SEISMIC VELOCITY INVERSION USING MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS

Vijay Pratap Singh

MINES PARIS

Collège doctoral

Ecole Doctorale n° **398** : "Géosciences et Ressources Naturelles"

exemplaire de soutenance

N° attribué par la bibliothèque

|__|__|__|__|__|__|__|__|__|__|

T H E S E

pour obtenir le grade de
Docteur de l'Ecole des Mines de Paris
Spécialité "Dynamique et Ressources des Bassins Sédimentaires"

présentée et soutenue publiquement par

Vijay Pratap SINGH

le 18 décembre 2006

APPLICATION DES ALGORITHMES ÉVOLUTIONNAIRES À LA DÉTERMINATION DE MODÈLES DE VITESSE PAR INVERSION SISMIQUE

Jury

M.	Bertrand DUQUET	Examinateur
M.	François JOUVE	Examinateur
M.	Gilles LAMBARÉ	Rapporteur
M.	Michel LÉGER	Examinateur
M.	Pascal PODVIN	Examinateur
M.	Marc SCHOENAUER	Directeur
M.	Médard THIRY	Examinateur
M.	Pierre THORE	Rapporteur

APPLICATIONS DES ALGORITHMES ÉVOLUTIONNAIRES À LA DÉTERMINATION DE MODÈLES DE VITESSES PAR INVERSION SISMIQUE

RÉSUMÉ

Enjeux :
Le pétrole ne se manifeste à distance par aucune propriété physique permettant sa découverte. C'est pourquoi l'exploration pétrolière consiste à imager par la méthode sismique les pièges susceptibles d'en contenir. Le but de la migration, ou rétropropagation numérique des enregistrements sismiques, est de former une image des structures géologiques en replaçant en profondeur les réflecteurs qui ont causé les échos enregistrés. Les variations de la vitesse de propagation des ondes, de 1500 m/s dans l'eau à 6000 m/s et plus dans les roches sédimentaires compactes, rendent cette tâche critique car un modèle de vitesse erroné donne une image très distordue. Le coût énorme des forages effectués sur des structures fausses impose l'obtention d'images précises du sous-sol et donc la détermination du champ des vitesses sismiques, surtout en contexte de piémonts lorsque les images sont peu lisibles.

Positionnement du sujet :
Toutes les méthodes de détermination des vitesses exploitent la redondance des données sismiques : chaque portion de réflecteur renvoie plusieurs échos correspondant à des couples source-récepteur dont le déport, la distance de la source au récepteur, diffère. Certaines méthodes telles que la tomographie fonctionnent bien lorsque les structures géologiques sont assez simples pour que les réflexions soient bien reconnaissables sur l'ensemble des enregistrements, mais ce n'est pas le cas dans les piémonts. Nous avons donc choisi la migration itérative, dont le principe est que, la Terre étant unique, les images obtenues avec les différents déports doivent être superposables. Ce critère ne suffisant généralement pas à déterminer les vitesses correctes, il est nécessaire d'introduire des informations géologiques. Pour l'optimisation du champ des vitesses, les méthodes de gradient étant d'implémentation fort lourde, nous avons choisi un algorithme évolutionnaire pour sa simplicité, son adaptabilité, et surtout son automaticité. De plus, la diversité de la population optimale donne une idée de l'incertitude qui entache le résultat.

Résultats :
Parmi tous les champs de vitesses possibles, bien peu ont une géométrie géologiquement acceptables, d'où l'idée de ne manipuler que des modèles satisfaisant au critère de coupe équilibrée. Une coupe est équilibrée lorsqu'elle est compatible avec les hypothèses de conservation des épaisseurs et des longueurs mesurées le long des couches. Dans une première partie, nous avons montré que l'on pouvait non seulement générer des modèles géométriquement plausibles, mais aussi les optimiser relativement à des données de pendage de couches ou de position de chevauchements disponibles à l'affleurement ou dans des puits. La seconde partie concernant l'optimisation des vitesses n'a pu être reliée à la première. Dans cette seconde partie, nous avons représenté le champ de vitesses par des grilles. Par le choix d'un algorithme évolutionnaire multi objectif, nous avons pu faire coopérer efficacement les critères de semblance et de semblance différentielle qui, tous deux, mesurent l'invariance de l'image migrée quant au déport. Nous avons amélioré le réalisme des solutions en les lissant dans la direction du pendage. Enfin, nous avons extrait, des écarts à cette invariance, des corrections des grilles de vitesse qui accélèrent notablement la convergence. Les résultats obtenus sur les données Marmousi, un cas synthétique réaliste, sont satisfaisants. Sur les données réelles de Mer du Nord, le dôme de sel reste un problème non résolu par les méthodes automatiques, mais ses environs sont bien imagés.

Transfert des résultats vers l'industrie :
Le principal intérêt de la méthode développée est son automaticité et sa souplesse. Son créneau est le dégrossisage rapide de problèmes difficiles, avant qu'un interprétateur ne reprenne la main avec des méthodes interactives plus poussées, mais aussi plus exigeantes en expérience et plus consommatrices de temps humain.

Mots clés :
Imagerie sismique, piémonts, analyse de vitesse, semblance, algorithmes évolutionnaires, algorithmes génétiques, multi objectif.

CHAPTER 1 ..... ***1***

INTRODUCTION ***1***

1.1. MIGRATION VELOCITY ANALYSIS 1
1.2. GLOBAL OPTIMISATION METHODS 2
1.3. REPRESENTATION 3
1.4. DOMAIN KNOWLEDGE 3
1.5. THESIS OVERVIEW AND CONTRIBUTIONS 4
1.6. SUMMARY OF CONTRIBUTIONS 6

CHAPTER 2 ..... ***7***

SURVEY OF EVOLUTIONARY ALGORITHMS (EAS) AND ***7***

MULTIOBJECTIVE EVOLUTIONARY ALGORITHMS (MOEAS) ***7***

2.1 INTRODUCTION 7
2.2. EA DEVELOPMENT HISTORY AND DISTINCTION BETWEEN EA'S 9
2.3.1 Generational Evolutionary Algorithms *9*
2.3.2 Steady-State Evolutionary Algorithms *10*
2.4. REPRESENTATION 11
2.5. OPERATORS FOR THE REAL–VALUED EA 12
2.5.1 Crossover (or Recombination) operator *12*
2.5.1.1. Discrete recombination (DR) *13*
2.5.1.2. Blend Crossover (BLX-α) *13*
2.5.1.3. Fuzzy recombination (FR) *14*
2.5.1.4. Simulated Binary Crossover (SBX) *15*
2.5.2. Mutation *16*
2.5.2.1. Random mutation *16*
2.5.2.2. Gaussian mutation *16*
2.5.2.3. EXP mutation *17*
2.5.2.4. Polynomial mutation *17*
2.5.2.5. Non-uniform mutation *17*
2.6. SELECTION 18
2.6.1. Proportional Selection *18*
2.6.2. Ranking selection *19*
2.6.3. Tournament selection *19*
2.7. MULTIOBJECTIVE OPTIMIZATION 21
2.7.1 Introduction *21*
2.7.2. Multiobjective Optimisation Problem (MOP) *22*
2.7.3. Pareto Concept *23*
2.7.4. Pareto Dominance *24*
2.7.5 Pareto Optimality *24*
2.7.5.1. Non-dominated set *24*
2.7.5.2. Globally Pareto-optimal set *25*
2.7.5.3. Locally Pareto-optimal set *25*
2.7.6 Multiobjective optimisation approach *26*
2.7.6.1. Aggregate fitness *26*
2.7.6.2. Pareto based ranking *26*
2.7.6.3 ε-dominance ranking *28*
2.7.7 Overview of Multi-Objective Evolutionary algorithm (MOEA) *28*
2.7.7.1. Non-Pareto based approaches *29*
2.7.7.2. Pareto based approaches *30*
2.7.7.3 Nondominated Sorting Genetic Algorithm (NSGA) *31*
2.7.7.4 Strength Pareto Evolutionary Algorithms (SPEA): *32*
2.7.7.5 Pareto Archived Evolutionary Strategy (PAES) *33*
2.7.7.6 Pareto Envelope-based Selection algorithms (PESA) *35*

2.7.7.7 Non-Dominated Sorting Genetic Algorithm II (NSGA-II) ... *36*
2.7.7.8 Strength Pareto Evolutionary Algorithms 2 (SPEA2): ... *38*
2.7.7.10 Micro Genetic Algorithms ... *43*

CHAPTER 3 ... ***45***

VELOCITY MODEL DETERMINATION METHODS FOR COMPLEX SEISMIC DATA ... ***45***

3.1 INTRODUCTION ... 45
3.2 TIME DOMAIN METHODS ... 46
3.2.1 Tomography methods ... *46*
3.2.2 Full waveform inversion methods ... *47*
3.3. Depth domain methods ... *48*
3.3.1 Tomography migration velocity methods ... *48*
3.3.2 Image domain methods ... *50*
3.3.2.1 Image perturbation criteria ... *51*
3.3.2.2 Differential Semblance Optimisation ... *53*
3.3.3 Semblance function ... *55*
3.4. KINEMATICS OF THE IMAGE IN OFFSET AND ANGLE DOMAIN ... 57
3.4.1. Kinematics of the local–offset gathers for horizontal reflectors ... *57*
3.4.2. Kinematics of the Angle gathers for horizontal reflectors ... *59*
3.5 MIGRATION ... 59
3.5.1 Kirchhoff migration ... *60*
3.5.2. DOWNWARD CONTINUATION METHODS ... 62
3.5.2.1 Shot-profile migration (SPM) ... *63*
3.5.2.2 Source-receiver migration ... *64*
3.5.3 Comparison of Kirchhoff and wavefield-continuation method: ... *65*
CONCLUSIONS ... 66

CHAPTER 4 ... ***67***

REPRESENTATIONS FOR EVOLUTIONARY MULTI-OBJECTIVE SUBSURFACE IDENTIFICATION ... ***67***

4.1. INTRODUCTION ... 67
4.2 VORONOI-BASED REPRESENTATIONS ... 69
4.3 GEOLOGICAL KNOWLEDGE ... 71
4.3.1 Geology ... *72*
4.3.2 Kinematic Models ... *72*
4.3.3 The Contreras Model ... *73*
4.4. THE EVOLUTIONARY ALGORITHM ... 74
4.4.1 The Representation ... *74*
4.4.2 Initialization and Variation Operators ... *75*
4.4.2.1 Initialization ... *76*
4.4.2.2 Crossover Operator ... *76*
4.4.2.3 Mutation Operator ... *77*
4.4.2.4 The ε-Multiobjective Evolutionary Algorithm ... *77*
4. 5 FIRST RESULTS IN GEOLOGICAL MODELING ... 78
4.5.1. The Geological Identification Problem ... *78*
4.5.2 The Evaluation Functions ... *78*
4.5.3 A Three Fault Model ... *79*
4.5.4 Results on the Seven Fault Model ... *81*
4.5.5 Convergence test ... *83*
4.5.6 Discussion on foothill identification ... *84*
4.6 Velocity inversion of a foothill structure ... *84*
4.6.1 Two- step seismic velocity inversion ... *85*
4.6.2 Single step seismic velocity inversion ... *87*
4.7: BACK TO GRID THE REPRESENTATION ... 88
4.8 CONCLUSIONS ... 93

..... 94

CHAPTER 5 94

INGREDIENTS OF MIGRATION VELOCITY ANALYSIS..... 94

5.1 INTRODUCTION 94
5.2 THE FITNESS FUNCTION..... 95
5.2.1 Offset Domain Differential Semblance (ODS) 96
5.2.2 Modified Offset Domain Differential Semblance function 99
5.2.3 Angle Domain Differential Semblance (ADS)..... 101
5.2.4 Modified Angle Domain Differential Semblance (MADS) 106
5.2.5 DIFFERENTIAL SEMBLANCE OR SEMBLANCE ? BOTH! 108
AUTOMATIC GROSS VELOCITY ERROR ESTIMATION 109
5.3 INTRODUCTION OF RMO 109
5.3.1 RMO and Radon Transform..... 110
5.3.2 Envelop 116
5.3.4 Results and Discussion..... 117
5.4 STRUCTURAL TRENDS AND DIP 118
5.4.1 Sobel operator..... 118
5.4.2 Example of Application..... 120
5.5 CONCLUSIONS..... 122

CHAPTER 6 123

SEISMIC VELOCITY INVERSION USING MULTI-OBJECTIVE EVOLUTIONARY ALGORITHMS..... 123

6.1 Introduction 124
6.2 Multi-objective Evolutionary Algorithms..... 126
6.3 *ε*-MOEA FOR VELOCITY INVERSION..... 129
6.4 Customized hybrid ε-MOEA for velocity inversion..... 131
6.5. Main components of the customized ε-MOEA 134
6.5.1. Representation of velocity model 135
6.5.2. Objective functions..... 135
6.5.3. Dip smoothing..... 137
6.5.4. RMO correction 137
6.5.5. Synthesis of a parent 139
6.5.6. Reference Models..... 141
6.6 Implementation of Customized Evolutionary Components 142
6.6.1 Initial Population..... 142
6.6.2. Guided crossover 146
6.6.2.1 Horizontal synthesis of Parent..... 146
6.6.3 Reference Models..... 150
6.7. Results..... 158
6.7.1. Evolutionary Algorithms and parameters..... 158
6.7.2. Marmousi Velocity Model..... 161
6.7.2.1. 250m Grid sampling..... 162
6.7.2.2. Superimposition of migrated image 170
6.7.2.3 100m Grid sampling..... 171
6.7.3. L7 Model 175
6.8 DISCUSSION AND CONCLUSION..... 179

CHAPTER 7 181

CONCLUSION AND PERSPECTIVE..... 181

PERSPECTIVES..... 182

TWO DIMENSIONAL VELOCITY INVERSION..... 182

THREE DIMENSIONAL 183

Chapter 1

Introduction

To have significant geological information of earth requires an integration of different types of data and approaches. However one of the most successful approaches uses geophysical methods based on seismic data. Goal of seismic data processing is to convert a recorded wave field into a structural or lithological image of the subsurface. This requires a model of the wave propagation velocities of the subsurface. Nevertheless obtaining this model is often the most difficult processing step, in areas of complex structure such as foothill or salt body. The goal of this work is to develop an automatic seismic velocity estimation technique for such region using Migration Velocity Analysis (MVA) and global optimisation methods.

1.1. Migration Velocity Analysis

Most velocity estimation methods are based on the measurement of the kinematics of the reflections. An important difference among them lies in the way kinematics are measured, either directly from the data in the time domain, or after migration in the image domain. When geology structures are mild and lateral velocity variations are smooth, the kinematics measured on data space are usually robust and accurate. However in the presence of complex structure and/or strong lateral velocity variations, measurement of kinematic on the image space is more robust and accurate, even if migration velocity is

far from true velocity. Since migration focuses the energy, the incomplete focusing of reflection is a measurement of velocity error. Velocity estimation methods that use the focusing capability of migration to extract kinematics error are commonly known as Migration Velocity Analysis (MVA) methods.

MVA is an iterative process where each iteration is made of two distinct steps: (1) data are imaged by prestack migration and (2) the velocity function is updated based on the kinematics error measured on the migrated data. Due to the highly non-linear relationship between the velocity model and the focusing quality of image, velocity estimation is better solved when posed as optimisation problem. Both gradient and global optimisation methods have been used to estimate velocity. But because the objective function that is optimised during MVA is non-convex and has several local minima, the quality of initial velocity model is crucial for global convergence with gradient methods. On the other hand extreme computation cost is severe hindrance with global optimisation methods. The goal of this thesis is to develop a robust and efficient migration velocity estimation technique that uses global optimisation methods and remains computationally tractable.

1.2. Global optimisation methods

The objective of global optimization is to find the globally best solution of model, in the (possible or known) presence of multiple local optima. Since velocity optimisation problems in geologically complex regions are non-linear, non-convex and ambiguous, we have chosen Evolutionary Algorithms (EAs) for optimisation. EAs are stochastic search methods that mimic the metaphor of natural biological evolution. EAs operate on a population of potential solutions, applying the principles of blind variation plus survival of the fittest to produce hopefully better and better solutions.

Generic EAs have to search a large parameter space with a little exploitation of domain-specific information from previous iteration to find global optimal solutions. Consequently they are usually computationally expensive and/or have to give up on

precision. In this thesis attempts have been made to reduce the computational cost of EA by adding geological and geophysical knowledge to the component of the algorithms, at the representation level, in the variation operators and in the objective function.

1.3. Representation

Acute computation demand of global methods imposed to a concise representation of velocity model. A lot of attempts have been made for representations e.g. B-spline, Bi-cubic, horizontal layers, and Voronoi representation. It has been noticed that some representations are very concise, however unable to represent a real geological structure, whereas other which are geologically significant require a large number of parameters. Therefore in this thesis we made an attempt to represent a velocity model for foothill structure that is concise as well as geologically significant.

In this work, after some effort to design a concise and geologically meaningful representation, we finally concluded that the grid representation was the most flexible one, even though it implies a large number of unknown parameter and induces a high computational cost. Hence we started to look for domain specific ways to reduce the computational cost.

1.4. Domain Knowledge

Traditional methods extract a lot of information about the velocity model from the geological knowledge and the migrated data to correct the velocity model, as for example the geological knowledge that "generally velocity increase with depth, varition of velocity along layer is small" and "salt body have almost fix velocity". A lot of information about the velocity can also be extrated from the gathers using residual move out (RMO) curves. They also provide information about global as well as local goodness of a velocity model. Hence we decided to use these information in our approach and have therefore developed domain specific operators. The result is a customised hybrid algorithms.

1.5. Thesis overview and contributions

Review of EAs and MOEAs: Because we made the choice of EAs as optimisation methods we started by giving in chapter 2, an overview of Evolutionary Algorithms. First, concepts and development of evolutionary algorithms along with different components of EAs (crossover, mutation, and selection) are described. Multiobjective optimisation and related concept are then introduced and state-of-art MultiObjective Evolutionary Algorithms (MOEAs) are surveyed.

State of art in velocity optimisation tools: In chapter 3, we review the state of the art of seismic velocity estimation techniques. We discuss time domain techniques i.e. tomography and waveform inversion, depth domain techniques i.e. tomography migration velocity analysis, and fully automatic techniques based on Differential Semblance Optimisation (DSO).

Representation: In chapter 4, we present a cross-section balancing representation for foothills (Singh et al. 2005a and 2005b). The goal is to represent a subsurface structure in a geologically sound manner, thus obtaining a concise representation of velocity models. We successfully apply this representation to the seismic velocity inversion with optimised geological structure (Singh et al. 2005). However the limited success of a first attempt to simultaneously invert both geological and geophysical criteria leads to go back to grid representation. At this point, looking for an optimal grid size, some experiments on different grid sampling demonstrate that a too coarse sampling may lead to ambiguity.

Ingredients of velocity inversion: In Chapter 5, we modify the Differential Semblance function (Symes, 1991) in both offset and angle domains. Modified differential semblance functions are "more convex" than the original one, and less sensitive to migration parameter settings, even for large velocity errors (Singh et al. 2005) (we shall use those modified differential semblance function and semblance function simultaneously to optimize velocity model in Chapter 6). We also present an automatic

Residual MoveOuts (RMO) extraction technique from gathers to estimate the approximate velocity error (that will be used in Chapter 6 for correcting the velocity error of the models during their optimization). Structural trends of the geological model provide significant information about the velocity model. Here we extract this information and use it reduce the velocity variation along the layers.

Automatic seismic velocity estimation using MOEA: In Chapter 6, we present our main realization, the algorithm for automatic seismic velocity inversion, and some results obtained on both two problems. This velocity analysis method inherits the characteristics of wavefield extrapolation, mainly robustness in presence of large and sharp velocity contrast, as well as its ability to cope with multipathing. Since a global optimisation method is used, we are free from linearization of the wave equation. The basis of our algorithm is a Multi-Objective Evolutinoary Algorithm, but, in order to increase its efficiency, we first customize the algorithm itself, and also propose a new exploitation operator. The goal of the customization is to strive to have both the robustness of global methods and the efficiency of local optimisation methods. We present examples of migration velocity analyses mainly on Marmousi model, together with a few results on the North-Sea L7 model. We demonstrate that our automatic velocity analysis technique is able to cope with large velocity error and is as efficient as the gradient methods except in salt body.

Conclusions and Perspective: In Chapter 7, we conclude our research and discuss about the possible 3D extension of our approach. Also, there is a need for adding human information during the optimisation, and integrating with some other linearized approach (Save and Biondi, 2004; Shen et al., 2004) to get a precise model. For 3D extension one may require to smartly and efficiently use migration algorithms, exploit the information from migrated data for further improvement and generation of models.

1.6. Summary of Contributions

In this thesis our first contribution is the development of an automatic cross-section balancing algorithm for foothill structure (Chapter 4). It can be generalized to other structures. We made an effort to analyze the influence of different type of representations (i.e. Voronoi, geological, and grid). We notice that representation is mostly geologically dependent and that the number of parameter is not the main issue when choosing a representation. Our second and major contribution is the development of an automatic tool for seismic velocity estimation, using a customized MOEA, and domain-specific operator where we have introduced domain knowledge. Using such tool, we were able to solve model with a large number of unknown parameters at a reasonable computation cost.

Chapter 2

Survey of Evolutionary Algorithms (EAs) and Multiobjective Evolutionary Algorithms (MOEAs)

This chapter provides a quick overview of evolutionary algorithms (EAs). First we give a general and historical introduction of EAs, and then we introduce variants (i.e. crossover, mutation) and selection operators. Finally we introduce the concept of multiobjective optimisation, and give a brief outlook of state of the art MOEAs.

2.1 Introduction

The buzzword doing the beats at all hierarchical levels of the industry today is optimisation. Calculus had been the reigning emperor of optimisation techniques until recently, when global optimisation techniques have been put to use. Among those various techniques, one of the most promising is Evolutionary Computing (EC), which mimics the natural process of evolution. It is based on Charles Darwin's theory of evolution, where nature selects the best genetic settings to survive in the next generation and some random change may occur during next generation birth. Evolutionary algorithm similarly selects the best performing solutions from the current population, and uses the variation operators of crossover and mutation to generate further solutions. An important feature of biological evolution is robustness - which is what evolutionary algorithm (EA) strives to achieve.

Essentially, EAs are a method of "breeding" solutions of a optimisation problem by means of simulating evolution. Since it is inspired by the natural selection and genetics, evolutionary computation borrows much of its dialect from genetics, cellular biology and evolutionary theory. In EC, a candidate solution is known as an *individual*. The collection

of current individual in the system is collectively known as the *population*. The actual encoding of an individual's solution is known as its *genome* (or *chromosome)* and representation is known as *genotype*. The way solution operates when tested in the problem environment is known as the individual's *phenotype*. When the individuals are modified to produce new individuals, they are said to be *breeding*. After the evaluation, a individual gets a mark, known as its *fitness*, which indicate how good a solution it is. The period of evaluation and assignment of fitness to a individual is known as *fitness assessment*. The whole process of finding an optimal solution is known as *evolving a solution*.

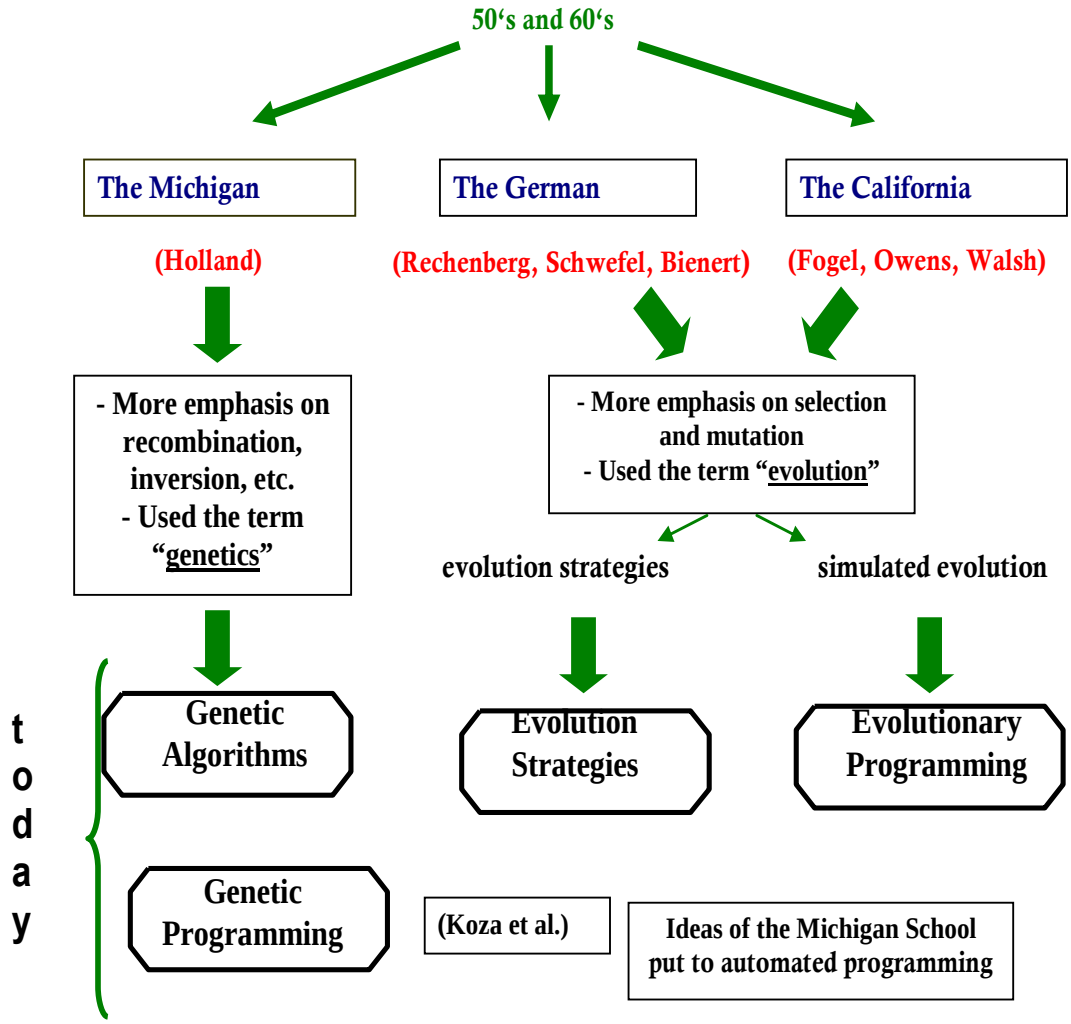


Figure 2.1 Developments of evolutionary

2.2. EA Development history and distinction between EA's

Historically, four evolutionary computation paradigms have emerged. They are: (i) evolution strategies, (ii) evolutionary programming, (iii) genetic algorithms, and (iv) genetic programming. Though during the last few years, these paradigms have crossbred and their lines of distinction have blurred, each of them had different features (Figure 2.1). The main difference between these evolutionary algorithms lies in their representation and genetic operators (Bäck, 1996). The operators are closely related with the underlying representation scheme of each evolutionary algorithm.

Genetic algorithms typically work on fixed-size bit-strings using crossover as its main operator. Evolution strategies are based on vectors of real values for representation and use mutation as the main operator. Evolutionary programming usually manipulates graphs using mutation as the single genetic operator. Genetic programming represents individuals as trees of flexible size. Another difference lies in the way selection is applied: GA use propositional selection and generational replacement, though rank based selection has become more popular over the years. Evolutionary strategies use deterministic replacement and no parent selection while evolutionary programming uses tournament replacement.

From now onward we will not detail all possible variation of EAs (see instance Eilven and Smith, 2003) but only the ones that are uses within Multi-objective EAs. There are two common high level, conceptual procedures made use of in evolutionary computation. The first one is the traditionally used generational EA, whereas the second approach is steady-state EA which is progressively a more popular newcomer.

2.3.1 Generational Evolutionary Algorithms

First, a set of random individuals (models) is generated. Then, each individual in the initial population is evaluated and fitness is assigned to them. The better individuals from the members of an initial or old population are selected for breeding and form a new population. This new population is evaluated and then mixed with the already existing

old population. An altogether optimal new population is created from this assembly (old + new population). This process is continued until an ideal individual is discovered or resources are exhausted. The pseudo code of the generational Evolutionary is as follows:

Algorithms : Generational Evolutionary Algorithm

Begin EA;

t = 0; *// Initializing time*

Random P(t); *// Initialize a usually random population of individuals:*

Evaluate P(t); *// Evaluate the fitness of all individuals*

While not done do *// Testing for termination criterion*

t = t + 1; *// Increasing time*

P'' = select P(t); *// Select a sub-population for offspring production*

Recombine P''(t); *// Recombine the "genes" of selected parents*

Mutate P''(t); *// Stochastically perturb genes of the mated population:*

Evaluate P''(t); *// Evaluate the new fitness*

P = survive P(t), P''(t) *// Select the survivors from actual fitness:*

End while;

End EA;

2.3.2 Steady-State Evolutionary Algorithms

In contrast to the generational EA, where a whole offspring population is created in every generation, in steady state EA only one or a few individuals are created and immediately integrated back into the parent population in each generation. The term steady-state means that in one step only a small change takes place without the whole population changing. The pseudo code of the steady-state EA is given below

Algorithms: Steady-state Evolutionary Algorithm

Begin EA;

t = 0; *// Initializing time*

Random P(t); *// Initialize a usually random population of individuals*

Evaluate P(t); *// Evaluate the fitness of all individuals*

While not done do *// Testing for termination criterion*

t = t + 1; *// Increasing time*

P''$_i$ = select P(t); // ***Select a few parents*** *for offspring production:*

Recombine P$_i$''(t); *// Recombine the "genes" of a few selected parents*

Mutate P$_i$''(t); *// Stochastically perturb genes of the mated parents*

Evaluate P$_i$''(t); *// Evaluate the new fitness of offspring*

P = survive P(t), P$_i$''(t) *// Select the survivors from actual fitness:*

End while;

End EA;

Of course the practical meaning of the word "generation" is fairly different in both cases: important population modification for generational evolutionary algorithms, modification of a few individual in for steady state evolutionary algorithms.

2.4. Representation

The choice of representation is one of the most critical point of the design of any EA, since it will likely have a strong impact on the algorithm's overall performance. The EA design decision should be parsimonious in defining the representation, to limit the size of search space and to avoid generating potentially infeasible solutions. The design should be based on the physics of the problem and it should also be constrained by the domain knowledge. To avoid the generation of infeasible solutions, each parameter should be constrained by the feasible range. In Chapter 4 few geological representation techniques and related issues are described.

Once the model is chosen, then might be different way to encode it (i.e. binary, real, integer, finite state automata). For a given problem, certain encoding (or encodings) may result in a relative compact search space, which is beneficial. For example binary encoding for discrete search space, real encoding for real valued parameters are most favorable. Similarly, structural encoding, such as grammatical encoding, is one of the most efficient ways to represent network topologies.

Our discussions about the EA operators are w.r.t real value. The various operators of the real–value EAs are briefly described in the following section.

2.5. Operators for the real–valued EA

In real-coded EA, parameter, considered as genes, are used directly to form a genotype. The use of real–valued EAs for real function optimisation facilitates the problem of coding and decoding genotypes and phenotypes. A genotype represents a solution, and population is a collection of such solutions. The operators (selection, recombination and mutation) modify the population of the solutions of any representation to create a new (and hopefully better) population. The various cross-over and mutation operators are briefly explained in the following subsections.

2.5.1 Crossover (or Recombination) operator

The recombination operator combines the genes of two or more parents to generate better offspring. The main purpose of a crossover operator is to recombine the partial good information from two or more parents so as to generate better offspring. Crossover occurs during evolution according to a user-definable crossover probability. The crossover plays a central role in EAs, in fact it may be considered as one of the algorithms defining characteristics. It is one of the components to be borne in mind to improve the behaviors of EAs (Liepins et al., 1992).

In real parameter EA, the main challenge is how to use the real parameter vectors to create a new pair of offspring vectors. In what follows, a real parameter crossover is

described followed by few real-parameter mutation operators. A good overview of many real parameter crossover and mutation operators can be found in Herrara et al. (1998)

2.5.1.1. Discrete recombination (DR)

DR corresponds to a standard uniform crossover in the binary case. Geometrically it is represented in Figure 2.2. Only cross sites are allowed to be chosen at the variable boundaries (green point). The offspring $O_{1i} \in (P_{1i}, P_{2i})$ for each $1 \leq i \leq n$. This crossover operator does not have adequate search power because the set of the possible values for each parameter is unchanged. Therefore we need mutation operator to change the set of possible values.

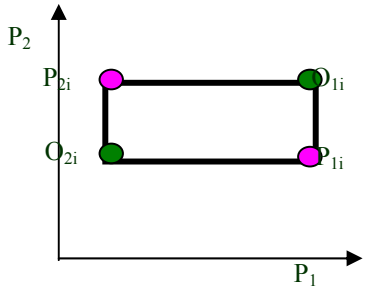


Figure 2.2: *Discrete recombination on two decision variables. Parents are represented in pink and offspring are represented in green. This operator does not have adequate search power.*

2.5.1.2. Blend Crossover (BLX-α)

The BLX cross-over was proposed by Eshelman and Schaffer (1993). For the ith parameter, and two parents P^t_{1i} and P^t_{2i} the blend recombination (BLX- α) creates at each generation t one offspring O^{t+1}_i that can be represented as follow:

$O^{t+1}_i = P^t_{1i} + \beta (P^t_{2i} - P^t_{1i})$,

where β is a random variable in the interval [-α , 1+ α]. The value α defines the size of area for possible offspring (Figure 2.3). If α value is more than 0 than it add exploration property whereas α =0 adds the exploitation property. If α is set to zero, this recombination creates a solution inside the range defined by the parents (see Figure 2.3) given area. It is also called arithmetic by Michalewig and intermediate by Rechenberg and Schwafel.

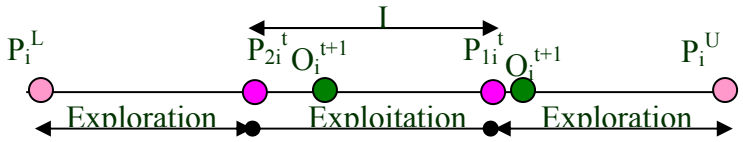


Figure 2.3: *The BLX- α operator. Parents are represented in pink, offspring in green and upper and lower limit in rose colour. BLX-a uniformly picks new individuals with values that lie in* $[P_{1i}- \alpha I, P_{2i}+ \alpha I]$.

If the distance between the parents solution is small, the difference between the offspring and parents solution is also small. This property makes the search operator partially adaptive.

2.5.1.3. Fuzzy recombination (FR)

Fuzzy recombination operator was proposed by Voigt et al. (1995). The probability that the offspring has the value O_i is given by a bimodal distribution

$p(O_i) \in (\varphi(P_{1i}), \varphi(P_{2i}))$

With triangular probability distribution $\psi(r)$ having the modal values P_{1i} and P_{2i} with

$P_{1i} - d.|P_{2i} - P_{1i}| \le r \le P_{1i} + d.|P_{2i} - P_{1i}|$

$P_{2i} - d.|P_{2i} - P_{1i}| \le r \le P_{2i} + d.|P_{2i} - P_{1i}|$

for $P_{1i} \le P_{2i}$ and $d \ge 1/2$. Geometrically, it is represented in Figure 2.4

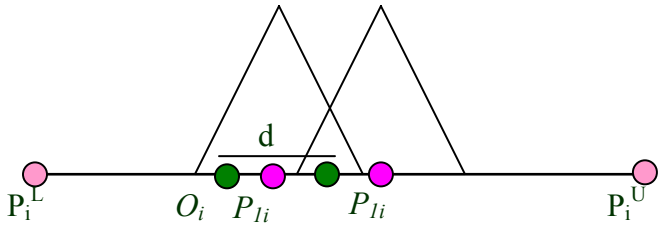


Figure 2.4: *The FR operator, parents are in pink colour and offspring in green, lower and upper limits in rose colour. Each triangle denotes the probability of the offspring to resemble each of its two parents.*

This operator gives importance to the creation of a solution near to its parents. The distribution can be changed with the parameter d. If d is large the solution will be away from the parents and vice versa.

2.5.1.4. Simulated Binary Crossover (SBX)

Simulated binary crossover (SBX) operator for real variables was first introduced by Deb and Agrawal (1995). This operator gives similar results to those that would be given if the parents were binary encoded, and binary single crossover were to be performed. This operator respects the interval schemata processing, in the sense that common interval schemata of the parents are preserved in the offspring's. The SBX crossover puts the stress on generating offspring in proximity to the parents. So, the crossover guaranties that the range of the children is proportional to the range of the parents, and also favors that near parent individuals are more likely to be chosen as children than individuals distant from the parents. These crossovers are self-adaptive in the sense that the spread of the possible offspring solutions depends on the distance between the parents, which decreases as the population converges.

The procedure of computing offspring P_{1i}^{t+1} and P_{2i}^{t+1} from the parent solutions P_{1i}^{t} and P_{2i}^{t} is as follows. First, a random number u_i is generated between 0 and 1, thereafter a spread factor $\beta_i = | P_{2i}^{t+1} - P_{1i}^{t+1} | / | P_{2i}^{t} - P_{1i}^{t} |$ is calculated using specified probability distribution function given below. These probability distribution functions are used to create an offspring using the following relation (Deb and Agrawal, 1995):

$P(\beta_i) = 0.5 (\eta_c + 1) \beta_i^{\eta_c}$, if $\beta_i \leq 1$;

$P(\beta_i) = 0.5 (\eta_c + 1) / \beta_i^{\eta_c}$, otherwise.

η_c (distribution index) is a non-negative real number. β_i is calculated by equating area under the curve equal to u_i using the following relation.

$$\beta_i = (2u_i)^{1/\eta_c + 1} \quad \text{if } u_i \leq 0.5 ;$$

$$\beta_i = (1/2(1-u_i))^{1/\eta_c + 1} \quad \text{otherwise .}$$

After getting the β_i from the above relation the offspring is calculated as follows

$$x_i^{(1, t+1)} = 0.5 [(1 + \beta_i) x_i^{(1, t)} + (1 - \beta_i) x_i^{(2, t)}] ,$$

$x_i^{(2, t+1)} = 0.5 [(1 - \beta_i) x_i^{(1, t)} + (1 + \beta_i) x_i^{(2, t)}]$.

A large value of η_c gives a higher probability to generate 'near parent' offspring (Figure 2.6).

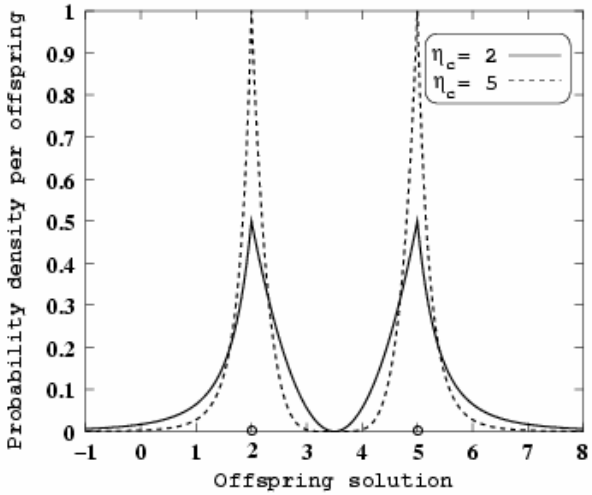


Figure 2.6: *A large value of η_c has a higher probability to generate offspring similar to the parents and vice versa (Deb and Agrawal, 1995).*

2.5.2. Mutation

Mutation provides a mechanism for maintaining diversity in a population. In one way, it acts as a safeguard against premature convergence by randomly changing the value of one or more allele in a chromosome. However, for real-coded genetic algorithms, it often plays the main role. Several mutation types are in use and we describe below few widely used ones.

2.5.2.1. Random mutation

Each variable that is going to be mutated is assigned a value, laying within its feasibility range (see Michalewicz, 1992).

2.5.2.2. Gaussian mutation

This mutation is similar to the previous one, except that the mutation step ΔPi is calculated according to a Gaussian' distribution. Smaller mutation steps are much more probable then large mutation steps. This is the standard in evolutionary strategies

(Rechenberg 1973, Schwefel 1977, Bäck 1995a). $O_i^{t+1} = r_i (P_i^U - P_i^L)$ where r_i is a random number between [0,1].

2.5.2.3. EXP mutation

This mutation type comes from the idea that the role of mutation at the beginning is to make larger jumps, whereas later on, as the search progresses, fine tuning is achieved by making smaller jumps.

2.5.2.4. Polynomial mutation

Polynomial mutation was proposed by Deb and Goyal (1995). If x_i is the value of the i^{th} parameter selected for mutation with a probability p_m and the result of the mutation is the new value y_i obtained by a polynomial probability distribution $P(\delta) = 0.5(\eta_m + 1)(1 - |\delta|)^{n_m}$ x_i^L and x_i^U are the lower and upper bound of x_i respectively and r_i is a random number in [0,1]

$$y_i = x_i + (x_i^U - x_i^L)\delta_i$$

$$\delta_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1 & \text{if } r_i < 0.5 \\ 1 - |2(1 - r_i)|^{1/(\eta_m+1)} & \text{if } r_i \geq 0.5 \end{cases}$$

The distribution is controlled by the parameter η_m (distribution index).

2.5.2.5. Non-uniform mutation

Non-uniform mutation was first introduced Michalewicz (1992). According to this, the probability to have the value y after mutation for the ith parameter is

$$y_i = x_i + \tau(x_i^U - x_i^L)\left(1 - r_i^{\left(1 - t/t_{\max}\right)}\right)$$

where t_{max} is the maximum number of generations, b a user is specified control parameter and τ a random bit, 0 or 1. The exponent of r_i approaches 0 as t approaches t_{max} and y_i is closer to x_i. This allows reducing the search closer to the optimum value as the evolution proceeds.

2.6. Selection

An evolutionary algorithm performs a selection process in which the best fitting members of the population survive, whereas, the "least fit" members are eliminated. In a constrained optimization problem, the notion of "fitness" partly depends on whether a solution is feasible (i.e. whether it satisfies all the constraints), and partly on its objective function value. Selection provides the driving force in an evolutionary algorithm and the selection pressure is a critical parameter. If the selection pressure is too high the search will terminate prematurely, and if the pressure is too low progress will be slower than necessary. There exists a variety of selection algorithms: Goldberg and Deb (1991) performed some analysis on some of the most commone algorithms used in Genetic Algorithms (GAs).

2.6.1. Proportional Selection

In proportional selection, the individuals are selected according to their relative fitness values. The selection probability of ith individual I_i^g at generation is defined as

$$\mathbf{P(I_i^g) = f(I_i^g) / \Sigma^{\lambda}_{i=1} f(I_i^g).}$$

This is a probabilistic selection method in which every individual having non-zero fitness will have a chance to be reproduced. This selection scheme is adopted by the simple genetic algorithm and believed to be the most similar mechanism that occurs in nature. One problem with the fitness-proportional selection is that it is directly based on the fitness. Assessed fitness is rarely an accurate measure of how "good" an individual really is.

2.6.2. Ranking selection

Ranking selection is a selection method which assigns selection probabilities solely on the basis of the rank i of individuals, ignoring absolute fitness values. In (μ, □) uniform ranking (Schwefel, 1995), the best μ individuals are assigned a selection probability of 1/ μ while the rest are discarded:

$$P(I_i^g) = \begin{cases} 1/\mu & 0 \le i \le \mu \\ 0 & \mu \le i \le \lambda \end{cases}$$

2.6.3. Tournament selection

This is the most popular selection mechanism. It is popular because of it is simple, fast and has well understood statistical properties. Tournament selection (Blickle and Thiele, 1995) is performed by choosing parents randomly and reproducing the best individual from this group. When the number of parents is q, this is called the q-tournament selection. The standard values of q are 2 and 7. Value 2 is used in GA whereas 7 is used in GP. Value 2 is weakly selective whereas value 7 is highly selective.

There are many other types of crossover operator like unimodal normally distributed crossover (UNDX) (Ono and Kobayashi, 1997), simplex crossover (Tsutsui et al., 1999), fuzzy connective based crossover (Herrara et al., 1995) and uniform average crossover (Nomura and Miyoshi, 1996). The comparisons of these crossover operators are mainly context dependent. The issue of exploration and exploitation make a crossover dependent on a chosen selection operator. Beyer and Deb (2001) argued that in most situations selection operator reduces the diversity. The reduction of diversity due to the selection operator can be related to the exploitation property of selection operator. Hence, in general, crossover operator should enhance the population diversity. Such a balance between the descent and ascent of diversity will allow EA to have an adequate search property (Figure 2.5) Based on this argument; two postulates have been recently suggested by Beyer and Deb (2001).

First population mean should be invariant before and after the crossover.

Second population diversity may increase after the crossover.

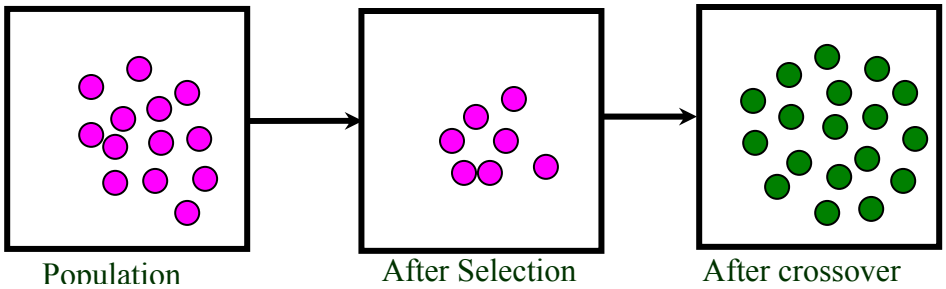


Figure 2.5: *When selection is reducing the diversity, the variation should increase the diversity. Balancing the two allows EAs to have adequate search property.*

Since real parameter crossover operator directly manipulates two or more real number to create one or more number as an offspring, one may wonder if there is a special need for using another mutation operator. The confusion arises because both operators seem to perform the same task, i.e. perturb every solution in the parent population to create a new population. The distinction among various operators lies in the extent of perturbation allowed in each operator. Although it is highly debated in literature, Deb (2000) believed that distinction between these two lies in the number of parent's solution used for perturbation process. If only one parent is used for perturbation then it is called mutation On the other hand, if more than one parents are involved for perturbation then it is called crossover.

2.7. Multiobjective optimization

2.7.1 Introduction

Multiobjective optimisation is different from single-objective optimization in that it involves consideration of more than one, and often conflicting, objective functions. Many real world design problems involve multiple, usually conflicting optimization criteria. Often, it is very difficult to weigh the criteria to build a global criterion. Multiobjective evolutionary algorithms use the principle of Pareto optimality, which states that, a model is "Pareto optimal" if it is not possible to improve it with respect to any criterion without worsening it with respect to at least one other criterion. This in turn allows the user to choose among many alternatives.

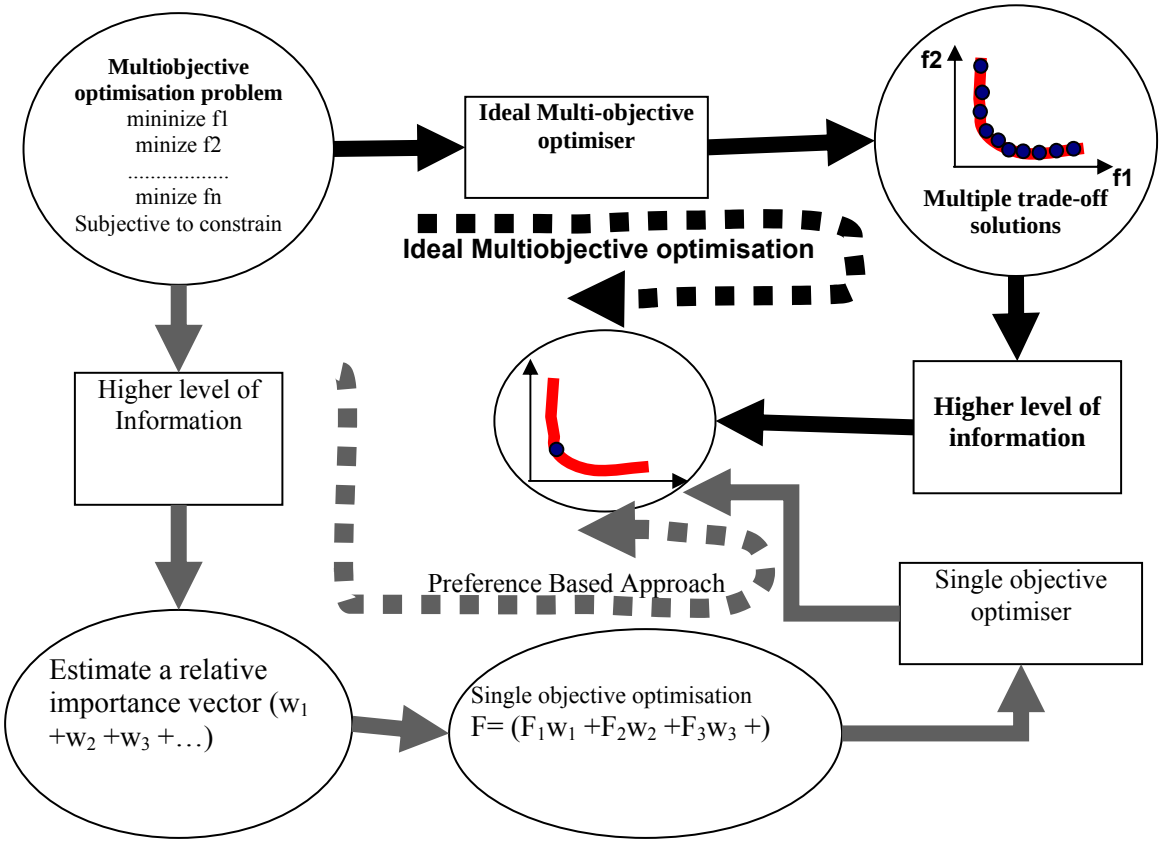


Figure 2.6: *Ideal Multiobjective Optimisation and traditional preference based optimisation. Traditional preference based methods require multiple run and need to define weight for obtaining trade of solution, whereas ideal multiobjective optimisation gives possible trade of solution in a single run.*

Multiobjective is not only different in terms of number of objective function but also in term of number of search spaces. Single objective optimisation has only parameter search space, whereas multiobjective optimisation has two search space parameter and objective space (Figure 2.7). As a result multi-objective optimizations produce a number of compromise solutions, known as the Pareto-optimal solutions. It is not a consequenc of algorithms, it is a choice. The task in a multi-objective optimization problem is to find as many Pareto-optimal solutions as possible. Classical optimization methods are not efficient in finding multiple Pareto-optimal solutions. The major difficulty with all classical methods is that they require multiple run and need to define weight to each objective to obtain multiple Pareto-optimal solutions (Figure 2.6) and they sometime can not find whole front (concave front).

Evolutionary algorithms (EAs) are ideally suited to solve multi-objective optimization problems, simply because multiple Pareto-optimal solutions can be captured in a single population by suitably modifying the EA selection operators. Additionally, evolutionary algorithms are less sensitivity to the shape or continuity of the Pareto front (Coello, 2001).

With multiobjective optimization problems, knowledge about the Pareto-optimal set helps the decision maker in choosing the best compromise solution. For instance, when simulating a mountain front, a geologist works on an existing model space exploring multiple model possibilities to arrive at the best possible model. Thereby, the model space is reduced to a set of optimal trade-offs. However, generating the Pareto-optimal set is computationally expensive and often infeasible, because of the complexity of the underlying application, which prevents exact methods from being applicable. EAs also do not guarantee the identification of optimal trade-offs but try to find a set of solutions that are (hopefully) not too far away from the optimal front (Figure 2.6).

2.7.2. Multiobjective Optimisation Problem (MOP)

A multiobjective problem (MOP), also called multi-criteria optimization, or multi-performance or vector optimization involves a vector of N parameter $\vec{x}$, a vector of M

objective functions, and a vector of L dimensional constraint $g_l(\vec{x})$. The objective function as well as constrains are functions of the parameter. Since minimizing $F_m(\vec{x})$ is equivalent to maximizing - $F_m(\vec{x})$, we may assume minimization in all cases without any loss of generality.

Minimizing $F_m(\vec{x}) = (f_1(\vec{x}), f_2(\vec{x}), \ldots\ldots\ldots f_m(\vec{x}))$ m=1, 2...M;

subjected to $g_l(\vec{x}) = ((g_1(\vec{x}), g_2(x), \ldots\ldots\ldots gl(\vec{x})) \leq 0)$ l= 1,2…L;

where $\vec{x} = (x_1, x_2, \ldots\ldots\ldots x_n)$ $x_i^L \leq x_i \leq x_i^U$ i= 1,2,…..N

Each parameter x_i has to take a value within a lower bound x_i^L and an upper bound x_i^U . If any solution $\vec{x}$ satisfies all the L constraint $g_l(\vec{x})$ and all the 2N variable bounds, it is called as a feasible solution, and infeasible in other cases. In the presence of constraints, the entire parameter space need not be feasible.

