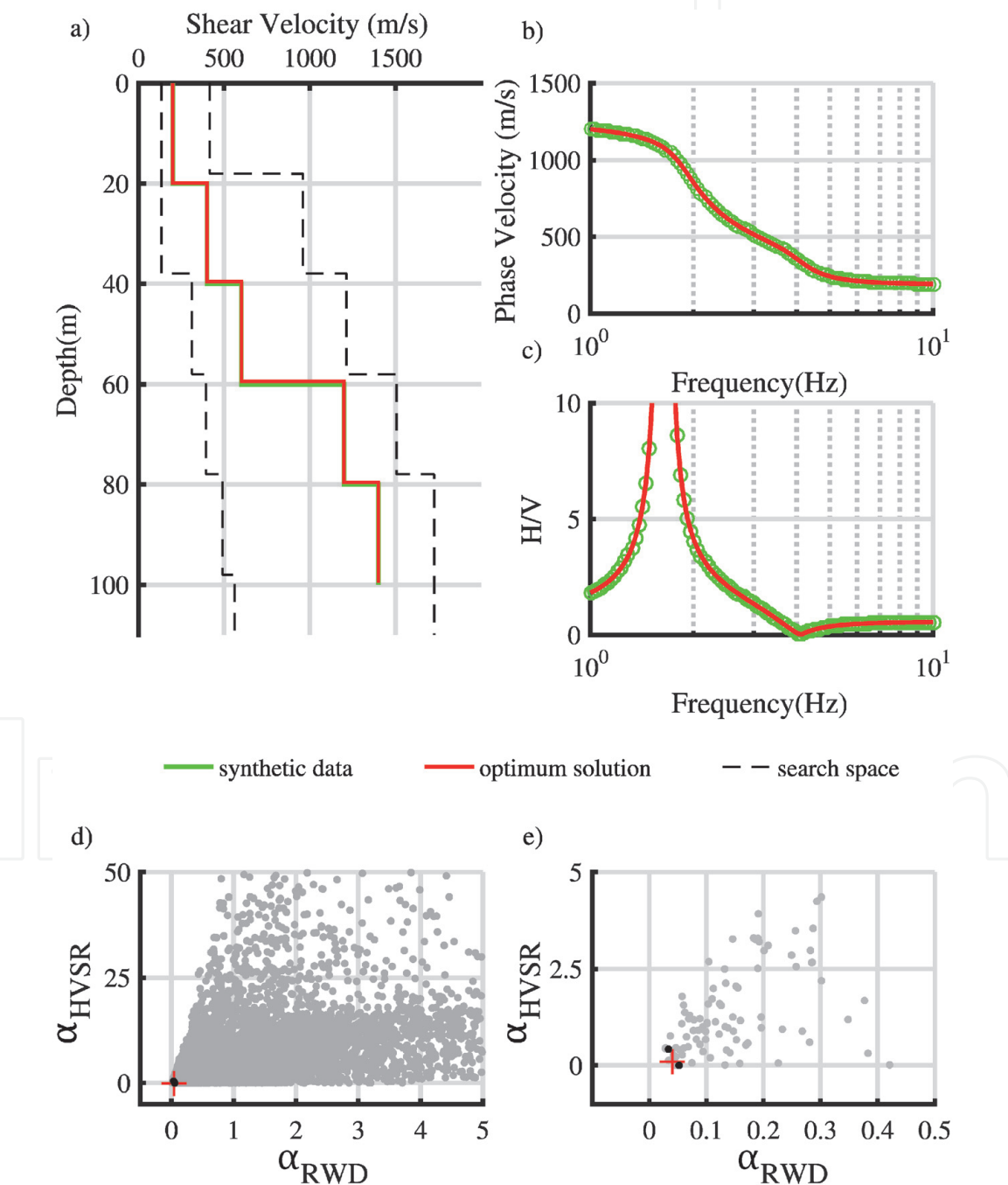


Figure 10. Results for SM-2. Synthetic model and a model obtained from the Pareto optimum particle and the parameter search space (a); the fit between the observed and calculated RWD (b), and HVS (c); the Pareto optimum solution marked as + and Pareto front (dark dots) with the dominated particles (light dots) for all iterations (d); and the Pareto optimum solution (+) with Pareto front and dominated particles at the last iteration (e) [8].