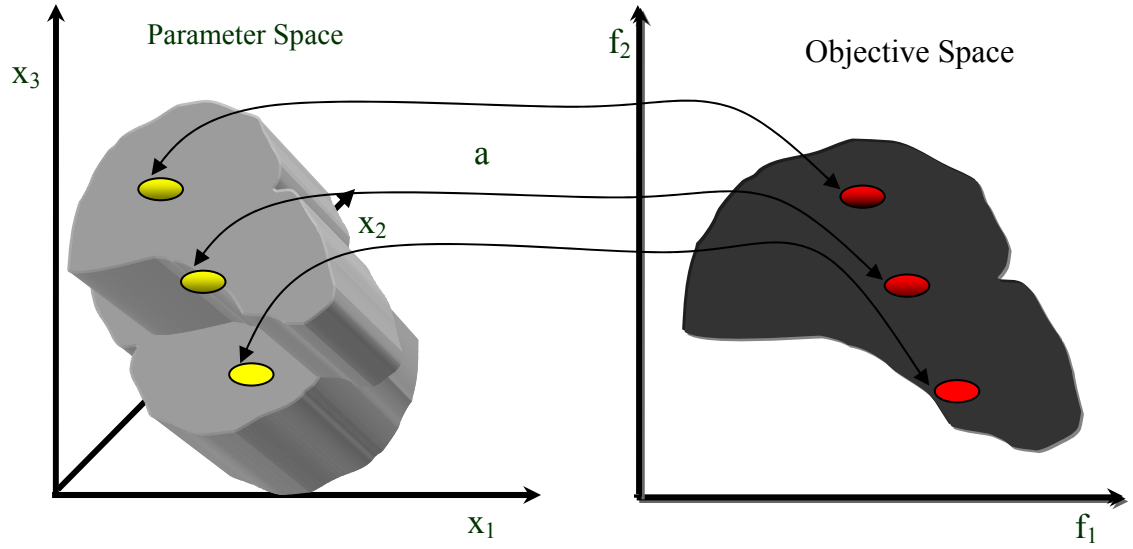


Figure 2.7: *Representation of parameter space and corresponding objective space.*

2.7.3. Pareto Concept

The concept of Pareto-optimum was first introduced by Vilfredo Pareto (Pareto 1886). The key Pareto concepts for the minimization problem are defined as follows:

2.7.4. Pareto Dominance

For any two decision vector $\vec{a} = (a_1, a_2, ...a_n)$ and $\vec{b} = (b_1, b_2, ...b_n)$,

$\vec{a} < \vec{b}$ ($\vec{a}$ dominates $\vec{b}$) if $F_m(\vec{a}) < F_m(\vec{b}) \quad \forall m, 1 \le m \le M$

$\vec{a} \le \vec{b}$ ($\vec{a}$ weakly dominates $\vec{b}$) if $F_m(\vec{a}) \le F_m(\vec{b})$

$\vec{a} \approx \vec{b}$ ($\vec{a}$ is indifferent to $\vec{b}$) if $F_m(\vec{a}) \ne F_m(\vec{b}) \wedge F_m(\vec{b}) \ne F_m(\vec{a})$

Based on the Pareto dominance concept in a multi-objective problem, a solution for which there is no way of improving any-objective without worsening at least one other objective is called a Pareto-optimal solution. A graphical representation of Pereto optimal (non-dominated) and dominated solutions is depicted in Figure 2.8.

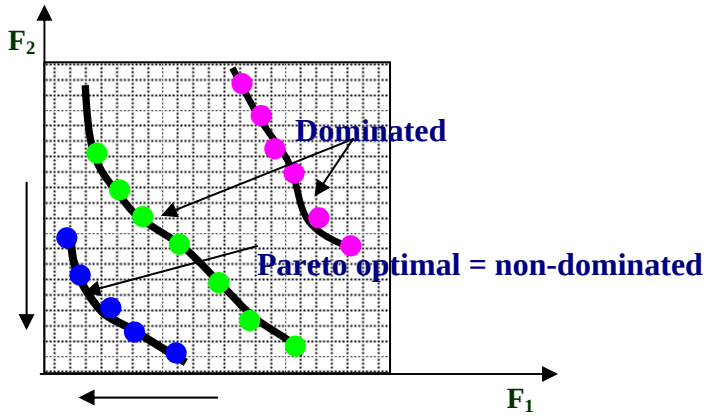


Figure 2.8: *Representation of Pareto optimal solutions and dominated solution. Blue front is the final Pareto optimal solution, whereas green and pink fronts were Pareto optimal at particular instant. Now they are dominated by Blue front solution.*

2.7.5 Pareto Optimality

2.7.5.1. Non-dominated set

Among a set of solutions S, the subset of non-dominated solutions S' are those that are not dominated by any member of S. In Fig. 2.8, blue dots represent the non dominated solutions and they are called Pareto-optimal solutions. Fig. 2.9 shows the Pareto-optimal sets for different possible combinations of maximization and minimization functions.

2.7.5.2. Globally Pareto-optimal set

The non-dominated set of the entire feasible search space S is the globally Pareto-optimal set.

2.7.5.3. Locally Pareto-optimal set

If for every member x in a set P, there exists no solution y (in the neighborhood of x such that $||y-x|| \infty \leq \varepsilon$, where ε is a small positive number) dominating any member of the set P, then the solutions belonging to the set P constitute a locally Pareto-optimal set (Miettinen, 1990 ; Deb, 1999c).

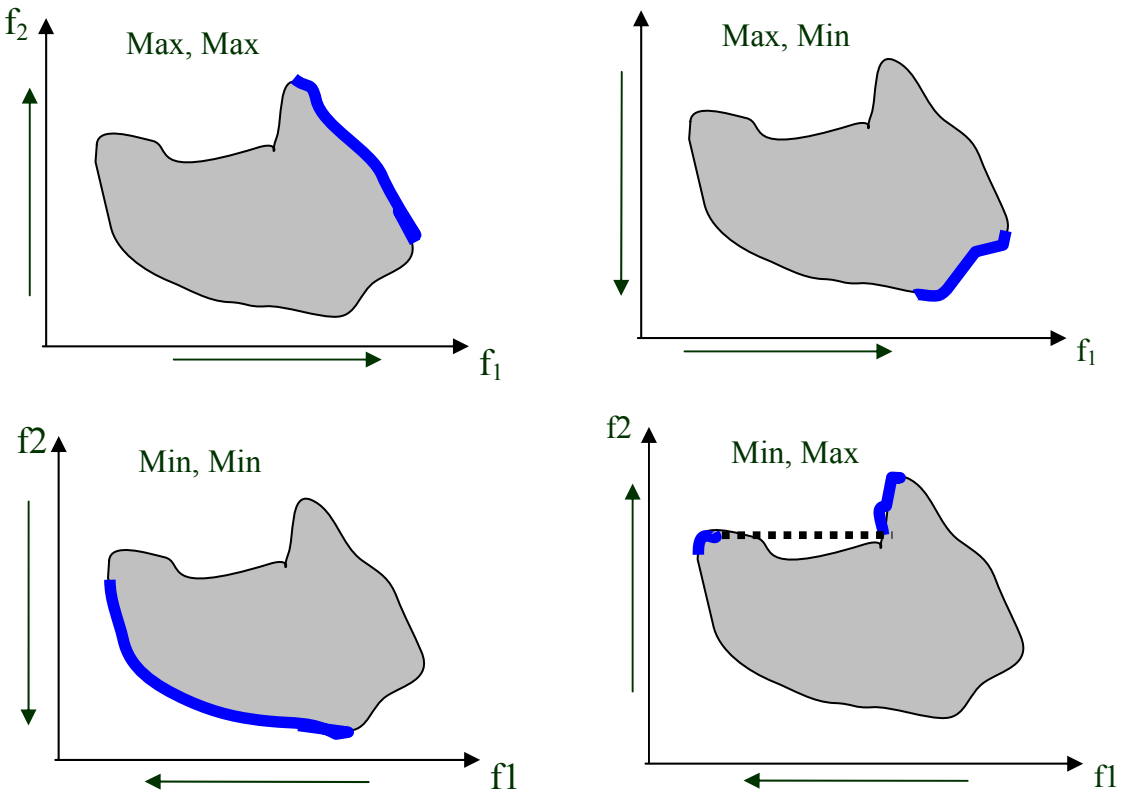


Figure 2.9: *Representation of Pareto front shape in different Min –Max condition of objective function.*

2.7.6 Multiobjective optimisation approach

To solve multiobjective problem with EA, the only things that need to be modified is the selection. In multiple-objective fitness assessment, an individual receives separate assessments in each of the criteria of interest, and the system must determine how to select individuals based on some function of these criteria. There are two common strategies for doing this, aggregate fitness and Pareto ranking. Now a day a recently developed approach based on ε-dominance concept is routinely used. Here we present aggregate fitness, Pareto ranking and ε-dominance based strategy for selection.

2.7.6.1. Aggregate fitness

Here the strategy is to join each individual's assessments into single aggregates fitness by which the individual is selected. The most straightforward aggregation approach is simply to add the various assessments as a weighted sum (Hajela and Lin 1992). An alternative way to do this is to set the standardized fitness to the maximum (worst) of the various fitness assessments (Wilson and Macleod 1993). A third aggregation technique known as lexicographic ordering is used with ranking and tournament selections (Fourman, 1985). Lexicographic ordering assumes that there is an order of importance among the criteria. Here, individuals are first sorted (for ranking or tournaments) by the most important criterion. Ties are then broken by ranking by the second criterion, then the third criterion, etc.

2.7.6.2. Pareto based ranking

In this set of method, individuals are ranked according to some Pareto based mechanism. There are two common techniques which result in Pareto ranking. The first technique, by Goldberg (1989), consists in assigning rank "1" to all the individuals who are not dominated rank '"2" to nondominated remaining individuals and so on. This ranks individuals in layers based on their domination of others (Figure 2.10a). The second technique by Fonseca and Flemming (1993) simply sets an individual's standardized fitness as the number of individuals in the population which dominate that individual (Figure 2.10 b). In this method ties may be broken by random choice.

The Pareto ranking procedures based on the above two procedures are summarized in Table 2.1.

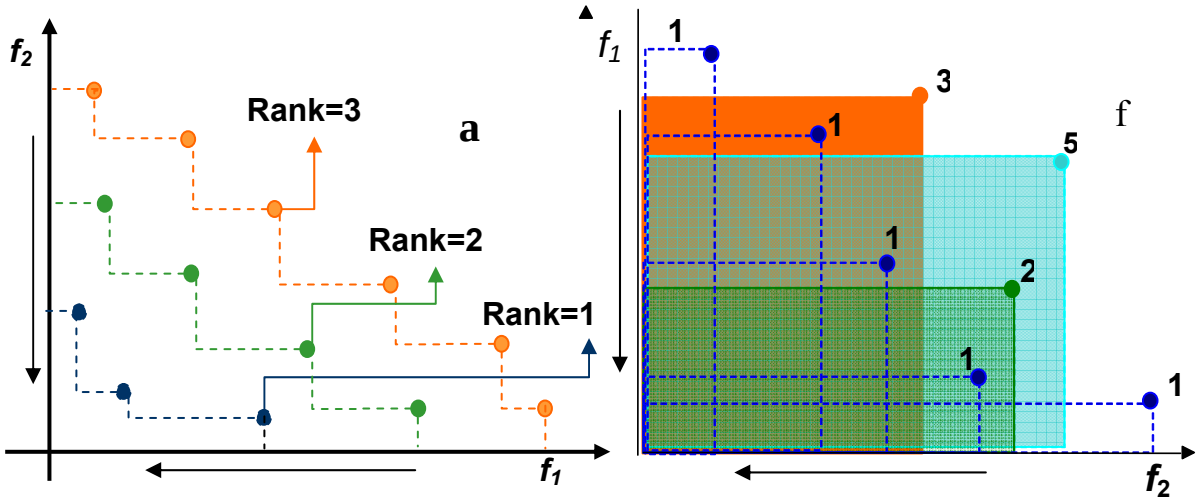


Figure 2.10: *Pereto ranking according to Goldberg's (1989) at left and according to Fonseca and Fleming (1993) at right.*

Table: *Steps of Pareto ranking for Goldberg's (1989) and Fonseca and Fleming (1993).*

procedure:	**procedure:**
Goldberg's method (1989)	**Fonseca and Fleming method (1993)**
input: individual f_k, k=1, 2, …, popSize	input: individual f_k, k=1, 2, …, popSize
output: fitness eval(fk)	output: fitness eval(f_k)
step 1: rank 1 is given to the nondominated individuals	step 1: rank 1 is given to the nondominated individual
step 2: removing them from contention.	step 2: removing them from contention.
step 3: finding the next nondominated individual, removing them from contention, rank 2 is assigned to them.	step3: finding the next nondominated individual, removing them from contention, rank equally to the number of its dominating individuals plus one.
step 4: process continues until the entire population is ranked, and output fitness eval(f_k).	step 4: process continues until the entire population is ranked, and output fitness eval(f_k).

2.7.6.3 ε-dominance ranking

The ε-dominance sorting is done in two steps: (1) sorting of non-dominance solutions (2) ε-box ($\varepsilon_i * f_i$, where f_i^{min} is the minimum possible value of the i_{th} objective and ε_i is the allowable tolerance in the i^{th} objective, below which the values are insignificant for the user.) is created for each solution and after that ε non-dominance box solutions are sorted. This approach maintains the diversity along the Pareto front and help in fats convergence. The concept of ε-dominance is illustrated (Figure 2.11).

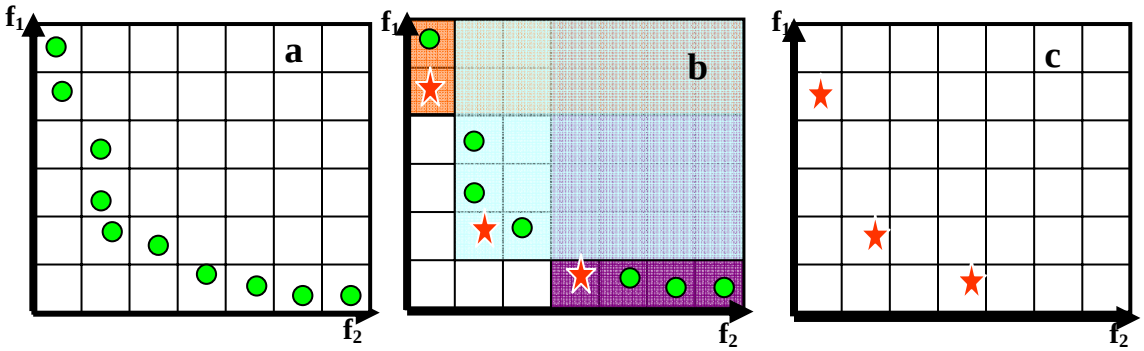


Figure 2.11: *Illustration of ε-dominance concept.(a) dominance solution (b) the ε-dominant solution by star and ε-dominant region by background colour (c) After sorting ε-dominant solution.*

2.7.7 Overview of Multi-Objective Evolutionary algorithm (MOEA)

Multi-objective problems (MOP) were first discussed by Rosenberg in 1967. The first reported implementation and test of a multi-objective evolutionary approach was the Vector Evaluated Genetic Algorithm (VEGA) by Schaffer (1984). Since this branch of EA has attracted many researchers dealing with non-linear, non-convex and integer-variable multi-objective optimization problems.

Recently, there has been a surge in research on new, and particularly genetic/ evolutionary multi-objective optimization algorithms and their applicability to various optimization problems (Coello, 2001; Corne et al., 2000; Deb et al., 2002; Jensen, 2003; Knowles and Corne, 2000; Tan et al., 2002; Zitzler and Thiele, 1999). Fonseca and Flemming (1993) classified the MOEA into three groups, namely: (i) plain aggregation based (ii) population-based non-Pareto, and (iii) Pareto-based approaches. The main

approaches used to handle the MOP are Non-Pareto and Pareto based approaches. The evolution of MOEA is shown in Figure 2.12

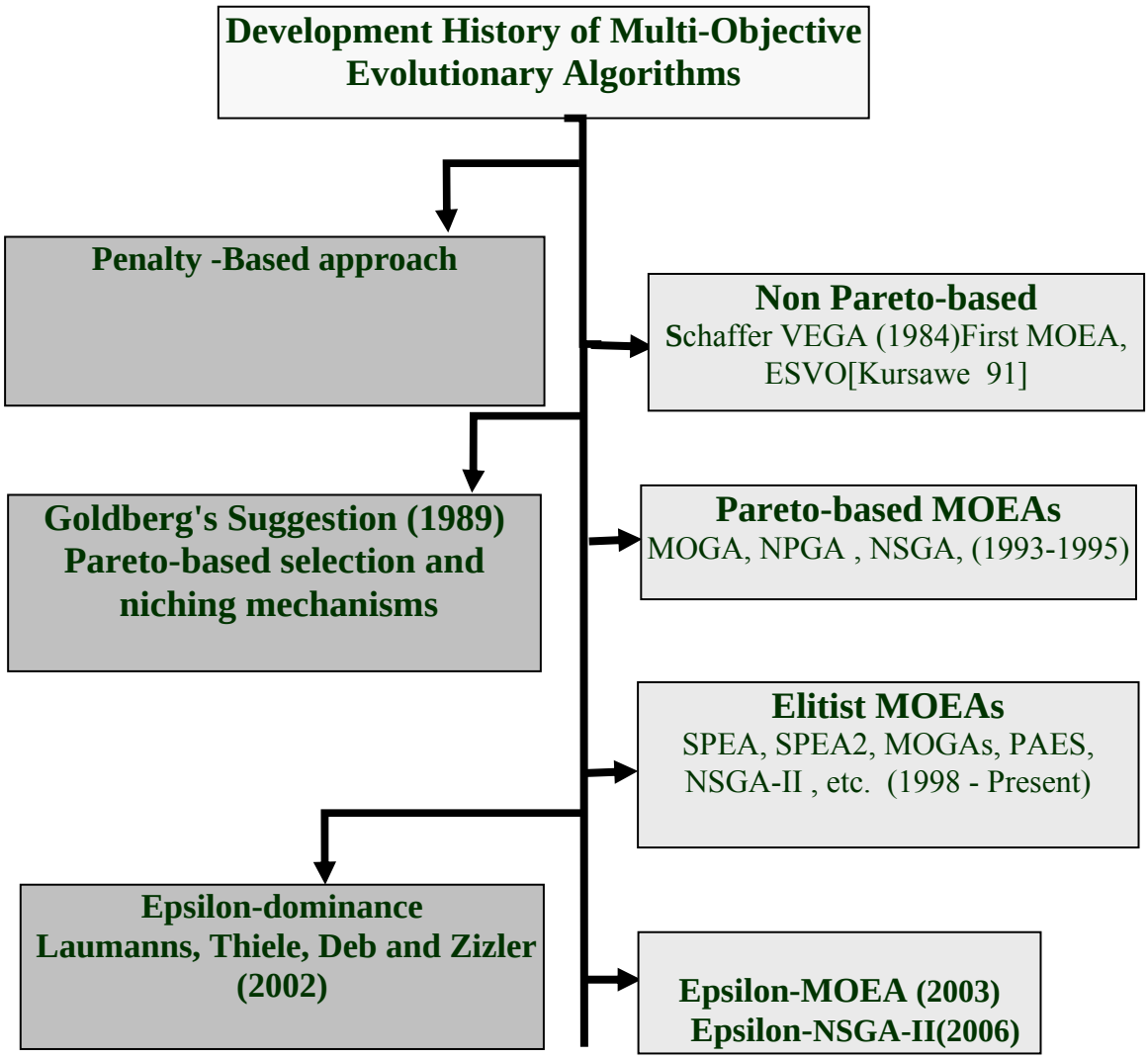


Figure 2.12: *Key developments of multi-objective evolutionary algorithms.*

2.7.7.1. Non-Pareto based approaches

The Vector Evaluated Genetic Algorithm (VEGA; Schaffer, 1985) is a non-Pareto based approach. It is a straight forward extension of single objective GA. In each generation, GA population is randomly divided into subpopulations, equal to the number of

objectives. Each subpopulation is assigned a fitness based on different objective functions. In this way, each objective function is used to evaluate some members in the population. It is reported that this method tends to accumulate results at extremes to the solution space, often yielding poor convergence of the Pareto front (Fourman (1985)). A more recent algorithm, based on scalarization with a weighted sum function, is proposed in Ishibuchi and Murata (1998) where the weights are chosen at random. Recently, Coello and Christiansen (1999) proposed two different methods based on aggregated functions and min-max optimization.

2.7.7.2. Pareto based approaches

The major objective of MOEA is to find a set of well-distributed solutions close to the true Pareto-optimal front. The goals in the development of MOEA are i) convergence to the true Pareto-optimal front, ii) maintenance of a well-distributed set of non-dominated solutions and iii) achieving both the above tasks with computational efficiency. To fulfill above criteria many approach has been used.

These methods use the concept of Pareto optimality explicitly. Many successful evolutionary multi-objective optimization algorithms were developed based on the two ideas suggested by Goldberg (1989): Pareto dominance and niching. Pareto dominance is used to exploit the search space in the direction of the Pareto front. Niching explores the search space along the front to keep diversity. The well-known first generation Multi-objective Evolutionary Algorithm's (MOEA) is Multiobjective Genetic Algorithm (MOGA) (Fonseca and Flemming, 1993), Niched Pareto Genetic Algorithm (NPGA) (Horn et al., 1994), Non-dominated Sorting Genetic Algorithm (NSGA) (Srinivas and Deb, 1994), Strength Pareto Evolutionary Algorithm (SPEA) (Zitzler, Laumannas and Thiele, 1999), etc. In the recent past, the first generation MOEA were modified using more effective approaches such as rMOGAxs (Purshouse and Fleming, 2001) NSGA-II (Deb et al., 2001), SPEA2 (Zitzler, Laumannas and Thiele, 2001), Generalized Regression GA (GRGA) (Tiwari and Roy 2002) and so on.

Pareto-based ranking correctly assigns all nondominated individuals the same fitness, however, this does not guarantee that the Pareto set is uniformly sampled. In order to avoid such a problem, Goldberg and Richardson (1987) proposed the additional use of fitness sharing. The main idea behind this is that individuals in a particular niche have to share the available resources. The more number of individuals located in the neighborhood of a certain individual; the more its fitness value is degraded. Detailed discussions of MOEA approaches can be found in Coello et al.(2002) and Deb (2001).
We now briefly describe below the most frequently used first generation MOEAs (NSGA, SPEA, PAES) and the second generation MOEAs (PEAS, NSGAII and SPEA2).

2.7.7.3 Nondominated Sorting Genetic Algorithm (NSGA)

This was first proposed by Srinivas and Deb (1994). In NSGA the population is first ranked using the Goldberg's (1989) Pareto ranking. As a result a large fitness is assigned to the individual in the first non-dominated front, namely the set of the non–dominated individual with rank 1 (Figure 2.13). Better non-dominated sets are emphasized systematically and NSGA progress towards the Pareto-optimal region front wise. The flow chart of NSGA algorithms is shown in the Figure 2.13. Moreover, performance sharing in parameter space allows phenotypically diverse solutions to emerge with NSGAs. NSGA includes some fundamental MOEA components, but is now surpassed by other state of the art algorithms.

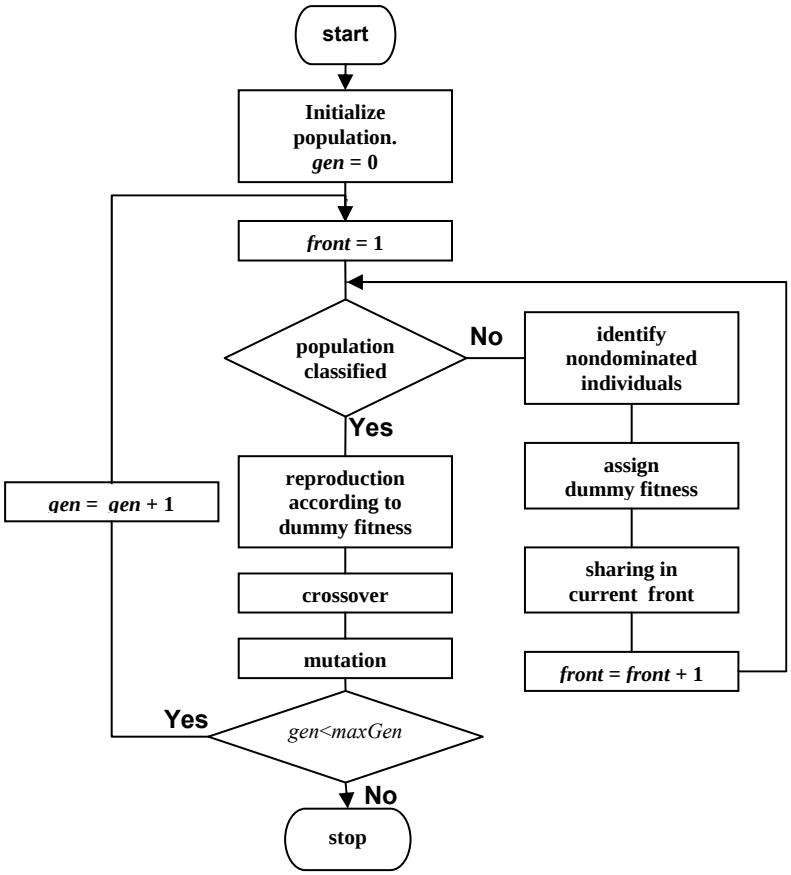


Figure 2.13 *Non-Dominated Sorting Genetic Algorithm*

2.7.7.4 Strength Pareto Evolutionary Algorithms (SPEA):

This was first introduced by Zitzler and Thiele (1999). This algorithm introduced an elite-preserving strategy by using an archive P' which contains the non-dominated solutions found previously. A clustering method (average linkage method) based on the objective space was implemented to preserve the diversity in the population. The flow chart of SPEA is shown in Figure 2.14

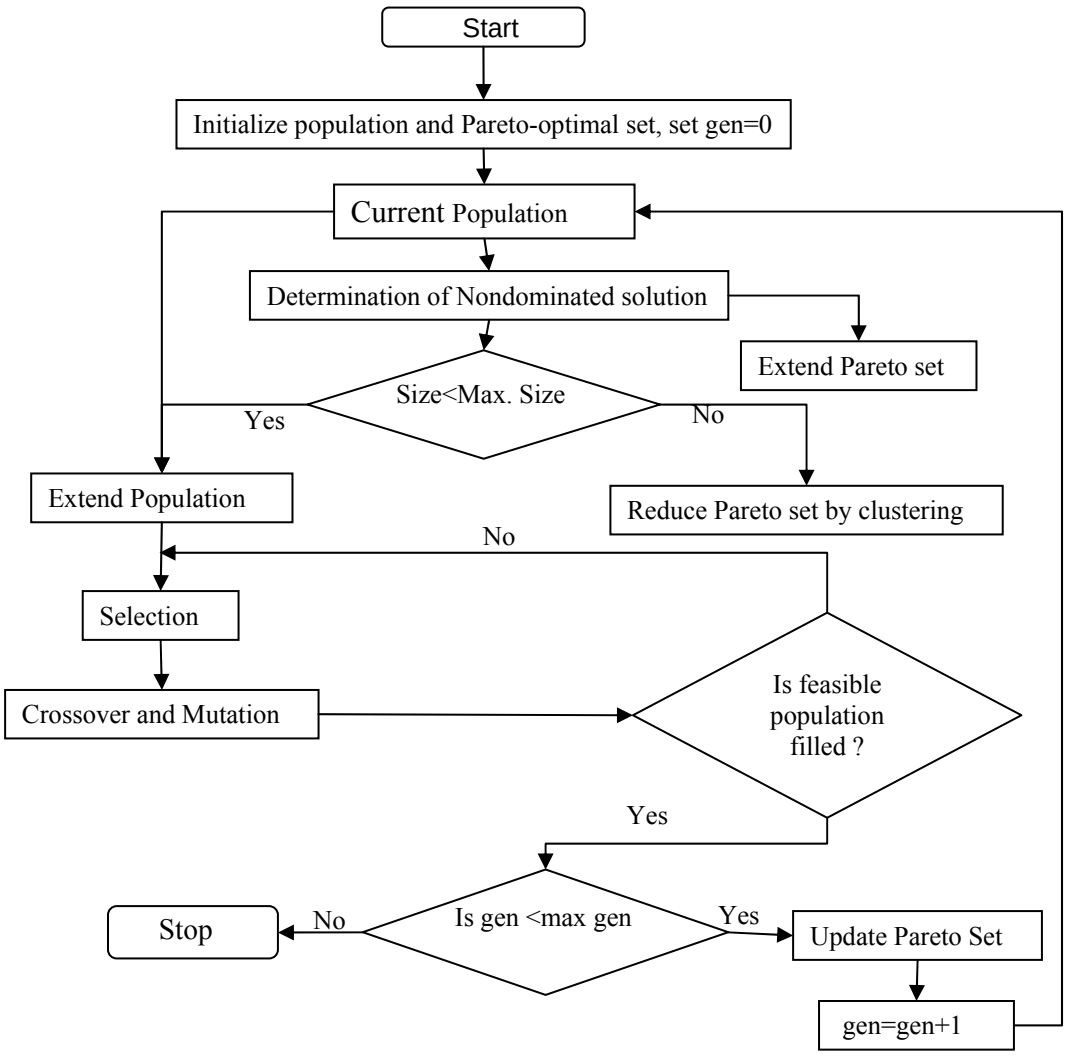


Figure 2.14: *Strength Pareto Evolutionary Algorithms*

In this algorithm, at each generation, nondominated individuals are copied to the external non dominated set (Archive). For each individual in the Archive, the strength value is proportional to the number of solution to that this individual dominates. The fitness of the member of the current population is computed according to the strength of all the Archive solutions that is dominates.

2.7.7.5 Pareto Archived Evolutionary Strategy (PAES)

This was first introduced by Knowles and Corne (2000a). The basic idea in this is shown by a flow chart (see Figure .2.15).

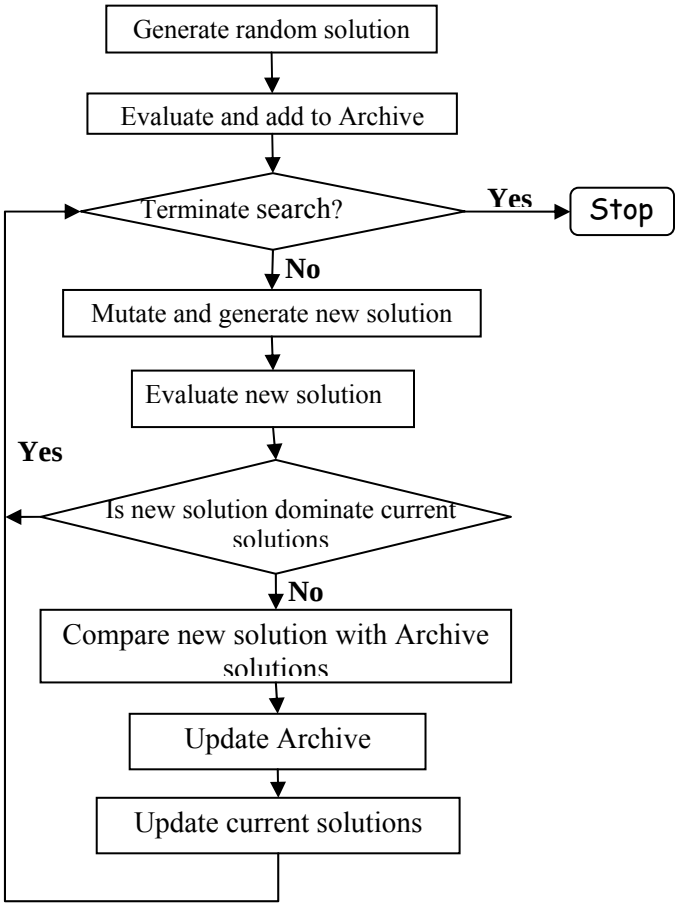


Figure 2.15: *Flow chart of Pareto Archived Evolution Strategy* (Knowles and Corne, 2000a)

This consists of a (1+1) evolution strategy (i.e., a single parent that generates a single offspring) in combination with Archive that records some of the non-dominated solutions previously found.

A new crowding method is introduced in this algorithm to promote the diversity in the population. The objective space is divided into hypercube by a grid in a recursive manner, which determines the density of individuals. Each solution is placed in certain grid location based on the values of its objective functions. A map of such grid is maintained, indicating the number of solutions that reside in each grid location. The zone with lower density is favored. Moreover the procedure has lower computation complexity

than traditional niching methods (Knowles and Corne, 2000b). They have also proposed (1+λ) and (μ+λ) variation of PAES. Nevertheless they argue that the use of a population did not in general, improve the performance but rather add extra computation cost to the algorithms (Knowles and Corne, 2000b).

2.7.7.6 Pareto Envelope-based Selection algorithms (PESA)

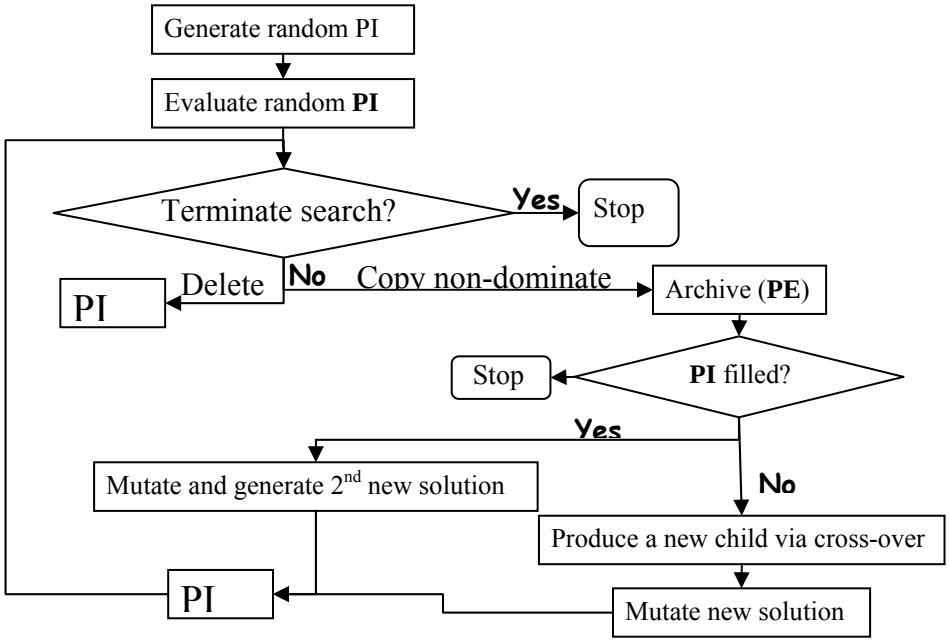


Figure 2.14: *Pareto Envelope-based Selection Algorithms.*

PESA was proposed by Corne et al. (2000). PESA not only use the same PAES hyper-grid division of objective function space to maintain the diversity, but also its selection mechanism. The same crowding measure is used to decide what solution to introduce into the Archive. This Approach uses a small internal population (PI) and a large external population (PE). A revised version of this algorithm, PESA-II is similar to PEAS except that it uses region based selection. The PESA algorithm is shown in Figure 2.14

2.7.7.7 Non-Dominated Sorting Genetic Algorithm II (NSGA-II)

The nondominated sorting genetic algorithm (NSGA-II) is a well known and extensively used algorithm based on its predecessor NSGA. It was formulated by Deb et al.(2000) as a fast and very-efficient MOEA that incorporates the features mentioned earlier, i.e. an elitist archive and a rule for adaptation assignment that takes into account both the rank and the distance of each solution regarding others.

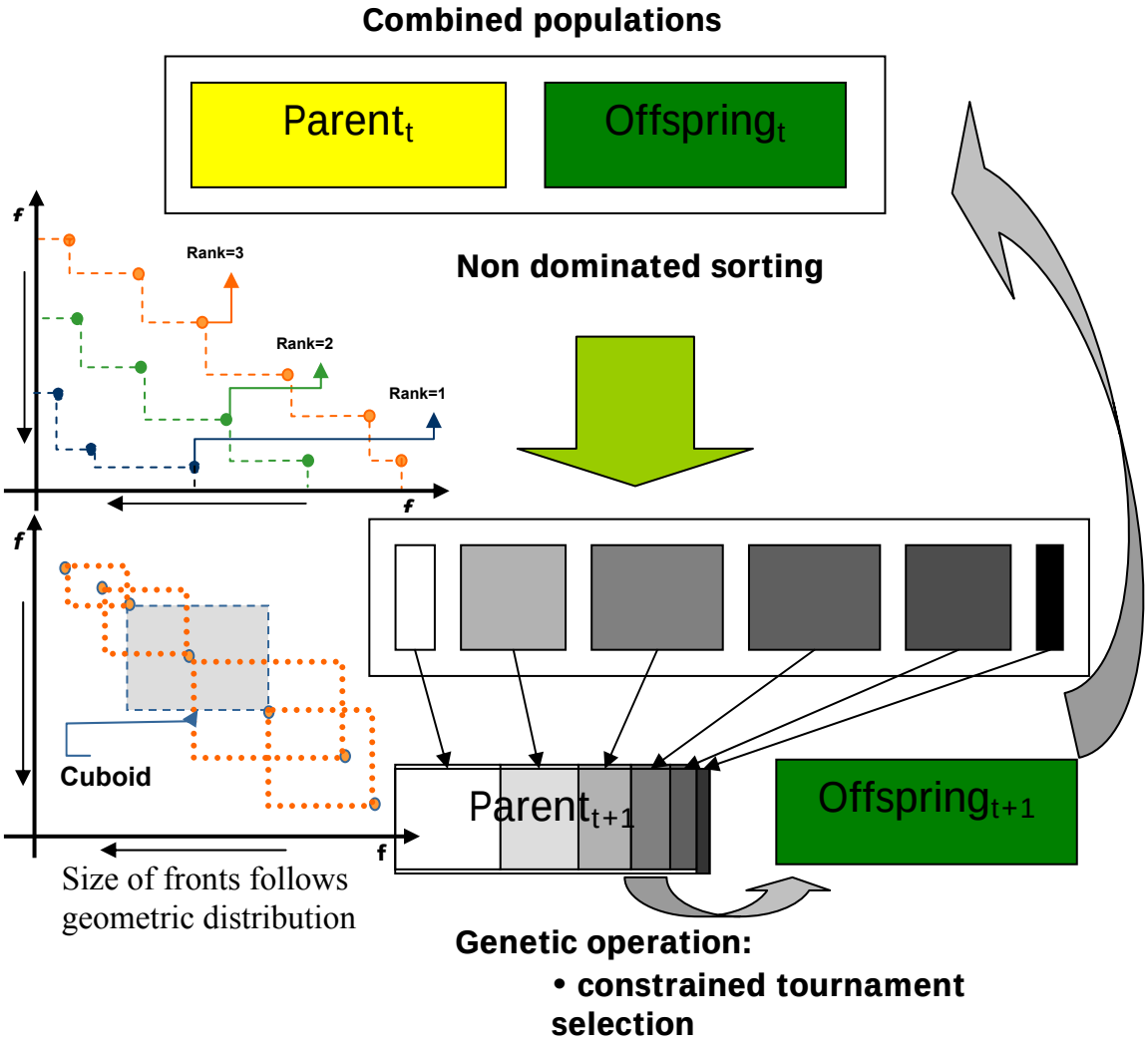


Figure 2.15: *A schematic diagram of NSGA-II.*

Figure 2.15 demonstrates the meaning of rank in a minimization case. The value of adaptation is equal to its rank. When comparing two solutions belonging to the same rank, isolated solutions prevail over non-isolated ones. If both solutions are not extreme, the one with the bigger crowding distance (i.e. the perimeter of the rectangle or "L1

norm" calculated between the two nearest neighbors) wins (see Figure 2.18). In this way isolated solutions and less crowded areas are encouraged. A schematic representation of NSGA2 is shown in Figure 2.16.

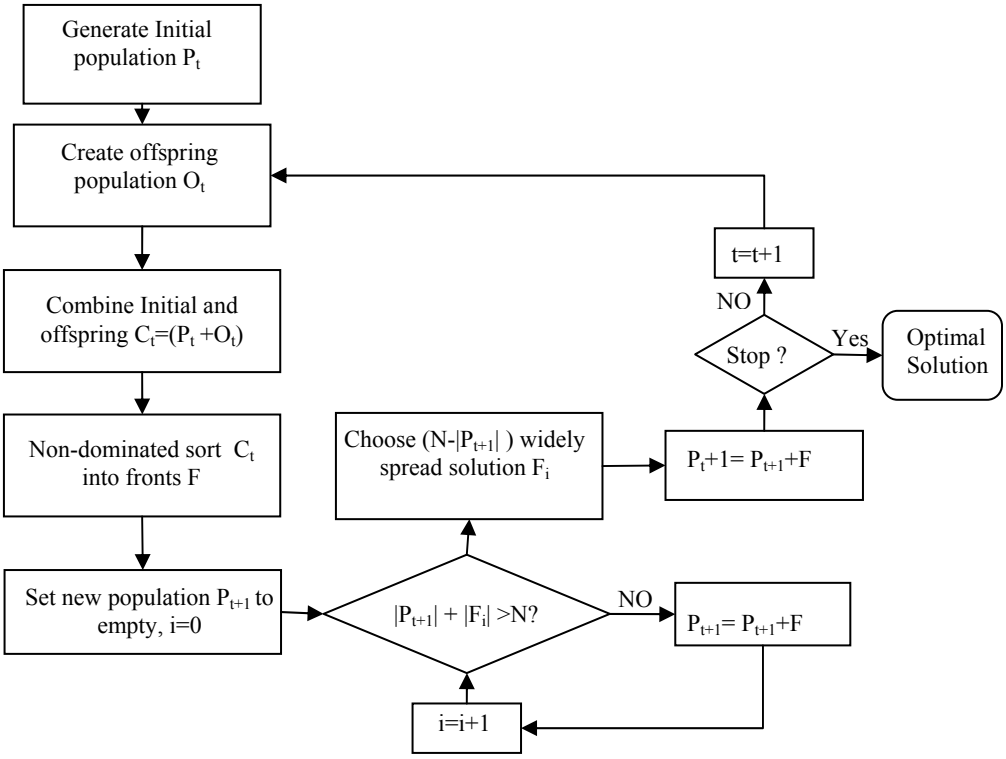


Figure 2.16: *Flowchart of NSGA-2 algorithms. It is one of the most efficient MOEA.*

In NSGA-II, the offspring population is first created by using the parent population through a crowded tournament selection, where the better individual in the parent population and the "elites" are selected so as to maintain the diversity in the population. Selected individuals will then go through cross-over and mutation operations to form an offspring population. Both offspring and parent populations are combined and sorted into non-dominated fronts. Among individuals in each front, there is no unique best solution. Each one of them performs better in some objective than the other individual, but worse in remaining objective. However, the individual in worse front are dominated by all individuals in the better front and the next generation is then filled with individuals from the sorted fronts starting from the best. If the front is only partially filled in the next generation, crowded tournament selection is invoked to ensure the diversity. This strategy

is called "niching ". Accordingly, once the next generation population has been filled, the algorithm loops back to create an offspring population from this new parent population. Figure 2.16 shows the various procedures followed in NSGA-II.

2.7.7.8 Strength Pareto Evolutionary Algorithms 2 (SPEA2):

A improved version of SPEA called SPEA2 was proposed by Zitzler et al.,(2001b). SPEA2 uses an improved fitness assignment scheme, a nearest neighborhood density estimation technique and a new archive truncation method. In SPEA2 the size of Archive is fixed. The flowchart of SPEA2 is shown in the Figure 2.17

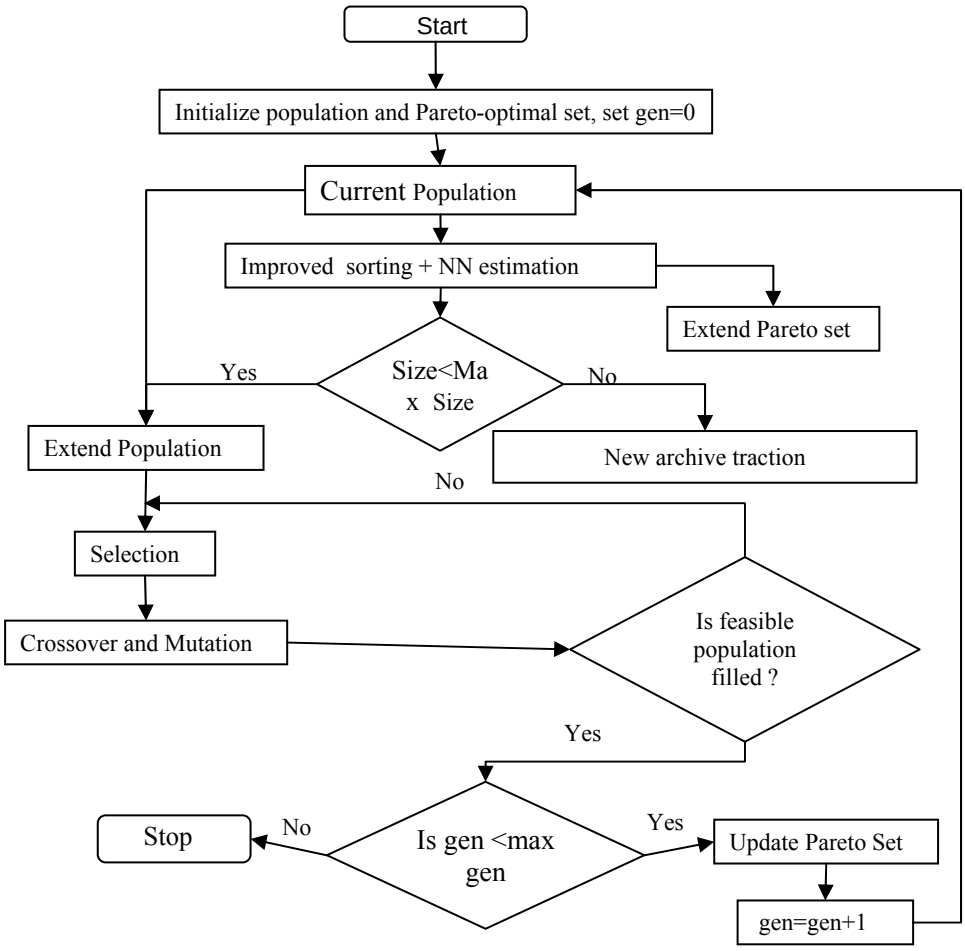


Figure 2.17: *Flow chart of SPEA2. This is improved version of SPEA. Improved fitness assignment and new archive tractions method is introduced.*

In the following section we brief describe the ε-MOEA, we used for the foothill structure and velocity optimization.

2.7.7.9 ε-Dominance Based Multi-Objective Evolutionary Algorithm

To achieve a fast convergence and well distributed solution with computational efficiency (Figure 2.19) is the major goal of all the MOEAs.

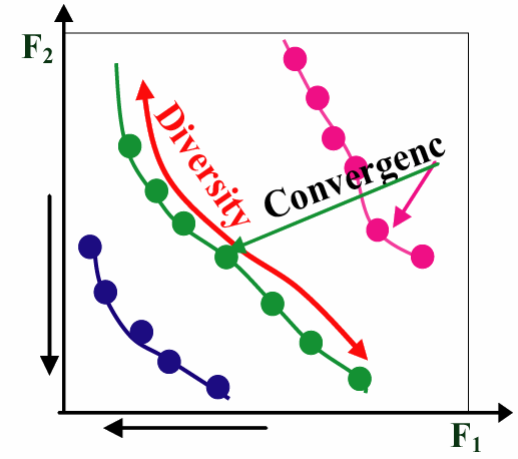


Figure 2.19: *The goal of MOEA is to achieve fast convergence toward the Pareto front and diversity along that front.*

There are many algorithms that try to achieve the above goals. Though some converge very fast towards the Pareto-optimal front they end up with a sparse distribution of the solution (i.e. PEAS). On the other hand, there are some algorithms that give dense distributions of solutions at the Pareto-optimal front, but they are computationally highly expensive (i.e. SPEA2).

To achieve the above goals in an efficient manner, a variant of MOEA, called ε-MOEA was proposed by Deb et al (2005).

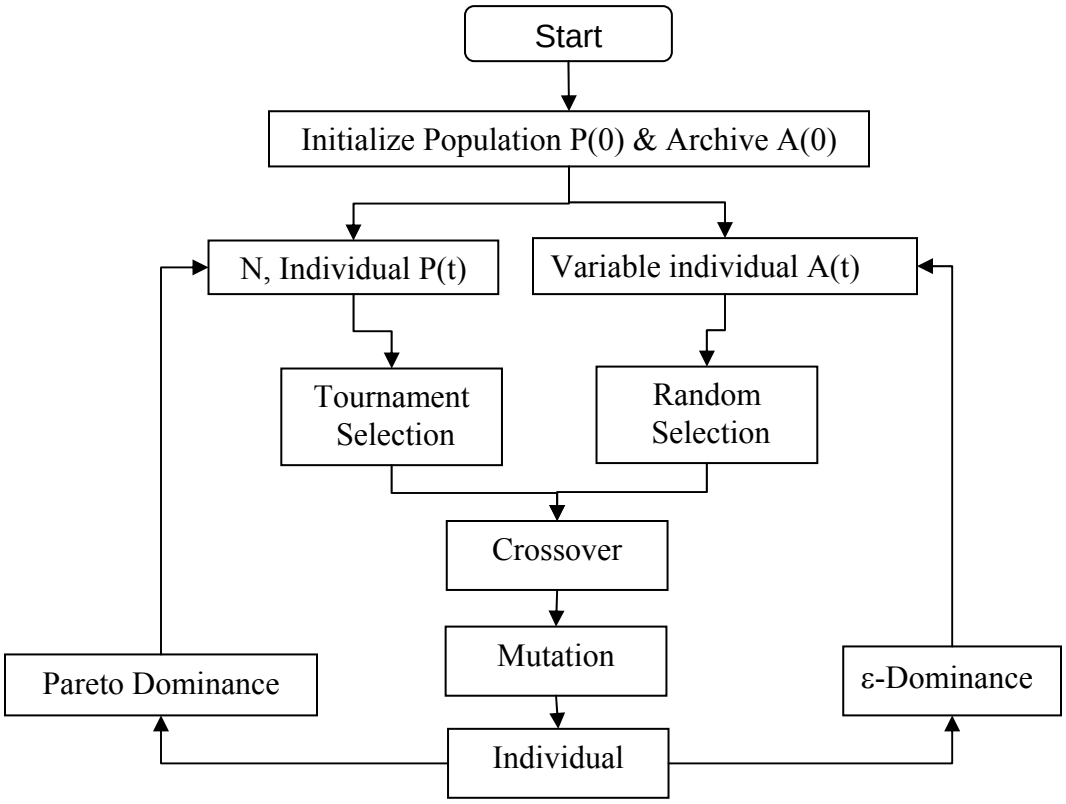


Figure 2.20: *Flow chart of ε-MOEA. It is a steady state MOEA.*

ε-MOEA uses two co-evolving populations: an EA population P(*t*) and an archive population A(*t*). The initial archive population E(*0*) is assigned from the initial population P(*0*) using ε-non-dominated solution. Thereafter two solutions, one from the P (*t*) and other from A(*t*) are selected using population and archive selection procedure and an offspring solution *o* is created. This offspring *o* can then enter into each population using population acceptance procedure or archive acceptance procedures, which are as follows:

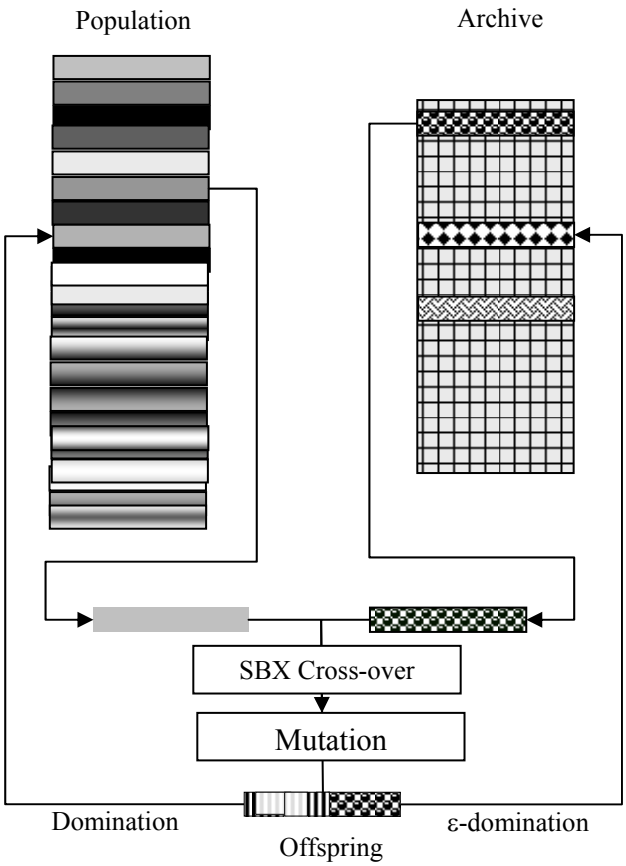


Figure 2.21: *Epsilon -MOEA flowchart. One solution from each Population and Archive are selected for generation of offspring by cross-over and mutation. After evaluating the offspring they get their place in Population and Archive only if the offspring is dominant or ε-dominant.*

Population_selection procedure

To chose a solution from population P(*t*), two population members from P(*t*) are picked up at random and dominance is checked. Of the two, the dominant one is selected, however in the absence of a dominant member, one of the two is selected randomly.

Archive_selection procedure

Random selection procedure is applied for selecting a solution e from A (time).

Population_acceptance procedure

For the population acceptance, the offspring is compared with all the population members. If the offspring dominates one or more population member, then it replaces one of them randomly. On the other hand, if the population members dominate the offspring,

the individual is not accepted. When both the above mentioned conditions fail, then the offspring replaces a randomly chosen member of the population, thereby ensuring that the EA population size remains unchanged.

Archive_acceptance procedure

For the archive acceptance, the offspring o is compared with the archive population using ε-dominance criterion (Laumanns et al.(2002)). Discretization of objective space f_i

$f_i^d = INT(|f_i| / \varepsilon_i)$ for minimizing f_i

ε_i is the allowable tolerance in the i^{th} objective, below which the values are insignificant for the user. The identification array divides the whole objective space into hyper-boxes, each having ε_i size in the i_{th} objective (see Figure 2.22).

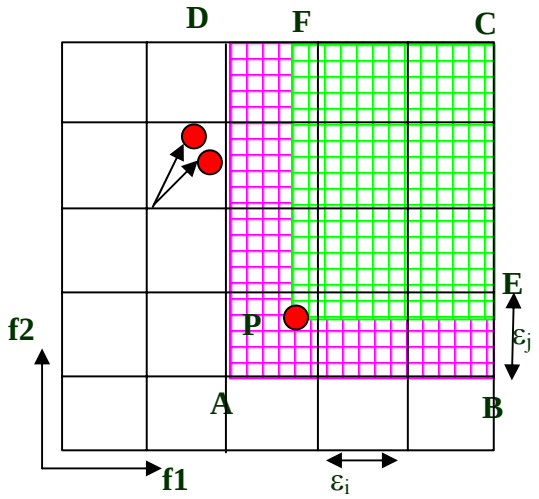


Figure 2.22: *The ε dominance acceptance procedure. Region ABCD is ε-dominant region for solution P whereas original dominance region is PECF. The ε-dominant solution nearest to the axis is selected.*

In Figure 2.22, solution P is a ε-dominant in the entire region ABCDA, where as the original domination definition allows P to dominate only the region PECFP. The identification arrays of P (Figure 2.22) are the coordinate of point A in the objective space. This discrete Pareto front held with possible ties. Ties removal by distance criterion (inside a box, small distance is preferred, Figure 2.22). This insures that each hyper-box is occupied by only one solution. Tie also removed with strict dominance. It provides two properties (1) the well distributed solution can be maintained (2) the total

number of archive solution in the final Pareto-optima will be bounded. This avoids the need to prefix the upper limit of archive size. The archive will get bounded according to the chosen ε-vector. To summarize

ε-MOEA has the following properties.

It is a steady-state MOEA.

It emphasizes on non-dominated solution.

It maintains the diversity without redundancy in the archive by allowing only one solution in each hyper-box on the Pareto-optimal front.

It is an elitist approach.

By choosing appropriate ε-vector, emphasize on the selected objective can be given.

2.7.7.10 Micro Genetic Algorithms

This was introducing by Coello Coello and Toscano Puliodo (2001). This is a GA of small population and reinitialization process. The flowchart of micro-GA is shown in Figure 2.23. The initial population is made of two subpopulation called as replaceable and non-replaceable. The non-replaceable population will never change and provide help in maintaining diversity whereas replaceable population will experience change during the each cycle of the micro-GA. In the beginning of each cycle both subpopulation take parts in generation of new population. After each micro-GA cycle the non-dominated solution copied into the extern memory (Archive). The member of external memory updates replaceable subpopulation after certain interval.

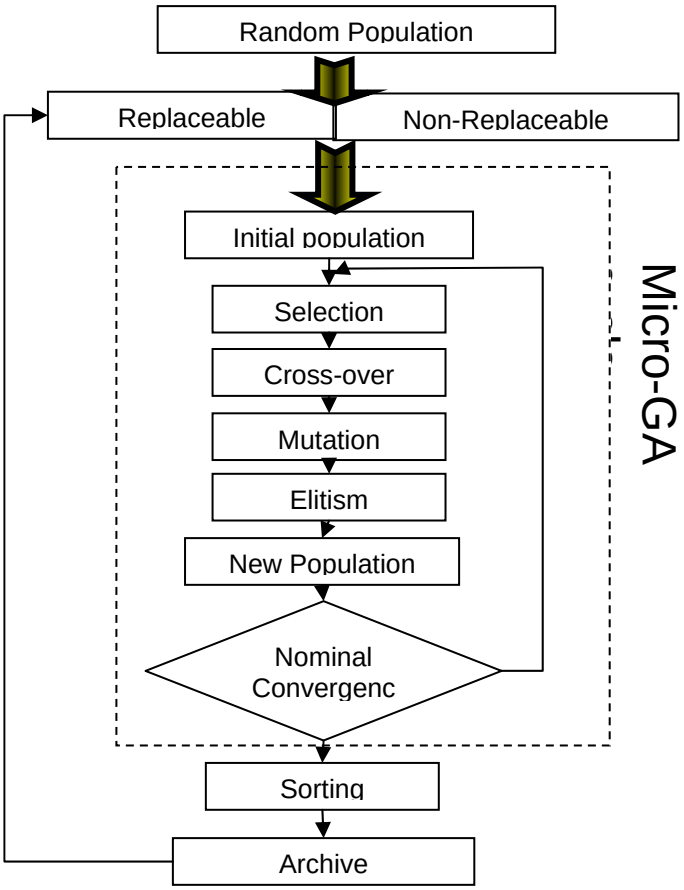


Figure 2.23: *The flowchart of Micro-GA (Coello & Pulido, 2001). Micro-GA use a small population and divide it in two sub-population, replaceable and non-replaceable during evolution.*

Conclusion

In this chapter we made an attempt to give an overview of EAs and more particularly MOEA. We also discussed the genetic operator (i.e. crossover, mutation, selection etc.). We notice that SBX crossover operator is an adaptive crossover that does not require explicit mutation (since crossover includes some implicit mutation, see Figure 2.6); hence we chosen to use it in our application. It has been shown that, ε-MOEA is one of the most efficient MOEA (Deb et al 2005). It maintains the diversity and has elitist approach. Hence it will be our choice for the foothill and velocity optimisation. Not only this, MOEA also provides a set of Pareto optimal solutions, the knowledge of which helps decision maker in choosing the best compromise. Hence we are going to use ε-MOEA with SBX crossover operator in our optimisation process.

Chapter 3

Velocity model determination methods for complex seismic data

This Chapter provides a quick overview of velocity estimation techniques generally used in geologically complex regions. First we give general overview of time domain methods and then a mix of time and depth domain methods i.e. tomographic migration velocity analysis methods and finally we discus about the depth domain methods and parametric curves. of common image gathers. We discuss state of the art automatic seismic velocity analysis techniques using Wave Equation Migration Velocity Alalysis (WEMVA). Finally we give flavor of migration algorithms.

3.1 Introduction

Seismic velocity model estimation for complex seismic data (complex surface and sub-surface structure, irregular data coverage, large lateral velocity variation) is one of the most challenging tasks. Due to the non-linear relationship between seismic data and velocity model, it is usually posed as an optimisation problem. For this optimisation usually iterative methods have been used. Depending on how the objective function is constructed for velocity optimisation, it can be divided into two groups: data domain methods and image domain methods. In the data domain methods error is mostly measured on the time scale, whereas in the image domain methods error is measured on depth scale. Hence it can also be divided as a time domain methods and depth domain methods.

3.2 Time domain methods

In data domain methods, objective functions are formulated to measure the deviation of predicted data from the observed data. Measured deviation is mostly the measurement of time difference between observed and predicted data. Goal of optimisation problem is to minimize the data residual.

$$\left\| d^{obs}, d^{pre}(p) \right\|_d^2 \quad (3.1)$$

where $\| \ \|_d$ indicates an appropriate norm of data domain and $d^{pre}(p)$ denote the predicted data from parameter vector p.

Early *tomography methods* and *Full waveform inversion technique* are examples of data domain methods.

3.2.1 Tomography methods

Early velocity estimation techniques attempted to find the velocity models that best fitted the data. When the observed data is simplified and restricted to the arrival times, and interval velocity is estimated by fitting model travel time to measured traveltime using an inversion procedure, the method can be categorized as reflection tomography method. Traveltime reflection tomographic inversion (Bishop et al. 1985) can resolve the velocity model determination problems. It assumes that the travel times of reflection are readily available. This is not true in practice, and actually obtaining accurate and robust measures of reflected traveltime is one of the main challenges of reflection tomography.

Though picking of the individual traveltime directly from the data is conceptually simple, and it has been used in earliest tomography methods (Bishop et al. 1985; Stroke and Clayton, 1991). However access of kinematic information (travel times) associated with reflection events can be anywhere from difficult to impossible for complex seismic data. On the other hand picking 4D reflection can be easily done in depth domain (depth domain more interpretable) (Lailly and Ehinger, 1991). Current industrial approach mostly realizes on depth domain.

3.2.2 Full waveform inversion methods

Seismic waveform inversion is potentially a very powerful tool to extract all possible information (compressional wave velocity, Poisson's ratio and impedance contrasts) from seismic data, by minimizing the differences between observed and synthetic seismic data. However optimisation is quite difficult even in case of a simple structure and to the best of our knowledge no effective optimisation method (gradient or global methods) is available. The gradient based waveform inversion tries to fit the observed data by solving the wave equation. The data residual is formulated by linearzing the wave equation with respect to velocity, which is limited by the first-order Born linearization. If the phase differences between the modeled and recorded wavefields are larger than a fraction of the wavelet, then the assumption made under the Born linearization is violated and velocity inversion methods diverge (Woodward, 1992; Pratt, 1999; Dahlen et al., 2000). Consequently, calculation using a gradient method requires a good guess of initial velocity model, which is one of the severe obstacles in geologically complex regions. The gradient calculation is formulated as a adjoint of Born modeling. The Physical meaning of adjoint of Born modeling is understood as a projection of data residual back to the model space through downward continuation of source and residual wavefields.

To avoid the Born linearization, a decoupling approach is used where low and high wavenumber information, inverting for each parameter set alternately (Hicks and Pratt, 2001). For low frequency, such projection produces reasonable search direction for updating velocity. For high frequency, the back projected data residual reconstructs the image asymptotically, and thus is equivalent to reverse time prestack depth migration. Hence the increasing frequency scheme where data correspondence to the low frequency is first fitted and then gradually increased to higher frequency (Pratt 2001). Time domain implementations of such methods have been presented by Tarantola 1997, Tarantola & Vallette, 1982.

The wave form inversion approach is limited by the lack of good guesses of initial velocity model and absence of low frequency content in seismic data. Global optimisation methods are not constrained by the local linearization of the wave equation. Thus the local minimum of the objective function may be avoided by performing a global

search on the objective function (Sen and Stoffa, 1991). Global optimisation methods carry out a systematic exploration of the multidimensional search space using for example, Monte-Carlo, genetic algorithms or simulated annealing algorithms. Although these techniques are capable of handling non-linear behavior by inverting, for both low and high wavenumbers, they require number of computation of the same order of forward modeling as there are model parameters involved. Hence high computation costs prohibit the use of large data set with complex velocity models, so global methods can not be used.

3.3. Depth domain methods

In geologically complex regions due to strong alter as well as vertical velocity changes, the reflections are not only hyperbolic, but may be completely uninterpretable and thus no continuous event can be interpreted in time domain seismograms. To overcome these difficulties, it is interesting to use a method based on a combination with some depth domain based access to kinematic information, i.e. interpretation in the prestack depth migrated domain instead of prestack time domain. In the depth domain, events are more interpretable than in the time domain, even migration is performed with only an approximate velocity. This is because migration removes the propagation effects from the data and thus enhances the coherency of events. Hence recent trend of velocity analysis is more toward depth migrated image domain methods, where linearity to the data is expressed by the depth migrated image.

4D reflection data picking can be easily done in depth domain and also CIG after migration can be described by a family of curves (Section 3.4). Therefore a 4D interpretation reduces to 3D interpretation (zero offset or shot migrated cube + 1 parameter for each (x,y,z) describing curve). As a result tomographic migration, velocity analysis approaches are mow routinely used in oil industry whereas differential semblance based approaches nonetheless need some more research to get mature.

3.3.1 Tomography migration velocity methods

Access of kinematic information (travel times) in complex seismic data is difficult. Therefore new industrial trend of tomography is shifting from data domain to image domain for the measurement of kinematic information. New tomography methods

incorporate migration to access more accurate and robust kinematic information. In these methods, traveltime error is measured on the migrated image and not from the data. When either the reflector geometry is complex or the wavefronts are distorted by a strong lateral velocity variation, the analysis of kinematics is more robust in the image domain after the migration than in the data domain methods before migration. The CIGs obtained after the migration is more robust and accurate in providing the kinematic error, even if the migration velocity is not a good approximation of the true velocity.

Migration velocity analysis is an iterative process of interval velocity estimation. It consists of the following steps (Figure 3.1). (1) Data is migrated with the current best estimate of interval velocity. (2) The Prestack image is analyzed for kinematics errors on CIGs. (3) The measured kinematics errors are inverted into the interval velocity update. Different algorithms differ in how to update the velocity on the basis of measurement of residual curvature (or flatness of gathers) or image difference.

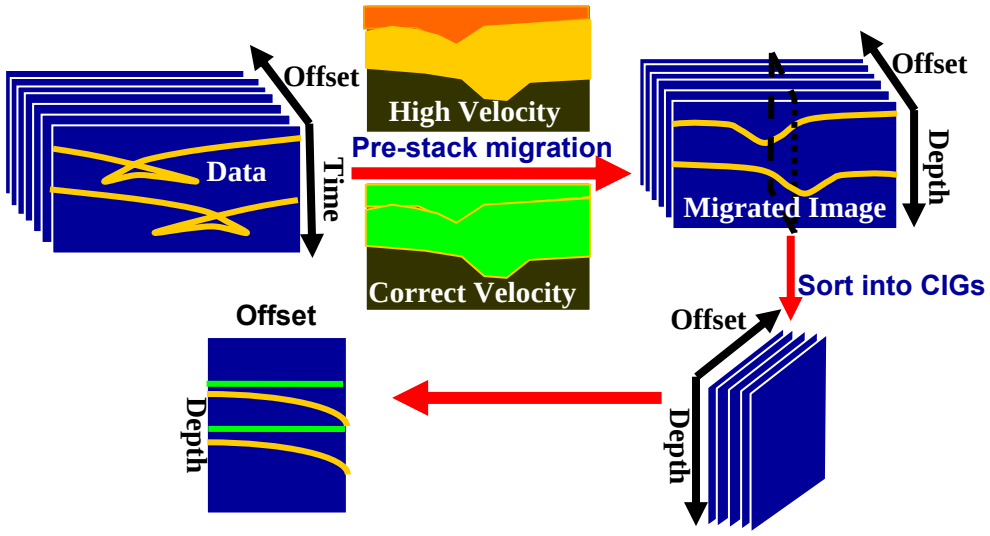


Figure 3.1: *Illustration of migration velocity estimation process. (1)Data is migrated with the current best interval velocity model (2) the velocity errors are measured on the CIGs obtained after migration. (3) The measured kinematics errors are used for velocity update.*

For a correct velocity model the gather should appear flat. For example Marmousi data migrated with correct velocity model produces flat events on coherency panels (or gathers) (Figure (3.2)).

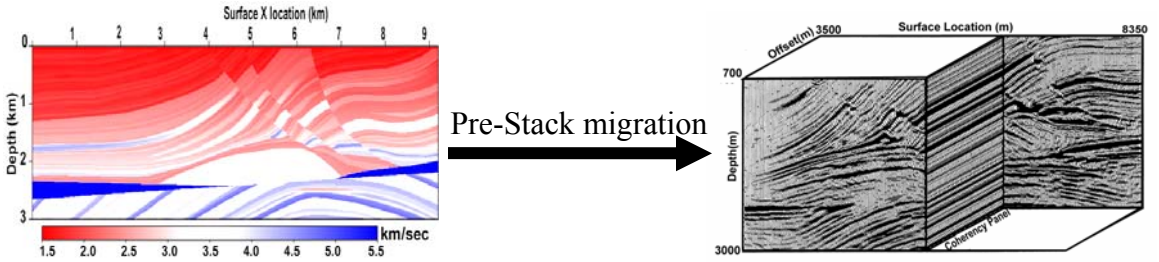


Figure 3.2:*Pre-stack migration of Marmousi data with correct velocity model. On the coherency panel, event they are appearing flat.*

In the tomographic migration velocity, the aim is to flatten the migrated CIGs, or in other words, to minimizing the RMO function measured from the CIGs, $\bar{n}_{RMO}$. Hence one needs to minimize following the objective function:

$$J_{tomo}(v) = \left\| \overline{n_{RMO}(v)} \right\|^2 + . \quad (3.2)$$

Different authors also include some regularization term, constrain handling technique. RMO function can be directly picked from migrated CIGs for each value of reflection angle(s) or offset(s). However a more robust and automatic approach to measure RMO function is to use RMO curve or residual- migration analysis described in the next section (Section 3.4). Automatic RMO estimation technique is described in chapter 5. Differential semblance optimisation (DSO) is another viable approach for flattening the RMO function.

Still tomographic migration velocity methods require picking, which is quite heavy and interpretative. Hence some attempts have been made for automatic velocity inversion in the image domain.

3.3.2 Image domain methods

Here the objective function measures the quality of the reconstructed image. The goal of optimization is to maximize the quality of image i.e. to accurately focus the energy. In other words it amounts to minimizing the unfocused energy.

$$\min_c \left\| O_I(I) \right\| \qquad (3.3)$$

where $\|..\|$ indicates the proper norm of image domain and O_i provide the required measurement of the image domain. A proper semblance measurement criterion can make image domain methods less vulnerable to local minima during velocity inversion.

In the presence of complex wave propagation, ray based migration are often not capable of producing high quality images, while wavefield continuation methods yield better results. Recently some attempts have been made to optimize the velocity using wave equation migration velocity analysis (WEMVA) (Shen et al., 2003; Sava and Biondi, 2004). Sava and Biondi (2004) WEMVA is based on the linear relationship between the image perturbation and velocity perturbation which is a linearized version of DSO, whereas Shen et al., (2004) use the concept of optimal focusing (a non-linear version of DSO). Both approaches have used gradient optimisation technique. In the next section we will summarize the gradient computation technique for both methods.

3.3.2.1 Image perturbation criteria

This method tries to maximize the quality of the migrated image. It is also similar to the differential semblance optimisation (Symes and Carazzone, 1991). It is based on the general principle that a change in the velocity model will result in a change in the image. Suppose that we are given a downward-continued migrated image $I_c(x,z,h)$ which is the function of space (x, z) and subsurface offset h, and assume that the true background velocity is c. Due to the background velocity change δc *results in* the new migrated image is I. Hence in the linear approximation, the perturbation in images $\delta I(x)$ *i*s related to the velocity perturbation by

$$I_c(x) - I(x) = \delta I(x) = \frac{\partial I}{\partial c}\delta c \qquad (3.4)$$

$$I_c(x) = I(x) + \frac{\partial I}{\partial c}\delta c \equiv I(x) + L\delta c \qquad (3.5)$$

where $L = \frac{\partial I}{\partial c}$

The goal of wave equation velocity analysis is to solve this equation for δc, given a suitable image perturbation $\delta I(x)$. A linearized inverse problem based on the first order perturbation of image can be framed as

$$\min_{c} \left\| \delta I - L\delta c \right\| \quad (3.6)$$

Here the major difficulty is to obtain the image difference δI because the true velocity (c) is unknown. The image perturbation δI is estimated using prestack residual migration technique as

$$\delta I \approx \left. \frac{dI}{d\rho} \right|_{\rho=1} \delta\rho \quad (3.7)$$

Here ρ is the ratio of velocity to which the background velocity is migrated. Using the chain rule of differentiation, one can write

$$\delta I \approx \left. \frac{dI}{dk_z} \frac{dk_z}{d\rho} \right|_{\rho=1} \delta\rho \quad (3.8)$$

This image perturbation consists of three parts, first is the derivative of the image with respect to depth of the wavenumber ($\frac{dI}{dk_z}$) and two weighting factors, derivative of the depth wavenumber with respect to velocity ratio ($\frac{dk_z}{d\rho}$) and the magnitude $\delta\rho$ *of the* perturbation from the reference to the improved image.

The image derivative in Fourier domain $\frac{dI}{dk_z}$ is straight forward to compute at ρ=1,

$$\left. \frac{dI}{dk_z} \right|_{\rho=1} = -izI \quad (3.9)$$

This derivative of the image is represented by the imaginary part of migrated image scaled by depth.

Secondly, the image derivative of depth wavenumber $\frac{dk_z}{d\rho}$ is formulated through the Double-Square-Root equation (Sava, 2003).

$$\left. \frac{\partial k_z}{\partial \rho} \right|_{\rho=1} = \frac{\mu}{2\sqrt{\mu^2 - |k_s|}} + \frac{\mu^2}{2\sqrt{\mu^2 - |k_r|}} \quad (3.10)$$

Where

$$\mu^2 = \frac{\left[4k_{z0}^2 + \left(|k_r| - |k_s|\right)^2\right]\left[4k_{z0}^2 + \left(|k_r| + |k_s|\right)^2\right]}{16k_{z0}^2} \quad (3.11)$$

k_r, k_s, *and* k_{z0} are spatial wavenumbers for the sources, receivers and vertical component corresponding to the current background velocity respectively.

Finally, an optimum velocity ratio ρ^* is picked, according to the semblance or differential semblance criteria through the repeated residual migration, which is suitable only for constant velocity ratio. The perturbation of velocity ratio is estimated by

$$\delta\rho \approx \rho^* - 1 \qquad (3.12)$$

After estimating all the three parameters, the image perturbation can be evaluated by combining equations*(3.9)*, (3.10) and*(3.12)*. Once the image perturbation δI is calculated, the velocity perturbation δc can be solved from the linear least square problem *(equation*(3.6)*)*. Since the nonlinear objective function is not clearly defined, it is hard to analyze the convergence. However this image perturbation approach maintains the singularity of the image whereas the residual image tends to remove them.

3.3.2.2 Differential Semblance Optimisation

The basic principle of DSO is similar to that of conventional MVA; i.e. it relies on measurement of horizontality of gathers. However, instead of minimizing the RMO of the image at each angle (or offset) with respect to the normal-incidence (zero-offset) image, the DSO method minimize the relative difference in image depth between neighboring angle (or offset). Differential semblance (DS) function (Symes and Carazzone, 1991; Symes, 1994; Chauris and Noble, 1998; Plessix et al. 2000; Pratt and Symes 2002; Mulder and Ten Kroode, 2002) has been widely used for velocity estimation. The DS function has also been used for the optimization of waveform tomography (Symes, 1994, Plessix et al. 2000,Pratt and Symes 2002).

Recently Shen at al. (2003; 2004) proposed WEMVA method based on Claerbout's principle of survey sinking. According to Claerbout's principle, the correct velocity model is the one in which all the energy is optimally focused at zero subsurface offset and zero time. If the velocity model is not correct, then migration will not focus the energy. The failure of migration to focus the image in subsurface offset is an index of velocity error. This index is the remaining wavefield after annihilating the focused energy

from the migration. The remaining wavefields can be measured on the migrated gathers. Shen at al. (2004) proposed the following relation to measure the unfocused energy

$$DS = \frac{1}{2}\|hI\|^2 \quad (3.13)$$

This is a mean square of the image volume scaled by the subsurface offset (h). For detail discussion about this formulation please see Shen at al. (2004). In angle domain it is equivalent to measuring the flatness of the migrated gather (equation (3.14)).

$$DS_\theta = \frac{1}{2}\left\|\frac{\partial}{\partial\theta}\Re I\right\|^2 \quad (3.14)$$

Where $\Re$ is Radon transform from offset to angle θ (Sava and Fomel, 2003) and DS_θ is minimum, when velocity is correct. Detailed analysis of DS properties is presented in chapter 5. For the sake of completeness here we are presenting the main components of this approach. For detail discussion please see Shen et al. (2004).

The measurements of the residual wavefield on the gather do not explicitly involve the correct image. Hence Shen et al. (2004) solved the slowness perturbation directly in the adjoint sense. The gradient at depth k could be presented by

$$\left(\nabla_c J\right) = \sum_{s,w}\left(\frac{\partial H}{\partial c^k}S^k\right)^* S^k_{adj} + \left(\frac{\partial H}{\partial c^k}R^k\right)^* R^k_{adj} \quad (3.15)$$

Where $\forall k \in [2, N_z]$, the adjoint field S_{adj} (equation (3.16)) and R_{adj} (equation (3.17)) can be written as

$$S^{k-1}_{adj}(x,s,\omega) = H\left(c^k\right) * S^k_{adj}(x,s,\omega) + \sum_h h^2\tau(h)I^k(x,h)R^k(x+h,s,\omega) \quad (3.16)$$

and

$$S^{k-1}_{adj}(x,s,\omega) = H\left(c^k\right) * S^k_{adj}(x,s,\omega) + \sum_h h^2\tau(-h)I^k(x,h)R^k(x+h,s,\omega) \quad (3.17)$$

Here $H\left(c^k\right)$ is the operator that allows to propagate wavefield from depth $k\Delta z$ to $(k+1)\Delta z$ with a velocity c^k.

$$S^{k+1}(x,s,\omega) = H\left(c^k\right)S^k(x,s,\omega) \quad (3.18)$$

$$R^{k+1}(x,s,\omega) = H\left(c^k\right)R^k(x,s,\omega) \quad (3.19)$$

where S stands for the source and R stands for the receiver. I^k is the image at depth k. The detail derivation of the equation is presented in the appendix (A).

Like other gradient methods, this method is also limited by the Born approximation, as a result it requires a good guess of initial velocity model and a very fine migration parameter sampling to calculate gradient. Computation of gradient is expensive and equivalent to four migrations per iteration. These methods are well suited for the final velocity refinement once a good guess of initial velocity is obtained by other methods.

Flatness criteria can also be measured by semblance function. Properties of semblance function are discussed in the next section.

3.3.3 Semblance function

Ssemblance function has often been used for velocity optimization (Jin and Madariaga, 1994; Docherty et al., 1997; Mansanne et al. 2001) using global methods. It is based on the flatness on the coherency panels. A measure of flatness of events in the semblance panel is given by the sum of the total energy of the traces. Semblance is measured on angle gather and can be presented as:

$$S = \int_x dx \frac{\left[\int_z dz \left(\int_\theta I d\theta \right)^2 \right]}{\left[\int_z dz \int_\theta I^2 d\theta \right]} \qquad (3.20)$$

Where I is the migrated image, z is the depth and θ is the reflection angle. This semblance relation can be obtained in offset domain by replacing angle parameter θ by offset parameter h. The response of semblance function for a flat reflector is shown in the Figure (3.3).

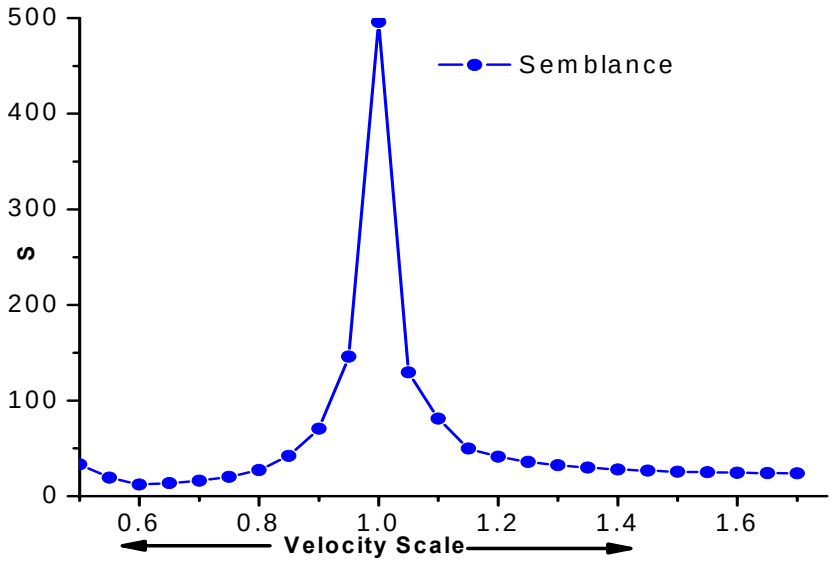


Figure 3.3: *The response of semblance function (equation (3.20)) of a flat reflector embedded in a homogeneous medium for different velocity scaling factor. For a small velocity error the semblance measurement is very sensitive to the velocity error factor, especially near to the true velocity field (velocity factor=1).*

One of the most attractive features of semblance function is its strong sensitivity near the true velocity. For a large velocity error CIGs are highly non-flat, different events may interfere and local minima can appear in the semblance objective function. Therefore semblance may be unable to make suitable adjustment to the velocity (Claerbout, 1985). Semblance and differential semblance measurement tell us about the flatness of the gather; however, they do not provide any direct information about the amount of error in each location. Whereas RMO function gives the approximate amount of error at each location. A vast majority of MVA methods and tomographic methods are based on the measurement of RMO in CIGs. This RMO function can be directly picked from the migrated CIGs, However a more robust way for measuring this RMO function is to use either the RMO or residual-migration analysis. Measurement of RMO provides a way to get rid of picking and helps in automation. In the next section we are presenting parametric curve for RMO estimation, from the gathers obtained by shot profile wave equation migration.

3.4. Kinematics of the image in offset and angle domain

The migration velocity analysis methods use the kinematics information contained in the prestack partial image to update the velocity. The vast majority of MVA methods are based on the measurement of Residual MoveOuts (RMO) in CIGs. RMO analysis of migrated CIGs is equivalent to the stacking velocity analysis of common mid point (CMP) gathers in the data domain. First we will analyze the kinematics of the image in offset domain.

3.4.1. Kinematics of the local–offset gathers for horizontal reflectors

Let a reflector be located at depth **z**. This reflector (x, Z) can be associated with the pair of source (S) and receiver (R) at the surface in coordinates (x-H, 0) and (x+H, 0) respectively. These S and R are separated by surface offset 2H with respect to the mid-point (x, 0) (Figure 3.4). Hence timetime taken to travel from S to R via the Reflector (x, Z) is

$$T_{H,Z} = \frac{2\sqrt{H^2 + Z^2}}{c} \qquad (3.21)$$

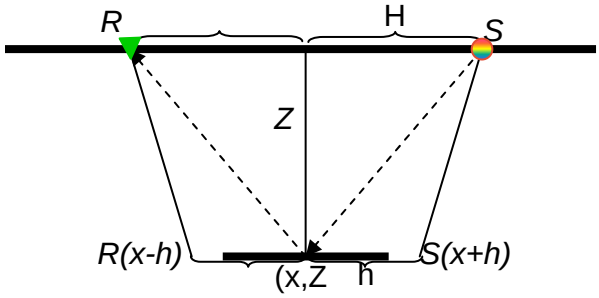


Figure 3.4: *Kinematics of the image in offset domain. Image at the point (x,Z) is obtained by correlation of* $S(x+h,S,\omega)$ *and* $R(x-h,R,\omega)$ *at each point x in the model. The point of correlation x is not necessary on the surface.*

For c the constant true velocity, the migration travel time in offset is denoted by sub-surface co-ordinates [x, h] and is expressed as the sum of the migration traveltime from the source (x+H,0) to a sub-surface point (x+h, z) and receiver (x-H,0) to a subsurface point (x-h, z) ,

$$T_{H,z} = \frac{2\sqrt{(H-h)^2 + z^2}}{v} \qquad (3.22)$$

where v is the constant migration velocity. The constructive interference occur at sub-surface coordinate [*h,z*] only when $T_{H,Z} = T_{H,z.}$ from the above equations we obtain

$$(h-H)^2 + z^2 = \gamma^2\left(H^2 + Z^2\right) \qquad (3.23)$$

$$z^2 = \gamma^2\left(H^2 + Z^2\right) - (h-H)^2 \qquad (3.24)$$

where $\gamma = v/c$ is the ratio of the migration velocity and the true velocity.

An image is formed when depth z is stationary with respect to the change in surface offset H. Which means *dz/dH*=0,

$$0 = 2\gamma^2 H + 2(h-H) \qquad (3.25)$$

i.e. H=h/(1 - γ^2), This equation gives the relation between surface offset (H) and subsurface offset (h). Replacing the surface offset H by subsurface offset (equation*(3.24)*) we derive

$$\frac{h^2}{(\gamma^2 - 1)Z^2} + \frac{z^2}{\gamma^2 Z^2} = 1 \qquad (3.26)$$

This equation (3.26) is characterized by the kinematics of the common image in offset for some fixed mid-point x. When the migration velocity *v* is less than the true velocity *c,* we have $\gamma \prec 1$, and equation (3.26) becomes a hyperbola. The moveout starts at an undermigrated depth z <Z and is curved toward the smaller depth. When the migration velocity *v* is greater than the true velocity *c,* we have $\gamma \succ 1$, and equation (3.26) became an ellipse. Here moveout starts at an overmigrated depth z>Z and is curved toward the larger depth. At $\gamma = 1$, migration velocity equals to the true velocity, all the energy will fall at depth point (Z, 0). Thus, the curvature of the moveout curve on wave equation offset gathers indicates velocity errors, which is the desired behavior for velocity analysis application.

3.4.2. Kinematics of the Angle gathers for horizontal reflectors

The angle domain relation can be directly obtained from the offset domain relation (3.26) by replacing the offset parameter (*h)* to the apparent angle parameter.

The apparent angle tanφ= *-h/Z*

$$z^2 = \gamma^2 Z^2 - \frac{\gamma^2 h^2}{(\gamma^2 - 1)} \qquad (3.27)$$

$$z^2 = \gamma^2 Z^2 + \frac{\gamma^2 Z^2 \tan\varphi^2}{(\gamma^2 - 1)} \qquad (3.28)$$

The linearization version of RMO function for a flat reflector is (Biondi and Symmes , 2004)

$$Z(\gamma - 1)\tan^2\varphi = RMO \qquad (3.29)$$

and for inclined reflector (geological dip α) is

$$RMO = \frac{(\gamma - 1)}{\cos\alpha} \frac{\sin^2\varphi}{(\cos^2\alpha - \sin^2\varphi)} Z\,. \qquad (3.30)$$

Similar kinematics relation has also been derived for 3D Angle domain CIG (Biondi and Tisserant 2006). In chapter 5, an automatic approach of RMO extraction from the gather will be presented. These gathers are produced by either Kirchhoff migration (ray tracing) or wave field extrapolation. In MVA tomography, generally ray tracing based approach is used whereas wavefield extrapolation methods are mostly suited for geologically complex region. Both migration approaches are routinely used in the velocity estimation process. Here we are reviewing them for sake of completeness.

3.5 Migration

When either the reflector geometry is complex or the wavefronts are distorted by a strong lateral velocity variation, the analysis of kinematics is more robust in the image domain after the migration than the data domain methods before migration. Migration (Figure 3.1) can be defined as a "map" from data space to image space and the image is a reflectivity picture.

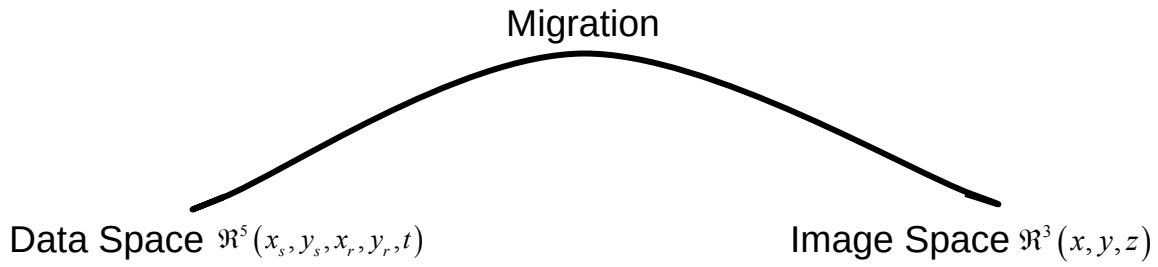


Figure 3.5: *A schematic diagram showing transformation from data space to image space. Here migration maps data space to image space.*

Migration algorithms can be divided into two main classes: Kirchhoff methods and downward-continuation methods. A very short description of both methods and their compressive advantages and disadvantages is given below.

3.5.1 Kirchhoff migration

Kirchhoff migration is based on the integral solution of the wave equation and has a kinematics and amplitude part to the solution. To perform a Kirchhoff migration (Figure 3.6) one first needs to compute travel time through a velocity depth model. Then to each travel time of the computed travel time table, a sample of the input data is matched. The amplitude of this sample is then scattered along the migration impulse response corresponding to a wave front or a location of an equal travel time. Amplitude decay is then considered along the impulse response curve. Impulse responses are then stacked to recreate a seismic reflector. Here, the Green's function is computed by ray tracing: it consists of kinematics and amplitudes parts. The kinematics parts define the traveltime of a diffraction shape, while the amplitude part provides amplitude weighting that is applied along the diffraction. Modern inversion techniques (Bleistein 2001) also result with the same Kirchhoff type solution but require “true amplitude” type processing and may apply different amplitude weightings along the diffraction.

Kirchhoff methods are an asymptotic approximation of Green's function and typically use a single arrival (usually the most energetic). It has proven to be a robust migration method because it is computationally efficient and can handle irregular acquisition geometry (Bancroft, 2004). Further, it can be target oriented, that is image at a specific depth level or levels can be obtained without requiring to imaging the entire volume. In a

high-fidelity Kirchhoff implementation, honors correctly the extremely large apertures (e.g. 12 to 15 km radius) and image overturn dips. Its reliance, however, on ray-traced traveltime means that it is sensitive to ray behaviors, which, in complex geology, can be erratic and can result in migration noise in poorly illuminated areas. Strong lateral velocity variations, however induce multipathing i.e. appearance of multiple raypaths connecting the source and the receivers location with the image point (Stolk and Symes 2004). To overcome these difficulties a lot of attentions has been given to wave field extrapolation (or downward continuation) migration – that implements differential solution of one-way wave equation performing the downward extrapolation of the wavefield.

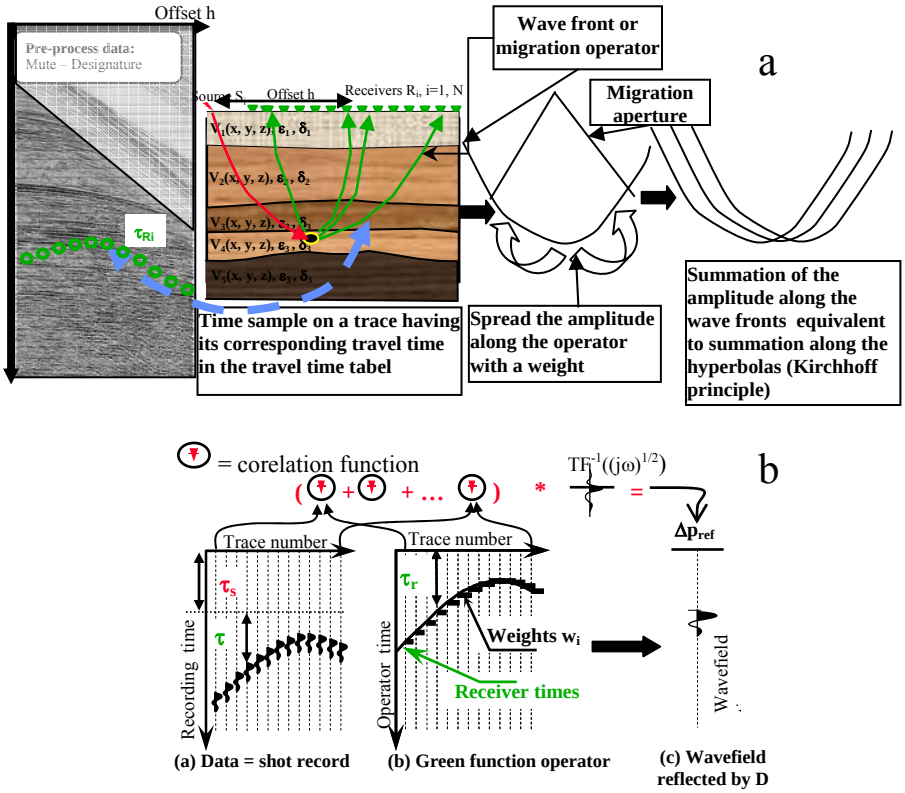


Figure 3.6: *In the Kirchhoff migration (a) , for an image point, Green functions (b) are computed through the velocity model for all raypaths from image point to source and receivers. The process yields time and weight for each offset, indicating where corresponding sample should be read from the data set. All the samples are then weighted and summed to produce the estimated reflectivity at the image point. The process is repeated for all the image point and shots.*

3.5.2. Downward continuation methods

Downward continuation can be defined as the process in which the surface recorded data is stepped down into the subsurface. In this method an image in the subsurface is constructed by the downward continuing of data through small depth step and invoking the imaging condition at each depth level (Berkhout 1981). The imaging condition can be defined as the process by which the reflectivity information is extracted from the extrapolated data (downward continuation). These methods are also called recursive methods because the wavefield at each depth level is computed from the wavefield at previous depth level using one-way or full (two-way) wave equation. There are different implementations of downward continuation migration methods: shot record implementation, plane wave implementation, s-g implementation..... They all have in common a wavefield propagator kernel. The accuracy of these kernels directly impacts the efficiency of the migration algorithms and the quality of the resulting migrated image.

The wavefied extrapolation using the full wave equation can theoretically model arbitrarily steep reflectors and arbitrary lateral variations in velocity. Unfortunately, it remains too expensive to use, as it requires computation in the time domain on a very fine grid to avoid numerical dispersion. To overcome some of these difficulties, an approach based on the one-way wave equation has been developed. Different wavefield propagation methods differ primarily in the numerical domain used to solve one-way equation. Selection of domain to solve one-way wave equation depends on the available computational resources, geological complexity and required quality.

The wave equation methods also differ by the way of implementation of the imaging condition. The two most commonly used approaches are shot profile migration (SPM), and source receiver migration. Source receiver is also commonly referred as the survey sinking or the double-square root method, and despite its name, it is commonly applied in the midpoint-offset domain (Claerbout, 1971,1985).

3.5.2.1 Shot-profile migration *(SPM)*

The shot-profile migration forms a subsurface image by an interferometric extraction of energy from the extrapolated source and receiver wavefields. Claerbout's imaging principle (Claerbout, 1971) states that a reflector exists at a point where the upgoing and the downgoing wavefields coincide in time and space (Figure 3.7). He proposed to obtain the reflector maps by crosscorrelating upgoing (*R)* and downgoing (S) wavefields in the earth. The upgoing and downgoing wavefields can be obtained by continuing downward into the earth, the recorded wavefield and source function respectively.

$$I(x,z)=\sum_{s}\sum_{\omega}S(x,z,\omega,s)\overline{R(x,z,\omega,s)} \qquad (3.31)$$

where x and z are the surface and depth positions, respectively, ω is the frequency, s is the shot position and $\left(\bar{}\right)$ is the complex conjugate.

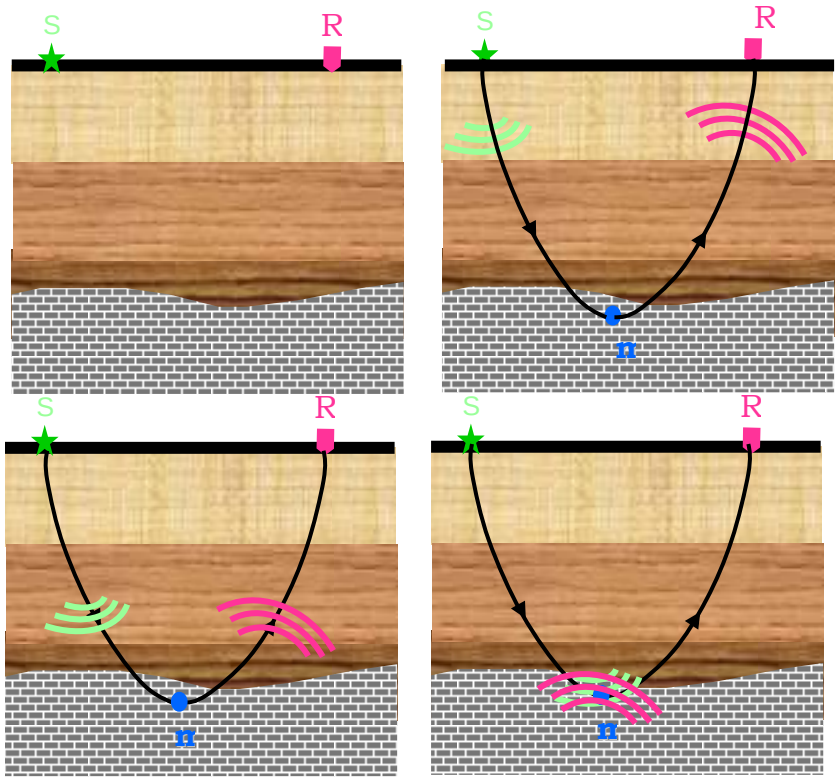


Figure 3.7: *Principle of wave equation migrations. Propagation the wave downward to the image point and crosscorrelating of the upgoing (R) and downgoing (S) wavefields in the image point to obtain the reflectivity.*

In shot profile migration, each shot can be migrated independently, and PSDM image is formed by stacking of the individual image.

Rickett and Sava, (2002) generalized the equation (3.31) by cross correlating the wave-fields and shifting them horizontally with respect to each other. The prestack image then becomes a function of the horizontal relative shift, which has the physical meaning of a *subsurface half offset* (*xh*). It is defined as

$$I(x,z,h) = \sum_{s}\sum_{\omega} S(x-h,z,\omega,s)R(\overline{x+h,z,\omega,s}) \quad (3.32)$$

In equation (3.32), $I(x,z,h)$ is the common image gather at an horizontal offset x.

Sava and Fomel (2003) presented a simple method for transforming the offset domain common image gather into Angle Domain Common Image Gathers (ADCIGs) by a slant stack transformation applied to each offset domain CIGs as given below:

$$Ix\ (z,\ x,\ \tan\gamma\ ) = \text{SlantStack}\ [\ I(x,z,h)\ ] \qquad (3.33)$$

where γ is the aperture angle of the reflection.

This transformation from offset domain CIG to ADCIG is based on the following relationship between the aperture angle and the slope, $\partial z/\partial xh$, measured in image space:

$$-\partial z/\partial x_h = \tan(-k_{xh} / k_z) \ ; \qquad (3.34)$$

where k_{xh} and k_z are respectively the half-offset wavenumber and the vertical wavenumber. The relationship between $\tan\gamma$ and the wavenumbers suggests that the transformation to ADCIGs can be accomplished in the Fourier domain by a simple radial-trace transform (Sava and Fomel, 2003).

3.5.2.2 *Source-receiver migration*

Source receiver migration is based on the survey sinking principle (Claerbout,1985). At each depth level, the downward-continued shot gathers are sorted into the receiver gathers, which are downwards, continued to the same depth level. The extrapolated data is equivalent to the data that would have been recorded if all the sources and receivers were placed on that level. The reflectivity is then estimated by invoking the zero time and zero-offset imaging conditions.

It has been shown mathematically that properly implemented DSR and SPM have equivalent accuracy and produce equivalent results (Wapenaar and Berkhout, 1987; Biondi, 2002). Based on algorithm considerations and imaging results, there are different areas of applicability for different imaging formulations. SPM are well suited for land and ocean bottom data, while DSR-based wave equation migration is best for marine streamer data.

3.5.3 Comparison of Kirchhoff and wavefield-continuation method:

In following table pros and cons of both Kirchhoff (integral methods) and wavefield-continuation (differential methods) are summarized (Jones & Lambaré, 2003).

Kirchhoff method	wavefield continuation methods
• Kirchhoff and Gaussian beam are the best known and they are usually implemented in time domain. • A distinguishing feature is the separation of calculation of travel time from imaging thus a subset of image can be computed without needing to image the entire volume.	• Finite difference wavefield continuation is the best known, in conjuction with phase shift plus corrections. Each depth slice of the wavefield is computed from the previously computed slice, thus essentially the entire image volume needs to be formed. Dip response is dependant on the order of expansion used (Thus it is potentially costly)
Strength: • It delivers a subset of the image volume, including offset (thus it is cost effective for the iterative model building) • good dip response • can yield sub-set of the full two-way solution (turning waves)	**Strength:** • image all arrivals • simple amplitude treatment • A full two-way implementation can yield all wave-path, including prism waves turning waves and (perhaps) multiples
Weaknesses: • Inherently kinematic • usually delivers only one arrival path	**Weaknesses:** • Images whole volume (thus costly) • obtaining good dip response is expensive

• velocity field coarsely sampled for travel time computation, then arrival times interpolated back to seismic spacing. • no readily extendible to a full-wave solution	• does not readily produce pre-stack data • thus difficult to achieve cost–effective iterative model building without 'restrictive' assumption (e.g. mono-azimuth) • very expensive to invoke a full two-way solution (but can use approximate two pass one-way)

Conclusions

In this chapter we reviewed state-of-the-art velocity estimation techniques for velocity analysis in geologically complex region. Travel time tomography requires a 4D picking of data which is difficult in time domain for geologically complex. This picking can be easily done on depth domain because it is more interpretable. Still picking is heavy (4D interpretation). Moreover observation on the gathers can be described by a family of curves which reduces a 4D interpretation into 3D. There are many implementation, most of them base on ray tracing method (sensitive to sharp velocity variation) to flatten the Common Image Gathers (CIGs). However Differential semblance optimisation (DSO) methods do not need picking, the objective function is also based on the flatness of the CIGs, directly linked to velocity errors. Hence in our approach we are going to use DSO based approach using wave equation migration.

Chapter 4

Representations for evolutionary multi-objective subsurface identification

There are two problems dealing with subsurface identification: The geological problem, tries to identify the subsurface structure using either data from surface geologic surveying or well logging; the geophysical problem aims at identifying subsurface velocity distribution from seismic data. In both problems however the first issue to be addressed is that of the subsurface representation.

In this chapter we are investigating different representation techniques for subsurface structure (i.e. Voronoi, spline, grid and geological). After a brief survey of Voronoi representation (section 4.2) we discuss its drawbacks and propose a new geological representation technique for foothill structure (section 4.3). An implementation of this representation is presented (section 4.4) and is tested on an artificial geological inverse problem in foothills structure (section 4.5). However, because the extension of the geological up to the geophysical problem was not efficient enough (section 4.6) we turn back to the grid representation (section 4.7) that we analyzed in depth here. We discuss that velocity representation is a subjective issue and it should be considered according to the geological complexity, available data information and computational resources.

4.1. Introduction

Representation of subsurface structure is one of the most critical issues in subsurface identification, as it controls the shape and size of the model parameters. The traditional approach for geological modeling is to repeatedly evaluate geological models with respect to balancing principles, namely, rock volume and bed-length preservation. These

geometrical hypotheses, which are only approximations to reality, mean that geologic structures behave locally like a folded paperback book, with layers slipping on each other like pages during folding. This geological modeling, which may be backward (Moretti et al., 1989) or forward (Endignoux et al., 1989), is very cumbersome and human time consuming.

Geophysical modelling, on the other hand, has taken advantage from the progresses of numerical modelling, and several works address the problem using classical gradient-based identification methods. Because the problem either has to be oversimplified, or is ill-posed, several works used Evolutionary Algorithms (EAs), that are known to be well suited for solving complex, mathematically ill-posed identification problems (Schoenauer and Sebag, 2002). However, a critical question is the choice of a representation (and of the associated variation operators (crossover and mutation)). In the case of subsurface identification, state-of-the-art works either assumed some expert knowledge about the geometry of the subsurface structure (e.g. horizontal layers (Stoffa & Sen, 1991), or more complex geologically-driven topology (Boschetti et al., 1996)), or used global automatic models that lack geological soundness (Mansanné et al., 2002) (e.g. the subsurface structures are not balanced).

Mansanné et al., (2002) made an attempt identify subsurface by representing a blocky subsurface structure using Voronoi sites (Figure 4.1). The underlying hypothesis in their work was that the sub-surface structure can be partitioned into regions of homogeneous velocity, i.e. velocity is supposed to piecewise constant. The idea was to evolve both portioned and real-value velocities. Whereas spline (Jervis et al (1996); Docherty et al. (1997)) coefficient encoded into a binary strings were used to represent subsurface structure.

The central issue when addressing such subsurface structure inversion problem is the trade-off one has to make between the complexity of the representation and the accuracy of the model in the resulting search space. For e.g. representing a grid of 200x200=40,000 points can be represented as a full 40,000–long vector in full parametric

approach (one velocity per element of the mesh), packed into 20x10=200 spline coefficients if the spline approximation is preferred, or into a few dozens of Voronoi sites.

The Voronoi representation is a compact unstructured representation. It has first been proposed in (Schoenauer 1995), as a first attempt toward unstructured representation for topological optimal design (TOD). Since then it has been used for different identification problems (Schoenauer et al., 1996; Schoenauer et al., 1998). A brief introduction of Voronoi diagram is given below before we highlight a typical situation where geological knowledge is crudely missing.

4.2 Voronoi-Based Representations

Voronoi Diagram: Consider a finite number of points $V_0,\ldots V_N$ (the Voronoi sites) of a given subset of R^n (the design domain). To each site V_i is associated the set of all points of the design domain for which the closest Voronoi site is V_i, termed Voronoi cell. The Voronoi diagram is the partition of the design domain defined by Voronoi cells. Each cell is polyhedral subset of the design domain, and any partition of design domain of R^n into polyhedral subset is the Voronoi diagram of at least one set of Voronoi sites (see Preperata and Shamos, 1985 for detailed introduction to Voronoi diagrams, and a general presentation of algorithmic geometry).