responses as seen in **Figure 9b**, **Figure 9c**, **Figure 10b** and **Figure 10c**, further strengthened our confidence in MOPSO modeling such seismological data. The most conspicuous observation to emerge from the result of both models was the distribution of dominated particles and Pareto front. Dominated particles in **Figure 9d** show a clear and balanced distribution for SM-1, however, as in **Figure 10d** dominated particles tend to towards the α_{RWD} axis for SM-2. This results showed that if only HVSR optimization was done, many local minima and non-unique solutions could be obtained in modeling of data obtained by response of such as SM-2 model. In contrast to earlier findings in [27], deviation from symmetry and divergence from an objective function axis were not only related to a improperly accomplished modeling, but also to the characteristic of HVSR optimization in case of model such as SM-2 that its optimization indicated non-uniqueness solutions. In addition, as shown in **Figure 9b**, spreading of Pareto front in SM-1 modeling showed a concave shaped curve that means an objective function could not further minimized without maximized to other objective function. However, for SM-2 model, Pareto front showed a similar distribution to L-shape, indicating that both objective functions were minimized independently and simultaneously.

7. Conclusion

This chapter considered is an overview of methods and parameters generally used in PSO and Pareto-based MOPSO, in the first step. The parameters and methods used in the literature are reliable but do not have an obvious superiority to each other. In spite of that MOPSO has been widely applied in many real world engineering problems, a few attempts have been made in order to modeling geophysical data. Until our previous study, this methodology have not been applied modeling seismological data. A set of solutions demonstrated in this chapter support the idea that MOPSO provides a powerful methodology for joint modeling of data having different sensivity. The present findings have important implications in order to solve weighting problem encountered in joint modeling approach. A clear benefit of MOPSO in the prevention of weighting-sum approaches could be clearly identified in this analysis. A further important implication is that divergence of particles from an objective function axis is not only related to properly defined parametrization and accomplished modeling, but also to non-uniqueness solutions. Our investigations into this point are still ongoing and seem likely to confirm our hypothesis. The evidence from this study implies that MOPSO is considered to be very attractive for joint modeling geophysical data in the future. However, further work needs to be performed to confirm whether MOPSO is beneficial to joint modeling of different types of geophysical data.

Acknowledgements

I am grateful to Prof. Dr. Abdullah Karaman and Assoc. Prof. Dr. Ekrem Zor for their help and valuable suggestions and discussions. Support was given by Istanbul Technical University and The Council of Higher Education (YÖK), who funded the work in its initial stages.

IntechOpen

IntechOpen

Author details

Ersin Büyük

Department of Geophysics, İstanbul Technical University, İstanbul, Turkey

*Address all correspondence to: ebuyuk@itu.edu.tr

IntechOpen

© 2021 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. 