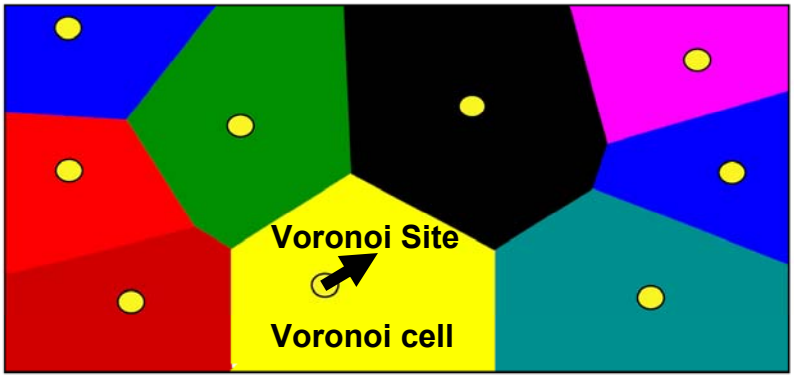
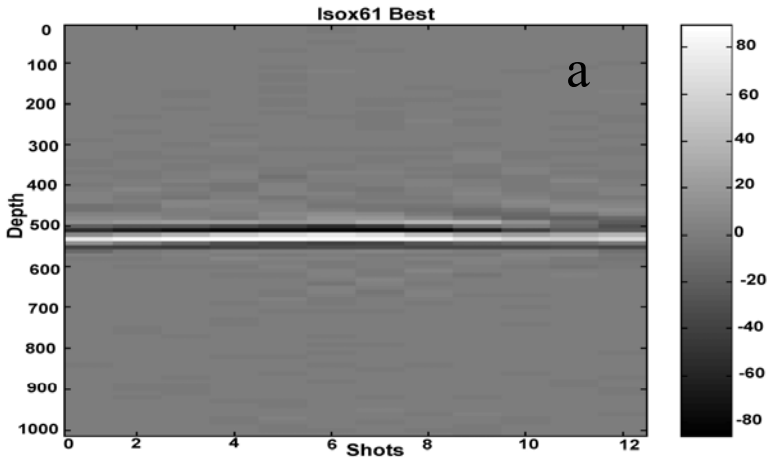


Figure 4.1: *The yellow dots represent the sites. Voronoi diagram divides the sites into the regions (cell), such that the borders of the regions are equidistance from the two nearest sites.*

A 2D Voronoi velocity model will consists of several sites and each sites is represented by three parameters, two space co-ordinates (x, y) and one velocity parameter. These Voronoi sites can adopt any shape. Also it is straightforward, the extension from 2D to 3D velocity models requires one more space co-ordinates and few more Voronoi sites to represent.

Mansanné et al. (2002) optimised a simple 2D subsurface structure using genetic algorithm; they represented subsurface by Voronoi model and applied geophysical criteria (semblance function) to measure the fitness. Numerically, they were able to optimize (maximum semblance or flat gathers) the models, although the optimised subsurface structure was same times geologically insignificant (Figure 4.2). They experienced the need of some guidelines to optimize the subsurface structure and avoid the generation of absurd geological models.



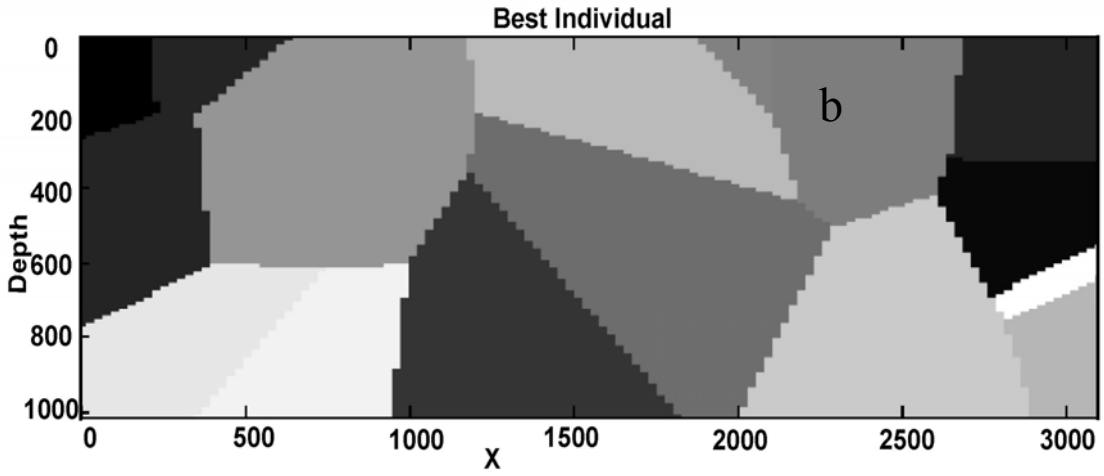


Figure 4.2: ***(a)*** *Flat gathers obtained from the velocity model.* ***(b)****This Flat gather obtained from a geologically absurd velocity model (Mansanné et al. (2002)).*

4.3 Geological knowledge

In this work we made an effort to generate geologically significant structure and incorporated geological information to guide the optimisation process. We propose an original indirect representation for subsurface identification. The morphogenesis process (from the encoded representation to the subsurface structure) can only generate balanced geometries. Moreover, it can be used to solve either or both the geological and the geophysical identification problems by EAs automatically, and without the need for any additional expert knowledge. The idea is to represent a subsurface structure by the combination of, first, an initial configuration (series of flat homogeneous layers of different geological nature), second, some initial faults in that simple structure, and third, a description of successive geological deformations along the ages. A kinematic model, that relates fold geometry to fault shape and displacement, is used to construct cross-sections of the geological structures. These cross-sections are automatically balanced by virtue of the above-mentioned balancing principles (volume and bed-length preservation) upon which the kinematic model is based. The resulting complex subsurface structure is then used to compute the identification criteria: comparison with surface or subsurface geological data, geophysical simulation of some seismic experiment, etc (next sections).

However, as already mentioned, the different types of available information may be conflicting. Therefore the problem is better reformulated into a multiobjective optimization problem: there is no unique optimal solution but a set of Pareto optimal solutions, also termed non-dominated non-inferior, admissible, or efficient solutions. Multiobjective evolutionary algorithms (MOEAs) as discussed in Chapter 3, more precisely, the Epsilon-MOEA [Deb-epsilon], have been used to tackle this multiobjective problem.

4.3.1 Geology

Complexity and ambiguity of mountain fronts are what makes them the most challenging fronts for a geologist or a geophysicist. Now the thrust regions (mountainous regions) are the next human quest for petroleum exploration. These fold-and-thrust belts are the manifestation of collision of tectonic plates. Most fold-and-thrust belts consist of foreland-progressing and hinterland-dipping thrusts (Boyer and Elliott, 1982) that may connect layer-parallel detachments (décollement horizons). The distribution of mesoscopic deformation features in fault-related folds has important consequences for hydrocarbon migration, trapping and production. Structural, environmental and stratigraphic factors control the spatial distribution and intensity of the deformation features (e.g. Fischer and Jackson, 1999) of mountains, under a given rheological stratigraphy (Woodward and Rutherford, 1989) and prevalent environmental conditions (Stewart and Alvarez, 1991; Jamison, 1992; Lemiszki *et al*., 1994). It is fault-fold kinematics which controls the distribution and intensity of deformation that develop in a rock sequence (Fischer *et al*., 1992; Storti and Salvini, 1996).

4.3.2 Kinematic Models

For the representation of mountainous structure mainly three types of kinematics models appear in the literature. ***1)*** *kink model* (Suppe 1983, 1990), ***2)*** *trishear model* (Erslev, 1991; Hardy and Ford, 1997; Allmendinger, 1998) and ***3)*** *force fold model* (Johnson and Johnson, 2000). Amongst the above mentioned three types, the kink model

is considered best suited for thin skinned tectonic regions. Since we are interested in such regions, hence kink model is being used.

A two-dimensional geometric model of fault-bend folding for a thin-skinned tectonic was first formulated by Suppe (1983) based on the conservation of layer thickness and bed length. He showed its applicability in the simulation of the Pine Mountain thrust sheet in the southern Appalachians, and to the fold and thrust belt of Western Taiwan. The use of kink-band style folding with non-deforming footwalls (characteristic of such geometric models) has been criticized by Ramsay (1991) as being unrealistic. Another fault-bend folding model was proposed by Contreras and Suiter (1990), however it was also based on the conservation of area. Since their introduction, these geometric models have received a lot of attention (Medwedeff, 1989; Jordan et al., 1993) and have been used extensively to predict thrust fault geometries at depths based on observed fold geometries. They have also been used in a forward modeling sense to predict hanging wall geometries above thrust faults (Mitra, 1990; Contreras and Suiter, 1990, 1997; Zoetemeijer, 1993; Hardy S., 1995).

4.3.3 The Contreras Model

The kinematic model used in the current simulation was introduced by Contreras and Suiter (1990). This model can be applied to regions being deformed by shortening, extension and also for the duplex systems (Contreras and Suiter, 1997). This is a two-dimensional model and is based on a coordinate transformation from less deformed state to a more deformed state (Figure 4.3). The medium is subdivided into domains of constant dip and homogeneous displacement vector fields that are delimited by the planes bisecting the fault inflections. The displacement occurs by translation. The displacement trajectory is of constant length for all the displaced particles throughout the medium and is parallel to the underlying active fault segment. This model also considers fault parallel simple shearing. A complete derivation Contreras fault bend folding (FBF) was provided by Contreras and Suiter (1990). The heterogeneity of the displacement vector field across axial planes introduces longitudinal and angular shear strains. Transformations from one

state to another state do not cause a change in area and also the deformation is isochoric, a characteristic typical of deformation by simple shear (Truesdell and Toupin, 1960).

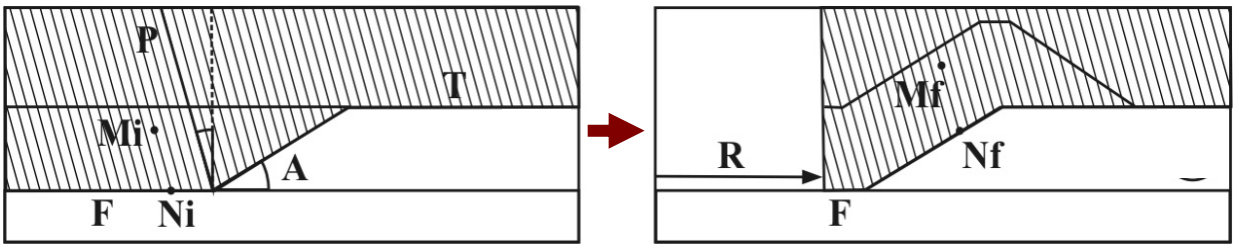


Figure 4.3: *Principle of the Contreras forward tectonic model. The co-ordinate transformation from less deformed state to more deformed state.*

4.4. The Evolutionary Algorithm

4.4.1 The Representation

As mentioned in the introduction, the main difficulty when tackling a geological or geophysical identification of subsurface structure problem by EAs is to define a representation: structured representations are either unrealistic in complex regions like foothills (e.g. a fixed number of horizontal layers) or require some very specific expert knowledge about the geometry of the subsurface structure (Boschetti et al., 1996). Unstructured representations, on the other hand, don't require any input from the expert – but they lack of the minimal geological common-sense and hence can lead to quasi-perfect numerical fit that are absurd from a geological point of view (Mansanné et al., 2002). Note that some alternative possibilities have been proposed, that define a global model with numerous local parameters (e.g. using spline nodes uniformly spread on the considered domain), but such representations have the same lack of geological soundness, while additionally resulting in a huge optimization problem for the sake of precision. Moreover, it seems very difficult indeed to try to constrain the solution proposed by unstructured representations with some geological rules (e.g. some simple rules could be that the underground velocity should increase with the depth – but interesting regions are precisely regions where such rule is violated!).

The representation proposed in this work deals with the above-mentioned problem by relying on a kinematic model of subsurface deformation from an initial simple state

(horizontal layers, that can be assumed to be universally true in all regions at some point in the past) and subsequent deformations based on one of the kinematic models discussed in section 4.3.3 that ensure the consistency of the structure at any time, including the final state of the model – the target subsurface structure.

In EAs terminology (see Chapter 2), the **genotype space**, in which the algorithm will actually search, and where the variation operators are defined, is here the space of initial configurations plus initial faults plus series of deformations (see figure 4.4 for the detailed description of the complex data-type). The **phenotype space**, or behavioral space, where the fitness of each genotype is computed, is the space of subsurface structures, obtained from the genotypes by applying a transformation, called the **morphogenesis process,** or also the numerical Contreras model described in section 4.3.3. The tectonic modeling presented in section 4.3.3 is sound because we will obtain only balanced subsurface structures by this morphogenesis process.

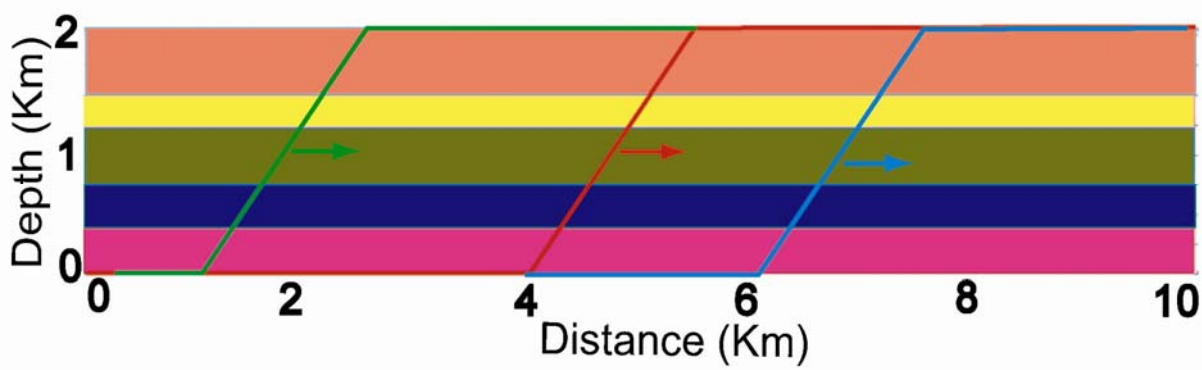


Figure 4.4 *A chromosome: Unknown parameters are the number of initial layers and, for each layer, its thickness; the number of faults and, for each fault, its origin, and number of segment, and, for each segment, its length and dip; and the displacements (for each fault, one horizontal displacement).*

4.4.2 Initialization and Variation Operators

First note that the representation proposed in this work is possibly a variable length representation (variable dimension parameter space). However, in the experiments presented in section 4.5, we have used a fixed length representation: the number of faults

and the number of segments per fault will be fixed (from the target model). We shall hence here only describe the operators of this fixed-length restriction.

4.4.2.1 Initialization

Here a complex structure of real values is being used, therefore the parameters are initialized uniformly on a given interval, with some geometrical constraints:

- Displacement of a fault can not be more than the difference of distance between the first segments of two successive faults.
- The length of first segment increases in the successive faults.

4.4.2.2 Crossover Operator

Because the representation is at the moment fixed length, the Simulated Binary Crossover (SBX) (Deb and Agrawal 95) recommended with the epsilon-MOEA algorithm has been used. During the crossover, the geometry constraints listed above are respected.

A quick parameter study (12 independent runs for each value of η) has been performed (on the 3-faults problem described in section 4.5.3). All final populations are merged and non-domination sorting is performed (result can be seen in Figure 4.5). It is found that high value of η is most suitable for this simulation, and thereon the distribution index η has been set to 15. Note that the recommended values of ε_1= 0.05 and ε_2= 0.05 were found robust enough and used in all experiments presented in section 4.5.3.

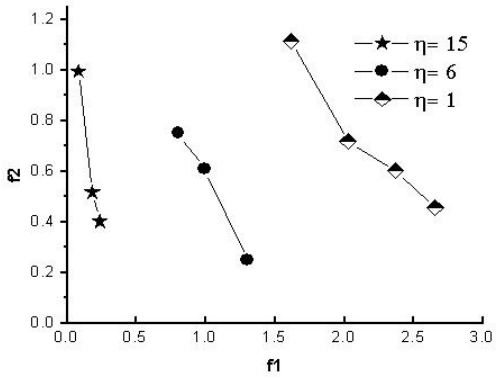


Figure 4.5 *Influence of the distribution index η of SBX crossover on the results (results of 12 runs). Higher value of distribution index η seems more efficient for foothill inversion.*

However, because the SBX crossover can hardly be applied on variable-length genotypes, some more variable-length specific crossover operators could be used in the variable-length case (add number of fault and segment).

4.4.2.3 Mutation Operator

The SBX crossover is known to be self-adaptive in the sense that the spread of the possible offspring solutions depends on the distance between the parents, and decreases as the population converges (Beyer and Deb 2001). Hence it is generally used without any mutation (the use of mutation together with SBX in the context of the problems presented in section 4.5 did not seem to bring any benefit indeed, and was abandoned). However, because the variable-length case requires additional use of mutation, self-adaptive Gaussian mutation (à la ES) is used on all real-valued parameters, while specific variable-length mutation operators are used to modify the numbers of faults and segments in each fault.

4.4.2.4 The ε-Multiobjective Evolutionary Algorithm

As mentioned in the introduction, various sources of data will be used for identification. Hence, the identification problems will be turned into Multi-Objective Problems (MOPs). Examples of such problems can be seen in section 4.2 for the geological modeling problem, or in (Mansanné 2002) for purely geophysical identification using both the semblance and the least-squares error on seismic signals. But the ultimate goal of the present work is to simultaneously use all available data, geological and geophysical, leading to even more objectives.

For the optimization of foothill structure (section 4.5), we have chosen the recently proposed, ε-MOEA (Deb et al., (2003)) because it achieves all the above desired properties at a minimum computation cost. The code of this algorithm is graciously offered by the authors at URL http://www.iitk.ac.in/kangal/soft.htm. This algorithm is already described in Chapter 2 (Section 2.7.7.9). For sake of completeness, here once again we are summarizing it.

ε-MOEA is based on the principle of ε-dominance that relaxes the strict Pareto-dominance principle. It uses two co-evolving populations: an EA population and an archive population A, and is a steady-state EA. The initial archive population A is assigned from the initial EA population using ε-non-dominated solution. Thereafter two solutions are selected, one from the EA population using tournament selection and the other from archive A using random selection. From those parents, an offspring O is created, and is used to update the EA population on the basis of strict dominance, and the archive population on the basis of ε-dominance. Only one solution in each hyper-box on the Pareto-optimal front is allowed, to favor diversity.

4. 5 First Results in Geological Modeling

4.5.1. The Geological Identification Problem

For identifying a foothill structure, a purely geological inverse problem is defined. In this inverse problem, the unknowns are the parameter of a structure, and the data are layer dip or fault location measured on the topographical surface or along a well (Figure 4.6). Experimental results are being presented on artificial problems: the "experimental" data are computed on a model that is in itself the result of an evolution using the Contreras kinematic model starting from a known initial configuration. Two models have been considered, a 3 fault and a 7 fault models.

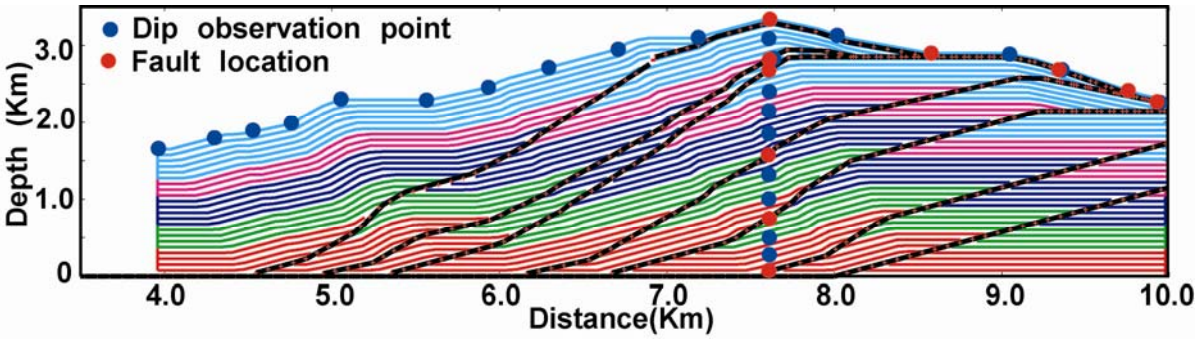


Figure 4.6 *Data are the measured fault location (red circles) and the dip of the layers (blue circle) along surface and well.*

4.5.2 The Evaluation Functions

Evaluation is the most costly and important step for real applications. Kink model (Contreras and Suiter, 1990) is used for the deformation of the model from the initial

state to final state. After deformation, dip and fault location parameters are evaluated for optimization purpose. In general, dip of faults, stratigraphic unit, and fault location are easily obtained from the field observations, well logs (dip-meter) or by remotes sensing data. For the evaluation of a model, a least-squares criterion measures the discrepancy between the field data and corresponding dip and faults location on the current model. The relation is shown by the following equation.

$$f1 = \sqrt{\sum_{i=0}^{nd} \left(Dip\_ob_i - Dip\_gen_i \right)^2}$$

$$f2 = \sqrt{\sum_{i=0}^{nf} \left(Fault_{position} Ob_i - Fault_{position} gen_i \right)^2}$$

The computational cost of the evaluation function is hence negligible when compared to that of the morphogenesis process (the Contreras model that computes the deformation of the subsurface structure). For instance, for the experiments presented in next sub-sections, the total computational cost on a Pentium 3.4GHz is about 4 hours.

4.5.3 A Three Fault Model

A target mountain front model, with five layers and three faults, is shown in figure 4.7 Because this model was generated from an initial genotype (as described in section 4.3.3) with the Contreras model, it is the exact solution to the identification problem. On this example, data is only available at the surface, light circles representing dip information, dark circles fault locations.

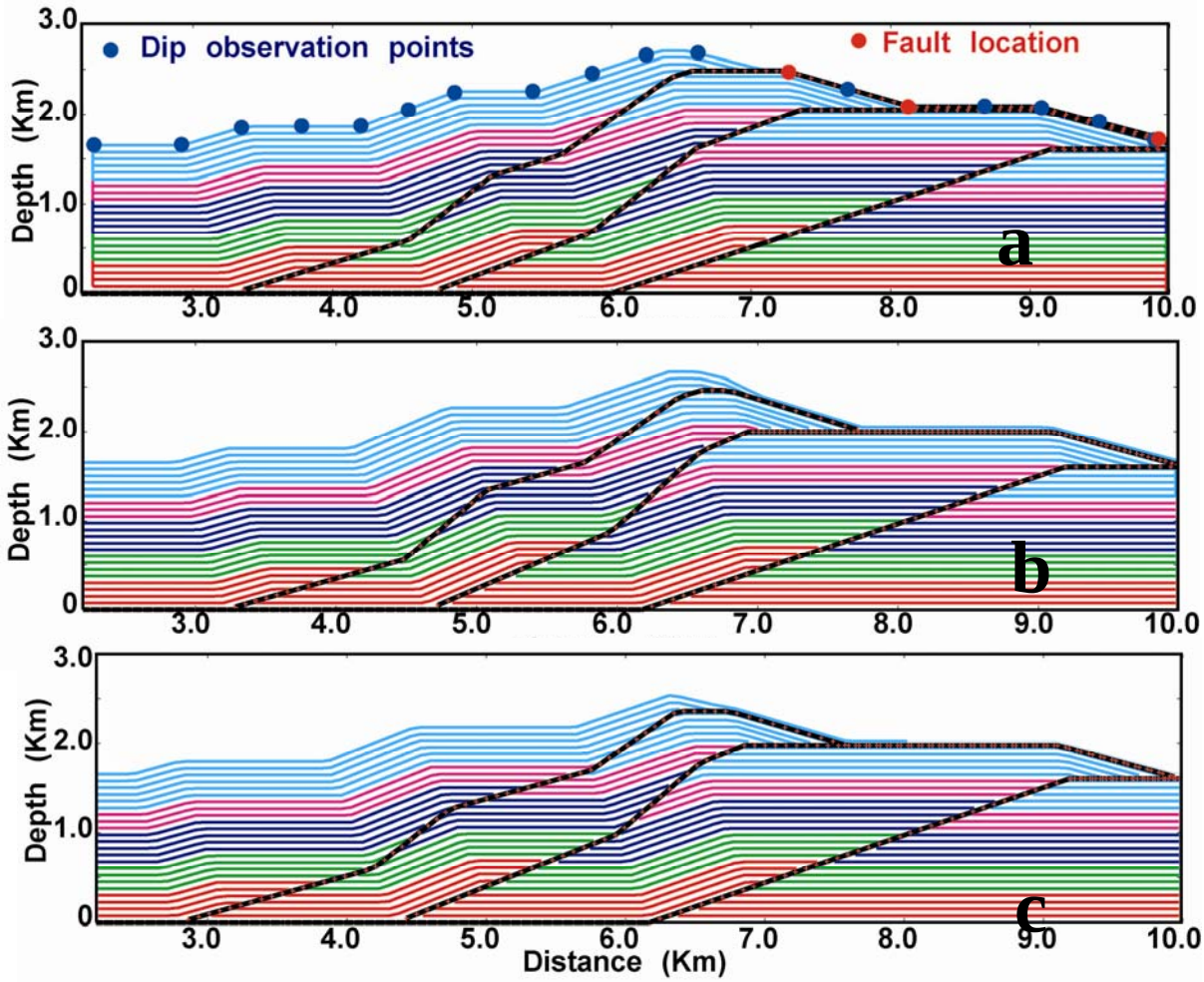


Figure 4.7: Good *results on the 3-faults model: (a) is the target structure. (b) and (c) are result structures. Structure (b) is similar to target structure in all respects. Structure (c) is also similar to the target structure but fault locations are slightly different. However, both structures coincide with the target at the observation points.*

For this experiment, the population size was 30, Maximum archive size 30, number of generations 300, and, as mentioned in section 3, the representation was restricted to fixed-length 3 fault genotypes.

Figure 4.7 shows some identified models that are very similar to the target model. Model 4.7 (b) is similar to the target model in terms of surface and subsurface fault location and topography, while model 4.7(c) is also similar to the target structure but its fault positions are slightly different. However, some others models have also be obtained, which are numerically quasi-optimal too, but visually quite different from the target model (see

figure 4.8). This leads to the conclusion that the problem is underdetermined: we need more information in order to uniquely define a solution.

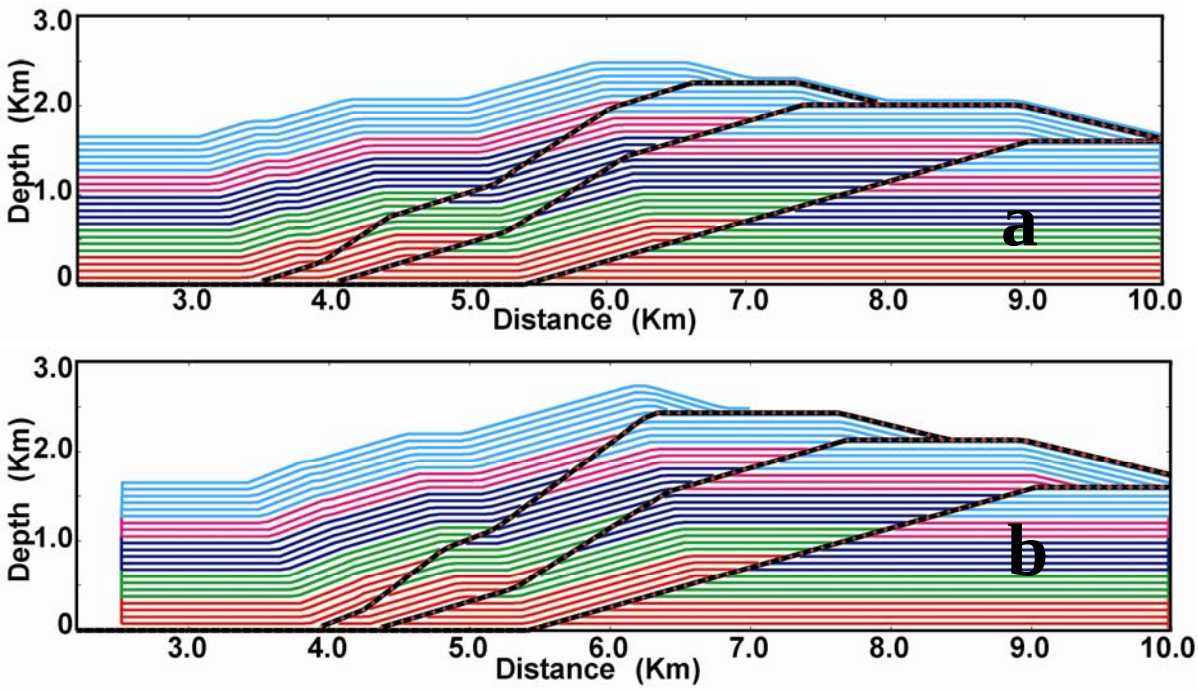


Figure 4.8 *(a) and (b) are the others optimized models. The models (a) and (b) are similar along the topographic surface, but visually different below.*

4.5.4 Results on the Seven Fault Model

A more complex seven fault model is shown in figure 4.9 (a). On this model the observation points are located both on the surface and in a well. For this experiment population size was 50, archive size 50, and number of iterations was 500. The identified models are shown in figure 4.9 (b) and (c). Again, they are very similar to the target model. In figure 4.9(b), the fourth fault is visually different from its analogue in the target model but the rest of the faults are both numerically and visually similar. Figure 4.9(c) also looks very similar to the target but for the sixth and fourth faults. Here also, as can be seen on figure 4.10, some identified models are quite different from the target model– though being quasi-optimal numerically, i.e. the available data on the surface and in the well are almost perfectly fitted.

This situation is very typical of ill-posed problems: we do not have sufficient information to ensure the uniqueness of the solution. Remember that some basic geometry constraints have been added to the representation (e.g. successive faults first segment length will be more than the previous one, see Section 4.4.2.1). Such additional knowledge did prove beneficial: without this information, the algorithm need around 1000 iterations to converge while only around 300 with the constraints. However, higher order of information like seismic, gravity or magnetic data, seem necessary to improve the results.

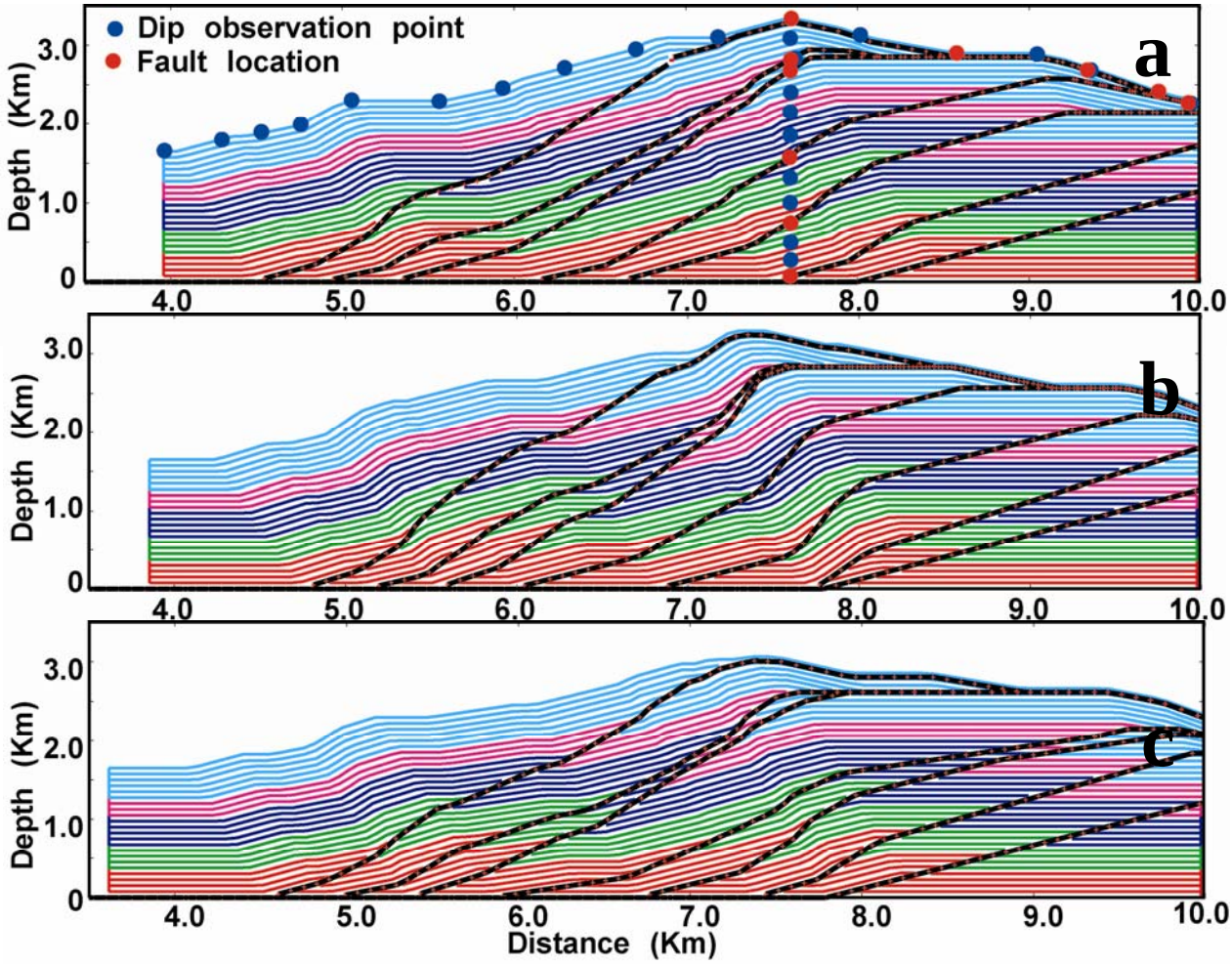


Figure 4.9 *(a) is the target structure, (b) and (c) are the numerically optimised structures. In spite of minor differences, models (b) and (c) are fairly similar.*

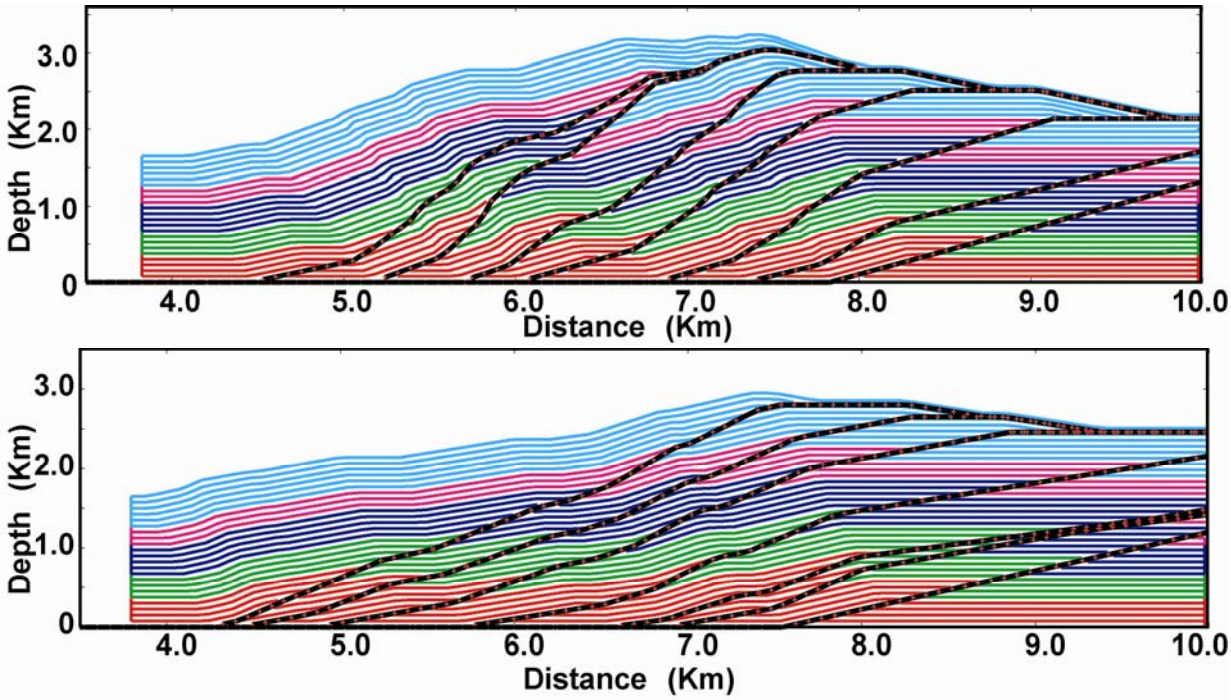


Figure 4.10 *(a) and (b) are other numerically optimal structures for the 7-fault model. But models 7(a) and 7(b) are obviously different from the target mode 6(a). However they fit rather nicely available dip and fault position data as well.*

4.5.5 Convergence test

Above experiments were performed with different random seeds to check the convergence of the algorithms. In the figure (4.11) symbol (R) represent the initial models and stars are the final models. We find that each time our algorithm converge toward the Pareto optimal solution. This shows that our optimisation process is independent of initial model.

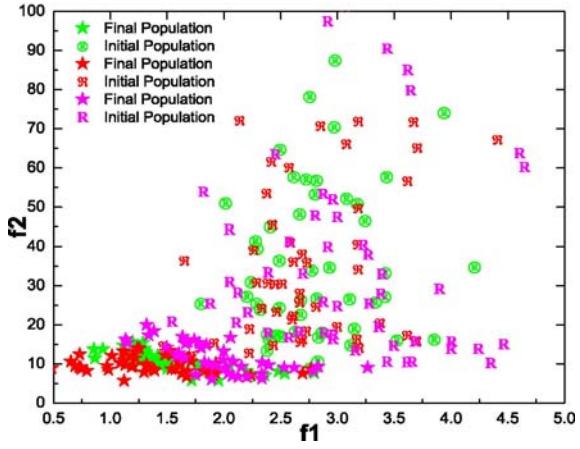


Figure 4.11: *The letter R (green, red pink) represents the random models at different initialisation and stars (green, red pink) represents the final population. In each initialisation population is converging toward the optimal solution.*

4.5.6 Discussion on foothill identification

The complexity and ambiguity of mountain fronts pose a significant challenge for both geologist and geophysicist in determining the geological and geophysical parameters. In traditional approach, mountain front identification needed either rigorous human interaction, because of absence of established numerical criteria to act as an inversion target, noisy surface, and subsurface data and little prior information, or lead to geologically unrealistic results because, except for very simple regions, the geometry of the subsurface structure cannot be accurately predicted.

We have proposed a new representation of subsurface structures by a set of automatically balanced geological models. It has two advantages: (1) it is geologically relevant (all structures are balanced), and (2) it does not require human guesses about the geometry of the unknown structure. Moreover, this is the first time, to the best of our knowledge, that multiobjective optimization is applied to a subsurface identification problem. The first results, obtained on synthetic geological identification problems, show the power of the proposed representation, even if we restricted this representation to fixed length for those final experiments.

Of course, this geological identification problem is ill-posed, and than more data is needed in order to reach good solutions with more robustness. We further perform experiment with the seismic data.

4.6 Velocity inversion of a foothill structure

One of the main goals of this thesis was to reconcile geology and geophysics by mean of EAs. After designing a geologically sound representation for sub-surface structure identification, there were at least two ways to proceed toward solving the complete geophysical identification problem: The first one is to first, perform the geological identification, and, then, freezing the sub-surface structure, identify the (constant) velocities in every layer; the second one is to add the velocity parameters into the geological representation (one velocity per layer), and to optimize both the subsurface

structure and the velocity distribution at the same time. We shall now discuss in turn these two approaches.

4.6.1 Two- step seismic velocity inversion

In our first approach we divide our velocity inversion into two steps. In the first step we performed geological inversion (as discussed in the previous sections of this chapter) and in the second step we perform the velocity inversion of the obtained geological model (Figure 4.12). In this second step, we assume that geometry of the foothill structure is known thanks to geological inversion and the only unknown parameters are then the velocities in every layer, that are assumed to be constant. We used semblance and differential semblance function (see Sections 3.3.3 and 5.2) to measure the fitness of a velocity model.

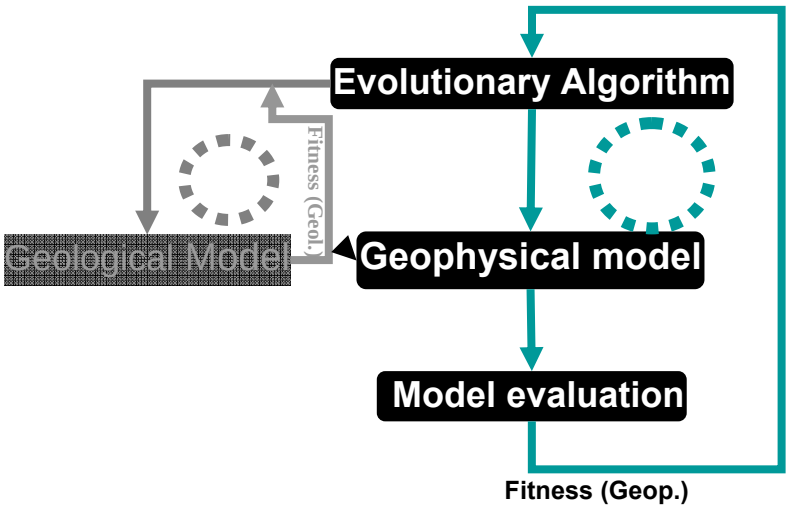


Figure 4.12: *Two step structural inversion. We assume that a geological model was inverted using geological criteria in a first step (in gray) Now, in the second step, we invert velocities using geophysical criteria.*

We have optimised a five layers synthetic foothill structure (Figure 4.13) using semblance and MODS criteria (see Section 5.2.2) with the multiobjective evolutionary algorithm ε-MOEA (see section 2.7.7.9).

We also used offset domain differential semblance (ODS) function in place of modified offset domain differential semblance (MODS, see section 5.2.2). Optimisation process took large number of evaluation and also need good parameter setting with ODS function. Whereas using MODS function, we were able to optimised in only few evaluation (100-150) and using half the number of shots and small frequency bandwidth. Results are shown in the figure (4.14).

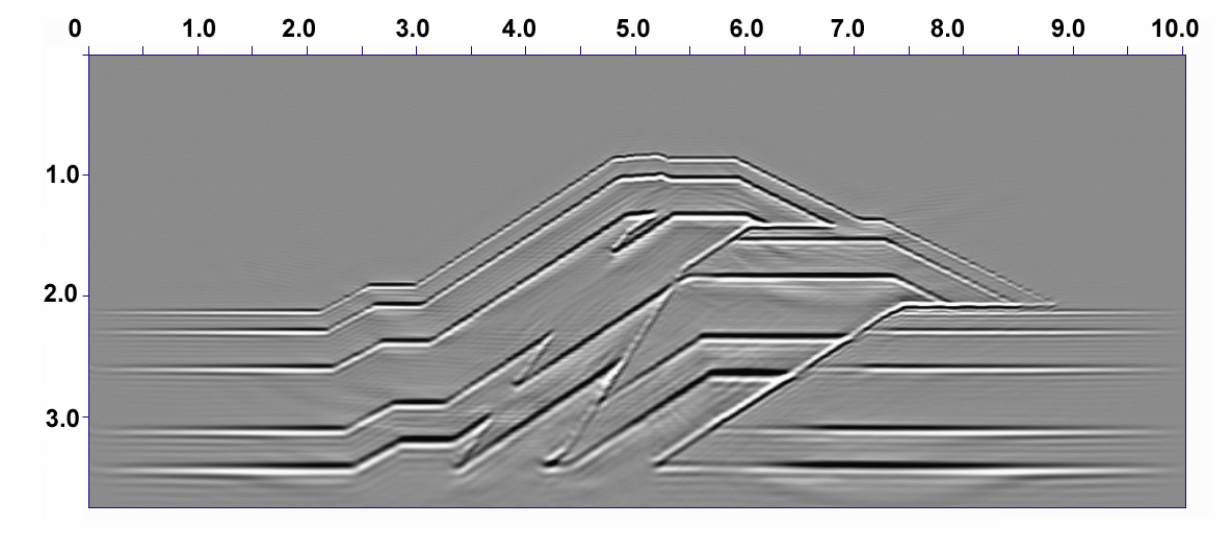


Figure 4.13: *This is the migrated image obtained by optimizing the layer velocities of a model with known geometry using MODS and DS function (see Chapter 5).*

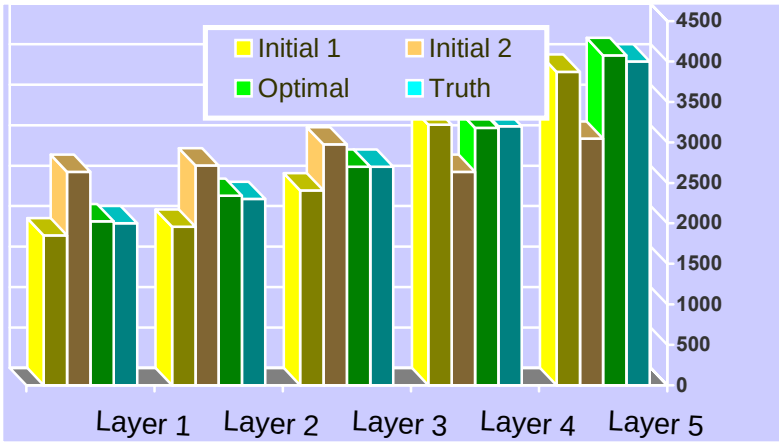


Figure 4.14: *For each layer of the model displayed in figure 4.13, this bar chart shows the velocity of two initial random models (yellow and orange), the velocity of one optimised model (green,) and the true velocity (sky blue). Optimised result is very close to truth.*

Doing this structural optimisation in two steps reduce the number of unknown variable, as a result optimisation speed is fast. This two step process can provides an option to verify the results and add some extra information during the optimisation. However velocity inversion completely depend on the models provided by geological inversion, which may results incomplete evaluation. This process may need human interaction and information. This process may also need human interaction and information.

4.6.2 Single step seismic velocity inversion

Since ε-MOEA provides a unique opportunity to optimize, models with different criteria independently, we decided, to evaluate a model both geologically and geophysically, in a single step (Figure 4.15). We used two geological objective functions *(see section 4.5.2)* and two geophysical objectives functions MODS and semblance to evaluate them. We generates horizontal layer model (see section 4.4.1) and add a velocity parameter corresponding to each layers. Then deform it using Contreras model (see section 4.3.3) and evaluate it using both geological and geophysical data and criteria.

We did many experiments, some with more than 5000 iterations. We were unable to get geologically significant results.

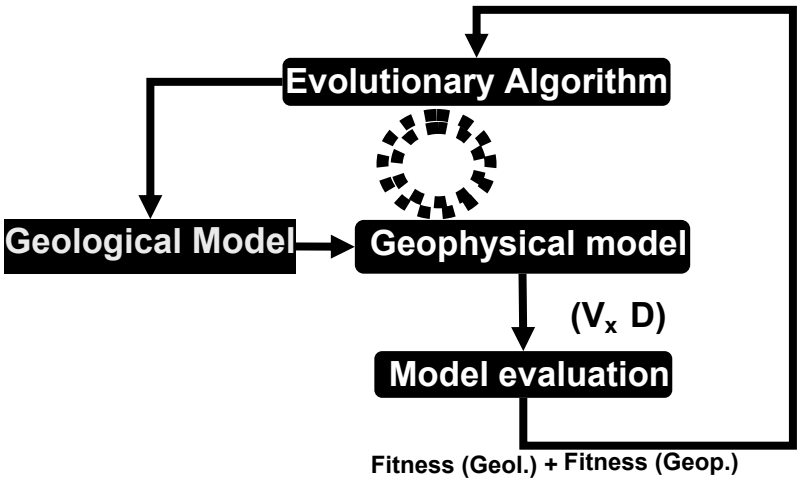


Figure 4.15: *Single step structural inversion. Here a model is evaluated both by geological (see section 4.5.2) and geophysical (MODS and semblance) objective function together using* ε-MOEA .

Doing both optimizations together increases the number of unknowns. Accuracy of geological parameters strongly influences the geophysical parameters. As a result optimization also requires a large number of evaluations. These large numbers of evaluations are not only because of number of unknown parameters but also due to increasing of the solution space dimension (4D, two geological and two geophysicist solution spaces).

We also realized that these approaches are limited by the geological and geophysical information and quite dependent on each other. Incorporation of information obtained from the seismic inversion to initial state (horizontal layer, faults etc.) needs a two step process. In the first step we need to introduce the information on a temporary geophysical space and then again go back to geological space to finally introduce the information, which is quite ambiguous. Since nature is very complex, and there are many circumstances where it is difficult to find a perfect geological simulator. One other difficulty that we realized is to find a real data with complete geological and geophysical information. Therefore we decide to go back to the classical grid representation.

4.7: Back to grid the representation

Grid representation is robust in terms of representing any geological structure. Moreover it provides a flexible and adaptable representation scheme and it is independent from the geological representational constrained (like, faulting, folding, erosion and deposition, etc.) though it requires a large number of grid points to represent complicated structures. Hence it may not be a good choice with global optimisation methods because of large number of unknown parameters. In spite of the fact that grid representation is not concise, we decided to use it because of its flexible and adaptable property.

First we are envisaged the effect of different size of grid representation on the velocity estimation. Representing velocity on a coarse grid is undesirable because it reduced the details that are present in the model. Whereas representing velocity on fine grid increases the unknown parameter for inversion. Some works have already addressed the effect of smoothing on the migrated image (Versteeg 1993, Gray 2001). However there is no any

definite answer to the question: how much one should smooth? Smoothing can be viewed as an averaging over a specific length. Velocity smoothing, though unnecessary for the finite difference migration, is usually a reasonable things to do before any depth migration, since we typically lack precise, detailed knowledge of the velocities inside the earth. Similarly for velocity inversion we would like to represent velocity on a fine grid. However, a large increase of unknown parameter enforces a coarse grid representation. Hence for a velocity estimation there is a need to find a compromise between the size of representation and corresponding accuracy.

Here we present some example of the Marmousi model for different grid size representations (Figure 4.16). To get the different grid size representation of Marmousi model, we smoothen it by different filter length and then resample it on same length of grid size. We applied damped least squares filter (Liu, 1994) to smoothen the original Marmousi velocity model. Smoothing is done on slowness (1/velocity) rather then velocity, to preserve the traveltime. We selected filter length of 100m, 200m, 300m and 400m to smooth and resample (Figure 4.16) the Marmousi velocity model. We applied shot profile wave equation migration on this velocity model to investigate the effect of grid size on the migrated image (Figure 4.17) and gathers (4.18).

First we notice that selection of a reasonable size of grid (e.g. 100m to 200m, Figure (4.16a and b)) does not make a noticeable impact on the migrated image even in complex region. This is similar to the Versteeg (1993) observation. However, large grid size (300m, 400m) representations deteriorate the quality of migrated image in the complex part. Whereas even a large grid size representation, does not have any significant impact on the simple part of migrated image.

Effect of the size of representation appears more significantly on the gathers than on the image. We can see that effect on the gathers start appearing since the beginning (Figure 4.17a). Effect of the size of grid is very significant on the complex part of the model. From Figure (4.17 b, c and d) we can see that in the complex part distortion of gather is very significant whereas in the simple part of velocity model distortion have almost no

effect

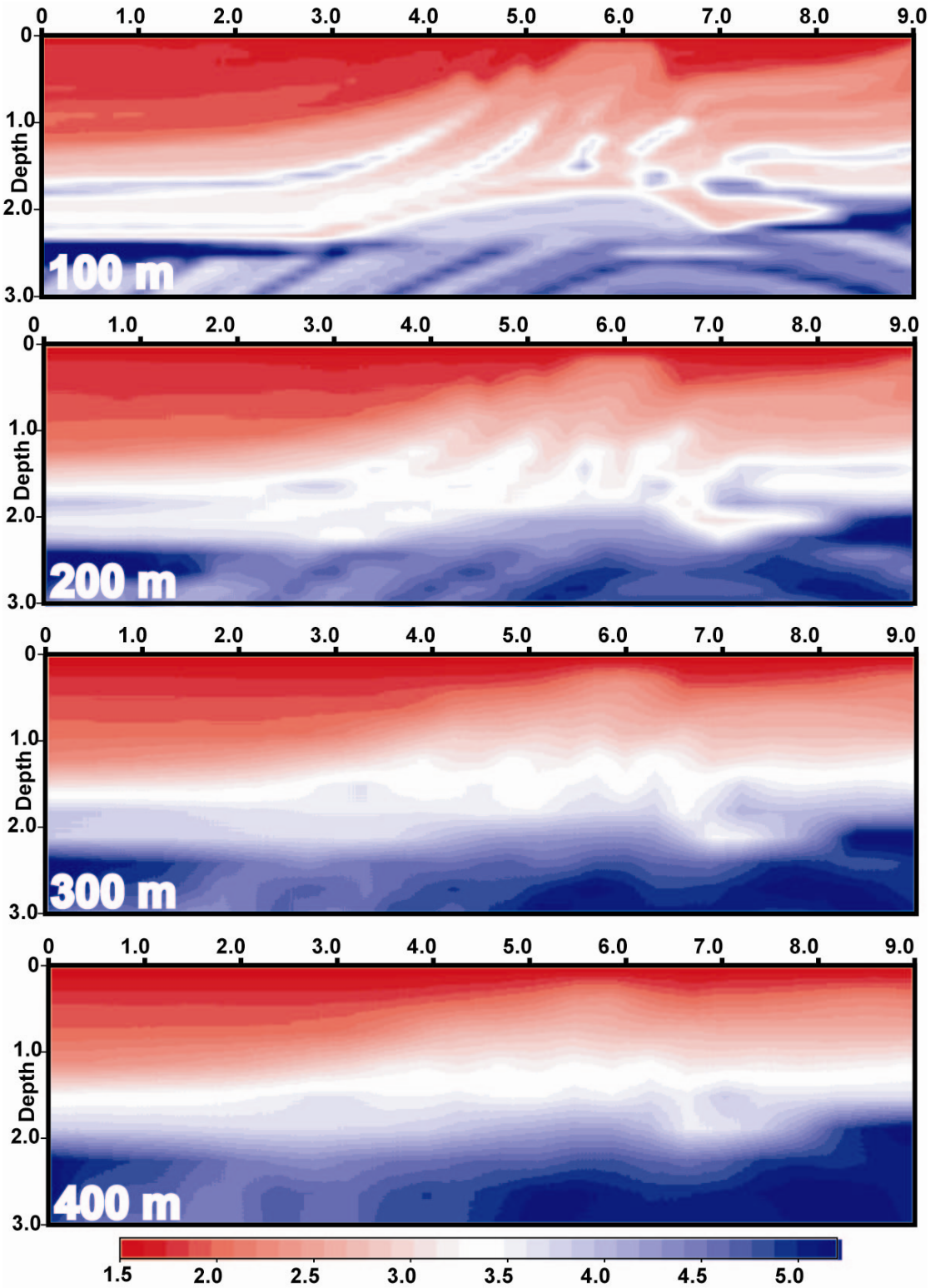


Figure 4.16 *Marmousi model smoothed with different filter size using damped least*

square technique (Liu, Z 1994), smoothing window size (a) 100 m (b) 200 m (c) 300 m (d) 400 m.

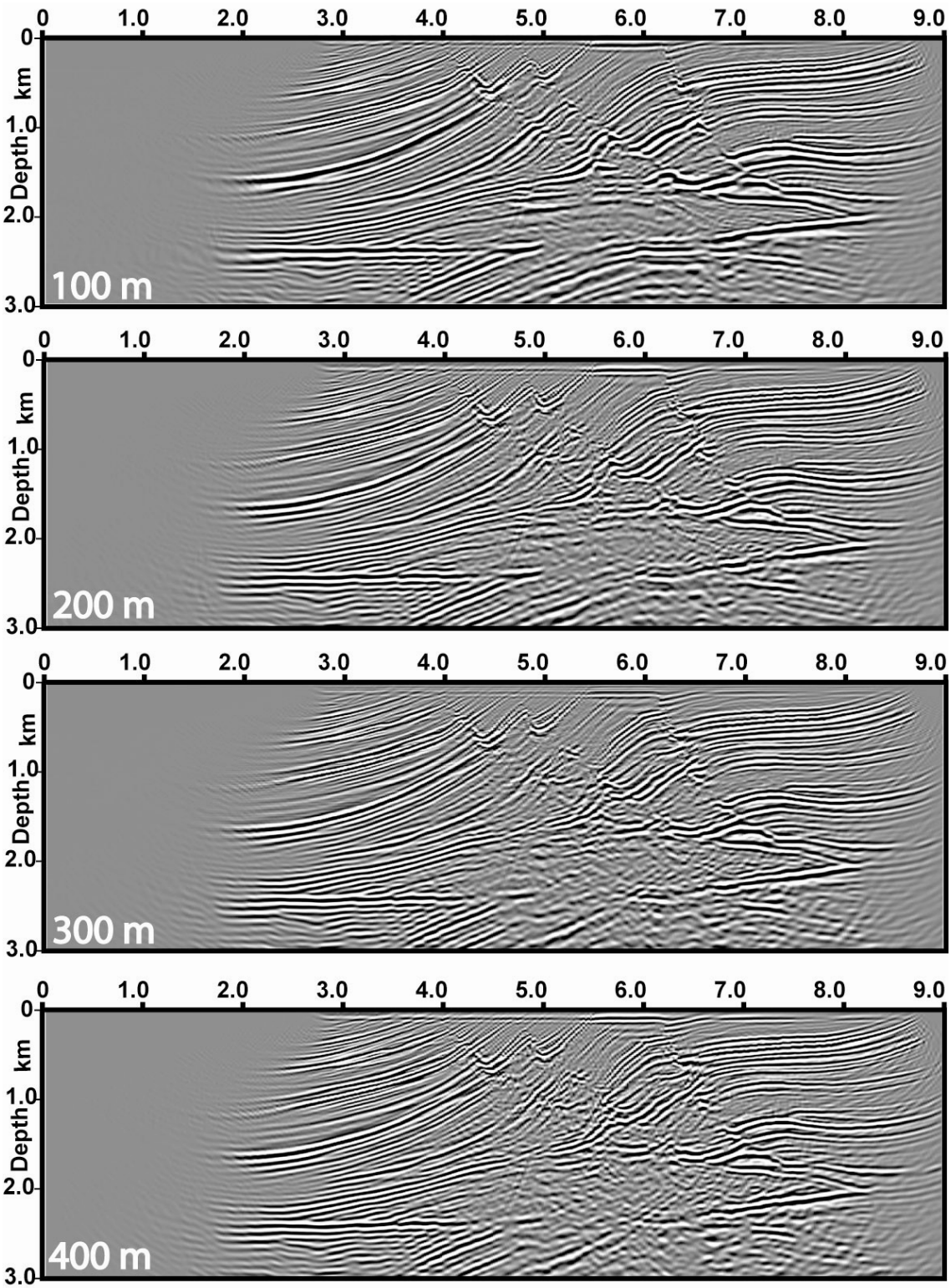


Figure 4.17: *Migrated image obtained for the smooth velocity models (Figure 4.16). Image quality and information content is decreasing with the increasing of the windows size.*

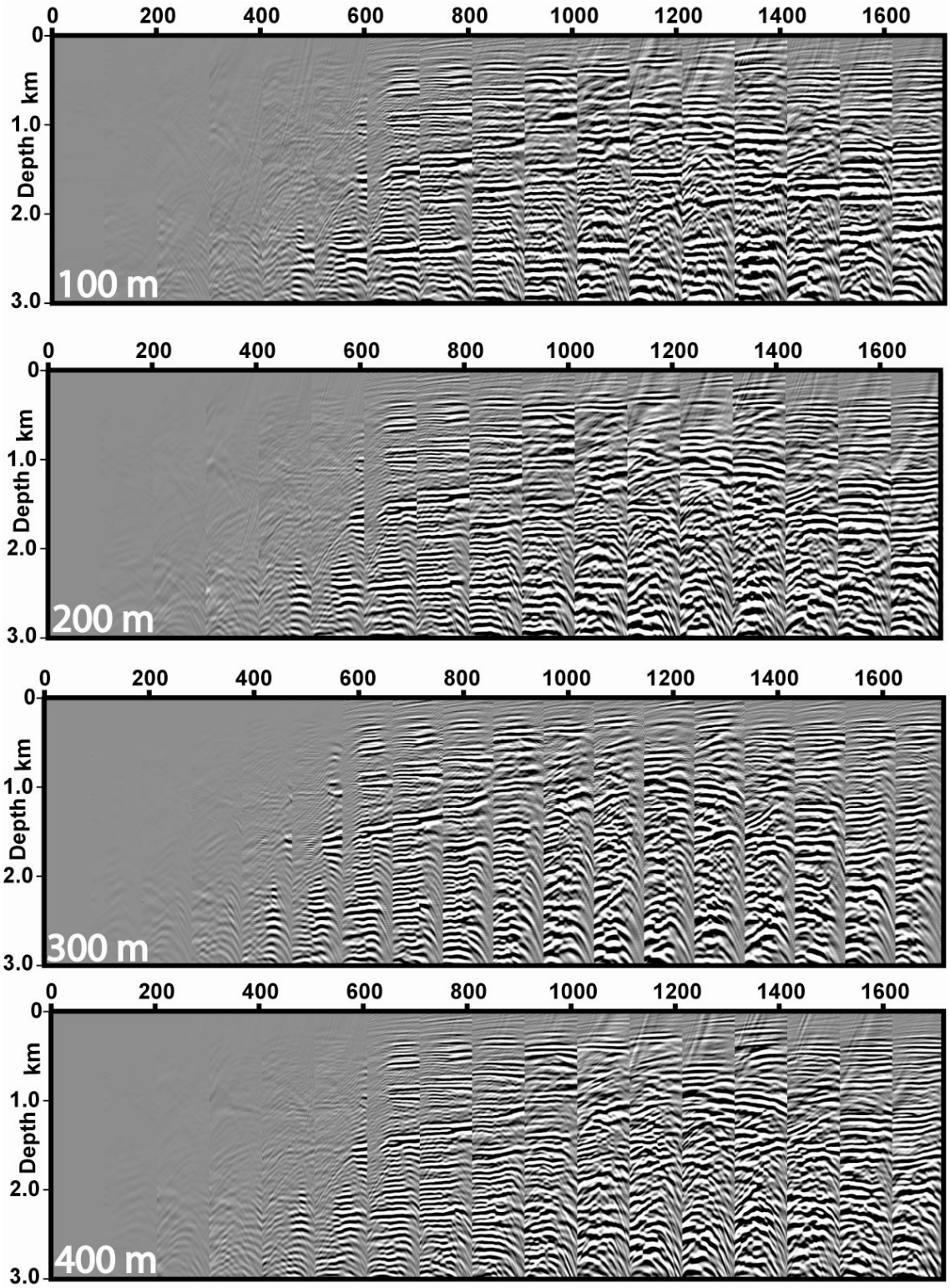


Figure 4.18: *Angle gather obtained for the velocity models (Figure 4.16). In the complex part of the velocity model gathers are not flat. These effects become severe with the increase of window size.*

In previous research work, it has been shown that even a coarse grid representation of velocity model can provide a good image (Versteeg, 1993) and it can speed iterative velocity estimation methods. Iterative methods rely on the horizontality of gathers for correct velocity. In figure 4.18 gathers are not horizontal in the complex part of the model smooth velocity. Hence fully relying on the horizontality of the gather in this complex part with a coarse grid may lead to an ambiguous result. Whereas coarse grid representation could be a good choice in the simple part the model. Hence consideration of grid size should be according to the complexity of the geological model. Moreover a mixed type of representation or variable length representation can be a good choice. In mixed type of representation one can do both coarse and fine grid representation, where simple part of geological model by coarse grid and complex part by fine grid representation. Whereas in variable length representation one can go from coarse to fine grid representation during the optimisation.

4.8 Conclusions

Representation of velocity model should be considered according to the assumed geological complexity, available data information and computational resources. Representing a salt body, where velocities are almost fixed can benefit from an adaptive and compact .representation like Voronoi diagrams. However finding the shape of salt body is in itself a challenge. On the other hand representation of foothill structure, where velocity variation adds the difficulty of finding the shape of the structure, a more flexible representation like grid should be preferred.

On the other hand, if much information about the sub-surface structure is known, then the geological representation proposed in this Chapter might also be a good choice. However, because we did not continue with the geological representation, we did not investigate its variable length extension. In the rest of the thesis we are going to use a grid representation.

Chapter 5
Ingredients of migration velocity analysis

In this chapter we discuss some geophysical technique that will be used to introduce domain knowledge in our algorithm in Chapter 6.

First we develop a robust objective function that measures the discrepancy to flatness of the coherency panels. These objective functions are a modification of the offset and angle domain Differential Semblance Function. These functions are nicely convex for a quite large range of high and low velocities, and are stable with respect to the frequency content and to the depth of the events.

We also develop a robust and automatic technique for gross velocity error estimaton using Residual MoveOut. We extract velocity ratio by picking the peaks of the envelope of the generalized Radon transform, of the angle or depth-offset gathers. Angle gathers appear to be less noisy than depth-offset gathers.

Finally we also extract dip information from the migrated image to reduce the velocity variation along the layers.

5.1 Introduction

In geologically complex regions, velocity inference from wave field continuation migrated images is vital for accurate seismic imaging. The purpose is to optimize, in a robust and automatic way, the seismic velocity field from migrated images, by using local or global optimization methods. Velocity estimation using migration is computationally expensive and requires the full exploitation of the seismic information.

Global optimization methods (*e.g.* Monte-Carlo or evolutionary algorithms) are expensive because they require the evaluation of many models in the search space. On

the other hand, local methods (*e.g.* gradient methods) were not much used because the calculation of the gradient is difficult and also two (Sava and Biondi, 2004) to four times (Shen 2004) more expensive than the migration itself. Global methods are able to cope with nonlinearities but did not come into practice because of their huge computational cost. We are trying to optimize the velocity using a global method: Evolutionary Algorithms (EAs) in order to make it converge quickly to the true velocity, first we need a good objective function that should be as convex as possible for a quite large range of high and low velocities, and also stable with respect to the frequency content and depth of the events. Second we need to estimate approximate velocity error, to improve the current velocity model(s) and third we want to introduce geological knowledge and informaion in order to accelerate the convergence of velocity models.

5.2 The Fitness Function

The choice of the objective function is crucial for automatic seismic velocity estimation. The least-squares seismic functions, semblance (S) and differential semblance (DS) are widely used. First we are analyzing the properties of the DS function for both offset and angle domains and then propose a modified differential semblance function. Then we will discuss the choice between S and DS.

After realizing the non-convex property of semblance function, Symes and Carazzone (1991) proposed the differential semblance (DS) objective function for velocity optimization. The DS function is also used for the optimization of waveform tomography in the time domain (Symes, 1994, Plessix et al. 2000), and in the frequency domain (Pratt and Symes 2002). DS combines the concepts from the least-squares seismic inversion, migration velocity analysis (flatness criteria of CIG) and travel time tomography. DS function measures the sum of difference of near trace energy. Many versions of DS function were proposed with small modifications. Here we analyze it for shot profile wave equation migration in both offset and angle domain.

5.2.1 Offset Domain Differential Semblance (ODS)

Recently Shen et al. (2003) presented a DS relation for the source receiver migration based on the Claerbout's principle of survey-sinking. This principle states that downward-continued source and receiver spatially coincide at zero time if the velocity model is correct. According to the Claerbout's principle the correct velocity model focuses all the energy at zero time and zero offset that is $I(x,z,h) = f(x,z).\delta(h)$, $\delta(h)$ being delta function, and hence $h.I(x,z,h) = 0$. Therefore Shen et al. (2004) proposed the following relation

$$ODS = \frac{1}{2}\int_x dx \int_h dh \left|hI\right|^2 \qquad (5.1)$$

$I(x,z,h)$ is the migrated image depending on abscissa x, depth z and offset h, which is minimum for the correct velocity model. The multiplication by offset wipe out wavefield at zero offset, differential semblances effectively removes focused energy from the migrated image. The remaining wavefield after removing all focused energy is a direct measure of model fitness.

We use data associated with a flat reflector (at depth 2000m) imbedded in a homogeneous velocity 3000m/s for the illustration of all DS functions. The offset domain differential semblance function using equation (5.1) for this flat reflector is shown in the Figure 5.1 for fine to coarse migration parameter settings

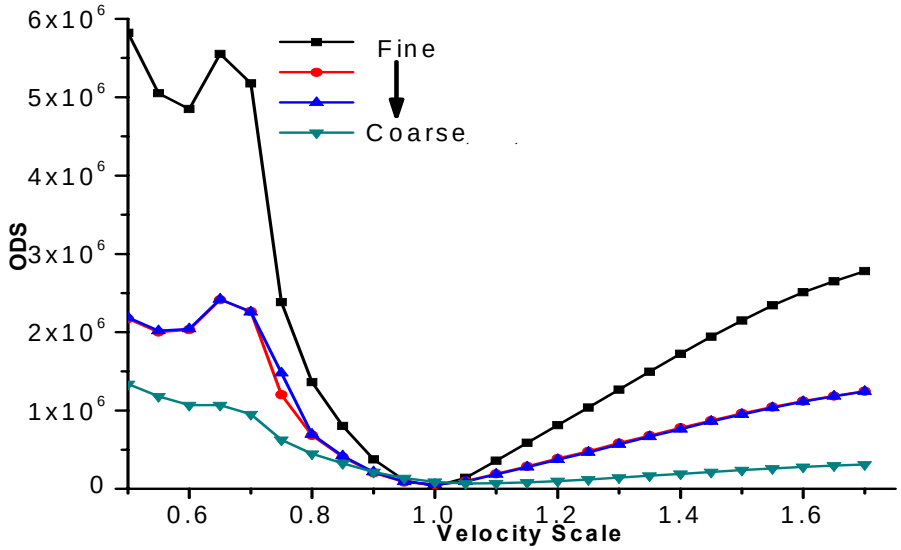


Figure 5.1: *DS plots with fine to coarse migration parameter setting for ODS function. the performance of ODS is degrading with coarse parameter.*

The abscissa corresponds to the ratio between the used velocity and the exact one. The black color curve represents a good parameter setting: frequency range 5 to 30Hz, all the shots, and 10 m propagation depth step. The red color curve differs by the frequency range, 10-20 Hz, the blue color curve by the propagation depth step, 40 m, and for the green color curve only half number of shots, is used. This parameter setting is also being used with MODS functions.

The ODS response is reasonably good in both high and low velocity regions, but the objective function is significantly diminished with the coarse migration parameter setting. Though ODS function were found more suitable than the ADS (angle domain differential semblance) for velocity optimization with the gradient method (Shen et al., 2004). We noticed following shortcomings of this *ODS* function:

1. *ODS* function has small sensitivity near the true velocity model,
2. its performance is significantly affected by the parameter setting, and
3. it seems very sensitive to the data and migration amplitudes since it was not scaled by any factor like $1/\int_z dz \int_h I^2 dh$.

Therefore we tried to improve this objective function.

Chauris and Noble (1998) proposed to introduce the scaling factor $1/\int_z dz \int_h I^2 dh$, in the ODS relation. We got the following relation (equation 5.2) after introducing the scaling factor.

$$ODS/SF = \int_x dx \frac{\left[\int_z dz \int_h dh |hI|^2\right]}{\left[\int_z dz \int_h I^2 dh\right]} \quad (5.2)$$

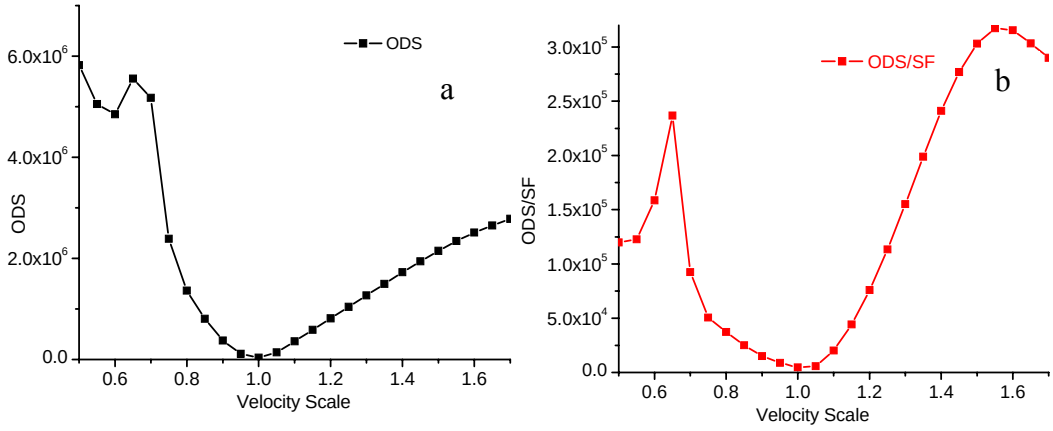


Figure 5.2: *For a simple homogeneous flat reflector the comparison of response of (a) ODS and (b) ODS/SF function. ODS/SF function is more sensitive for a large velocity error and less sensitive for small velocity error.*

The addition of denominator $\int_z dz \int_h I^2 dh$ reduces the effect of data and migration amplitude. Equation (5.1) is used by Mulder and Kroode (2002) for velocity optimization. They used true amplitude ray tracing migration and suppressed the noise by preprocessing the data, therefore their DS function does not need to have a denominator. Figure 5.2 (b) shows the improvement obtained after introducing scaling factor in the ODS function. ODS/SF function seems more sensitive for a large velocity error. However it has a less sensitivity for small velocity error. To further improve the ODS/SF objective function we propose a modified offset domain differential semblance function (MODS).

5.2.2 Modified Offset Domain Differential Semblance function

In this objective function we first introduce a new scaling factor $1/\int_z dz \int_h z^2 I^2 dh$ function to reduce the effect of data noise. Second, since amplitude decays with depth, the result will be more sensitive to shallow events than to deep events. To compensate this effect, first we introduce a depth factor z and obtain following relation

$$MODS = ZODS / SF = \int_x dx \frac{\left[\int_z dz \int_h dh \left|zhI\right|^2\right]}{\left[\int_z dz \int_h z^2 I^2 dh\right]} \qquad (5.3)$$

Response of the modified offset domain differential semblance (MODS) (equation (5.3)) function for a flat reflector is shown in the Figure 5.3 for a flat reflector

Comparing the Figures 5.1 and 5.3 for the different parameter settings shows that the MODS is performing very well in both low and high velocity regions. In contrast with the ODS function, there is no significant change on MODS function with reduced frequency, reduced number of shots and larger propagation depth step. Even there is no significant change in the valley of attraction. The response of the MODS function is little affected by the detracting the parameter setting (Figure 5.3). MODS function shows better sensitivity in the high and low velocity zone as compared to the ODS. It seems more stable than the ODS functions with respect to migration parameter setting. As the velocity scaling increases the proposed MODS function also increases which is very important for velocity optimization. We think that depth factor compensates the spherical divergence (associated with the used velocity model) and that scaling factor reduces the effect of data noise. Because the MODS function is not much affected by the migration parameter setting, we may use smaller frequencies and less shots, which is a key point to speed up the velocity optimization process.

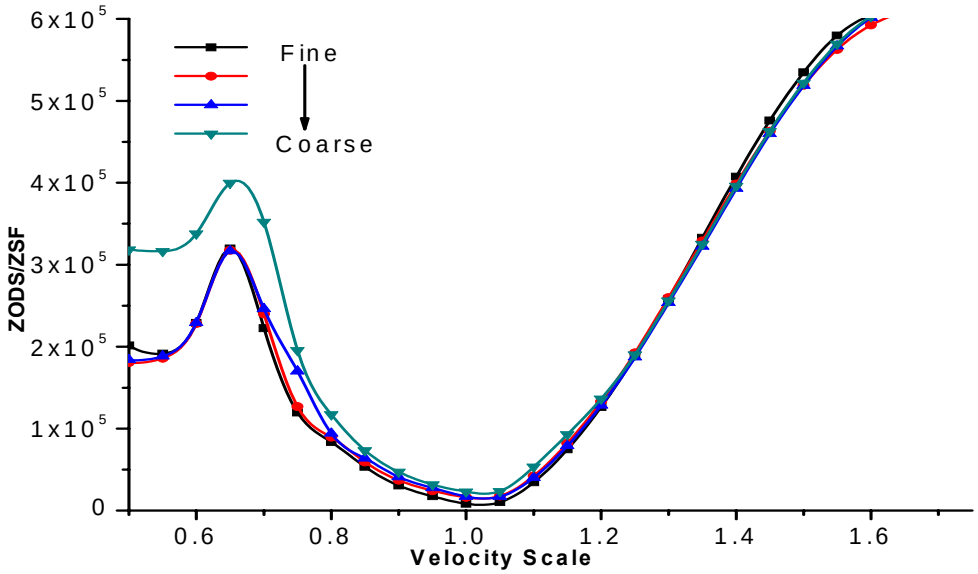


Figure 5.3: *MODS response for fine to coarse migration parameter settings. MODS function shows better sensitivity with high and low velocity error and is less affected by migration parameter setting.*

During our experiment we noticed that these DS functions are much affected by subsurface offset parameter. Increasing the subsurface offset also increases the computation cost. Hence initially we put maximum subsurface offset value 500 meters, and we observe that the DS function is behaving very well for low velocity factors whereas it starts to decrease for higher than 20% velocity factor. Therefore, we carried out an experiment for different subsurface offset parameter setting. In Figure (5.4) the responses of ODS (Figure 5.4a) and MODS (Figure 5.4b) functions for different subsurface offset parameter settings are shown.

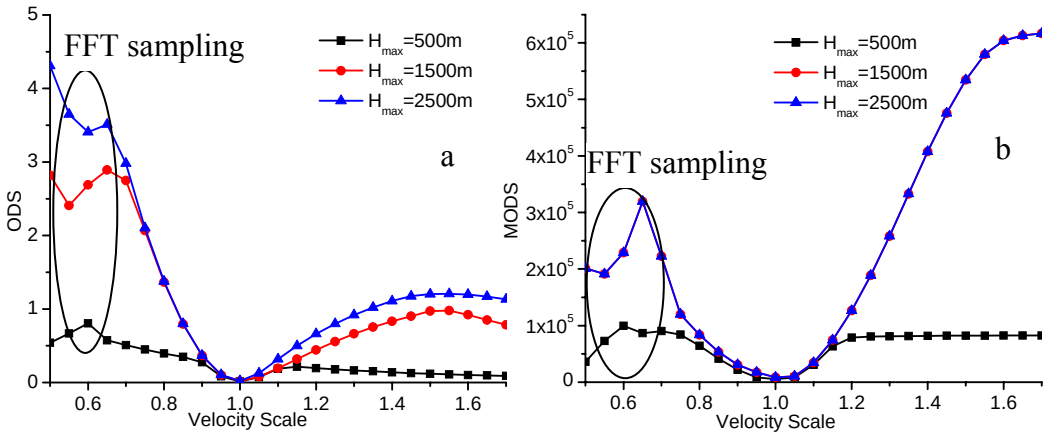


Figure 5.4: ***(a)*** *ODS.* ***(b)*** *MODS with depth offset 2500m (blue curve), 1500m (red curve) and 500m (gray curve). MODS function is less affected by sub-surface offset parameter setting. A fine fast Fourier transform (FFT) sampling is required for very low velocity scaling.*

We notice that increasing the maximum depth offset improves the response of DS function for larger velocity factor. Bu comparing the responses of these DS functions with respect to different maximum depth offset settings it is found that MODS is most stable. The response at 1500 and 2500 is similar in MODS while it is different in ODS. The performance of ODS is drastically affected by different maximum depth offsets. The zigzag curve in both ODS and MODS for very low velocity is due to the periodicity of the Fourier transform. This should be avoided during migration.

Now we also analyze the response of DS function in angle gather and proposing a modified angle domain differential semblance function.

5.2.3 Angle Domain Differential Semblance (ADS)

Sava and Fomel (2003) presented a simple method for transforming offset domain common image gathers into angle domain common image gathers (ADCIGs) by a slant stack transformation (Schultz and Claerbout, 1978) applied to each offset domain CIGs as given below:

$Ix\,(z, x, \gamma) = \text{SlantStack}\,[\,I\,(z, x, xh)\,]$ (5.4)

where γ is the aperture angle of the reflection. This transformation from offset domain CIG to ADCIG is based on the following relationship between the aperture angle and the slope,$\partial z/\partial xh$, measured in image space:

$-z/x_h = \tan(-k_{xh} / k_z)$; (5.5)

where k_{xh} and k_z are respectively the half-offset wavenumber and the vertical wavenumber. The relationship between tan and the wavenumbers suggests that the transformation to ADCIGs can be accomplished in the Fourier domain by a simple radial-trace transform (Sava and Fomel, 2003).

The differential semblance in angle domain equivalent to offset domain (equation 5.1) can be posed as

$$\text{ADS}=\int_x\int_z dz\int_\theta d\theta\left(\partial_\theta I\right)^2 ; \qquad (5.6)$$

where θ is the angle. ADS function measure the sum of difference of near trace energy. The response of ADS function for a homogenous flat reflector is shown in the Figure (5.5a) and for a Marmousi model in Figure (5.5b). For a homogeneous flat reflector its response is pretty good for a large velocity error whereas for a Marmousi model if the velocity error is more than ±25% the response of ADS function is not a consistent. The ADS response significantly deteriorate at complex part of Marmousi model (Figure 5.6).

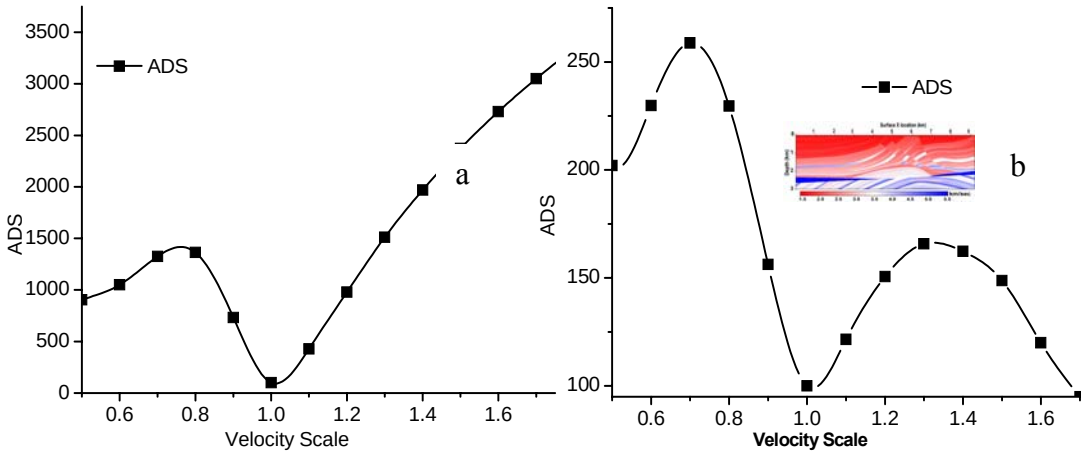


Figure 5.5: *Angle domain differential semblance function (a) response for a homogeneous velocity model is consistent for large velocity error (b) response for a Marmousi model is consistent upto ±25%.*

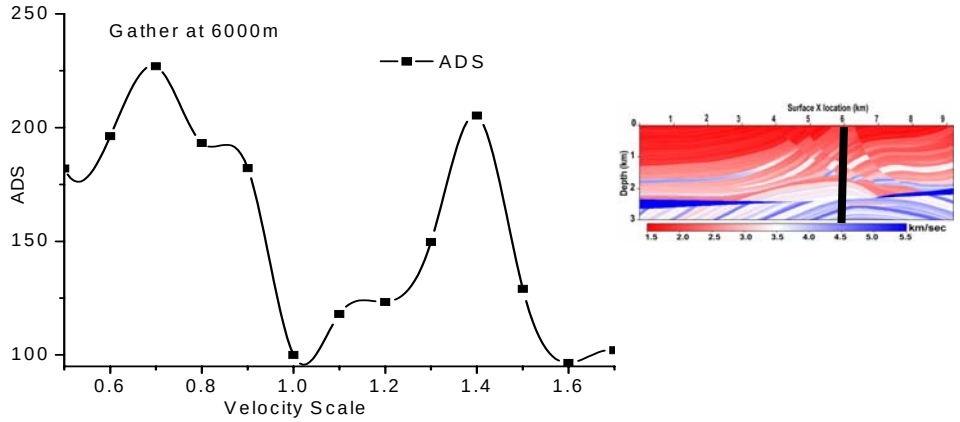


Figure 5.6: *Angle domain differential semblance function response of Marmousi model at 6000 m (most complex part of Marmousi model). The response is inconsistent with velocity error.*

ADS function is also affected by the data and noise. To improve ADS function we divide it by a scaling factor (SF) similar to Chauris and Noble (1998). We obtained following relation.

$$ADS/SF = \int_x dx \frac{\left[\int_z dz \int_\theta \left(\partial_\theta I\right)^2 d\theta\right]}{\left[\int_z dz \int_\theta I^2 d\theta\right]} \quad (5.7)$$

The SF ($\int_z dz \int_\theta I^2 d\theta$) is a square of the image.

The response for this function for a homogeneous flat reflector is shown in Figure (5.7a). The ADS/SF function for a homogeneous flat reflector shows significant improvement for low velocity error while no improvement for high velocity error. On the other hand response of ADS/SF function for a Marmousi model (Figure 5.7 b) shows significant improvement in both low and high velocity region. Moreover in the high velocity region it becomes sensitive up to +40% and after that it is flat. As compare to the ADS function ADS/SF function perform better for both low and high velocity model. Moreover ADS/SF performance is also better at 6000m of Marmousi model (5.8) for low velocity error whereas for high velocity error only a small improvement. We can conclude that improvement in ADS/SF function is because of SF.

This function is also affected by migration sampling (Figure 5.9a). However the influence of migration sampling is small as compared to ODS (Figure 5.1). This function is also affected by subsurface offset parameter setting (Figure 5.9b). Overall response of this objective function seems more stable than the ADS function.

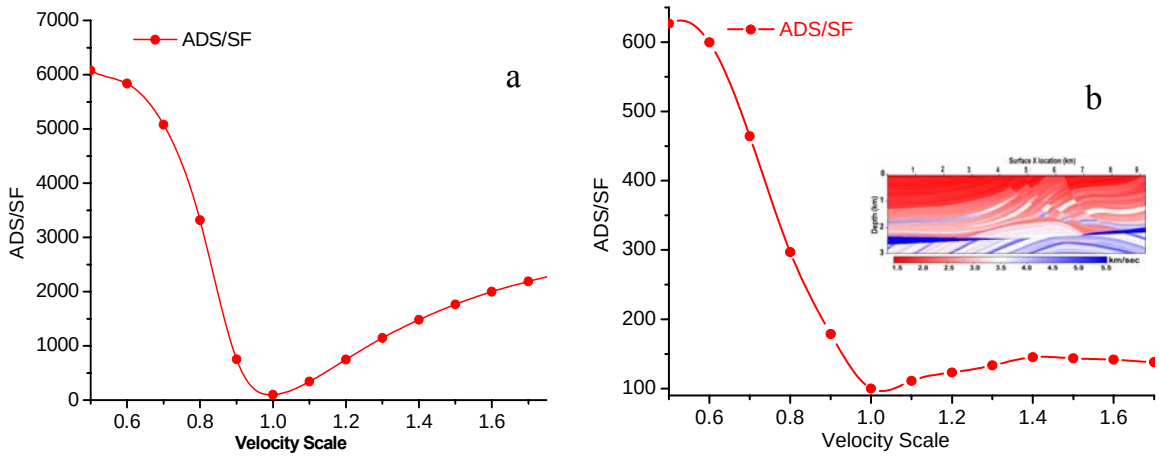


Figure 5.7: *ADS/SF function response (a) for a homogeneous velocity model it is consistent for large velocity error (b) the response for the whole Marmousi model is consistent upto +40% and after that flat.*

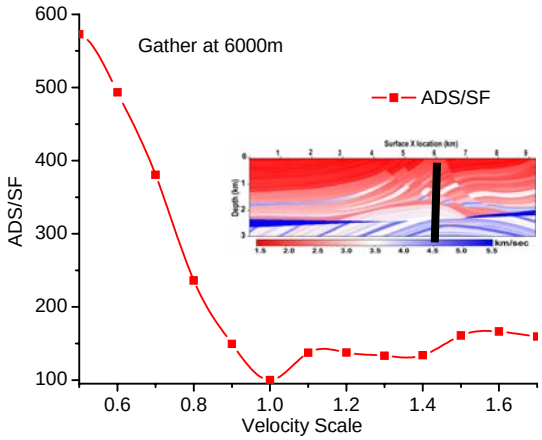


Figure 5.8: *ADS/SF function response of Marmousi model at 6000 m. The response is consistent for low velocity error whereas for high velocity error it is little consistent.*

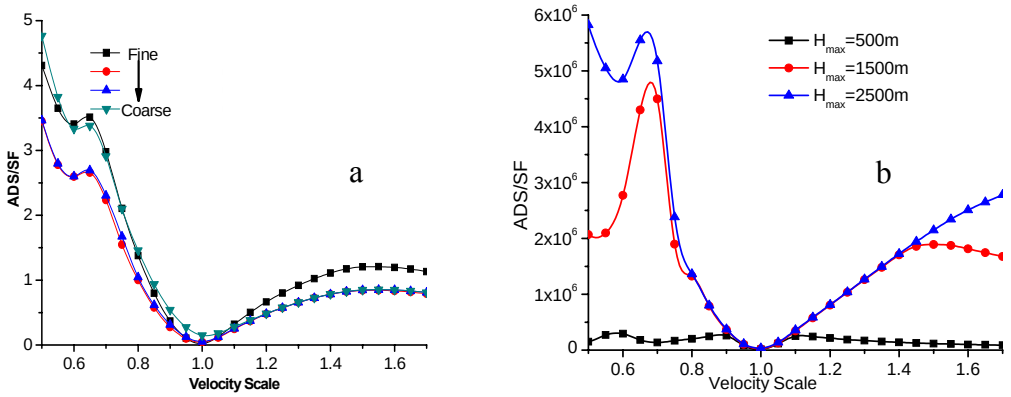


Figure 5.9 ***(a)*** *ADS/SF function for fine to coarse migration parameter settings. There is a small effect of migration parameter setting.* ***(b)*** *Similar to ODS function ADS/SF function is also affected by subsurface parameter setting.*

To further improve this objective function we propose to use a modified angle domain differential semblance (MADS) function. The improvements made in this function are similar to the MODS function.

5.2.4 Modified Angle Domain Differential Semblance (MADS)

We are also interested to see the effect of depth (z) factor on angle gather similar to offset domain. We introduced the z factor in ADS function and divide it by a SF, get the following relation.

$$MADS = (ZADS / SF) = \int_x dx \frac{\left[\int_z dz \int_\theta z\left(\partial_\theta I\right)^2 d\theta\right]}{\left[\int_z dz \int_\theta I^2 d\theta\right]} \qquad (5.8)$$

We introduce z factor to compensate the effect of spherical divergence (associated with the used velocity model) and scaling factor to reduce the effect of data noise.

The response of MADS objective function for a homogeneous flat reflector is shown in the Figure (5.10 a) and for whole Marmousi model (Figure 5.10b) and at the most complicated region of the Marmousi model (at X=6000m) is shown in the Figure 5.11). These response are obtained when every second shots and frequenct range 10-25Hz used.

This modified function is nicely convex for a large range of high and low velocities, and it is stable with respect to migration parameter setting. The modified function is not much affected by migration parameter settings, thus smaller frequencies and lesser shots can be used to speed up the velocity optimization process.

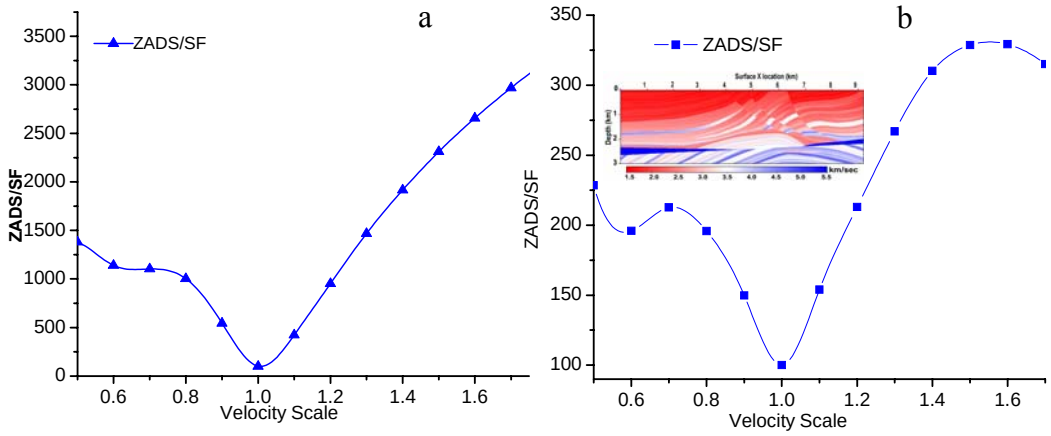


Figure 5.10: *MADS function performance (a) for a homogeneous velocity model it is consistent for both low and high velocity error (b) for a Marmousi model also it performance is pretty consistent for both low and high velocity error.*

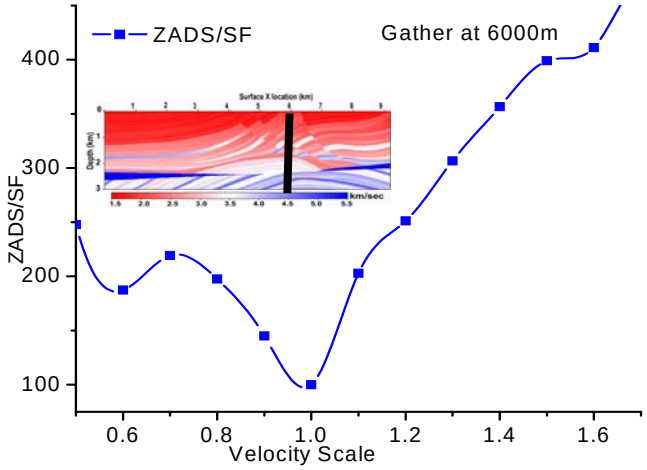


Figure 5.11: *MADS function response of Marmousi model at 6000 m. The performance is consistent for low velocity and high velocity error.*

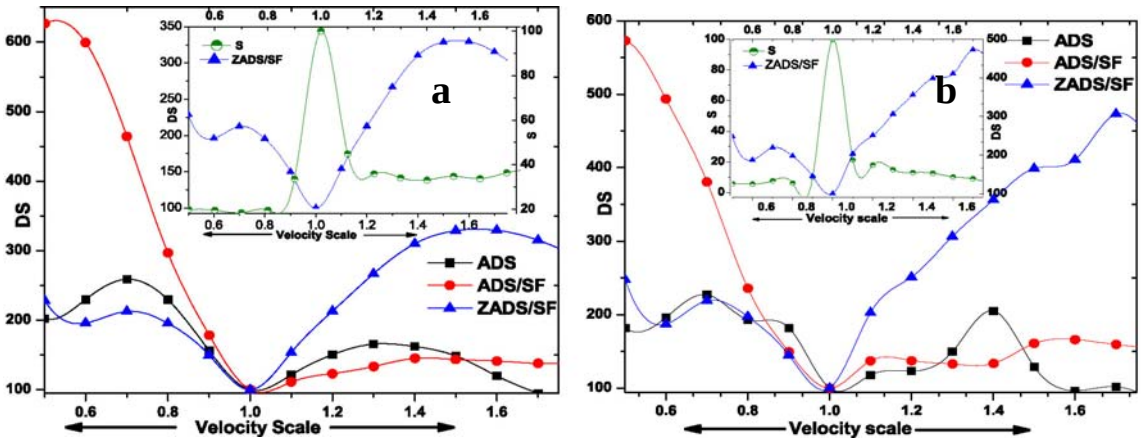


Figure 5.12: ***(a)*** *The response of different DS function for Marmousi model. The combination of semblance (S) and modified differential semblance (ZADS/SF) are in subplot. The semblance function has narrow and steep valley of attraction near to the true velocity model whereas modified differential semblance function has a wide and gentle valley of attraction.* ***(b)*** *The response of different DS function for Marmousi model at X=6000m. Modified DS function response is better than the others DS function even in very complex part.*

All DS functions and a combination of MADS and semblance function are shown in the Figure (5.12) for whole Marmousi model (5.12a) and at 6000m of Marmousi model (5.12b) for the purpose of comparison. These Figures clearly demonstrate that proposed objective function is performing quite well for both low and high velocity error and also in geologically complex region. Whereas ADS function is unstable for large velocity error and geologically complex region. Hence combination of MADS and semblance function seems to be a good choice for velocity analysis.

5.2.5 Differential Semblance or Semblance ? Both!

It has been suggested by Chauris and Noble (1998) that Differential semblance function could be used in the first few iteration and subsequently followed by semblance function. Since in the beginning of optimization, the initial models are far from the true models, at that time MADS function can help in the convergence because of its strong sensitivity for far velocity models. Whereas when the models are near to the true model the semblance function can help in the fast convergence because of its strong sensitivity for near

velocity models. Multi-objective evolutionary algorithms provide a unique opportunity to use both objective functions independently.

We have optimized a synthetic foothill velocity model (Figure 4.13) using the MODS and Semblance function with multi-objective evolutionary algorithms (MOEA) (Singh et al., 2005). The optimization process took a large number of iterations and also needed good parameter setting with ODS function. Whereas we are able to optimize the velocity model in few iteration and only using the half the number of shots and small frequency bandwidth with the MODS function (Figure 4.14).

Automatic gross velocity error estimation

5.3 Introduction of RMO

It had been found that the velocity estimation using migration method is computationally very intensive. One of the important reasons for this is lack of the exploitation of information that is present in the gather. Therefore we need a robust approach which could inform us about the approximate amount and direction of change required to update the velocity model.

Recently Sava (2003) made an attempt to get this information from a migrated image using Stolt residual migration. Stolt residual migration is applied in prestack domain as a velocity independent process for gross velocity error analysis from migrated image. The main advantage of residual prestack migration is its interpretative and structural dependencies. Residual prestack migration also reduces the effect of image dispersal between events that are imaged at the same physical location but with different aperture angle. However the main difficulty with residual migration is its computational complexity, interpretative and structural dependent nature which makes it very challenging to automate. It has been found that this technique may not work with large velocity variations and extreme complexity (Sava, 2003). Another effective and robust approach of gross velocity error analysis from migrated image is Residual Move Out

(RMO) analysis. Measuring the inconsistency in image amounts to compute the semblance scan as a function of one RMO parameter, and then pick the maximum of semblance scan (Biondi and Symes, 2004). The property of both methods is analyzed by Biondi and he showed that nonlinear RMO function for a flat event is equivalent to residual migration of flat events (Biondi 2003).

Methods have been developed to perform this analysis on both offset and angle domain CIGs for the Kirchhoff and wave equation migration. The offset-domain RMO analysis methods using Kirchhoff migration were developed first by Al-Yahya (1989).This technique was further developed by Lee and Zhang (1992), Lafond and Levander (1993), Liu and Bleistein (1995), Yan et al. (2001). The geometrical and mathematical characteristics of RMO for the prestack Kirchhoff migration were appraised for velocity by Zhu et al. (1998). The angle domain RMO analysis methods have also been proposed (Jiao et al., 2002). RMO properties for wave equation migration were analyzed by (Biondi and Symes, 2004; Shen P. 2004; Bartana et al., 2006; Sava and Fomel 2006).

5.3.1 RMO and Radon Transform

Here we are presenting an automatic approach of gross velocity error analysis using RMO technique for a source receiver wave equation migration. Our approach is tested on a variety of models ranked from simple to complex and ranging from small to large velocity contrast and it is found to be good. First we discuss the RMO function and their Radon transform for both offset and angle gather of a shot profile wave equation migrated image. Than the process of envelope creation and results are discussed.

RMO curvature analysis is based on the concept of generation of flat gather in CIGs by PSDM for correct velocity model regardless of the structure. Regardless of the domain in which the prestack partial images are defined, the RMO function is usually parameterized by single parameter (The ratio of migration velocity and true velocity). Inaccurate velocity will cause the moveout artifacts on the migrated image and the shape of the artifacts depends on the velocity contrast. The low velocity estimation will produce smile shape and for high velocity estimation frown. The depth offset (Shen, 2004) and angle gather (Biondi, 2003) could be presented by following relations.

$$\frac{h^2}{(\gamma^2-1)Z^2}+\frac{z^2}{\gamma^2 Z^2}=1 \qquad (5.9)$$

$$z^2=\gamma^2 Z^2+\frac{\gamma^2 Z^2\tan\varphi^2}{(\gamma^2-1)} \qquad (5.10)$$

Here h denotes subsurface offset z migration depth, Z true depth, and φ reflection angle. First we did the Radon transform of the gathers. The Radon transform of a gather converts a curve detection problem in image space to a peak detection problem in parameter space. The peaks are the parameter of a reflection, which inform about the background velocity model. The radon transform of the depth offset gathers had been used for the surface multiple removal (Duquet and Marfurt, 1999). In the above equation at γ=1, we obtained z=Z for all angle or offset. Introducing a variable ζ=γZ in both offset and angle RMO function and their discrete radon transform could be written as

$$R(\varsigma,\gamma)\approx\left(\sum_h G(\varsigma^2-\frac{\gamma^2 h^2}{\gamma^2-1})\right)^{0.5} \qquad (5.11)$$

$$R(\varsigma,\gamma)\approx\left(\sum_h G(\varsigma^2+\frac{(\gamma^2-1)\tan^2\theta}{\gamma^2})\right)^{0.5} \qquad (5.12)$$

Process of Radon transforms for offset and angle gathers are shown in the Figures 5.16 and 5.17 respectively.