References

- [1] Rao SS. Engineering Optimization: Theory and Practice. Fourth Edition. Hoboken, NJ, USA: John Wiley & Sons, Inc; 2009. 813 p. DOI: 10.1002/9780470549124
- [2] Van Den Bergh F, Engelbrecht AP. A study of particle swarm optimization particle trajectories. *Information Sciences*. 2006; 176: 937–971. DOI: 10.1016/j.ins.2005.02.003
- [3] Büyük E, Zor E, Karaman A. Rayleigh wave dispersion curve inversion by using particle swarm optimization and genetic algorithm. In: 19th European Geophysical Union EGU General Assembly; 2017; Vienna, Austria. p. 6911
- [4] Engelbrecht AP. *Computational Intelligence: An Introduction. Second Edition*. John Wiley and Sons, Inc; 2007. 628 p. DOI: 10.1002/9780470512517.
- [5] Blum C, Li X. Swarm Intelligence in Optimization. In: Blum C, Merkle D, editors. *Swarm Intelligence Introduction and Applications*. Berlin Heidelberg: Springer-Verlag; 2008. p. 43–85. DOI: 10.1007/978-3-540-74089-6_2
- [6] Pace F, Godio A, Santilano A, et al. Joint optimization of geophysical data using multi-objective swarm intelligence. *Geophysical Journal International*. 2019; 218: 1502–1521. DOI: 10.1093/gji/ggz243
- [7] Danek T, Leśniak A, Miernik K, et al. Pareto Joint Inversion of 2D magnetometric and gravity data-synthetic study. *E3S Web of Conference*. 2019; 133. DOI: 10.1051/e3sconf/201913301009
- [8] Büyük E, Zor E, Karaman A. Joint modeling of rayleigh wave dispersion and H/V spectral ratio using pareto-based multiobjective particle swarm optimization. *Turkish Journal of Earth Science*. 2020; 29: 684–695. DOI: 10.3906/yer-2001-15
- [9] Kennedy J, Eberhart R. Particle swarm optimization. *Neural Networks*. 1995; 4: 1942–1948. DOI: 10.1109/ICNN.1995.488968
- [10] Eberhart RC, Shi Y. Particle swarm optimization: Developments, applications and resources. In: *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*; 2001; 1: 81–86. DOI: 10.1109/cec.2001.934374
- [11] Fan H, Shi Y. Study on Vmax of particle swarm optimization. In: *Proceedings of the Workshop on Particle Swarm Optimization*. Indianapolis: Purdue School of Engineering and Technology, IUPUI. 2001
- [12] Ratnaweera A, Halgamuge SK, Watson HC. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Transactions on Evolutionary Computation*. 2004; 8: 240–255. DOI: 10.1109/TEVC.2004.826071
- [13] Ozcan E, Mohan CK. Particle swarm optimization: Surfing the waves. In: *Proceedings of the 1999 Congress on Evolutionary Computation, CEC*; 1999; pp. 1939–1944. DOI: 10.1109/CEC.1999.785510
- [14] Shi Y, Eberhart R. Modified particle swarm optimizer. In: *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*; 1998; pp. 69–73. DOI: 10.1109/icec.1998.699146
- [15] Kennedy J. The behavior of particles. In: Porto VW, Saravanan N, Waagen D, editors. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*.

- Berlin: Springer Berlin Heidelberg; 1998. p. 581–589. DOI: 10.1007/bfb0040809
- [16] Venter G, Sobieszczanski-Sobieski J. Particle Swarm Optimization. *AIAA Journal*. 2003; 41: 1583–1589. DOI: 10.2514/2.2111
- [17] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization. In: *Proceedings of the 1999 Congress on Evolutionary Computation, CEC; 1999*; pp. 1951–1957. DOI: 10.1109/CEC.1999.785513
- [18] Eberhart RC, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization. In: *Proceedings of the 2000 Congress on Evolutionary Computation, CEC; 2000*; pp. 84–88. DOI: 10.1109/CEC.2000.870279
- [19] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better. *IEEE Transactions on Evolutionary Computation*. 2004; 8: 204–210. DOI: 10.1109/TEVC.2004.826074
- [20] Coello Coello CA, Reyes-Sierra M. Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art. *International Journal of Computational Intelligence Research*. 2006; 2: 287–308. DOI: 10.5019/ijcir.2006.68
- [21] Eberhart RC, Shi Y. Computational Intelligence. In: Eberhart RC, Shi Y, editors. *Morgan Kaufmann*; 2007; p. 17–38. DOI: 10.1016/B978-155860759-0/50002-0
- [22] Kennedy J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 1999*. IEEE Computer Society; 1999; pp. 1931–1938. DOI: 10.1109/CEC.1999.785509
- [23] Engelbrecht AP. *Computational Intelligence: An Introduction*. John Wiley & Sons; 2005. p. 597. DOI: 10.1002/9780470512517
- [24] Janson S, Middendorf M. A hierarchical particle swarm optimizer. In: *2003 Congress on Evolutionary Computation, CEC 2003 - Proceedings*; IEEE Computer Society; 2003. pp. 770–776. DOI: 10.1109/CEC.2003.1299745
- [25] Deb K. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York USA: John Wiley & Sons; 2001. p 518.
- [26] Kozlovskaya E, Vecsey L, Plomerova J, et al. Joint inversion of multiple data types with the use of multiobjective optimization: problem formulation and application to the seismic anisotropy investigations. *Geophysical Journal International*. 2007; 171: 761–779. DOI: 10.1111/j.1365-246X.2007.03540.x
- [27] Dal Moro G. Insights on surface wave dispersion and HVSR: Joint analysis via Pareto optimality. *Journal of Applied Geophysics*. 2010; 72: 129–140. DOI: 10.1016/j.jappgeo.2010.08.004
- [28] Angeline PJ. Evolutionary optimization versus particle swarm optimization: Philosophy and performance differences. In: Porto VW, Saravanan N, Waagen D, et al., editors. *Evolutionary Programming VII*. Berlin: Springer, Berlin, Heidelberg; pp. 601–610. DOI: 10.1007/bfb0040811
- [29] Van Den Bergh F. *An Analysis of Particle Swarm Optimizers*. PhD Thesis, Department of Computer Science, University of Pretoria, Pretoria, South Africa, 2006; p. 284.
- [30] Coello Coello C, Pulido GT, Lechuga MS. Handling multiple objectives with particle swarm optimization. *IEEE Transactions on Evolutionary Computation*. 2004; 8:

256–279. DOI: 10.1109/
 TEVC.2004.826067

[31] Goldberg DE, Holland JH. Genetic Algorithms and Machine Learning. *Machine Learning*; 1988; 3: 95–99. DOI: 10.1023/A:1022602019183

[32] Van den Bergh F, Engelbrecht AP. A new locally convergent particle swarm optimiser. In: *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*. 2002, pp. 94–99. DOI: 10.1109/icsmc.2002.1176018

[33] Brits R, Engelbrecht AP, Bergh F Van Den. A Niching Particle Swarm Optimizer. In: *Proceedings of the 4th Asia-Pacific Conference on Simulated Evolution and Learning (SEAL 2002)*. Orchid Country Club, Singapore, 2002, pp. 692–696.

[34] Barrera J, Coello CAC. A review of particle swarm optimization methods used for multimodal optimization. In: Chee Peng Lim, Lakhmi C. Jain, Dehuri S, editors. *Innovations in Swarm Intelligence*. 2009, pp. 9–37. DOI: 10.1007/978-3-642-04225-6_2

[35] Kennedy J. Stereotyping: Improving particle swarm performance with cluster analysis. In: *Proceedings of the IEEE Conference on Evolutionary Computation, ICEC*. 2000, pp. 1507–1512.

[36] Pulido GT, Coello Coello CA. Using clustering techniques to improve the performance of a multi-objective particle swarm optimizer. In: Deb K, editor. *Genetic and Evolutionary Computation. Lecture Notes in Computer Science*: 2004; 3102: 225–237. DOI: 10.1007/978-3-540-24854-5_20

[37] Laumanns M, Thiele L, Deb K, et al. Combining convergence and diversity in evolutionary multiobjective optimization. *Evolutionary Computation*. 2002; 10: 263–282. DOI: 10.1162/106365602760234108

[38] Salazar-Lechuga M, Rowe JE. Particle swarm optimization and fitness sharing to solve multi-objective optimization problems. In: *Congress on Evolutionary Computation (CEC'2005)*. 2005; Edinburgh, Scotland, UK: IEEE Press; 2: 1204–1211. DOI: 10.1109/cec.2005.1554827

[39] Li X. A non-dominated sorting particle swarm optimizer for multiobjective optimization. In: Cantú-Paz E. et al., editors. *Genetic and Evolutionary Computation. Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg: 2003; pp. 37–48. DOI: 10.1007/3-540-45105-6_4

[40] Deb K, Pratap A, Agarwal S, et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. In: *IEEE Transactions on Evolutionary Computation*. 2002; 6:182–197. DOI: 10.1109/4235.996017.

[41] Deb K, Goldberg DE. An investigation of niche and species formation in genetic function optimization. In: *Proceedings of the third international conference on Genetic algorithms*. 1989; pp. 42–50.

[42] Raquel CR, Naval PC. An effective use of crowding distance in multiobjective particle swarm optimization. In: *GECCO 2005 - Genetic and Evolutionary Computation Conference*. New York, New York, USA: ACM Press, 2005; pp. 257–264. DOI: 10.1145/1068009.1068047

[43] Kennedy J. *Swarm Intelligence*. In: Zomaya A.Y. editor. *Handbook of Nature-Inspired and Innovative Computing*. Springer, Boston: 2006; pp 187–219. DOI: 10.1007/0-387-27705-6_6