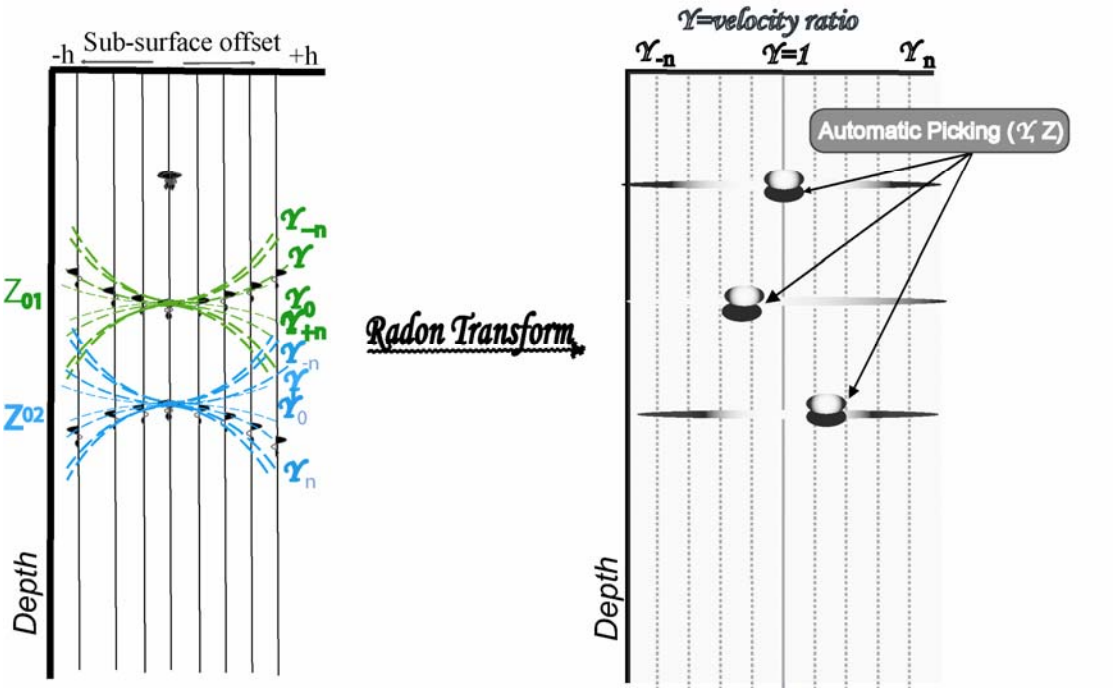


Figure 5.15: P*rocess of creating γ panel using Radon transform from the offset* gather

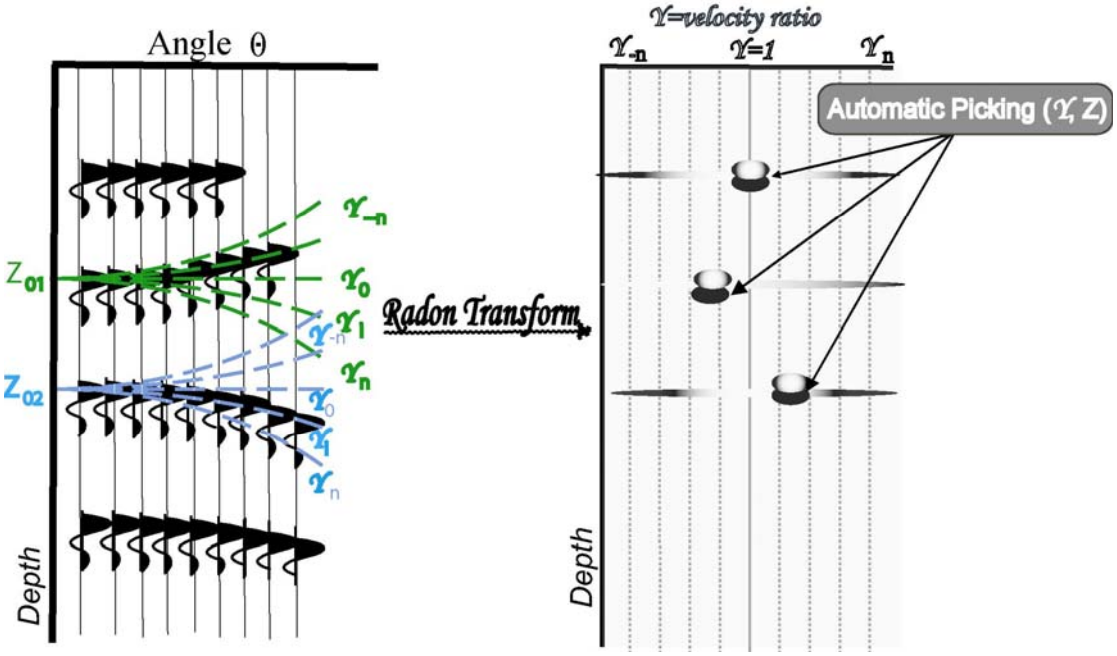


Figure 5.16: *Process of creating γ panel using Radon transform from the angle gather*

Figure 5.17: *Four layer horizontal velocity model.*

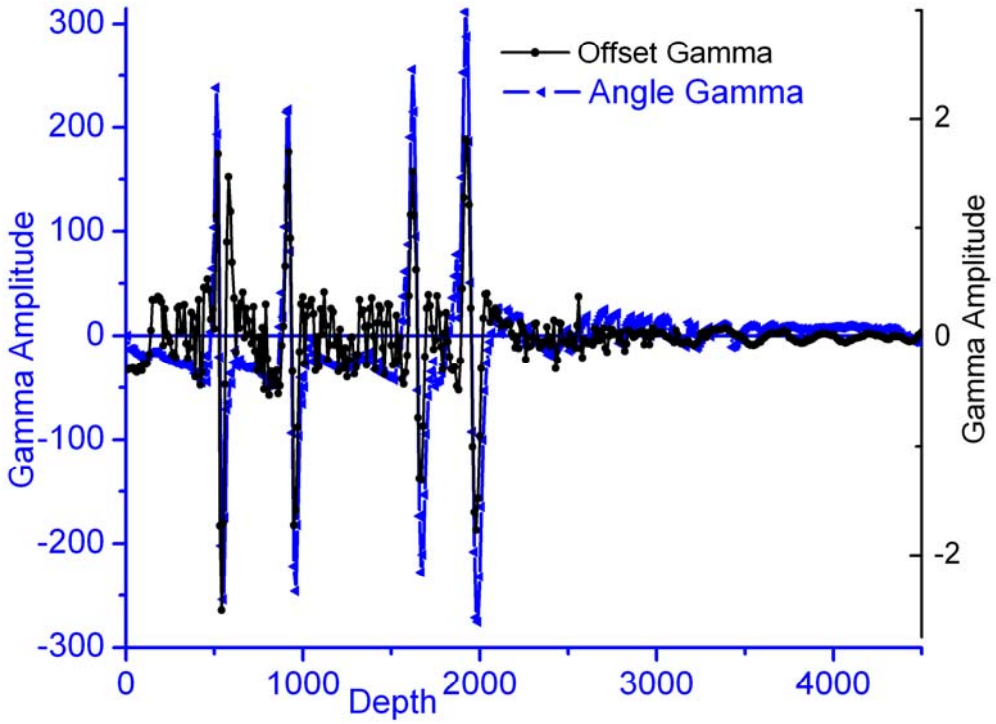


Figure 5.18: *offset and angle gathers γ value for four layer horizontal velocity model. The offset gamma amplitude is somewhat fluctuating.*

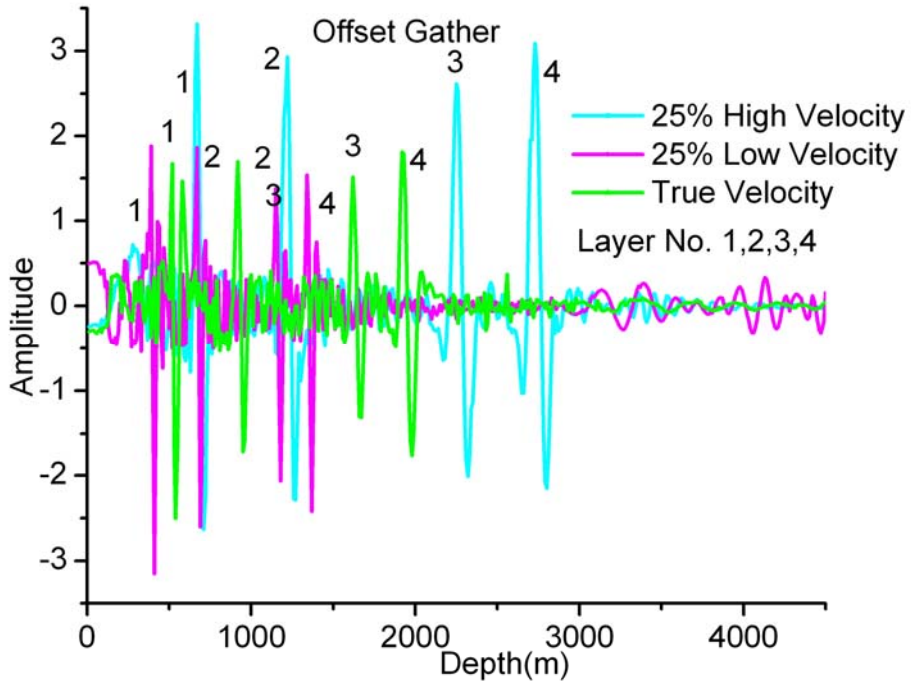


Figure 5.19: *The maximum γ value (γ amplitude) of COG Radon transform for each depth of a four layer velocity model for true 25% high and 25% low velocity.*

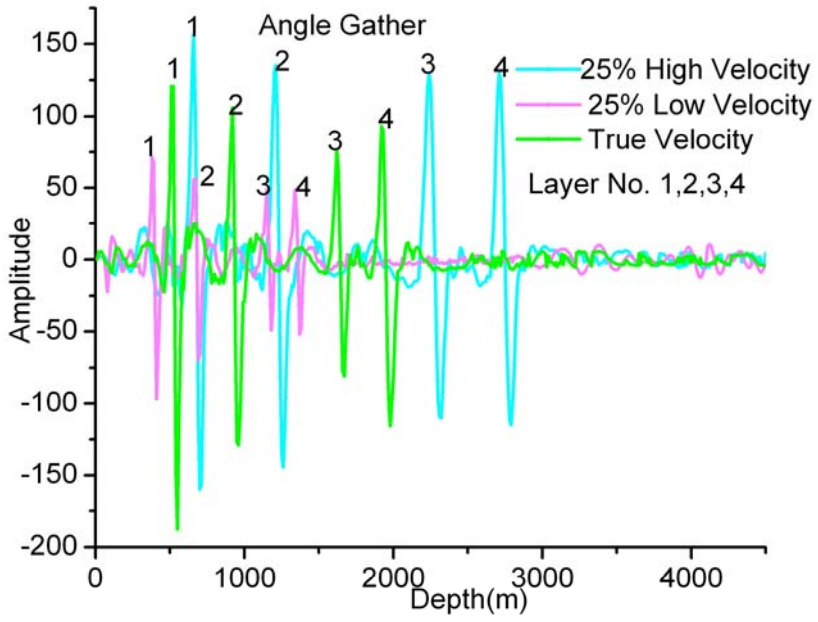


Figure 5.20: *The maximum γ value (γ amplitude) of CAG Radon transform for each depth of a four layer velocity model for true 25% high and 25% low velocity*

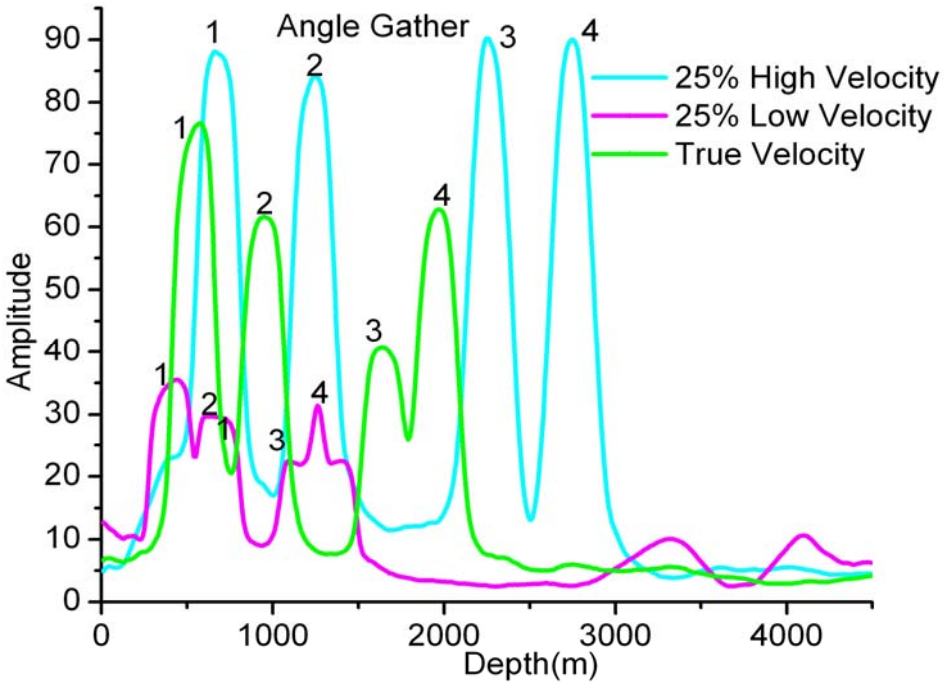


Figure 5.21: *The envelop of the CAG gathers obtained by equation (5.13).*

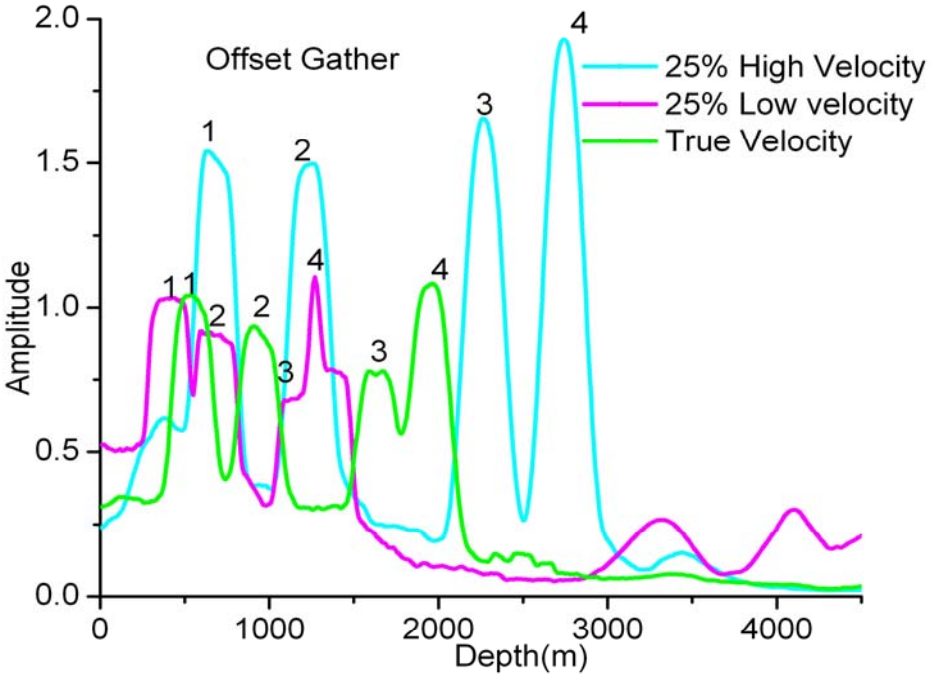


Figure 5.22: *Envelop of the COG gathers obtained by equation (5.13).*

Figure 5.19 and 5.20 demonstrate the peak amplitudes of the γ panels at each depth interval for four layer velocity model (Figure 5.17). After getting the γ panels the next challenge is to exactly identify the peaks. To exactly identify the peaks we did following steps.

1. We created envelops of the radon transformed signal using the Hilbert transform. The procedure is being described in the next section.
2. We applied a liner low-pass filtering.
3. Cloud separation scheme for pick identification

5.3.2 Envelop

Picking a burst on a zero-mean signal is difficult because the shape of the signal is (it a priori) unknown. For instance, picking symmetrical and anti-symmetrical signals are very different tasks. To solve that problem, the usual technique consists in computing the envelope *e(z)* of signal *f(z)* defined by following equation

$$e(z) = \sqrt{f(z)^2 + h(z)^2} \qquad (5.13)$$

with *h(z)* being the Hilbert transform of *f(z),* that is, * denoting convolution,

$$h(z) = f(z) * \frac{-1}{\pi z} = \int_{-\infty}^{+\infty} f(z-x).\frac{-1}{\pi x}dx \quad (5.14)$$

The Fourier transform of *h(z)* has the same amplitude than that of *f(z)*, but the phase is shifted by *π/4*. The convolutive method is computationally more efficient than the Fourier method if the impulse response is sufficiently short. In our case, very low frequencies are absent in *f(z)* , and thus we may apodize the hyperbola *-1/(πz)* with some [*-A,+A*] supported function *a(z)*,

$$h(z) = \int_{Z-A}^{Z+A} f(z-x).\frac{-a(x)}{\pi x}dx \qquad (5.15)$$

For sake of simplicity, we have chosen the parabola *a(z) =1 - z2/A2* truncated in [*-A,+A*]. To compute numerically equation (5.14) , we have the samples f_i of *f(z)*, and we need to sample hyperbola *-1/(π z)* . Unfortunately, direct sampling is very bad near the asymptote. Therefore, the method we used to get the value h_i of h*(z)* at $z=z_i$ is schematically as follows:

1. compute the derivatives of $f(z)$ at $z = z_i$ with Holberg numerical schemes up to the eighth order (Léger, M., 2000)
2. build Taylor expansion $T_1(z)$ of $f(z)$ around $z = z_i$,
3. multiply by the apodized hyperbola, which gives a new Taylor expansion $T_2(z)$,
4. and integrate $T_2(z)$ to get h_i.

This method is twice as expensive as the convolution by the standard discretized hyperbola, the length of the operator being the same (A =30), but the numerical error is maintained below 40dB between 4% and 34% of the sampling frequency instead of 4% to 10%.

5.3.4 Results and Discussion

We are presenting our results for a horizontal four layer velocity model (Figure 5.17). The maximum γ value of a velocity model for the γ = 1, 0.75 and 1.25 is shown in the Figure (5.19, 5.20). However comparing the Figures 5.19 and 5.20, we could see that the amplitude of COG radon transform is noisy compared to the CAG. Although CAG and COG should give the same response. The CAG is derived by the slat stack of the COG. We think that because of the slant stacking (summation along offset) the noise get suppressed. Hence CAG seems more suitable for the velocity analysis as compared to COG. The peaks of the layers are very prominent in CAG. The response of each layer has positive as well as strong negative amplitude because of phase change. The positions of the layer lie in between the positive and negative signal. Once we get the γ panel either for the offset or for angle domain the most important challenge is to find the peaks parameters $R(\varsigma, \gamma)$.

Envelops of the Figures (5.19) and (5.20) is shown in the Figures 5.21 and 5.22 respectively. The parameter associated with the peak amplitudes of each curve are the desired output which could be easily obtained by peak detection process. The low amplitudes peaks, which remain even after the linear smoothing can be suppressed by applying a threshold. In both Figures the low amplitude peaks associated with very low velocity signal may be because of the periodicity. The periodicity is obtained by the Fourier transformed applied to the data for low velocity. The time data should be pulled so that the periodic of seismic event can not be imaged in the considered depth region.

Since we are interested in the direction and approximate amount of change that need to be applied to the velocity model at particular place to improve the image. These peaks could be directly used in wave equation migration for velocity analysis (WEMVA). Because WEMVA estimates the perturbation of seismic velocity model from the perturbation of seismic image (Biondi and Sava, 1999; Shen and Symes 2003).

We have developed a robust and automatic technique for gross velocity error analysis using residual moveout. We get the velocity ratio by picking the peaks of the envelope of the generalized Radon transform of the residual moveout function of both angle and depth-offset gathers. Picking the envelope of the signal is more robust than picking the signal itself. Angle gathers appeared to be less noisy than depth-offset gathers. Except the gathers, migrated image also have a lot of information about the velocity. For example it is well known that velocity variation is very small or negligible along a layer. In the next section we are presenting a technique to extract the structural trend and dip of the layers.

5.4 Structural trends and dip

The structural trend of a geological model also provides significant information and has been used differently for velocity estimation. The basic assumption (see Delprat-Jannaud and Lailly 1992) is that the velocity follows structural dips or some other known trends and it can hence be incorporated as a term into the objective function (Delprad-Jannaut and Lailly 1992), Kaipio et al.,(1999) suggested using a prior structural information to create conditional covariance matrices. This has also been used as a model regularization operator as well as a preconditioner (van Trier,1990; Claerbout, 1992; Clapp et al.,2004). It has been found that this information significantly improves the convergence speed. This dip could be extracted from a migrated image using prediction error filter (Claerbout 1998a) or by geostatistical approch of covarience analysis (Clapp et al.,2004).

5.4.1 Sobel operator

Here, we are not using the dip information as objective finction, regularization or preconditioner. Our goal is to extract the dip inforrmation at each point of the migrated image and use this dip information to reduce the variation of velocity along the dip

direction. For identifying the reflection boundary and measuring the dip of the reflection, we have used the Sobel edge dectection tool (usually used in image processing). Edges characterize the boundaries in image. Sobel operator gives the direction and magnitude of these edges. It is described in the next section.

The Sobel operator performs a 2-D spatial gradient measurement on an image and so emphasizes the regions of high spatial frequency that correspond to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input gray scale image. The Sobel Edge Detector uses a simple *convolution kernel* to create a component of *gradient magnitudes*. Mathematically convolution of kernel *K* to image I can be represented as:

$$N(x,z) = \sum_{i=-2}^{2}\sum_{j=-2}^{2} K(j,i)I(x-j,y-i) \;; \qquad (1a)$$

Sobel Edge Detector uses two convolution kernels, one to detect vertical component K_x and another to detect horizontal component K_y

$$K_x = \begin{matrix} 1 & 2 & 0 & -2 & -1 \\ 4 & 8 & 0 & -8 & -4 \\ 6 & 12 & 0 & -12 & -6 \\ 4 & 8 & 0 & -8 & -4 \\ 1 & 2 & 0 & -2 & -1 \end{matrix} \;; \quad K_y = \begin{matrix} -1 & -4 & -6 & -4 & -1 \\ -2 & -8 & -12 & -8 & -4 \\ 0 & 0 & 0 & 0 & 0 \\ 2 & 8 & 12 & 8 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{matrix} \;; \qquad (2a)$$

Therefore we have the gradient magnitude (3a) and direction (4a):

Where $K=\mathrm{Sqrt}(K_x^2+ K_y^2)$ (3a) and $\theta = \tan^{-1}(K_y/ K_x)$; (4a)

The horizontal component K_x of a Sobel operator can be presented in a two separable component. The row components can be further divided into two components.

$$Kx = \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix} * \begin{pmatrix} 1 & 2 & 0 & -2 & -1 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 6 \\ 4 \\ 1 \end{pmatrix} * \left(\begin{pmatrix} 1 & 2 & 1 \end{pmatrix} * \begin{pmatrix} 1 & 0 & -1 \end{pmatrix} \right)$$

The column component of Sobel operator has strong normal smoothing whereas first component of row i.e. $(1 \quad 2 \quad 1)$ have small parallel smoothing. This shows that Sobel operators have embedded smoothing operation. If we normalized second component of row by 2, it becomes $\left(\frac{1}{2} \quad 0 \quad -\frac{1}{2}\right)$, which is equivalent to the second order numerical scheme of 1st derivative.

We can find similar expression of vertical component K_y. Though Sobel operator has embedded smoothing sometime it require more smoothing to reduce the noise. In certain implementations, this separable computation may be advantageous since it implies fewer arithmetic computations for each image point.

The Sobel operator represents a rather inaccurate approximation of the image gradient, but is still of sufficient quality to be of practical use in many applications. More precisely, it uses intensity values only in a 5×5 region around each image point to approximate the corresponding image gradient, and it uses only integer values for the coefficients which weight the image intensities to produce the gradient approximation. A even better approximation of gradient can be obtained by normalizing the filter (e.g. here by $1/128$).

5.4.2 Example of Application

We demonstrate dip estimation on the Marmousi model. The Migrated image o model is shown in the Figure 5.23. We migrated image is convolve to Sobel operator and estimate the dip (see figure 5.24). Migrated image and corresponding dip estimation is in fine scale (10mX25m) whereas we need 'dip' on the scale of velocity that may be 100mX100m or more. Hence we smooth it up to the scale of velocity (Figure 5.25) and use it to smooth the velocity along the dip direction.

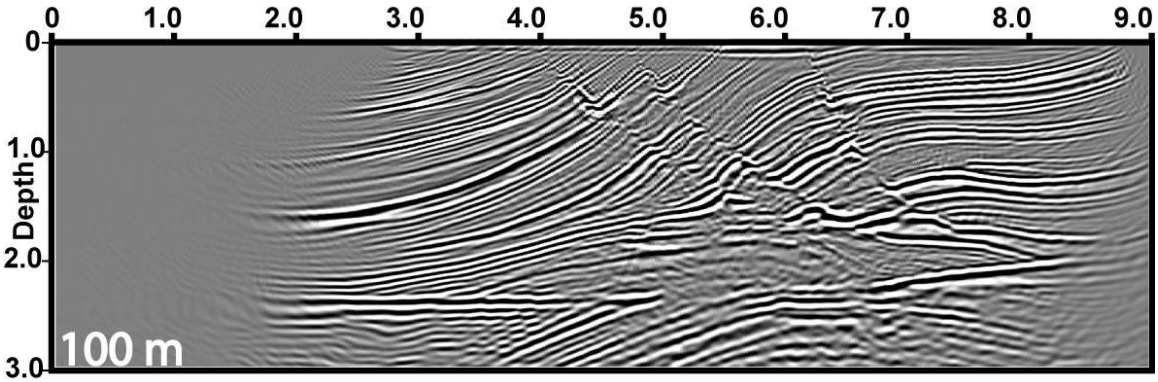


Figure 5.23: Migrated image of a Marmousi models.

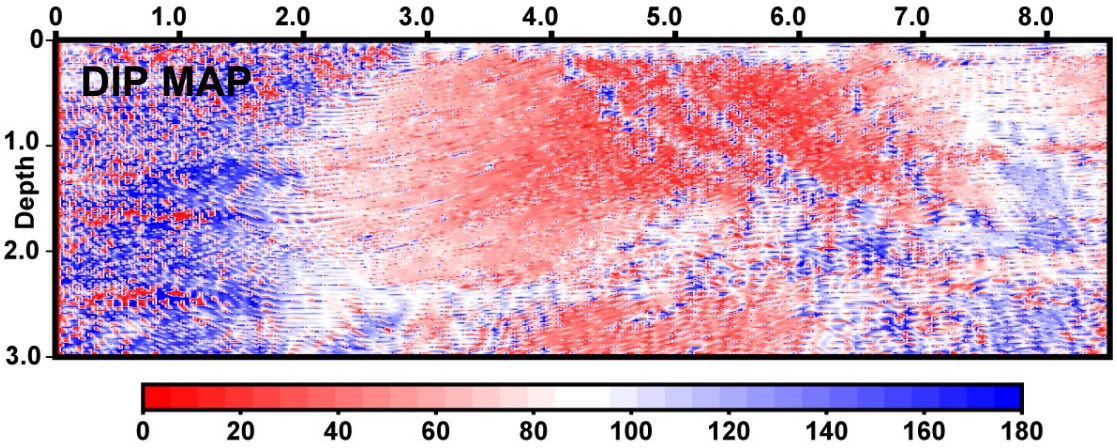


Figure 5.24: Dip extracted from the migrated image by convolving Sobel operator.

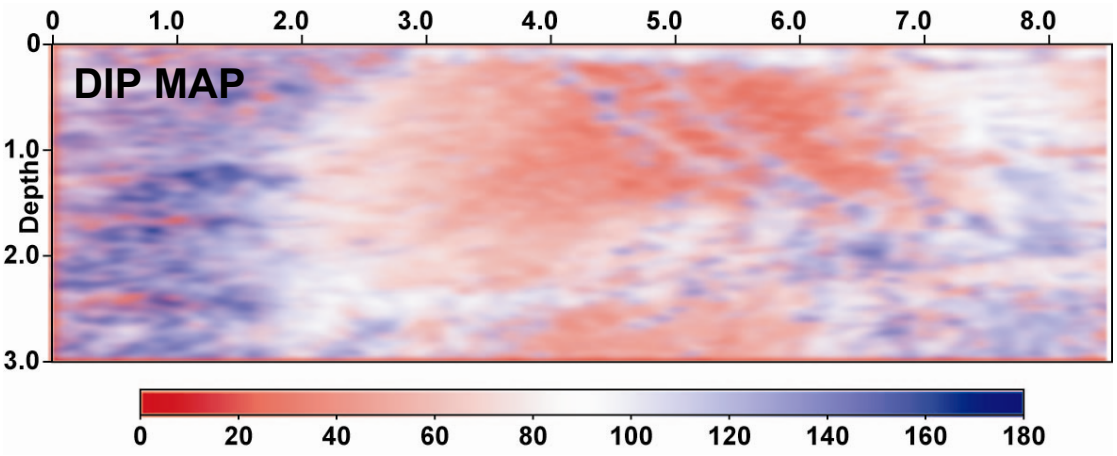


Figure 5.25: *Dip map is smoothed to reduce the small noise and convert this fine dip in to the scale of velocity model.*

5.5 Conclusions

In this chapter we presented modified differential semblance function for both offset and angle domain to measures the flatness of the coherency panels. These functions are nicely convex for a quite large range of high and low velocities, and are stable with respect to the frequency content and to the depth of the events. In next chapter we will use modified differential semblance function incombination with semblance function, as objective functions in MOEA. Both function together provide a roboust and accurate criteria for velocity estimation.

We also develop a robust and automatic technique for gross velocity error estimaton using RMO technique on the gathers. In the next chapter, we will use this RMO information to correct the velocity model before the crossover. This helps in generating good models and fast convergence.

We also extract dip information from the migrated image to smooth the velocity variation along the layers. In next Chapter , we will use this information in seismic velocity invesion.

Chapter 6

Seismic Velocity Inversion using Multi-objective Evolutionary Algorithms

This chapter presents our automatic seismic velocity inversion algorithm, obtained by assembling all ingredients presented in the previous chapter into an hybrid multiobjective evolutionary algorithms (MOEAs). Results of this algorithm on Marmousi and L7 data are also presented. Note that some results have been published as separate papers (Singh et al., 2006; Singh et al., submitted), and this chapter is built from these publications. On the one hand, this introduces some redundancies with the previous Chapters. But on the other hand, this chapter can also be read stand-alone, though more details can be obtained by referring to the corresponding mentioned sections.

For precise estimation of background velocity, to get a good subsurface image, methods like waveform inversion or prestack depth migration (PSDM) work well. However, these methods are limited by the first-order Born linearization, and an acute computational cost. To tackle these issues, we present a new automatic velocity estimation technique based on Multi-Objective Evolutionary Algorithms (MOEAs) using both the Semblance and the Differential Semblance as two distinct objectives, as discussed in section 6.5.2. Our approach is able to cope with large velocity errors, and the computational cost of this algorithm is comparable to that of gradients methods.

After an overview of the background in Secion 6.1, Section 6.2 presents a very brief introduction of MOEAs. In Section 6.3, the classical application of a standard MOEA (the so-called **ε**–**MOEA**) is introduced. Section 6.4 presents modified **ε-MOEA,** in which several domain-specific ingredients are added to ease the optimization process. Section 6.5 presents the main component of modified **ε-MOEA**. In Section 6.6, implementation of modified **ε-MOEA** and its components are presented. Finally, results on synthetic and real data are presented in section 6.7 and discussed in Section 6.8.

6.1 Introduction

One of the major interests in seismic exploration for oil is to correctly identify and localize the subsurface structure. Therefore, seismic processing demands precise estimation of background velocity so as to get a good subsurface image. The process of converting normal move out (NMO) and /or stacking velocity into interval velocity is unstable for the layers with lateral velocity variation (Lynn and Claerbout, 1982). Consequently conventional velocity estimation techniques do not work well in geologically complex regions, such as foothills and salt structure. It is also widely accepted that conventional seismic processing can not adequately overcome the above mentioned difficulties. However, tools like inversion or prestack depth migration (PSDM) work well. Common-image gathers (CIGs) produced by PSDM are sensitive to the velocity model and therefore CIGs could be used for more precise velocity estimation.

The residual move out (RMO) curvature analysis (Al-Yahya, 1989) is generally used for migration velocity analysis. This technique was further developed by Lee and Zhang (1992), Lafond and Levander (1993), Liu and Bleistein (1995), Yan et al. (2001). The geometrical and mathematical characteristics of RMO for the prestack Kirchhoff migration were appraised for velocity by Zhu et al. (1998). RMO properties for wave equation migration were analysed by (Biondi and Symes, 2004; Shen P. 2004; Bartana et al., 2006; Sava and Fomel 2006). RMO curvature analysis is based on the concept of generation of flat gather in CIGs by PSDM for correct velocity model regardless of the structure.

Presently, iterative PSDM methods are widely used for velocity analysis. They are also based on the flatness criteria. The flatness criteria has been measured on common receiver gathers (Al-Yahya, 1989), common offset gathers (Liu & Bleistein, 1995; Mulder & ten Kroode, 2002) , common scattering angle gathers (Brandsberg-Dahl et al., 1999) and common shot gathers (Symes & Carazzone,1991) generated by Kirchhoff migration. Kinematics migration artifacts in CIG typically arise in the prestacked image gathers when the medium is a strong refractor (Stolk & Symes, 2004). As a result, the flatness principle may not be valid in CIGs produced by Kirchhoff migration. To avoid these artifacts for general velocity models, wave equation migration is required for construction of CIG. Attempts have been made to estimate velocity using wavefields methods, such as the method of full waveform inversion (Tarantola 1984; Pratt 1999), diffraction tomography (Devaney & Oristaglio 1984), wave equation tomography (Woodward 1992; Luo & Schuster 1991), differential semblance optimization (Symes &Carazzone 1991) and wave equation migration (Sava et al.,2005; Sava & Biondi 2003; Shen P. et al., 2003).

Wave-equation based methods are limited by the first-order Born linearization. If the phase differences between the modeled and recorded wavefields are larger than a fraction of the wavelet, then the assumption made under the Born linearization is violated and velocity inversion methods diverge (Woodward, 1992; Pratt, 1999; Dahlen et al., 2000). Consequently, calculation of gradient in velocity optimization becomes one of the severe obstacles. Also, calculation of gradient is two (Sava and Biondi, 2004) to four times (Shen 2004) more computationally expensive than the migration itself. Hence, computation of gradients and a lack of good guess of initial velocity models in geologically complex regions inhibit the application of gradient optimization methods for automatic velocity estimation.

Conversely, global optimization methods (genetics algorithms, simulated annealing and Mont-Carlo method) are not constrained by the local linearization of the wave equation (Born limitation) and also are capable of coping with the non-linear relation of the

seismic data and velocity model (Jervis et al., 1996; Docherty et al., 1997; Mansane & Schoenauer, 2000). However, acute computational cost using global methods limits its routine use. This is because the global method needs to search a large parameter space with little or no domain knowledge. Therefore, there is an urgent need to customize such optimization methods such that, it could use the domain knowledge as well as to guide the optimization in the right direction without losing the generality of the method.

In this Chapter we present a new global optimization algorithm based on multi-objective evolutionary algorithms (MOEA) for automatic velocity estimation. This algorithm is able to cope with large velocity errors. Thanks to the customization of the MOEA according to domain knowledge that accelerate the convergence, the computational cost of this algorithm remains comparable to that of direct gradients methods.

First we are going to give concise overview of multiobjective optimisation. We also discuss the limitations and advantages of MOEA, which will help in understanding the proposed customizations.

6.2 Multi-objective Evolutionary Algorithms

Multi-objective optimization (MO) is concerned with finding solution that optimize (min or max) several contradictory objectives, and eventually meet some additional constraints. Suppose for instance that we want to minimize a vector of functions:

minimize $[f_1(x), f_2(x).....f_k(x)]$

subject to the m inequality constrains:

$g_i(x) \leq 0$ i=1,2,….m, (inequality constrain)

and the p equality constrains:

$h_i(x) = 0$, i=1,2,……..p, (equality constrain)

Where k is the number of objective functions f_i: $\boldsymbol{R}^n \rightarrow \boldsymbol{R}$. Here $x = [x_1, x_2, ...x_n] \in \mathbf{X}$ is a vector of decision variables and **X** is the feasible domain. We wish to determine from among the set F of all vectors which satisfy the constraints the particular set of values $x_1^*, x_2^*, x_3^*, ...x_n^*$ which yield optimum values of all the objective function. It is rarely the

case a single point simultaneously optimizes all the objective functions. Therefore, we normally look for "trade-offs", rather than single solution when dealing with Multiobjective optimisation problems.

A vector of decision variables $x^* \in F$ is said to *dominate* a vector x if $f_i(x) \leq f_i(x^*)$ for all i=1,….,k and $f_j(x) \prec f_j(x^*)$ for at least one *j*. A vector x^* is *Pareto optima,* or *non dominated, l* if there exists no feasible vector of decision variable $x \in F$ that dominates it, i.e. if it is not possible to decrease some criteria without causing a simultaneously increase in at least one other criteria. The set of Pareto-optimal solutions is also called the Pareto optimal set (Figure 6.10). The image of the Pareto optimal set under the objective function is called *Pareto front.* For a given positive **ε**, the notion of dominance can be relaxed to that of **ε**–dominance (see Chapter 2 for details).

Evolutionary Algorithms can, in principle, find solutions to problems with non-smooth, nonlinear objective functions and constraints, whereas smooth and differentiable objective function are required by gradient-based methods. Furthermore, because they evolve a population of candidate solutions, they offer additional advantages when tackling a multi-objective optimization problem, being able to identify many Pareto-optimal solutions in a single run. This explains why Multi-Objective Evolutionary Algorithms (MOEAs) are so popular today. Moreover, beside having no assumption w.r.t to problem space and they are easy to hybridized with other approaches (such as gradient or local methods), adding domain knowledge to improve their efficiency. MOEAs are also able to cope with noisy data . Different MOEAs are described in Chapter 2, and the choice of ε-MOEA was motivated by the study by Deb et al. (2003) in which it demonstrated similar performances than the best performing previous MOEAs but requiring much less computational efforts.

However, the disadvantage of MOEAs, and even ε-MOEA, is that they are usually much slower than local optimization methods -often by several order of magnitudes. As

problem size scales up (from, say, ten to a hundred or a thousand decision variables), an evolutionary algorithm is often overwhelmed by the dimensionality of the problem and is unable to find anything close to an optimal solution, whereas it is still possible to solve such large problems with local optimization methods.

The main reason of such high computational cost and inability to perform on a great number of variables is the blindness of those algorithms, i.e. the lack of exploitation of domain-specific information. Any efficient global optimization algorithm must carefully balance between two possible strategies to find the global optimum: *exploration* to investigate new and unknown areas in the search space and the *exploitation* to make use of the knowledge from previous iterations. In Figure 6.1 a schematic representation of various optimisation algorithms in term of degree to which they explore the parameter space and exploit information is given.

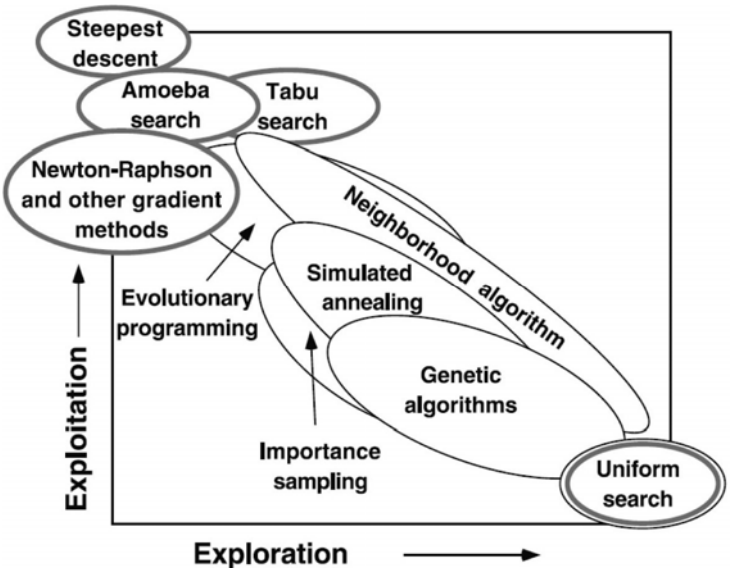


Figure 6.1: *A schematic representation of various search/optimisation algorithms in term of degrees to which they explore the parameter space and exploit information. Shaded borders indicate deterministic methods. Global optimisation methods have good exploration property, whereas local optimizations have good exploitation property. (Sambridge and Mosegaard 2002).*

These two strategies are contradictory, and therefore a good optimization algorithm must strive to find a balance between them. In the hybrid MOEA that we have designed (described in forthcoming section 6.4), an effort has been made to exploit and use domain-specific information, without reducing the exploration property of the original underlying MOEA, the ε-MOEA. Such ad hoc use of available expert knowledge from previous iterations is known to be mandatory for Evolutionary Algorithms to reach (fast) convergence toward the global solution even in large search spaces.

In next section, we will describe how we used the standard **ε-MOEA** algorithm to solve the problem of seismic velocity inversion, before presenting in section 6.4 the hybrid customized algorithm that we had to design to reach reasonable results.

6.3 ε-MOEA for velocity inversion

The standard **ε-MOEA** was described in details in Chapter 2. Here we will specifically describe its implementation for seismic velocity inversion.

A schematic diagram of ε-MOEA for velocity inversion is shown in Figure (6.2). Its step-by-step description follows:

- The *Population* is initialized with random models: a random value uniformly drawn in interval 1500 to 5500 m/s is set at each grid point. The evaluation process is applied to all models (Process of evaluation of each model is shown in Figure 6.2a)
- The ε-dominant models are copied into the *Archive*.
- ✓ One model is selected using tournament selection (see section 2.6.3) from *Population* and one model is uniformly selected from *Archive.* These models are called *Parents*.
- These *Parents* are used to generate *Offspring* using *Variation Operation* (SBX crossover and Polynomial mutation ; see sections 2.5.1.4 and 2.5.2.4)
- Each *Offspring* is evaluated, i.e. its semblance (S) and differential semblance (DS) are computed (Figure 6.2a).
- Each offspring is then compared with the models in the current *Population,* and it replaces the first one it *dominates (*see section 2.7.6.2*),* otherwise it replaces one randomly chosen member of the population.

- Each *Offspring* is also compared with *Archive* members, and replaces all those that it *ε-dominates* (see section 2.7.6.3).
- This process continues till the termination criterion is satisfied (here maximum number of iterations). (Go to step √)

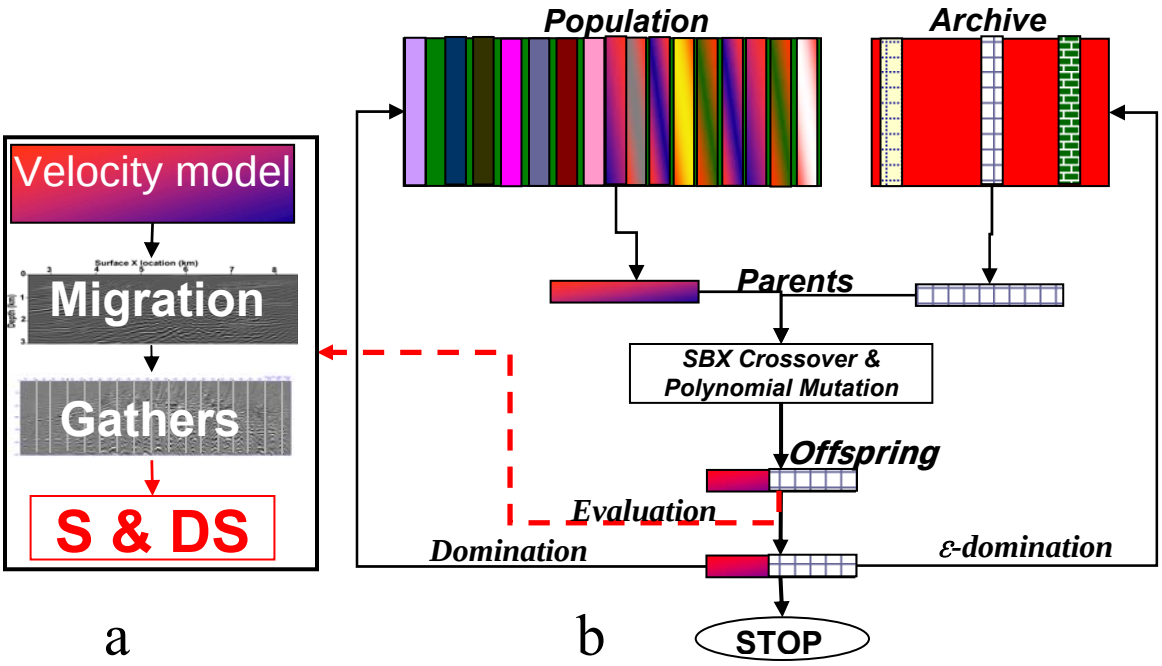


Figure 6.2: *(**a**) Process of **evaluations** of one velocity model. (**b**) Schematic diagram of standard ε-MOEA. One model is selected from the current **Population** and one from the Archive. They are used to generate an **Offspring** using the **variation Operators** (crossover and mutation). This **Offspring** is put in the **Population**, replacing one member it **dominates**, if any, or one randomly chosen otherwise. It get placed in the **Archive** only if it **ε-Dominates** at least one member of the current Archive.*

We performed velocity inversion for the Marmousi model using this algorithm. It was run many times for more than 3000 generations using different initializations. We noticed that there is very poor convergence. This was not surprising because number of unknown parameters was 457, which is utterly large for traditional ε-MOEA to solve in such a small number of iterations. Hence we decided to add *exploitation property* (a new guided crossover operator using some gradient-like information) as well as other domain-specific knowledge in several steps of the algorithm. Now we have presented standard ε-MOEA

we will present in the next section, the global framework of our customized ε-MOEA algorithm.

6.4 Customized hybrid ε-MOEA for velocity inversion

In customized ε-MOEA, we introduce *Reference model* to add exploitation property, *RMO* corrections that work like pseudo a gradient method and we add information by generating *good velocity model (initial & Parents)* and *dip smoothing*. A graphical chart of our proposed ε-MOEA is shown in Figure (6.3b). The customized part of this algorithm is shown with broken line (compare to Figure (6.2b)). The different steps of the algorithm will be described in the following, while detailed description of each step can be found in the corresponding mentioned section.

First we initialize *Population* and *Archive* by generating and evaluating (Process of evaluation and information extraction of each model is shown in Figure 6.3a) the random models (*Population (*Figure 6.3b)) and we copy the ε-dominant models into the *Archive (*Figure 6.3b). The *Population* is initialized with random models: a random value uniformly drawn in interval 1500 to 5500 is set at each grid point.

- The ε-dominant models are copied into the *Archive (*Figure 6.3b).
- *Reference models* are initialized by copying one of the *Archive* members.
- The population is divided in two parts: (1) *replaceable* (2) *non-replaceable*. **Non-replaceable member will not change during the evolution**. This non-replaceable part will remain a source of diversity all along evolution. Such procedure is inspired by the micro-GA (see section 2.7.7.10).
- ✓ **Four models** are randomly uniformly selected from the population with a 90% probability or among the *Reference models* with 10% probability.
  1. Each selected model is smoothened using the extracted ***dip* information** (see section 5.4 and section 6.5.3 for implementation).
  2. Each selected model is corrected according to the extracted ***RMO* information** (See section 5.3). *RMO information* works here like pseudo-gradient (see section 6.5.4 for implementation).

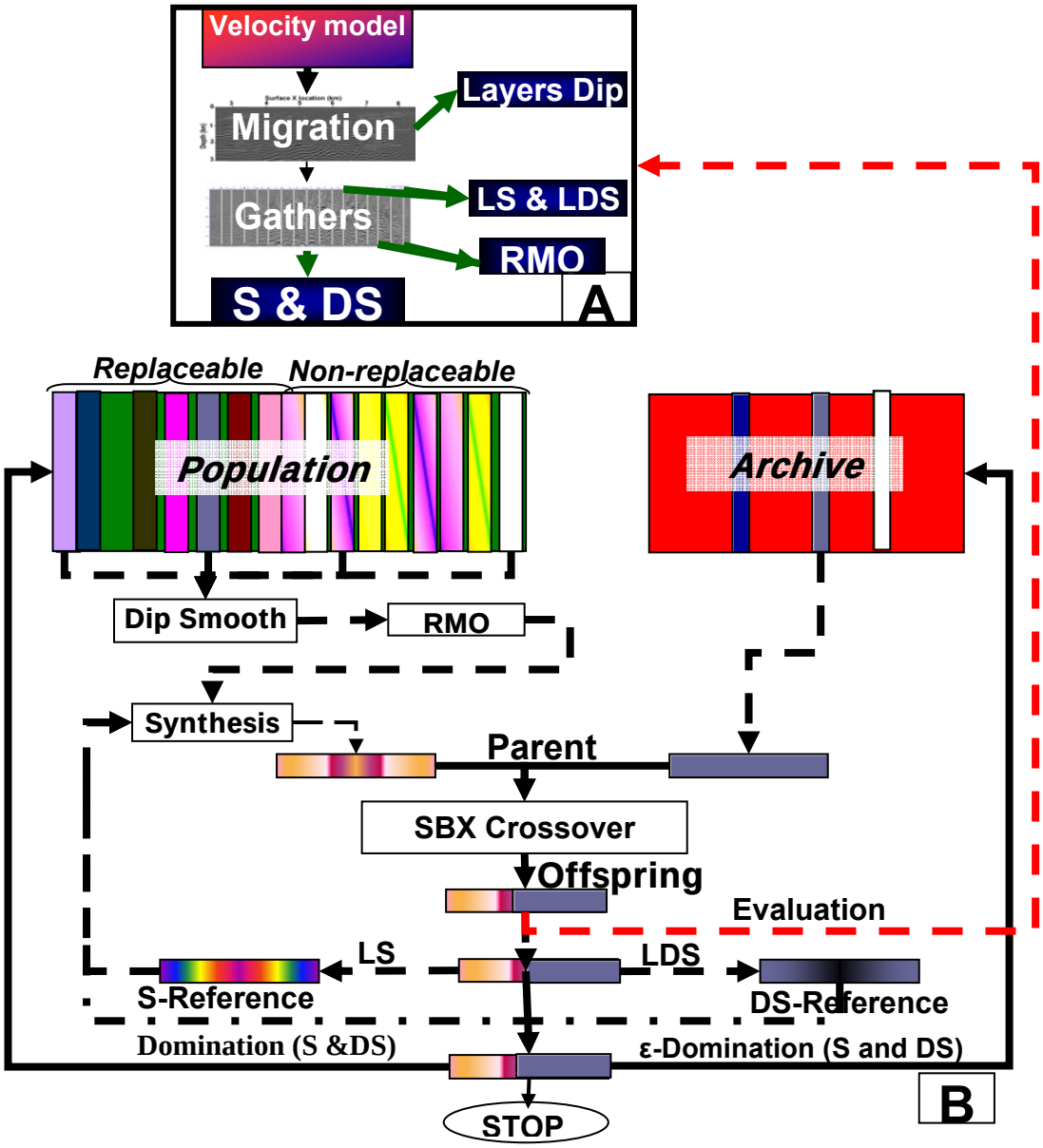


Figure 6.3: *(**a**) Process of **evaluation** and **information extraction** from a velocity model. (**b**) Schematic diagram of our **customized ε-MOEA**. Four models are stochastically selected from **Population** or **among the Reference models.** Each selected model is first smoothened using **Dip** information and then corrected using **RMO** information. A parent model is synthesized from these four models using **LS** or **LDS** criteria. **Synthesized Parent** and one uniformly selected parent from **Archive** are used to generate an **Offspring** using **variation Operation** (SBX crossover).*

*This **Offspring** is compared with **S-Reference** and **DS-Reference** using **LS** and **LDS** criteria. If the **Offspring** has better **LS or LDS** than the reference models for some panel, column in the velocity grid corresponding to this panel is copied into the reference model. This **Offspring** replaces one member in **Population** that it dominates, if any, or one random member otherwise, whereas it get placed in Archive only if it **ε-Dominate** at least one of its members.*

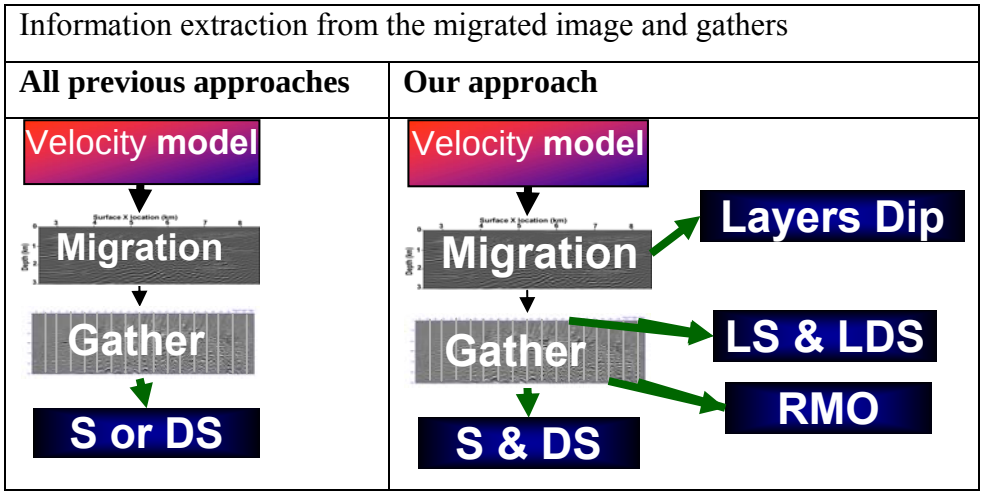
3. One Parent model is synthesized from these four models (for details, see section 6.5.5). In few words, The ***Synthesized Parent model,*** *results from* copying the velocity columns corresponding to the best gather of each model using either semblance (**LS**) or differential semblance (**LDS**) criteria (here L stand for local, i.e. for one gather).

- Once this Parent is synthesized, one other Parent is uniformly selected from the *Archive.*
- These two *Parents (Synthesized parent* and *Archive parent)* are used to generate *Offspring* using SBX crossover (see sections 2.5.1.4).
- These *Offspring* are evaluated, and domain information is extracted (Figure 6.3a and table 6.1).
- Each *Offspring* is compared with *S-Reference* and *DS-Reference* model using *LS* and *LDS* criteria respectively. If the *Offspring* has better *LS* or *LDS* than the Reference models, velocity corresponding to better *LS* or *LDS* of *Offspring* is copied to the corresponding *Reference models* (see section 6.5.6).
- Each offspring is then compared with the models in the current *Population, and if it dominates (see section 2.7.6.2)* any member of population*,* then it replaces that member otherwise it replaces one randomly chosen member of the population.
- Each *Offspring* is also compared with *Archive* members*, and replaces all those that it ε-dominates (see section 2.7.6.3).*
- This process continues till the termination criterion is satisfied (here maximum number of iterations). (Go to step √)

Because our main concern is to reduce the computational time of the inversion, we have adopted the micro-GA procedure that keeps half of the population fixed during evolution, as a diversity reservoir, and we are using only a small population (20-30). However, first experiments demonstrated that such an approach might not be able to generate diverse enough individuals to find a suitable solution. Hence our complete algorithm uses both the customized and the standard approaches: at each generation, the algorithm randomly

chooses to generate the offspring using either the standard approach or the customized approach with 40:60% probabilities respectively.

Table 6.1.: *Comparison with previous approaches. Standard EAs are weakly guided random search methods and hence convergence is rather slow. For example only global fitness of a model is measured in the previous approaches (i.e. Stoffa and sen 1992; Docherty et al, 1997; Mansanné et al., 2002). To accelerate this convergence in higher dimension parameter spaces we introduced more guiding information extracted from the domain knowledge. This information is, first, dip smoothing (it implicitely reduces the number of parameters), second, local fitnesses (LS, LDS) to reduce the coupling between parameters, and third, RMO based velocity improvemen (a step toward gradient methods).*



Before giving the experimental results obtained using this hybrid approach, we will first give more details about the implementations of the domain-specific procedures used therein.

6.5. Main components of the customized ε-MOEA

First we begin by describing the representation of velocity in section 6.5.1 and then one of the most important components of algorithms that is *objective functions* in section

6.5.2. Then we discuss about the *Dip smoothing* and *RMO correction* in the section (6.5.3 and 6.5.4). This will be followed by the construction of synthesized parents (in section 6.5.5) and reference models (in section 6.5.6) and at the last implementation in ε-MOEA.

6.5.1. Representation of velocity model

The representation of velocity model is one of the important issues in velocity optimization as it controls the shape and size of the model parameters. A detailed discussion about the velocity parameterization can be found in Chapter 4. We decided to use grid parameterization to represent our velocity models because any geological environment could be represented by a grid even though it requires a large number of parameters. In grid representation it is easy to incorporate the information obtained after the migration of velocity model because the wave equation migration is performed on a regular grid whereas it is difficult with other representations (Voronoi or geological). The usual drawback of having a large number of parameters (CPU time is large) will be balanced by the introduction of domain knowledge (Good initialization, RMO correction and Dip smoothing).

6.5.2. Objective functions

For an automatic velocity estimation through PSDM image gathers, where no picking is introduced, the choice of the objective function is vital. In this work we are using both semblance (S) and differential semblance (DS) function as a objective function. S and DS functions measure the global goodness of a velocity model (effect of *all the gathers* in a model), whereas we also use semblance and differential semblance criteria to measure the goodness of each gather (hereinafter we called as LS and LDS respectively, L stand for "local"). we will now summarize the properties of both S and DS functions for sake of completeness (detailed discussion in chapter 5) and providing arguments to substantiate the use of both functions.

Though both function measure the horizontality on the coherency panel. They differ by the way of measuring the flatness. Semblance measure the sum of the square of energy of the traces at each depth whereas differential semblance measure the sum of the, square of

difference of near trace energy at each depth level. Hence Semblance should be maximum and differential semblance minimum for true velocity. In term of sensitivity to velocity error, smoothness and noise both function are quite different. For small velocity error semblance function and for large velocity error differential semblance function has strong sensitivity. Hence Chauris and Noble (1998) recommend to use differential semblance function for first few iteration and subsequently followed by semblance function. It has been also noticed that differential semblance function is very sensitive for smoothness (Shen et al.,2004) and noise (Mulder and ten Krood 2002) whereas semblance function is not much affected by smoothness and noise however not sensitive for large velocity error. Hence single use of either semblance or differential semblance function may lead towards local minima. Consequently an independent and simultaneously measurement of both functions can lead towards global minima. Multi-objective evolutionary algorithms provide a unique opportunity to use more then one objective function, independently and simultaneously. Hence one can utilize the property of both objective functions. Since in the beginning of optimization, the initial models are far from the true models, at that time DS function will help in the convergence because of its strong sensitivity for far velocity models. On the contrary when the models are near to true model S function help in convergence because of its strong sensitivity for near velocity models. In the beginning when models are not smooth DS may produced local minima, however combination with the semblance can help in coming out from it. We decided to use MOEA so that we could exploit the property of both function and have more than one solution.

It had been found that the velocity estimation using migration method with traditional global optimisation process is computationally very intensive. One of the important reasons for this is the lack of exploitation of information present in the image and gathers. To create a more geologically feasible velocity model and to speed the convergence of the migration velocity estimation problem, it is necessary to add extra information during inversion so as to construct geologically feasible models. This extra information can be taken from well logging, geologist's structural model and preliminary stack or migration results. Other additional information could be from the migration of velocity models

(even from a wrong velocity model), such as their gross velocity error estimations as well as the structural trend of the geology. We thus need a robust approach which could extract approximate amount and direction of gross velocity error and structral trend of the geology. In the next section we show how such additional information can be extracted from migrated image and gathers and can be incorporated in to the velocity model to improve it.

6.5.3. Dip smoothing

As already disucced in section 5.4 the structural trend of a geological model also provides significant information and has been used differently for velocity estimation. The basic assumption (Delprat-Jannaud and Lailly 1992) is that the velocity follows structural dips or some other known trends and it can hence be incorporated as a term into the objective function (Delprad-Jannaut and Lailly 1992), Kaipio et al.,(1999) suggested using a prior structural information to create conditional covariance matrices. Here, we are not using the dip information as objective finction, regularization or preconditioner. Our goal is to extract the dip information at each point of the migrated image and use this dip information to reduce the variation of velocity along the dip direction. For identifing the reflection boundary and measuring the dip of the reflection, we have used the Sobel edge dectection tool. Edges characterize the boundaries in image whereas the Sobel operator gives the direction and magnitude of the edges.

We extract dip information from the migrated image. These migrated images are on a very fine scale grid, whereas we need dip information on a coarse scale velocity grid, therefore we smoothen migrated dip map up to the scale of velocity model. This dip smoothing can be viewed as some type of anti-aliasing. Once we obtained dip information to the scale of velocity model, we smooth the velocity model along the direction of dip.

6.5.4. RMO correction

Gross velocity errors from migrated images are generally estimated by the techniques of either Residual MoveOut (RMO) or Stolt residual migration (Sava 2003). In this work,

RMO technique is adopted as we are interested only in approximate amount and direction of change of velocity required at any place for correcting the velocity model. Here we are only reviewing it briefly, a detailed discussion can be found in Chapter 5. Regardless of the domain (offset or angle) in which the prestack partial images are defined, the RMO function is usually parameterized by a single parameter, γ, the ratio of migration velocity and true velocity. Inaccurate velocities will cause move out artifacts on the migrated image, and its shape also depends on the velocity contrast. Smile and frown shapes are generally produced respectively for low velocity and high velocity contrasts.

Here, we firstly perform the Radon transform of a gather (offset or angle gather). This enables us to work in parameter space (z,γ) rather than the image space where the curve detection problem is tedious. The peaks determined from the Radon transform basically are the parameters of a reflection which gives the necessary information about the background velocity model. Although these peaks are a measure of the combined effects of both the local and global velocity variations, we assume here that the major contribution to the peak is by the local part of velocity variation. The peaks thus determined can then be used to improve the velocity models locally.

To improve the velocity model we applied fraction of error correction using a triangular filter. For example if in any particular place of velocity model, we estimate a velocity error of ±5%, we improve it by only ±2.5% or less. We do not apply full correction, so that we could avoid the effect of far velocity error (i.e. global velocity error). There may be many way to apply these correction, we applied by using a triangular window shown below.

$$\pm\frac{\gamma}{W}\begin{vmatrix} 2 & 2 & 2 & 2 & 2 \\ 0 & 4 & 4 & 4 & 0 \\ 0 & 0 & 8 & 0 & 0 \end{vmatrix} \text{ ; Here } W=60$$

Here γ is the velocity error and W is the weight which decide the fraction of correction we want to apply. All the points above the place of errors is multiplied by corresponding fraction and added or subtracted from that point.

This is a step toward the gradient methods because we not only handle the value of the fitness's in the parameter space, but also vectors of correction.

6.5.5. Synthesis of a parent

Synthesis of a parent is based on the fitness of each gather of a model. For this we randomly select four models from the population. Each selected model is first improved on the basis of γ (RMO) information (see chapter 5). Then, these models are smoothened along the dip direction, which reduces the variance of velocity along the layers. Once this information is introduced, the next goal is to synthesize a parent's model from these four selected models.

The strategy of synthesis of parents model is shown in Figure 6.4. Firstly, we compare the selected models for each gather on the basis of LS (semblance of one gather). Then the velocity model corresponding to the best LS gather are copied into corresponding parent synthetic velocity model. We adapt same strategy to synthesis the parent using LDS (differential semblance of one gather) criteria. Here we can synthesize parent by three ways. First we could replace the model on the basis of best LS, second on the basis of LDS or third we could adapt Pareto dominance strategy for synthesizing the parents models. The schematic view of parent synthesize is shown in the Figure 6.4.

We can also adapt this strategy to synthesis a parent based on horizontal panel, where we need to compute LS and LDS corresponding to each depth windows and synthesis parents using above methods (Figure 6.7).

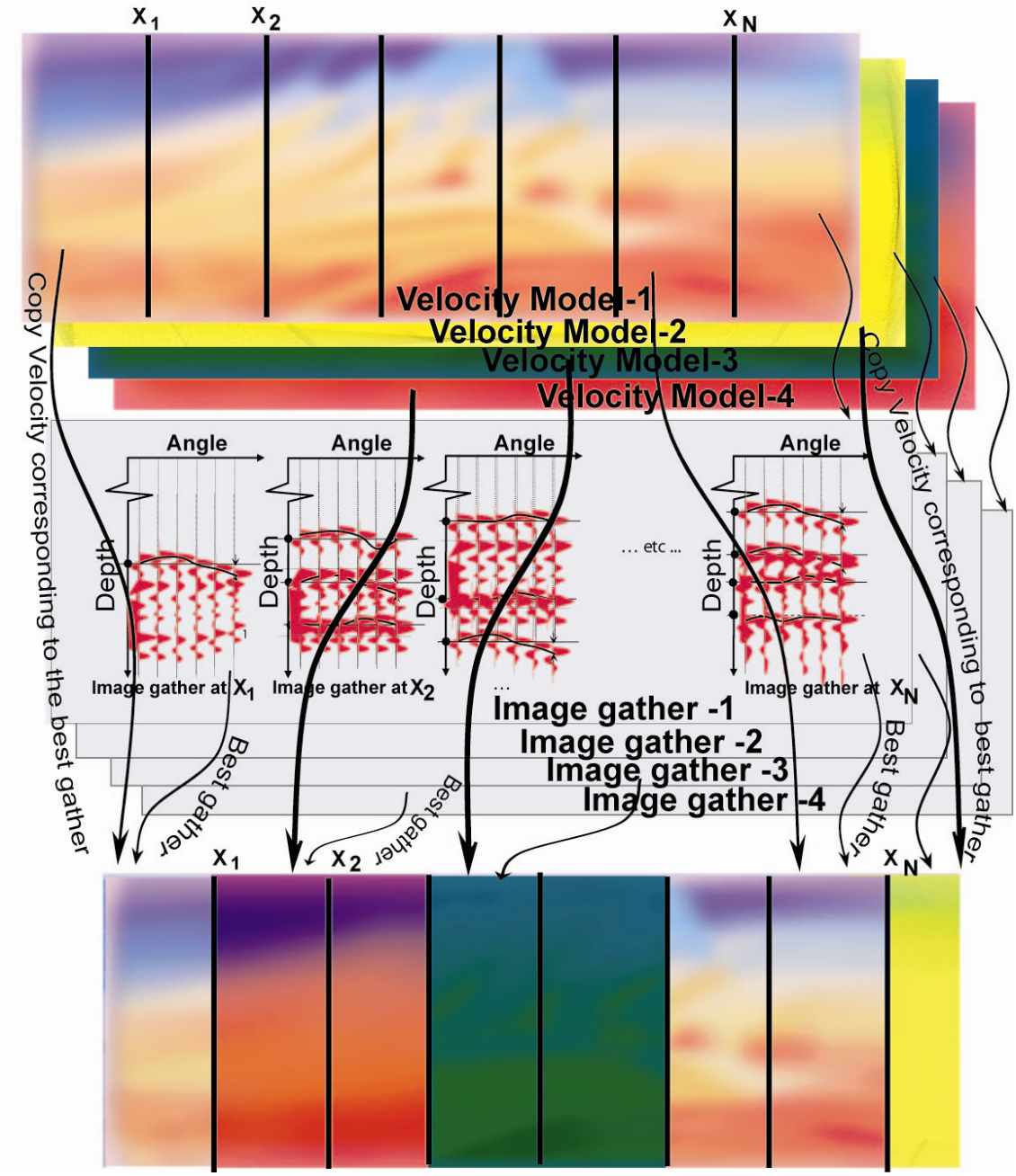


Figure 6.4: *Vertical Synthesis of parent. Each velocity model produces corresponding image gather. At each X position of all image gathers, local semblance (LS) and local differential semblance (LDS) are computed and compared with each other. Velocities corresponding to the best LS or LDS are copied into the synthesized parent model.*

6.5.6. Reference Models

It is well known that *N* inversions are cheaper than one N-times larger inversion. So we will apply this principle to our case by localizing, as much as possible, the problem at each abscissa. Indeed, the migrated image at some abscissa only depends on the velocity in the vicinity of this abscissa.

The global optimisation methods lacks by exploitation property. We decided to add the exploitation property in our ε-MOEA to make it more efficient. Generally globally good velocity models are preserving in the archive, where locally good model do not have such place, to be preserved. To exploits these locally good velocity models, we created two models which we hereby refer *Reference models*.

These reference models are initialized by copying the one archive member, when evolution of the velocity model start, each new offspring is compared with the reference models. One reference model is compared on the basis of semblance function and referred as *S-Reference model* (Figure 6.2), and other model is compared on the basis of differential semblance function and referred as *DS-Reference model*. *S-Reference* model is compared by offspring for each gather using LS criteria and the offspring velocity corresponding to best LS gather is copied in to the S-Reference model. Same procedure is applied of DS-reference model, only difference is the use of LDS criteria.

This enables us to better capture the local property. Reference models are then used in the process of crossover with other models. During the crossover, such reference models help in generating better models. Thus, reference models not only help in fast convergence but also introduce exploitation property into the MOEA.

6.6 Implementation of Customized Evolutionary Components

Once we have represented our model, found a better objective function and have the tools for providing the information during optimisation, we need now to develop a good optimisation strategy. As we intend to use two objective functions, multi-objective evolutionary algorithms (MOEA) is the preferred choice. Among MOEAs we can select a generational MOEA (NSGA-II(Deb et al 2000), SPEA-2(Zitzler and Thiele 2002), etc.) or a steady state MOEA (ε-MOEA (Deb et.al, 2003). Generally, steady state EAs are faster then generational EA . Because the superiority of ε-MOEA over NSGA-II and SPEA-II in term of fast convergence, diversity mainatenance as established (Deb et al., 2003) we decided to use it.

As already advocated, one way to cope with the main drawback of EAs (their high computational cost) is to add domain- specific knowledge.. This necessitates to develop a good exploitation operator as well as to gather useful information so as to proceed in the right direction. We have given an outlook of our customized ε-MOEA in (Figure 6.3),, in which a biased initial population (*Population*), information exploitation models (*Reference models*) and knowledge-specific crossover operator (*Dip smoothing, RMO correction and synthesising a parent*) are introduced. We will detailed these three specific parts in the following:

6.6.1 Initial Population

In MOEA, evolution starts from a population of specific size and made of random models. There are two practical issues in this approach (i) population size and (ii) initial population generation. These two issuues are discussed separately below.

Population size generally depends on the nature of the problem, but typically contains several hundreds or thousands of possible initial models. The population size has a major effect on the efficiency and performance of MOEAs. MOEAs usually do not work well for very small size population and a very large population size impacts the performance

of the MOEA. The purpose of large population is to have diversity in the parameter space. However, a large population needs the evaluation of a large number of models. Since our evaluation process is costly, we decided to evaluate less models and at the same time to maintain the diversity in the parameter space. To fullfill the above goals, firstly, a small number of models are generated in initial population and secondly this initial population is divided into two sub-populations: one sub-population is non-replaceble and the other sub-population is replaceble during the optimisation proces, in this way the diversity in the parameter space is maintained by the non-replaceble sub-population. A similar stretgy is adopted by Coello and Pulido (2001) for micro-multi-objective genetics algorithms (μ-MOGA). The replaceble part of the sub-population will (hopefully) converge to the Pareto front.

A example of population convergence for Marmousi velocity model optimisation on a 250m grid sampling is shown in the Figure 6.5, where initial population is shown by black colour balls and final population (after 600 iteration) in the blue colour balls. We can see that replaceable subpopulation of the final population is clustured (Figure 6.5 inside oval) after 600 iterations, it means, it does not have a significant diversity, however we can expect good parameter values from them. Nevertheless required diversity of final population is being provided by non-replaceble subpopulation which is still distributed (Figure 6.3, distributed blue balls marked as 600, outside the oval). Hence non-replaceble subpopulation is helping in maintaing the diversity of the population. whereas repaceble subpopulation is helping in the convergence.

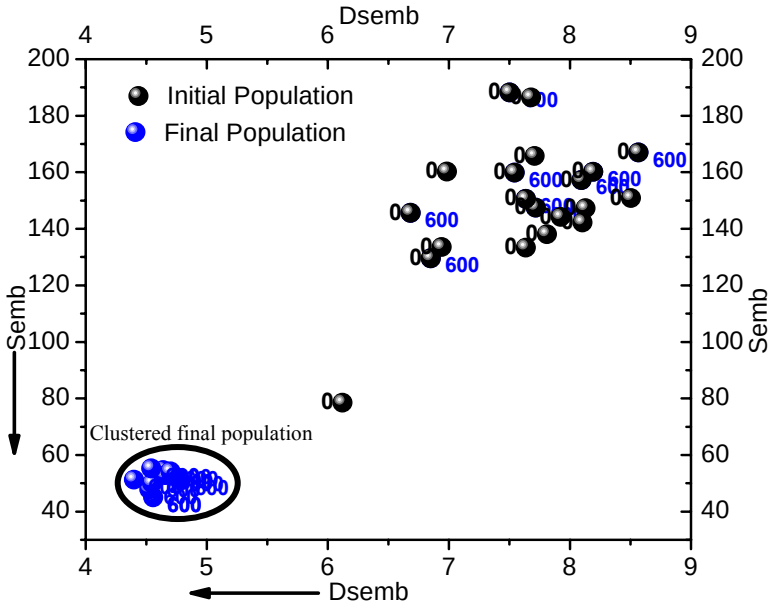


Figure 6.5: *Initial (semi-random) population (black balls marked with zero) and final population after 600 iterations (blue balls marked with 600) in the objective space. In the final population all the replaceable sub-population is converged and make a cluster (inside oval), whereas non-replaceable sub-population are not changing (marked as 600 blue colour, outside the oval) in the population and maintaining the diversity in the parameter space (for 250 grid sampling).*

The next important issue is the generation of the initial population. Generally, evolutionary algorithms start from random models. Such generation of completely random models is suitable for those problems where one does not have a prior information, knowledge and experience about the models. Initial population is an excellent place to embed knowledge from the problem domain. As a result a smart initial population can increase the likelihood of successful composing of the global solutions through the iterative process of information exchange.

In our approach, we have prior information about the possible velocity from geological studies, well logs and seismic preprocessing along with the well known fact that velocity generally increases with depth. Besides, we have information from gathers. Therefore we

need to generate approximate solutions by leveraging the resources that are generally available, i.e. domain knowledge and field data that characterize the system behavior. Consequently generation of knowledge and experience based semi-random models is more suitable. This process will speed up the convergence in addition to providing information to the system However, it should be pointed out that small populations with proper management, information and diversity are sometime better or as good as larger populations.

An example of semi-random model is shown in Figure 6.6 for optimizing Marmousi model. For generating this random model, we increase the upper bound of random velocity with depth. This is one the simplest information that we can add. A lot of velocity information can be extracted from the preprocessing of the seismic data, well logs and regional geology. We can also extract structural information from regional geology and time migration of the data. Both structural and velocity information can be added together in velocity generation. Despite the fact we are using only velocity information for Marmousi model optimisation. Addition of structural information as well as velocity information can be a very robust choice for salt body velocity optimisation.

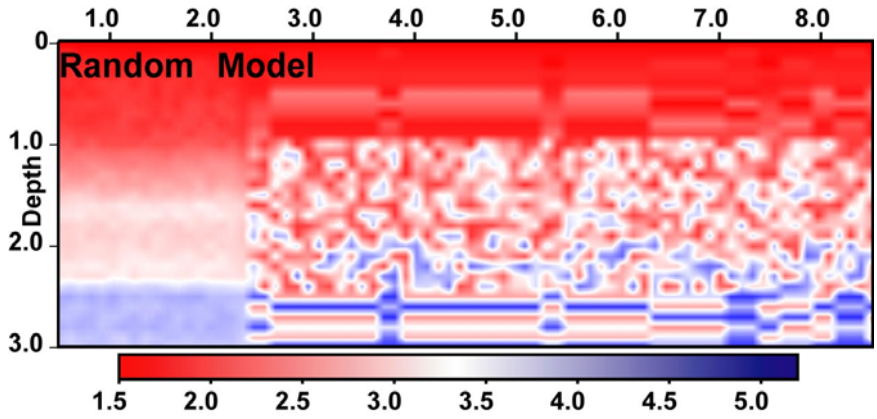


Figure 6.6: Semi-*random velocity model, upper bound of velocity is increasing with the depth. We are adding the well known domal knowledge that velocity increases with the depth.*

After generation and evaluation of initial velocity models, the process of evolution start which is based on the Darwinian principle of natural selection and blind variation. In MOEA variation operators are crossover and mutation. Since we have a lot of information about each velocity model, we decided to use the information, to guide the crossover operation. In the next section we are presenting this guided crossover.

6.6.2. Guided crossover

The philosophy of MOEA is that good parent models should generate good offspring models with positive probability. The offspring is generated by the crossover of parents and the main purpose of a crossover operator is to recombine partial good information from two or more models so as to generate possible better offspring models. Conventionally models for crossover are selected from a population by some stochastic techniques (tournament selection, roulette wheel selection etc.). As we have a small number of models in the population, in which part of the models is non-replaceable, selecting a good solution each time for crossover will not allow us to generate a different solution. Thus traditional selection criteria of a model may lead to premature convergence or reduce the convergence speed drastically.

In order to find the good models for crossover from a population, we decided to synthesize a parent model from a few randomly selected parent models. Among the many strategies that could be used to synthesize parent models, we have adopted two strategies: vertical and horizontal synthesis of parents. The process of vertical synthesis of parent has been already described in section 6.5.5 (Figure 6.4). In a similar way one can prepare a horizontal synthesized parent model.

6.6.2.1 Horizontal synthesis of Parent

Horizontal synthesis of a parent is based on the fitness of each horizontal depth windows of a gathers (Figure 6.7) in place of each gather. To synthesis a parent model, first we randomly select four models from the population then we smooth them along the dip direction, and also update them from γ (RMO) information. The semblance (LS) or differential (LDS) of each depth windows is used to synthesis a parent model (here L

stand for each depth window). We copy the velocity model corresponding to the best LS or LDS windows into the synthesized parent model (Figure 6.7).
We performed few experiment with both vertical and horizontal synthesis of parents. The combined synthesis has improved the convergence speed almost by a factor of two. Though it still require some more experiment to decide the factor of improvement. However it is sure that, mixture of both vertical and horizontal syntheses of parents brought better convergence than the single synthesis of parents (vertical or horizontal).

Once we have synthesized a parent from four parents either by horizontal or vertical synthesis we apply simulated binary crossover (SBX) (Deb and Agrwall, 1995). SBX is implicitly adaptive crossovers that generate offspring close to the parents for near parents, and spread offspring further from the parents if they are far away, automatically reaching a balance between exploration and exploitation. Hence specifically they do not require a mutation operator. By synthesizing one parent from several parents we have Increasing the probability of creating a parent of good attribute. Hence, if the parents of good attributes do the crossover they have a good chance to generate a good offspring. In this way just by using the philosophy of MOEA, we guide the crossover to the right direction without disturbing the exploration property of MOEA. The whole process of guided crossover is shown in the Figure 6.8.

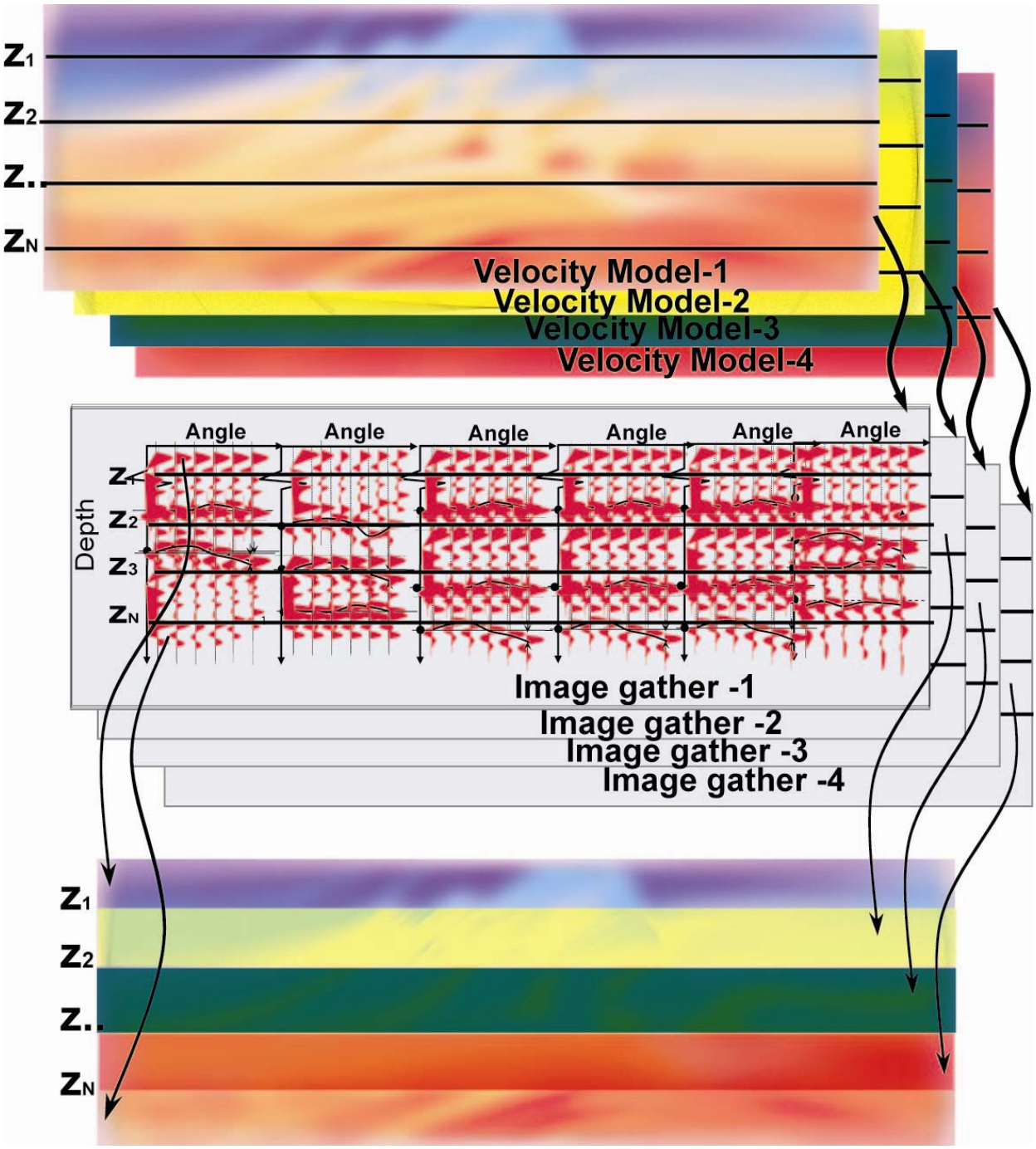


Figure 6.7: *Horizontal synthesis of a parent model can be done by copying the velocity models corresponding to the best horizontal panels (it may be based on semblance or differential semblance criteria of corresponding depth windows).*

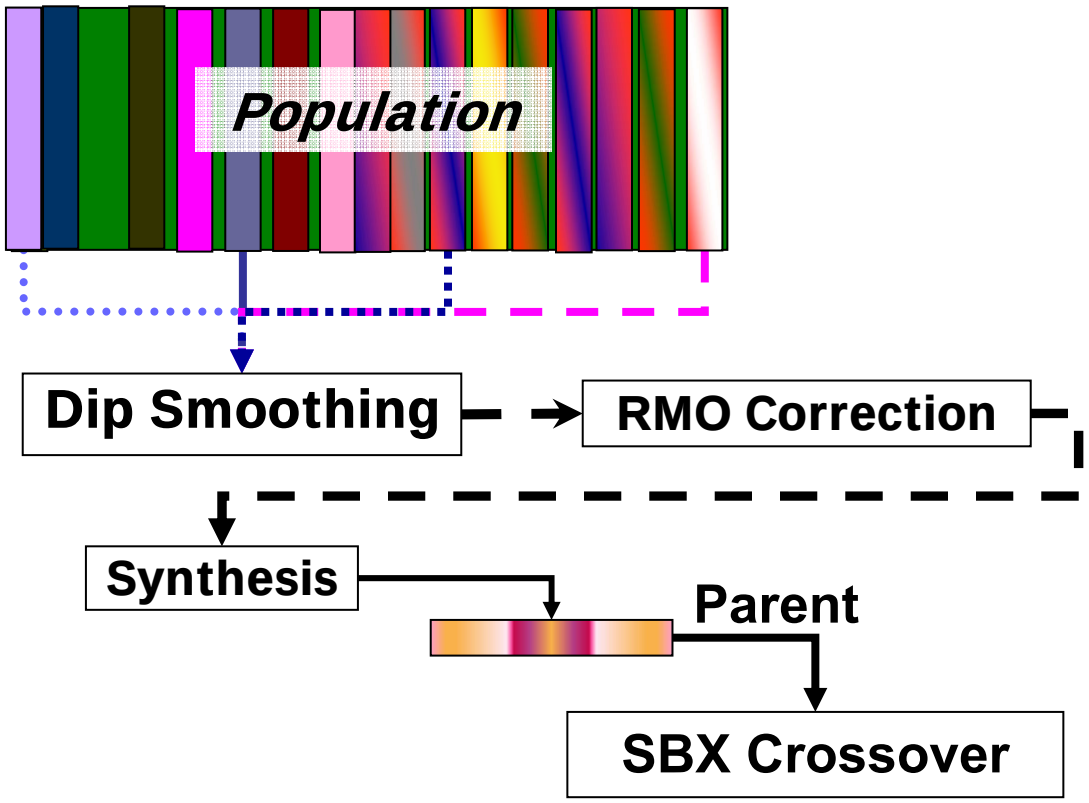


Figure 6.8: *The whole process of guiding the crossover. We uniformly select four parents velocity models from population and then we apply dip smoothing and RMO correction to each selected models. Then we synthesized single parent velocity model from these four parents' models (see Figure 6.4 and 6.7). We use synthesized parent model and archive parent models to SBX Crossover to generate an offspring. The whole process is called guided crossover.*

Though fixed sub-population is taking part during the synthesis of a new parent, we also crossover them separately with archive population. Since synthesis is based on the criteria of local fitness of a model, it is also necessary to do the crossover with global good and bad models. Crossover of non replaceable sub-population (random models) with archive models can help in producing a diverse model and maintain the diversity. Crossover with two reference models (collection of best velocity panels) will add the positive attribute to the offspring models and help in converging toward the global solution.

A guided crossover explores good regions by searching good gathers from randomly selected four models and synthesizing a parent. It exploits the good attributes of reference models and produces diversity by using the fixed sub-population. Therefore our crossover operator has all the good attributes required for a good crossover operator. Next section will present results based on the synthetic examples of Marmousi velocity model (Bourgeois et al. 1991) and one real field data.

6.6.3 Reference Models

The guided crossover operator is helping in generating a good offspring, however we also need some operator which can exploit the locally good properties of offspring and help in generating a good velocity model. We developed reference models to exploit these locally good velocity models. The process of creation of reference models are described in section 6.5.6. We created two reference models which are based on the collection of velocities corresponding to the best semblance(S-Reference) and differential semblance (*DS-Reference*) gathers.

Example of reference models for 250m and 100m grid sampling are shown in the Figures 6.9 and 6.10. Figure 6.9a shows the trapping of velocity corresponding to the best semblance panel whereas Figures 6.9b shows trapping velocity corresponding to best differential semblance panel for 250m grid sampling. In the beginning (0_{th} iteration) reference models are similar to random model. However reference models start improving with the number of iterations. Semblance reference (LS) model (Figure 6.9a) seems more sensitive to near surface velocity variations whereas differential semblance reference (LDS) (Figure 6.9b) sensitive for deeper part of the models. LDS reference models; strong sensitivity for deeper part of the model is due to "depth factor" that we have introduced in Chapter 5.

Figure (6.10a and 6.10b) shows the trapping of velocity corresponding to the best semblance and differential semblance panel for 100m grid sampling with the number of iteration. Reference models start improving since the beginning of evolution however the

process of improvement is slower than that of the 250m grid sampling. As reference model improves it also get smoothen. Here also we can see strong sensitivity of LDS for deeper part whereas LS shows strong sensitivity for shallow part of the model.

Concept of reference modes to capture locally good velocity models is similar to the Darwinian principle of survival of fittest, where best obtained velocity model corresponding to panel survival. This operator exploits the useful information that it generates during the optimisation, and adds exploitation property in MOEA. Hence it fulfills the requirement of exploitation operator in MOEA using almost negligible computation cost. Reference models also provide virtuous seeds for generating a new velocity model. As a result, it helps to increase convergence speed.

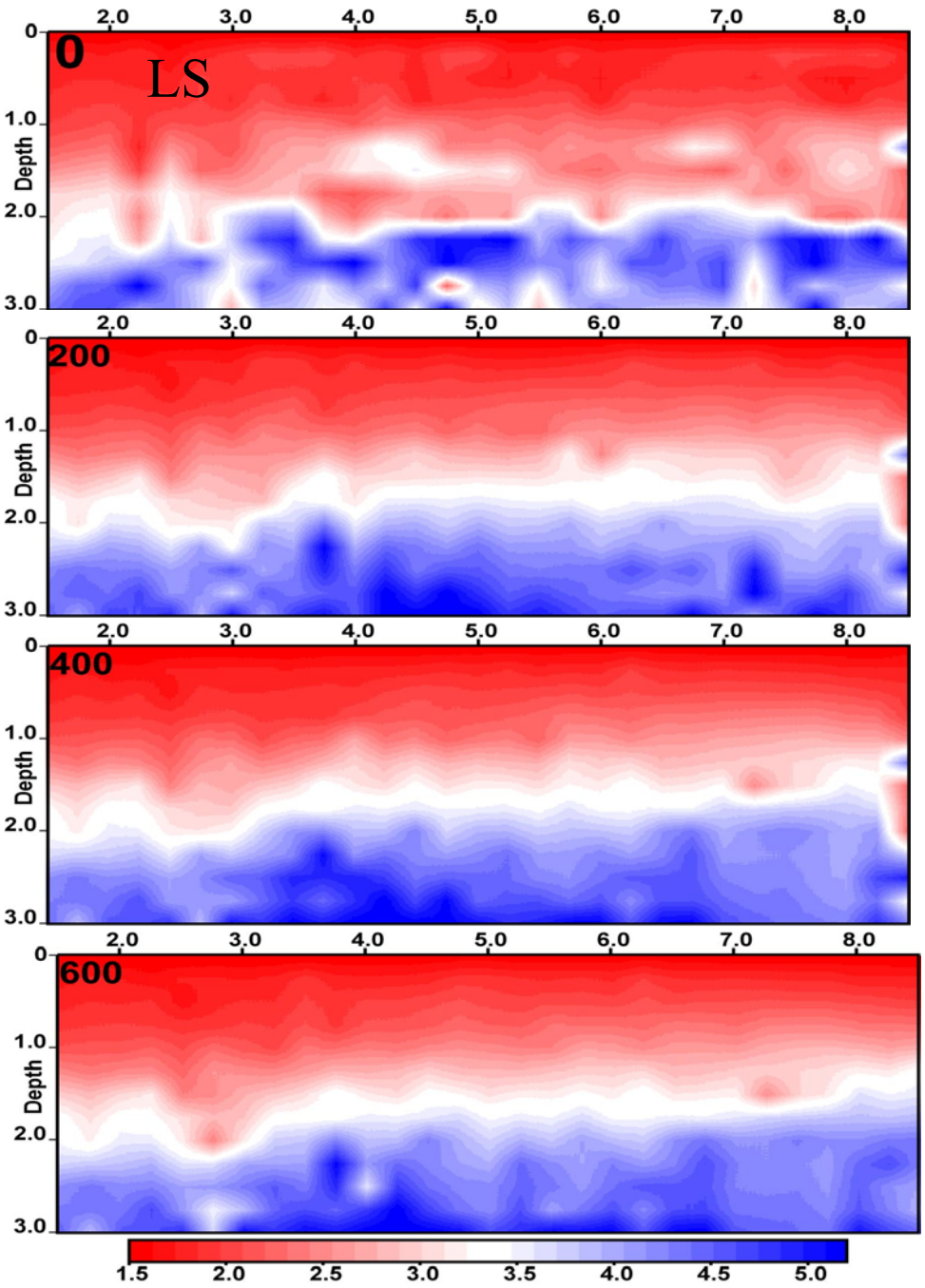


Figure 6.9 (a): *LS Reference model: capture the velocity corresponding to the best semblance panel (LS) (250m grid sampling). In the beginning (0th iteration) reference model is similar to the random model, it start improving with the number of iteration (100, to 600). It seem more sensitive for near surface velocity variation.*

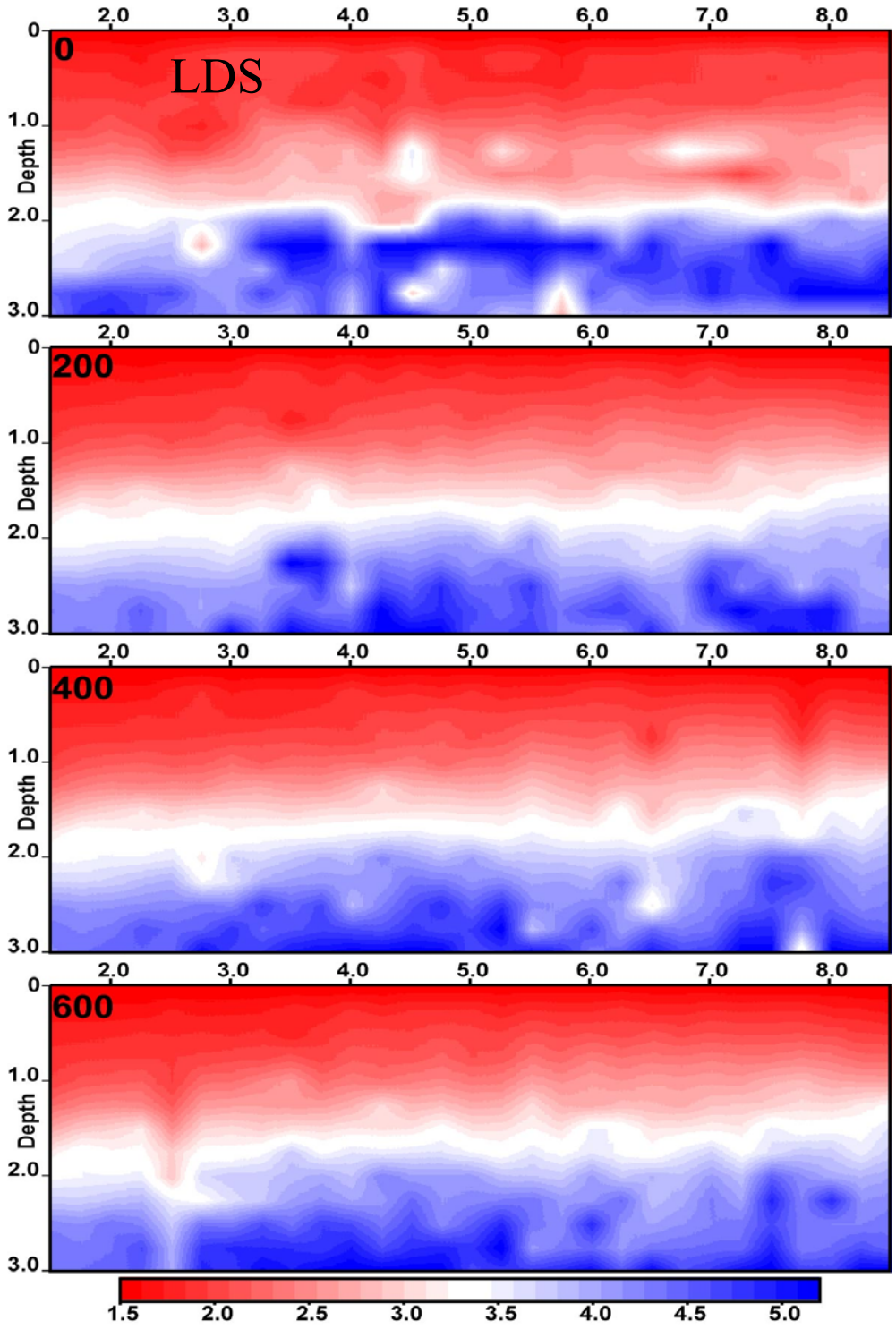
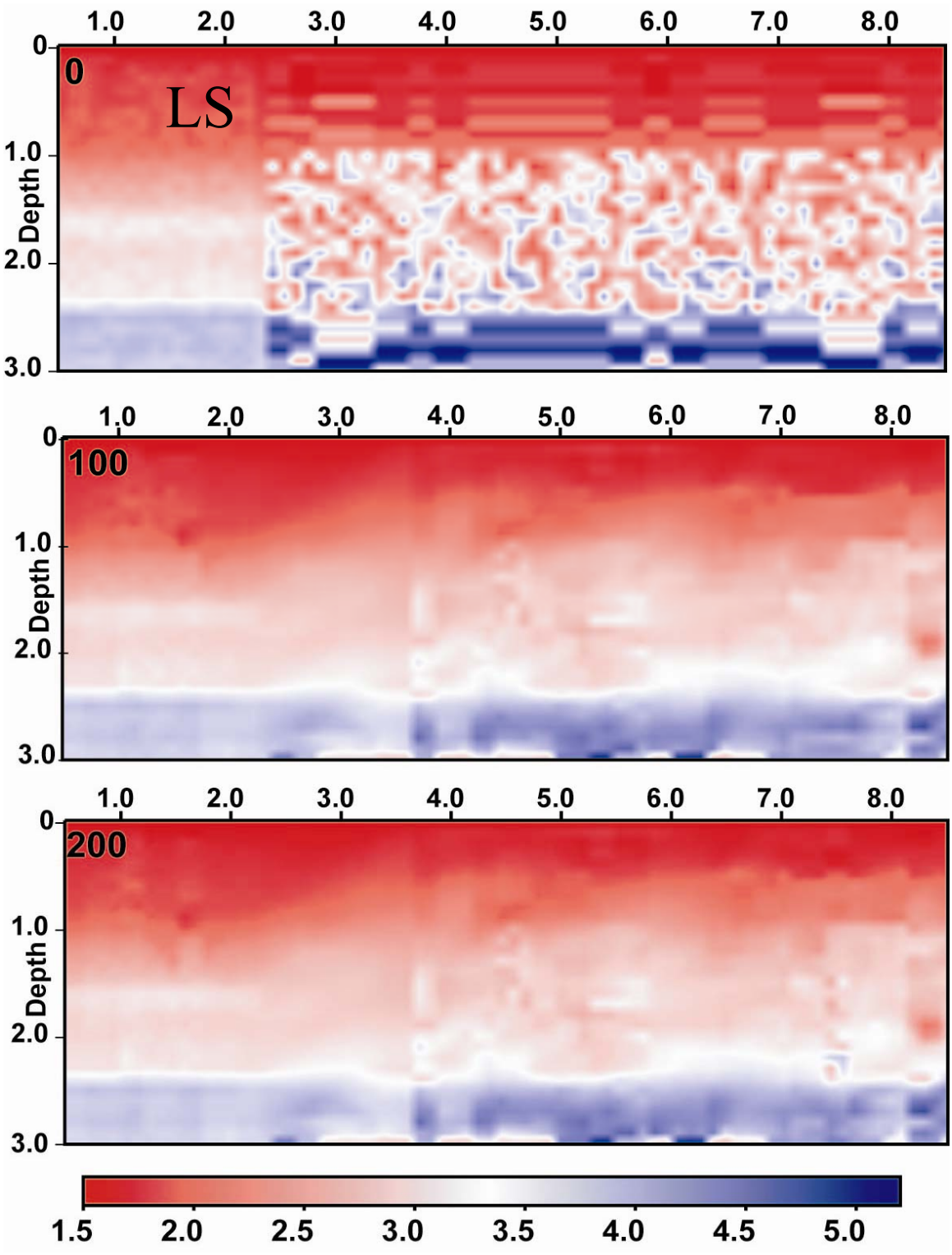


Figure 6.9 (b): *LDS Reference model: capture the velocity corresponding to the best differential semblance panel (LDS) (250m grid sampling). In the beginning (0th iteration) reference model is similar to the random model, it start improving with the number of iteration (100, to 600). It seems more sensitive for deeper part of velocity model.*


1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
0
LS
Depth
1.0
2.0
3.0
100
200
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

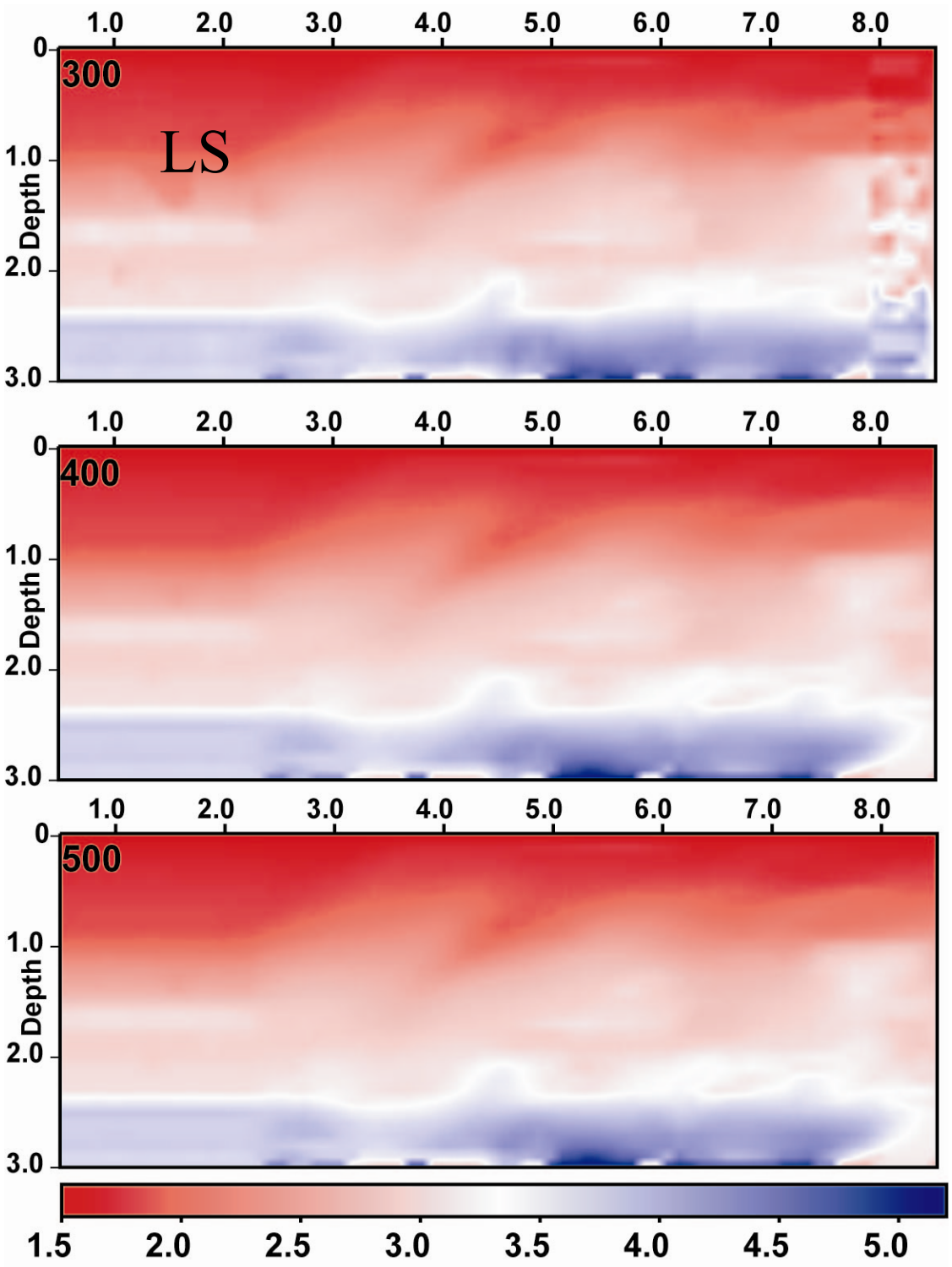
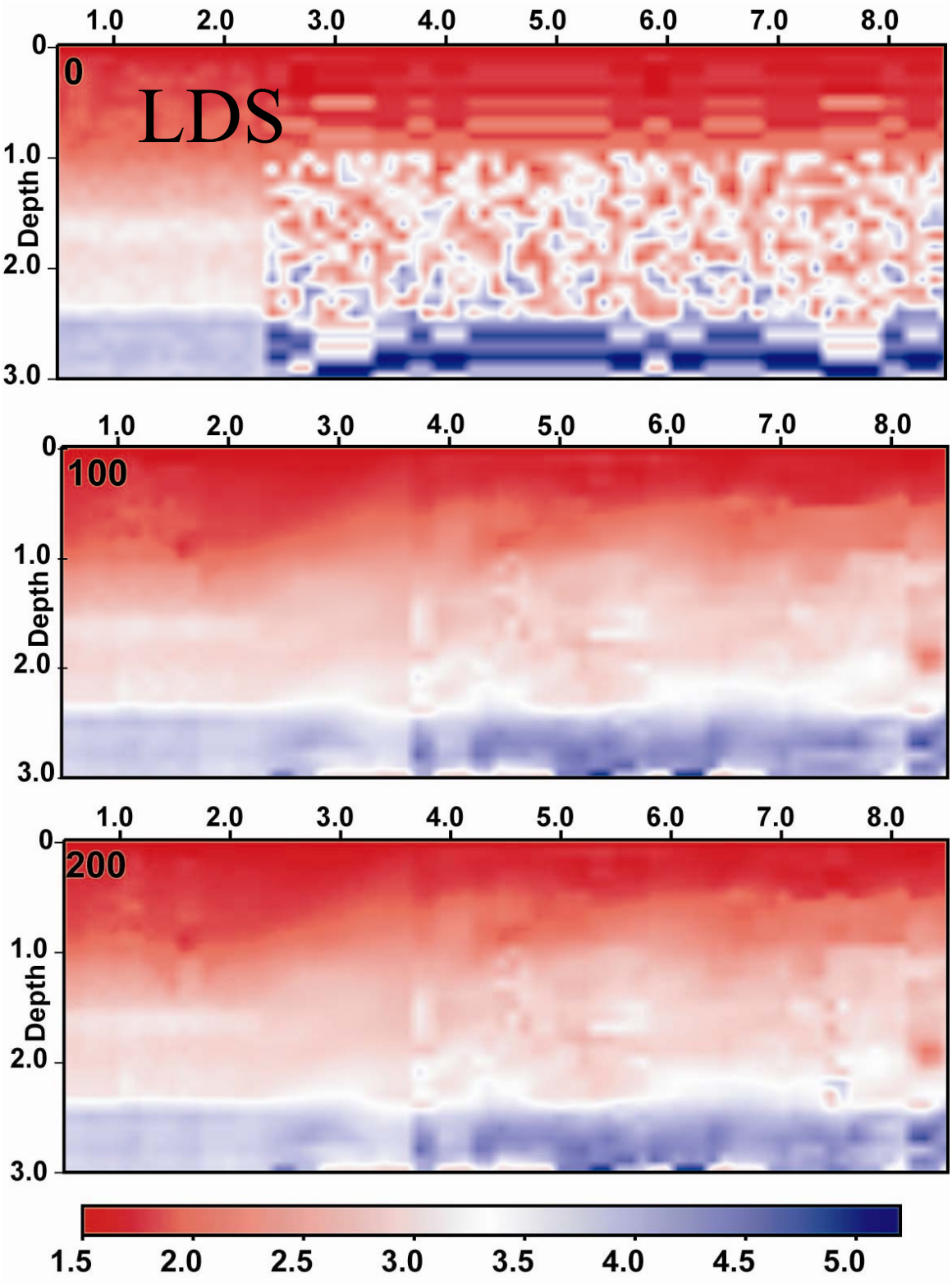


Figure 6.10 (a): *LS Reference model: capture the velocity corresponding to the best semblance panel (LS) (100 m grid sampling). In the beginning (0th iteration) reference model is similar to the random model, it starts improving with the number of iteration (100, to 600). By visualizing it seem more sensitive for near surface velocity variation.*


LDS
0
100
200
Depth
1.0
2.0
3.0
4.0
5.0
6.0
7.0
8.0
0
1.0
2.0
3.0
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0

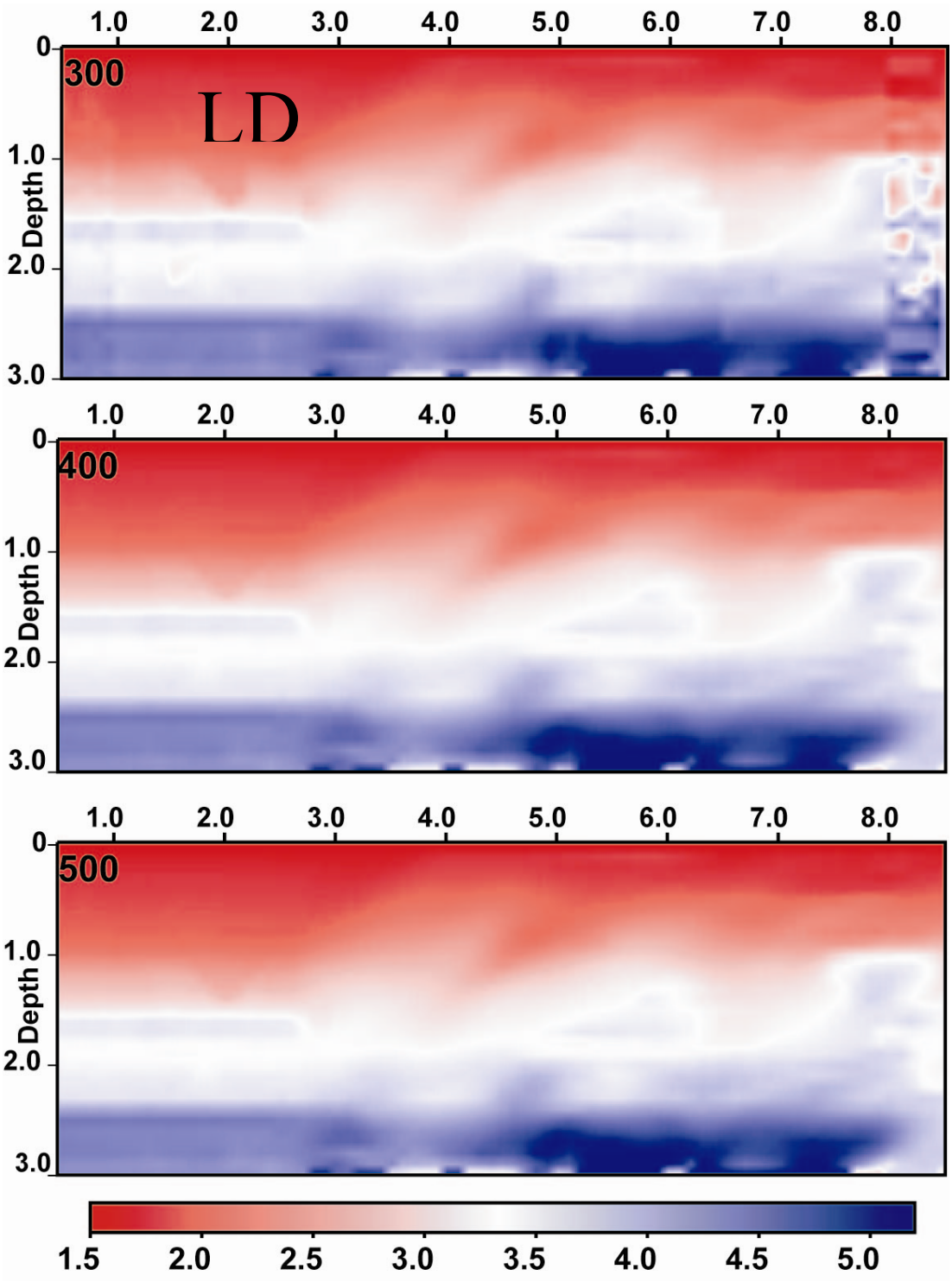


Figure 6.10 (b): *LDS Reference model: capture the velocity corresponding to the best differential semblance panel (LDS) (100m grid sampling). In the beginning (0th iteration) reference model is similar to the random model, it starts improving with the number of iteration (0, to 500). By visualizing it seems more sensitive for deeper part of velocity model.*

This concept can also be extended for constructing a reference models based on horizontal panel, similar to the synthesis of horizontal synthesis of parents (Figure 6.7). For constructing horizontal panel reference models, one need to compute semblance and differential semblance for each depth windows (Figure 6.7) of a offspring and then to compare with the horizontal reference model and copy the velocity corresponding to the best offspring depth windows into the reference models. Here also we can prepare at least two types of reference models based on the velocity model corresponding to both semblance and differential semblance function. Though we have applied this concept but we did not performed enough test. Therefore we are not putting any results. This concept can also be very useful in 3D, where we need to apply a layer striping approach (as discussed in Chapter 7).

6.7. Results

We are demonstrating our results on synthetic Marmousi model and North Sea L7 real data. Here first we are describing the algorithms parameter setting and then we show the results one by one.

6.7.1. Evolutionary Algorithms and parameters

To optimize any velocity model, there are three types of parameter settings. The first type is related to the representation of model, second type is related to the migration and third type is related to optimization, here MOEA. To optimize the velocity model of Marmousi data and L7 data, we represent it on regular grid as discussed in Chapter 4.

The MOEA related parameters are the number of models in population, crossover rate and coefficient of crossover and ε-parameters in objective space. We have taken a population of 20 to 30 models and with almost equal probability for cross over each with synthetic parents and normal parents. The ε parameters are related to the objective functions and divide the objective space into grid and provide an opportunity to fine tune the objective function according to their importance see Chapter 2 (section 2.7.7.9) and figure 6.11. The DS objective function is very sensitive when the initial models are far from the true models but its sensitivity fades out when models converge toward the true model. As a result, a coarse sampling (Figure 6.11) is chosen for DS function, thereby

resulting in shorting bad models and fast convergence. Conversely the S function is very sensitive when the models are close to the true one. Hence a fine sampling (Figure 6.11) is chosen for the S function. It can decide on the best model and converge quickly when the models are close to the solution. Although both S and DS functions measure the flatness criteria, a clever combination of the distinct properties of both functions with ε-MOEA reduce the computation cost.

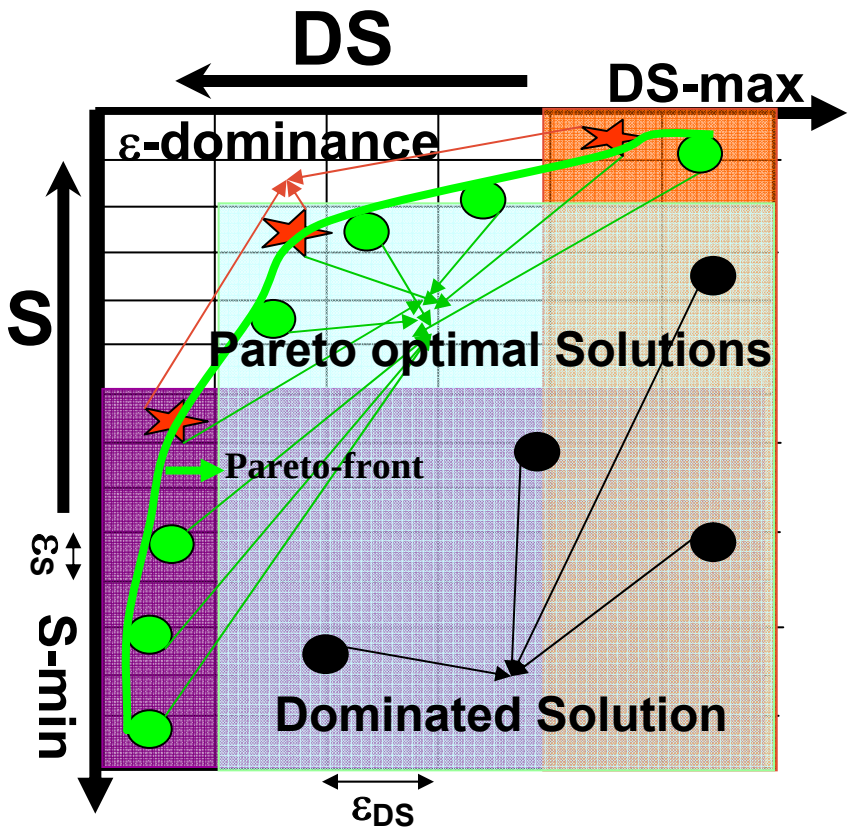


Figure 6.11: *Represent the ε-dominance concept and grid of the objective functions S and DS. The points represent the solutions in objective space. The solution shown by green points is a non-dominance or Pareto optimal solution and the green curve is Pareto-front. The red color stars represent the ε-dominance solutions. The black color points are dominated solution.*

For optimization, firstly, the velocity model was divided into many horizontal blocks. In these blocks, random velocity was generated such that the minimum velocity increases

with depth with maximum velocity limit of up to 5500m/s. Migration is then performed on these velocity models and the corresponding S, DS, LS and LDS functions, RMO and smoothing dips are calculated. The ε-dominate models are copied into the archive population. The LS and LDS function are also used to create the reference models. First reference model is created from population models by the combination of panels of good LS objective function and their velocity. Second reference model is created by the combination of panels of good LDS objective function and their velocity. After the generation of initial population, archive population and reference models, the process of optimization starts. The optimization procedure is shown in the (Figure 6.3). In this optimization process we have not used the mutation operator as it perturbs a solution obtained by the crossover in the hope of creating even better solution and also maintains the diversity of the solution. Since we have maintained a set of random models in the population to preserve the diversity, there is no need to have mutation operator. One other reason of not using the mutation operator is the use of SBX crossover operator, which also have desired property of mutation. Hence only crossover operator is used for optimization in this work.

We are using both original ε-MOEA and customized ε-MOEA. The ratio of apppling both algorithms for generating a offspring is 40:60. In customized ε-MOEA we take four parents and then apply RMO correction and dip smoothing on the models and then synthesize a single new parent. This new parents is a combination of panels of good LS or LDS functions and their corresponding velocity model. After synthesizing this parent, SBX cross–over is performed between the synthesized parents and archive selected parents. The SBX crossover is also used for the ε-MOEA. In the ε-MOEA we select one random parent from the population or from reference models and crossover it with the archive solution. After the crossover, new velocity models are generated and migration is performed on these new generated models. Once migration is performed we have the migrated image and gathers (offset and angle). Migrated image is used for visualizing and gathers are used for the S or DS calculation. Here the migrated image is used to get the information about the layers dip direction as well as dip smoothing. This process will reduce the variance of the velocity along the layers thereby producing a smooth image.

The gathers obtained after migration used for the RMO error estimation and LS and LDS are calculated for each panel. We associate the RMO errors and dip smoothing information with the models and have been used before crossover. After few hundreds of iteration we end up with optimized velocity models. Results of velocity optimisation on 250m and 100m grid sampling are presented in next subsection.

6.7.2. Marmousi Velocity Model

We have taken a complicated example of 2D-Marmousi velocity model to demonstrate the robustness of the approach. The Marmousi synthetic data set (Bourgeois et al., 1991) was first released as a blind test for velocity estimation. It is a complicated acoustic 2-D data set based on a profile of the North Quenguela through in the Cuanza Basin in Angola Versteeg (1993). The structural style is dominated by growth faults that arise from salt creep and cause the complicated velocity structure in the upper part of the model. The target zone is a reservoir located at a depth of about 2500 m. The model contains many reflectors, steep dips, and strong velocity variations in both lateral and vertical directions (with a minimum velocity of 1500 m/s and a maximum of 5500 m/s). The synthetic data set consists of 240 shots with 96 gathers each. Zero-phase source deconvolution was applied to the data used in this study as part of the preprocessing (Barut et al., 1991).

Here we are presenting our results using different grid sampling one on 250x250 m and the other one 100x100 m. The number of grids points is 432 for the 250 m grid sampling and 2700 for 100 m grid sampling.

The main migration parameters are the frequency range, the number of shots, the propagation depth sampling, the maximum subsurface offset and the size of gathers. These migration parameters control the migration cost. Since global methods require a large number of evaluations, reduction of migration cost will help in reducing the optimization cost. In this work, for migration we have used the frequency range 10-25 Hz, every 2^{nd} shot, propagation depth sample of 20 m, maximum subsurface offset of 3000 m and gather at every 250 m. Since we are using every 2^{nd} shots for migration the speed increases by a factor of two CPU time for a typical migration. At the same time, usage of a small frequency bandwidth and a reasonable propagation depth sample also

significantly reduces the migration cost. As small number of shots is used, shots artifacts appear in the gathers. We use here the robust objective function (MADS) which is less affected by the data and parameter setting. As a result of this, these artifacts will have negligible effect on the optimization process.

6.7.2.1. 250m Grid sampling

If the Marmousi model is represented by a coarse grid sampling of 250 x250m grid size and if smoothed using dip information the generated velocity models almost become equivalent to the 500x500m grid size. Only 200-400 hundred evaluation needs to be done for obtaining a velocity models. The computational cost for this 200-400 hundred evaluation is equivalent or less than the cost of 50-100 migration.

Here we are describing an example for 250m grid sampling optimisation. The parameter setting is already discussed. First step of optimisation is generation of random model, where we increase upper limit of velocity with depth. After looping of optimisation start, models gradually begin to improve. Reference models seem very efficient in capturing locally good velocity model (Figure 6.6a and b). In the beginning convergence speed of Archive (Figure 6.12) is very fast, whereas this speed decrease as models converges close to true model.

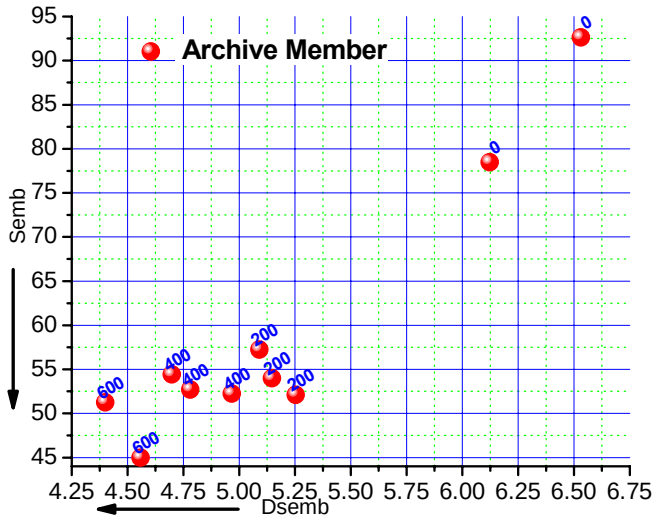


Figure 6.12: *Convergence of Archive with the number of iterations for 250m grid sampling. In the begning convergence is very fast but after few hundred iterations it* slow down.

The evolution of optimised velocity models and corresponding image and gather with the number of iteration for the archive (ε-Pareto optimal solution) (Figure 6.12) are shown in Figures (6.13, 6.14 and 6.15). Only two models from each step are shown. The purpose of showing these results is to give a feeling of the improvement of model with the number of iteration.

In the beginning there is a strong improvement in the near surface velocity model whereas this improvement shift to deeper part of velocity model in later iteration. This can be verified by seeing the improvement in image or gathers. In the complex part of the Marmousi model gathers are also very noisy even though migrated images are quite good. As we have already shown in Chapter 4 that velocity optimisation on a coarse grid may not be able to produced completely flat gather in complex part of the model because complex structures require a fine grid representation to accommodate lateral and vertical velocity variation.

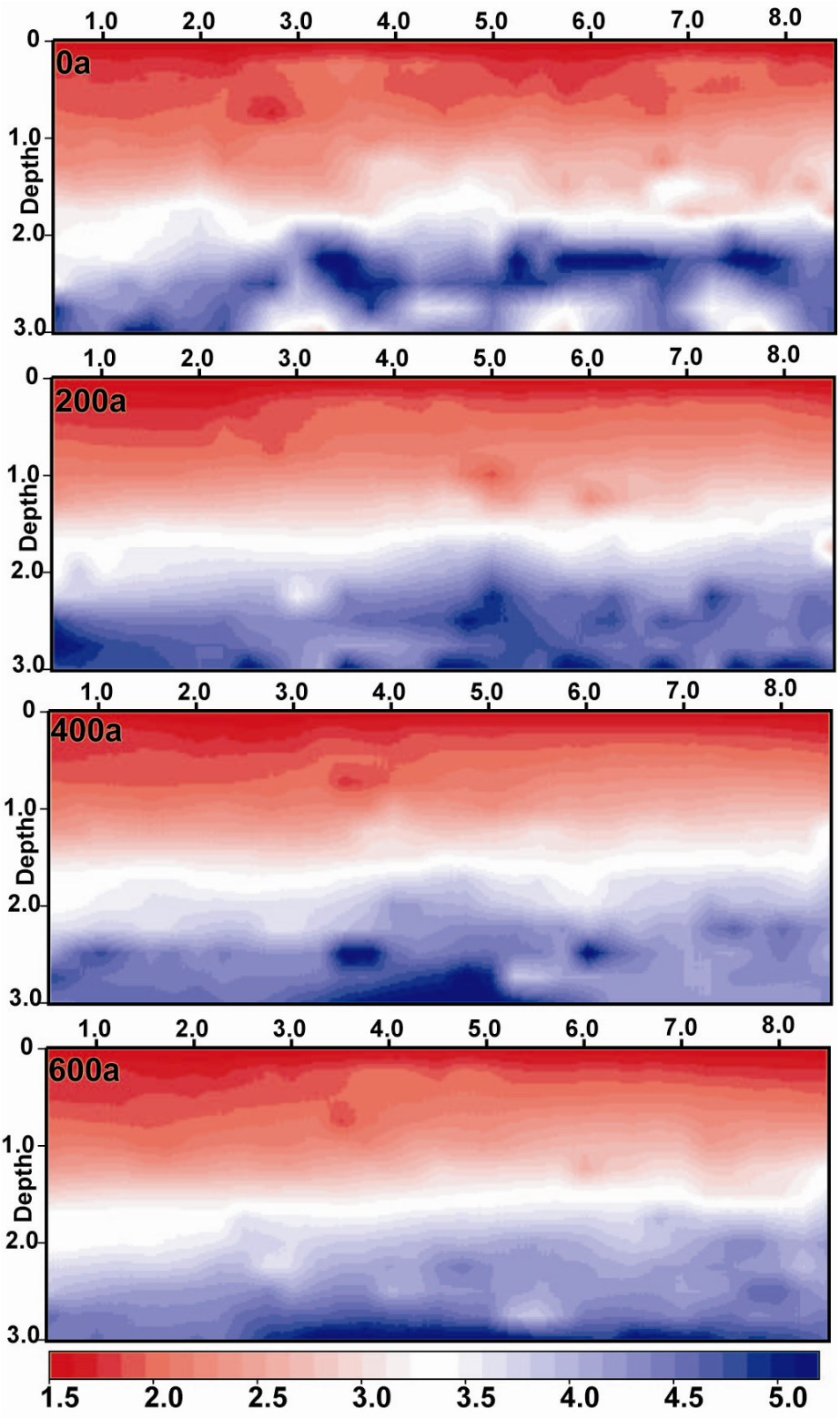


Figure 6.13 (a): *Evolution of velocity model with the number of iteration (first archive solution)*

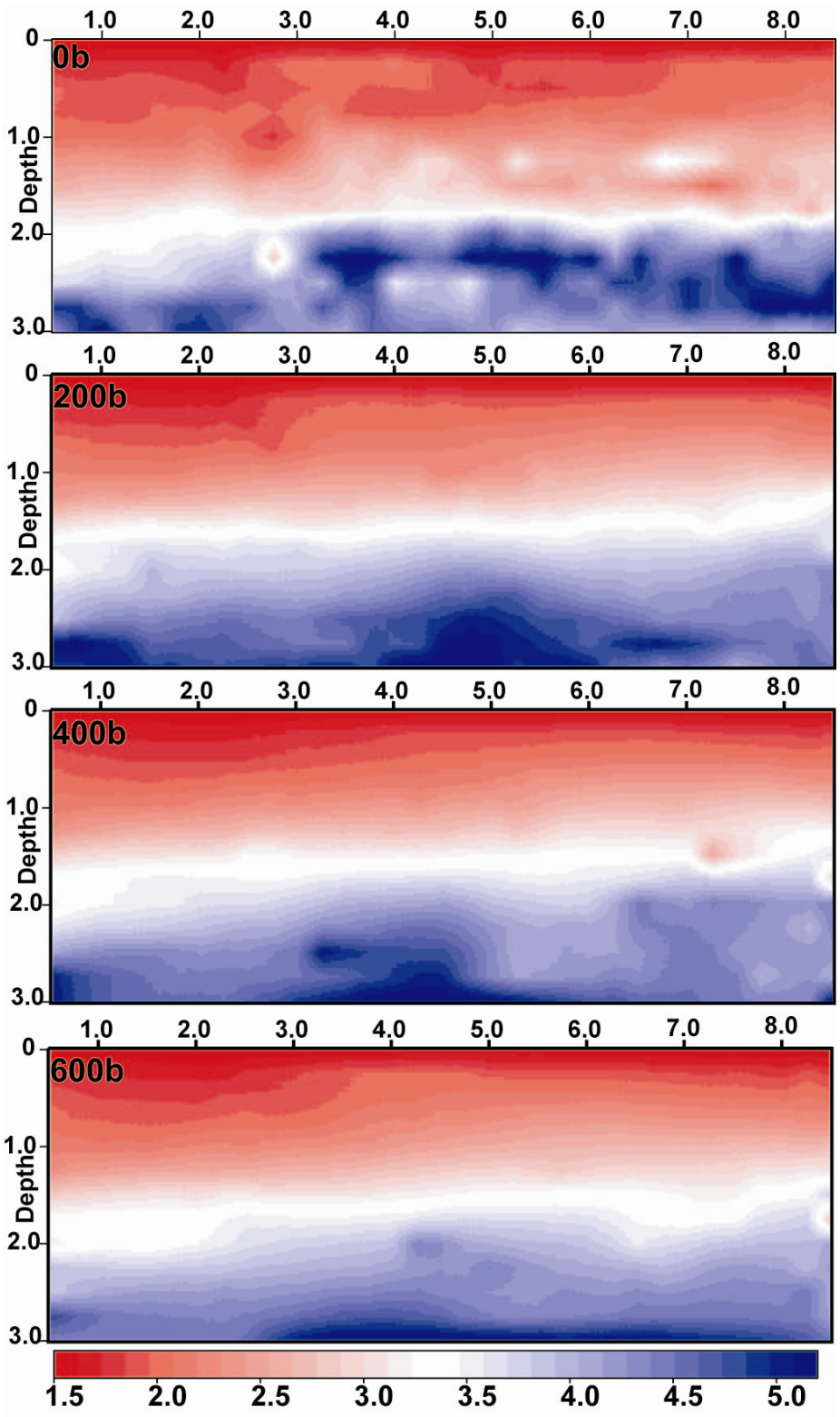


Figure 6.13(b): *Evolution of velocity model with the number of iteration (Second archive solution)*

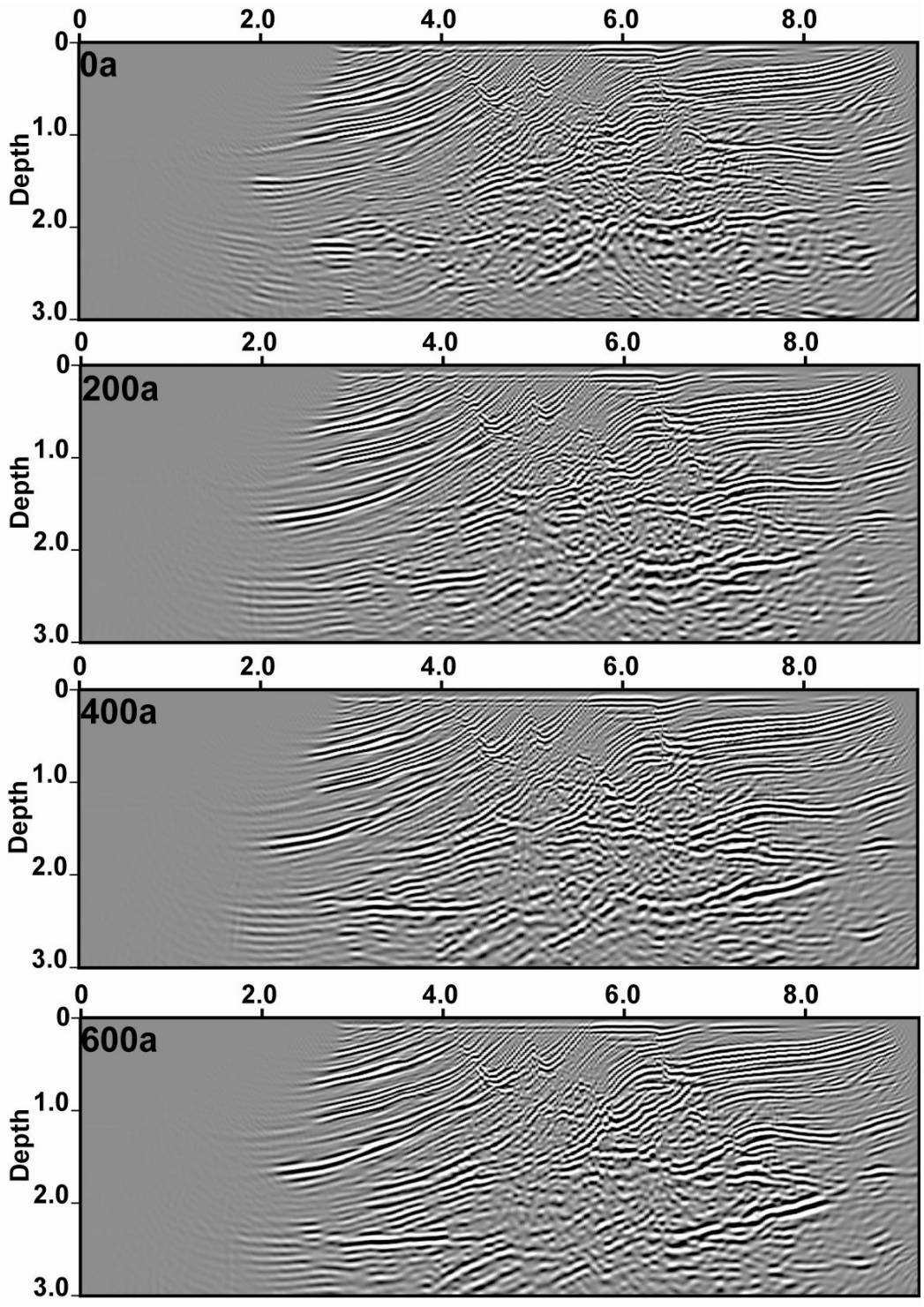


Figure 6.14 (a): *Migrated image obtained corresponding to the velocity model of archive (Figure 6.12)*

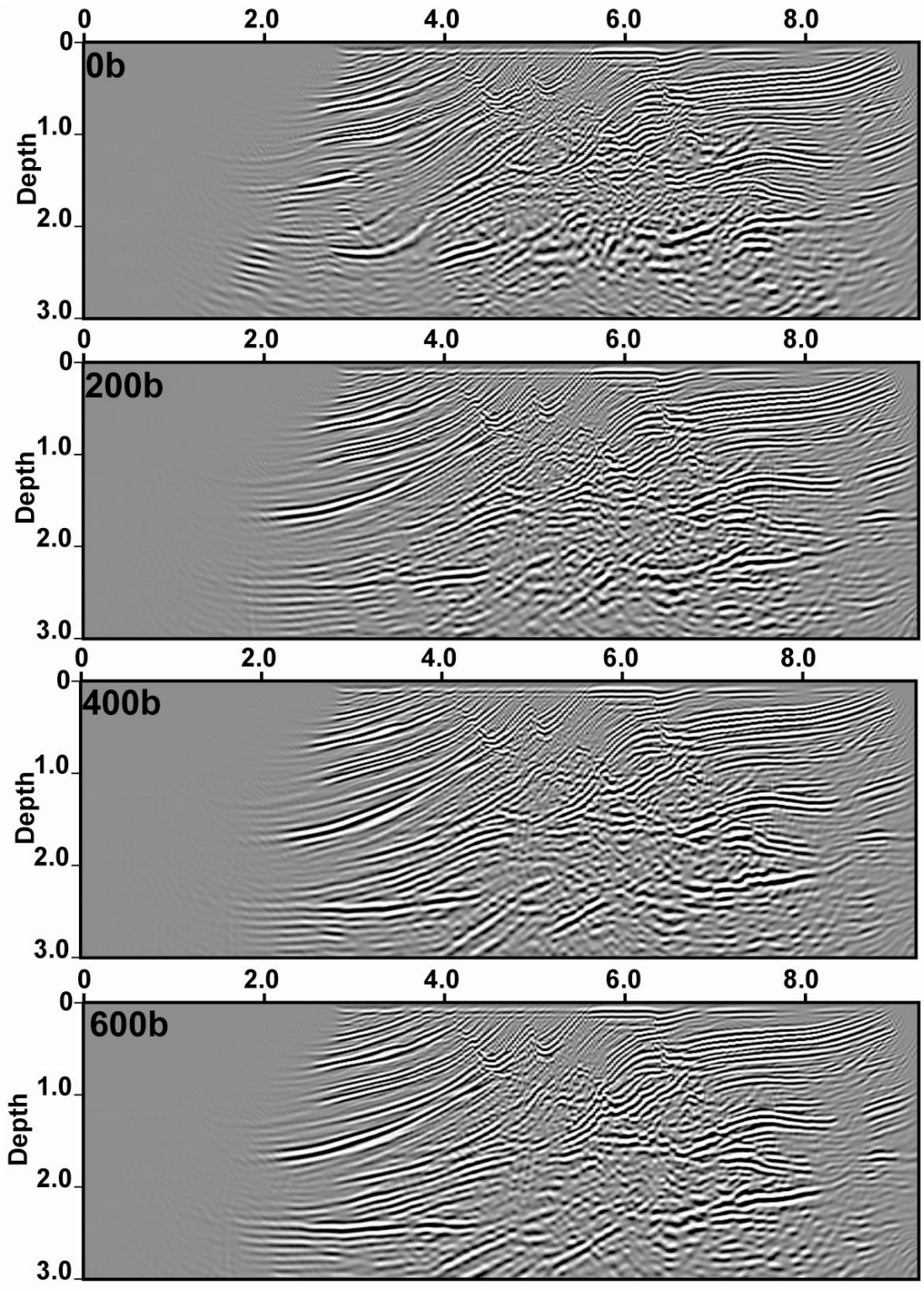


Figure 6.14 (b): *Migrated image obtained corresponding to the velocity model of archive (Figure 6.12)*

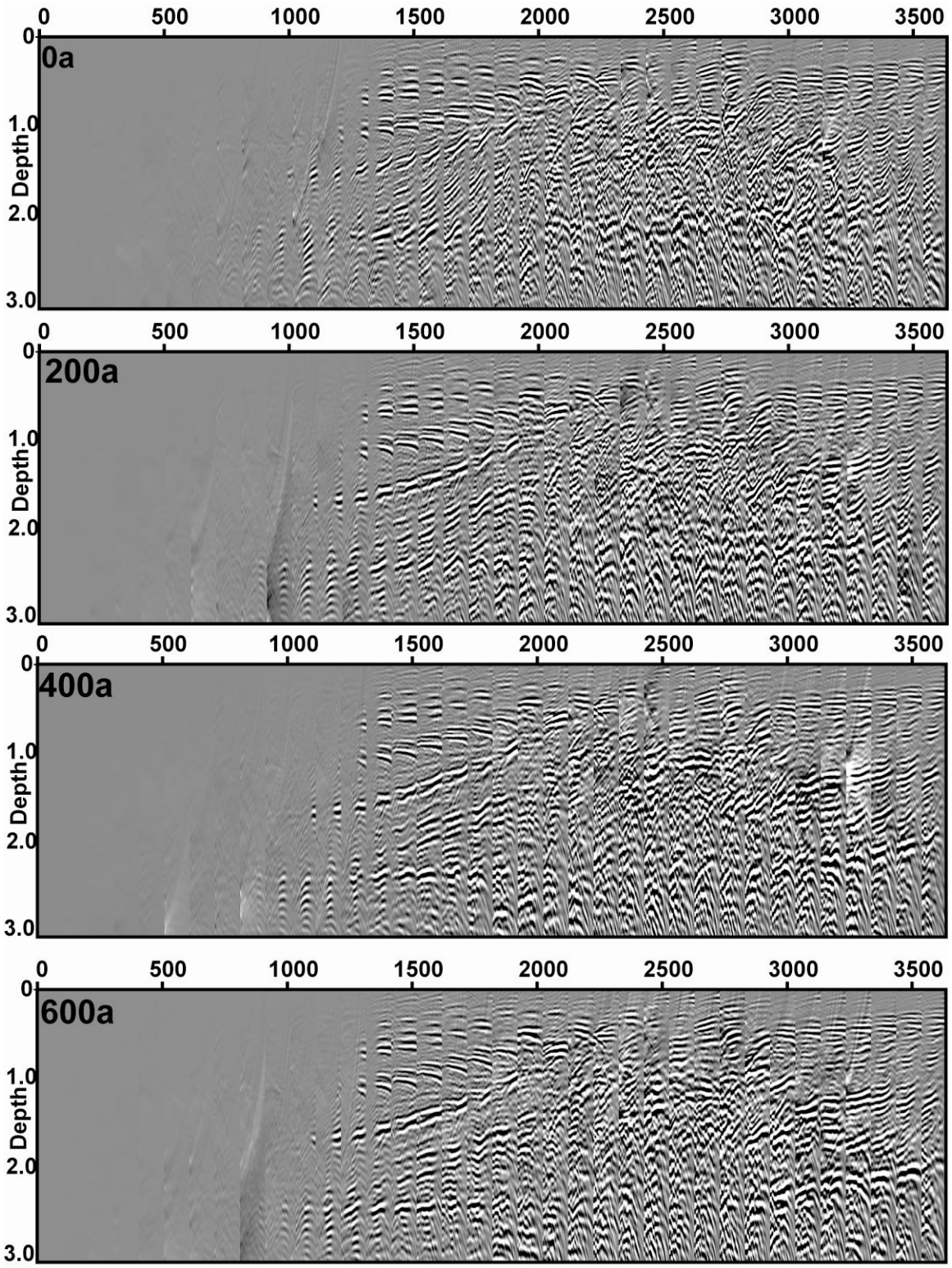


Figure 6.15(a): *Gathers obtained corresponding to the velocity model of archive (Figure 6.12)*

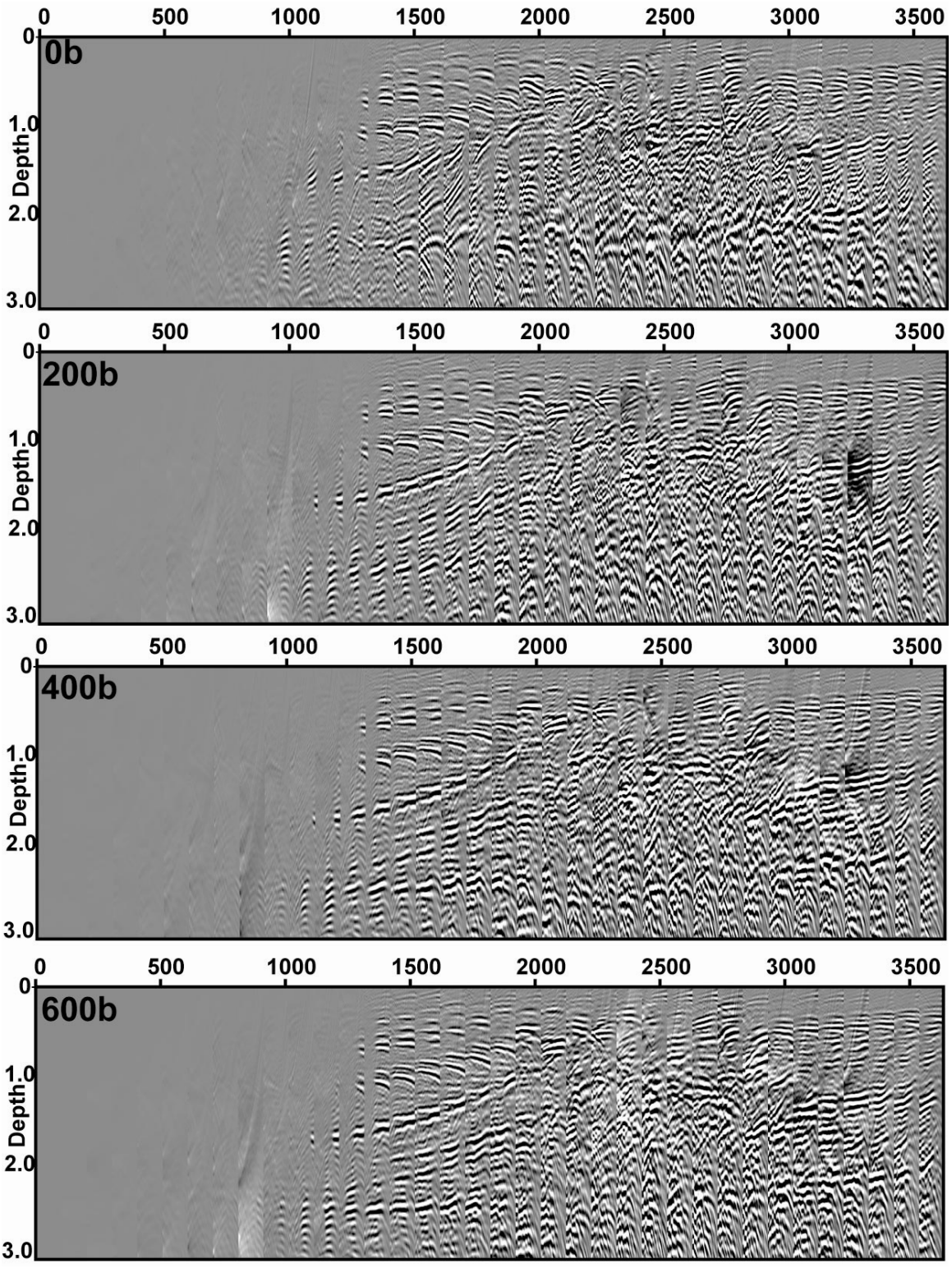


Figure 6.15(b):*Gathers obtained corresponding to the velocity model of archive (Figure 6.12)*

Since velocity improvement gradually moves toward deeper part of the model, it is not essential to do whole velocity modeling in one step. Whole velocity modelings in one step need a large computational cost whereas improvement is very small. Hence once can adopt the layer striping approach to make it faster.

6.7.2.2. Superimposition of migrated image

One of the most interesting properties of MOEA is its ability to produce more than one optimised solution. Hence it provides an option to a user to decide an optimised model on the basis of available information's (numerical, analytical or experience) or one can select partially good information from each optimised model. We noticed that after optimisation obtained velocity model are nearly similar (Figure 6.13a and b). One can take a mean velocity of the optimised model and hope to generate a better migrated image. We noticed those migrated images obtained from this velocity model are similar except around complicated part of Marmousi model. Hence to improve the visualization of the migrated image at complicated part, we made an experiment of merging the optimised migrated image. For example in Figure 6.16, we have merged two optimised migrated image 600a and 600b (Figure 6.14). After merging we are now able to see more detailed structural information even in complex part of the Marmousi model which is difficult to interpret by only one image. Though a geophysicist will prefer to generate a migrated image by taking the mean velocity of optimised model, we preferred image merging approach. Similar approach also used in Remotesensing image processing, where we add, subtract, multiply and divide the images of different frequency band to enhance the certain features.

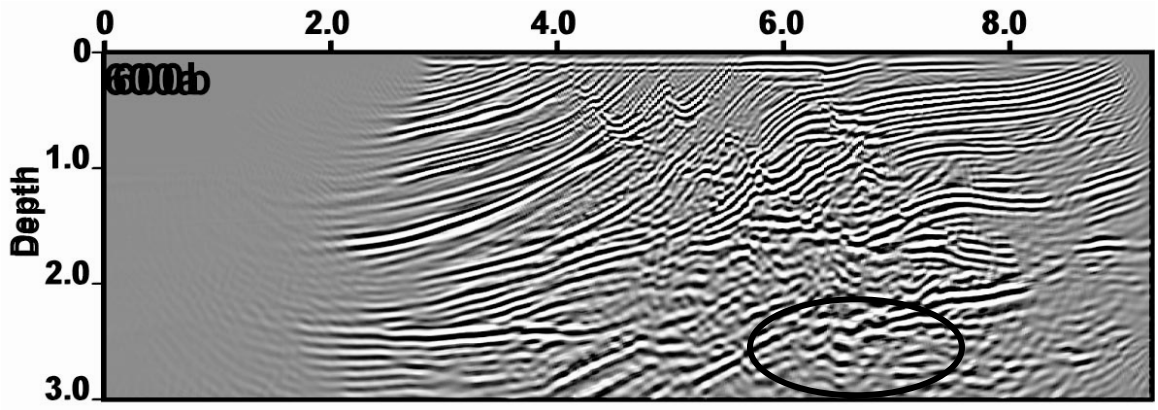


Figure 6.16: *Merging of the two optimised migrated image (600a and 600b). This image can be interpreted even at complex part of the model (ellipse).*

6.7.2.3 100m Grid sampling

100 m grid sampling requires 2700 unknown parameter to represent Marmousi model. Optimisation parameter settings are discussed above. Model generation and other strategy are similar to the 250 m sampling.

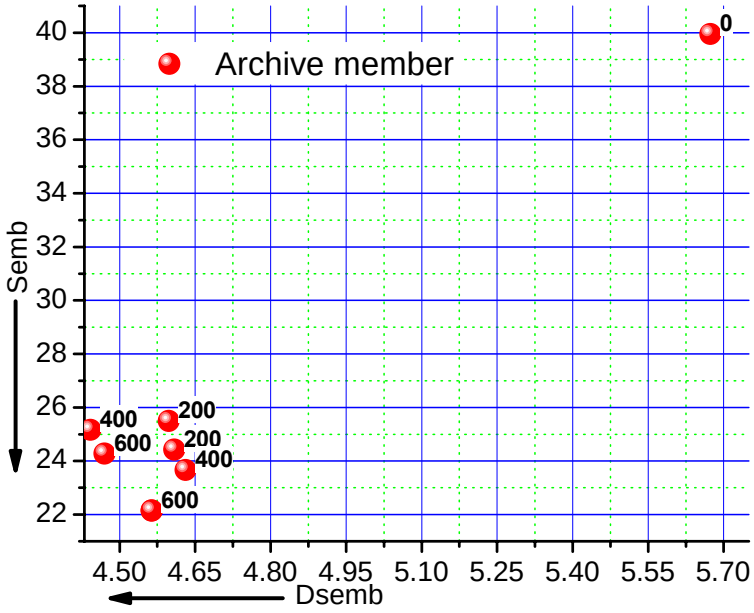


Figure 6.17: *Convergence of archive with the number of iteration for 100m grid sampling. In the beginning convergence is very fast but after few hundreds of iterations become very slow.*

We are smoothing velocity model along the dip direction, which reduced the variance of velocity model. As a result velocity model resolution is approximately 200 m to 300 m. Optimisation requires almost 600-800 evaluations. Increase in the number of evaluation is due to the increase in number of unknown parameters.

Convergence of archive with the number of iterations is shown in the Figure (6.17). Convergence is very fast in the beginning whereas it starts slowing after certain iteration. There are no drastic changes noticed in the migrated image either obtained by 250m or 100m grid (Figure 6.19). Whereas velocity model obtained by 100 m grid (Figure 6.18) gives a geological appealing velocity model. This gives us more confidence in velocity model at extra computational cost.

Here also we obtained more than one optimised solution and provides an option to a to decide an optimised model on the basis of available information's (numerical, analytical or experience) or one can select partially good information from each optimised model. On 100m scale even a single image is itself sufficient to interpret, though merging of the two optimised modem 500a and 500b (Figure 6.20) a better image.

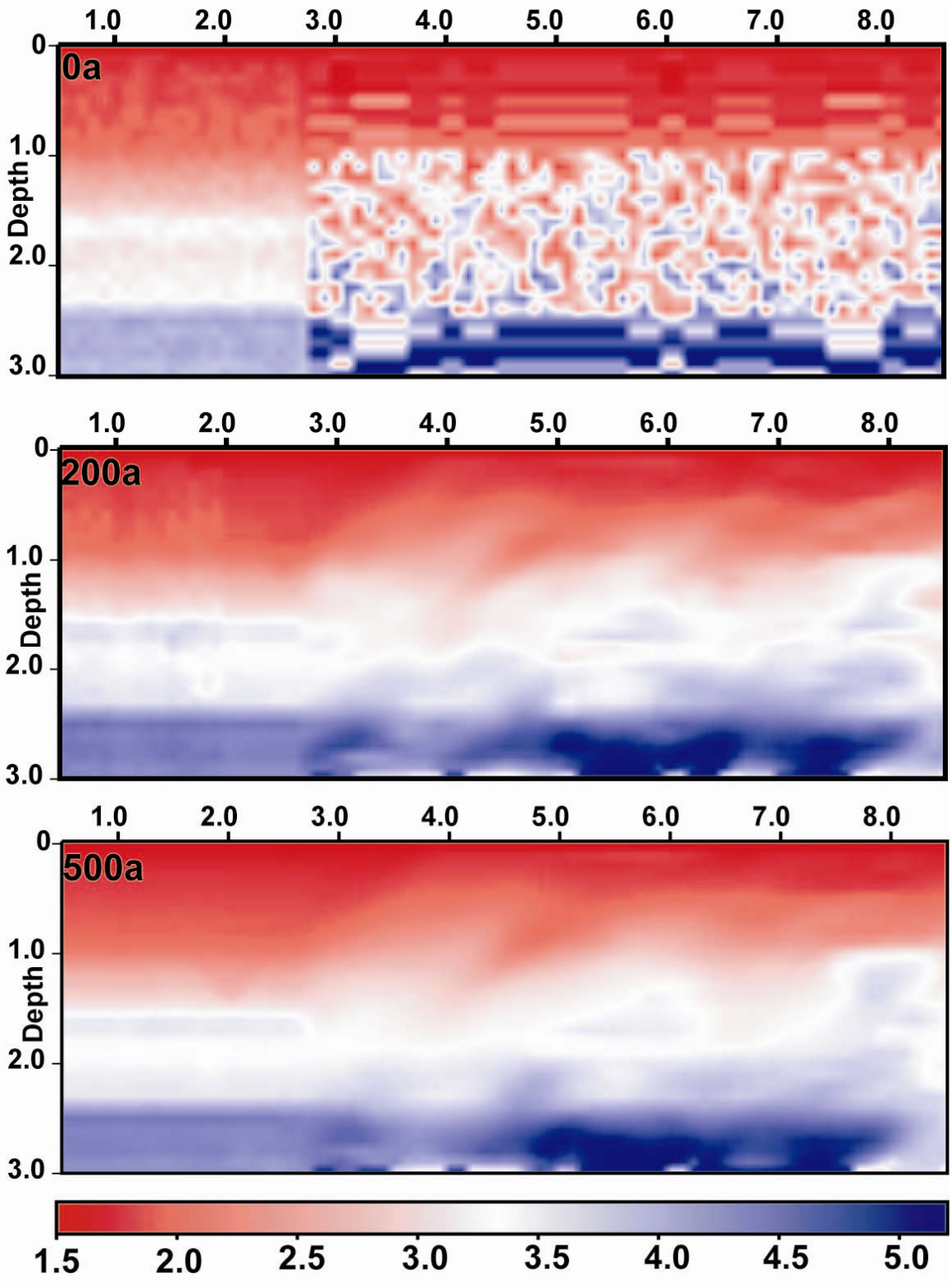


Figure 6.18: *Evolution of velocity model with the number of iteration.*

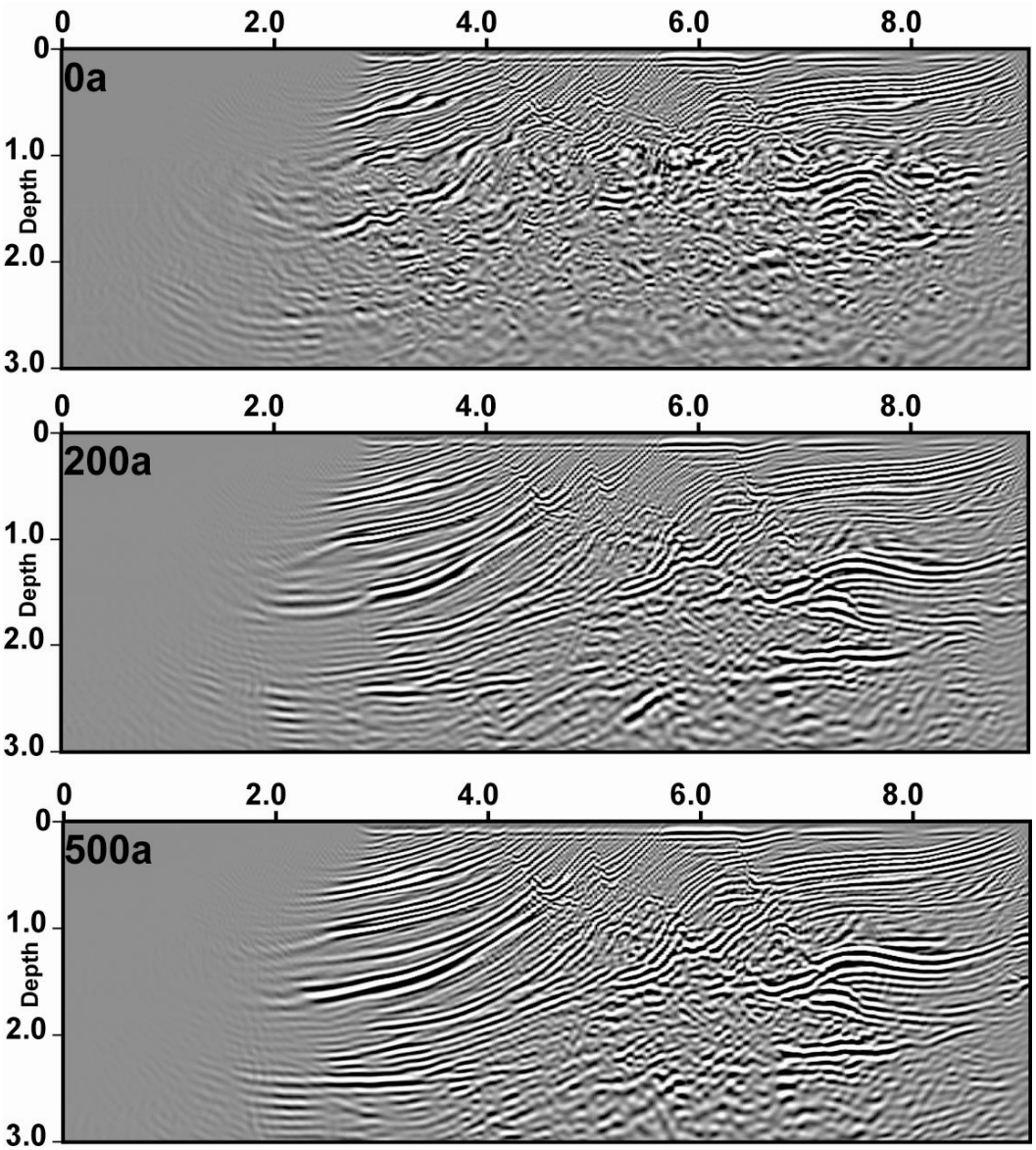


Figure 6.19: *Improvement in the image with the number of iteration.*

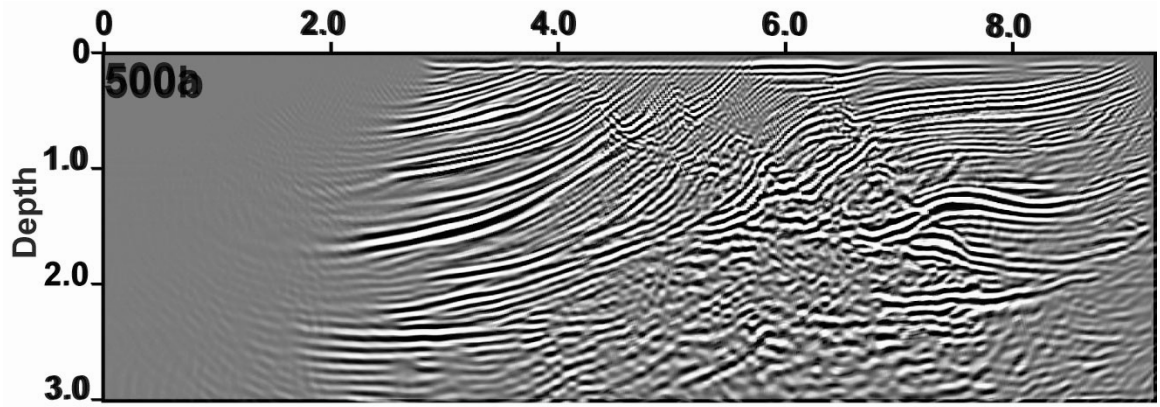


Figure 6.20: *Merging of the migrated image obtained after 500 iterations (500a & 500b).*

6.7.3. L7 Model

We have applied our algorithms to a real data set of North Sea. A 3d data was acquired by the petroleum company Elf (now Total). We have selected a 2D line from this data set. After optimisation, obtained velocity model, migrated image and gathers are shown in the figure 6.21, 6.22 and 6.23 respectively.

The optimised velocity models are not able to recover the velocity models around the salt body. Since depth velocity ambiguity is very large around salt body and also data do not have proper information, a coarse grid or blocky representation like Voronoi might here be a better choice than of regular grid. Because regular grid representation requires a large number of parameter as a result ambiguity increases whereas concise representation like Voronoi requires small number of parameters, they can be less ambiguous. These velocity models are also looking blocky, since we have used a coarse grid of 200 m to represent the velocity model. A fine grid representation can be a good choice to recover a small velocity error in the simple part of L7 models. However it requires an even larger number of unknown parameter with a very high computation costs. Since we have a set of optimised solution a final solution can be derived by the combination of all or one optimised solution can be selected based on the other available information.

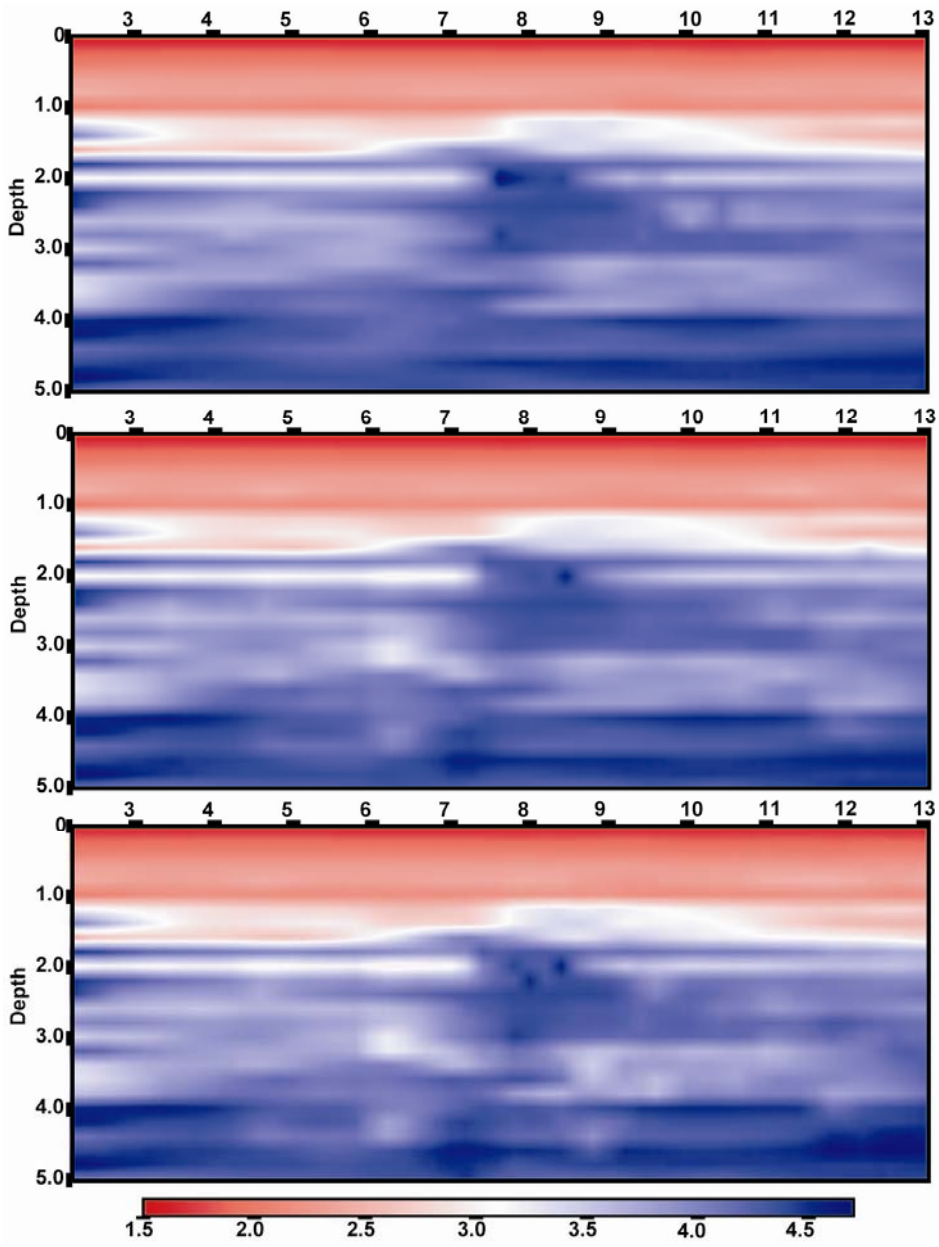


Figure 6.21: *Optimised velocity models for L7 data set. These velocity models are unable to estimate the salt velocity, whereas in other places they are able to estimate a good velocity model.*

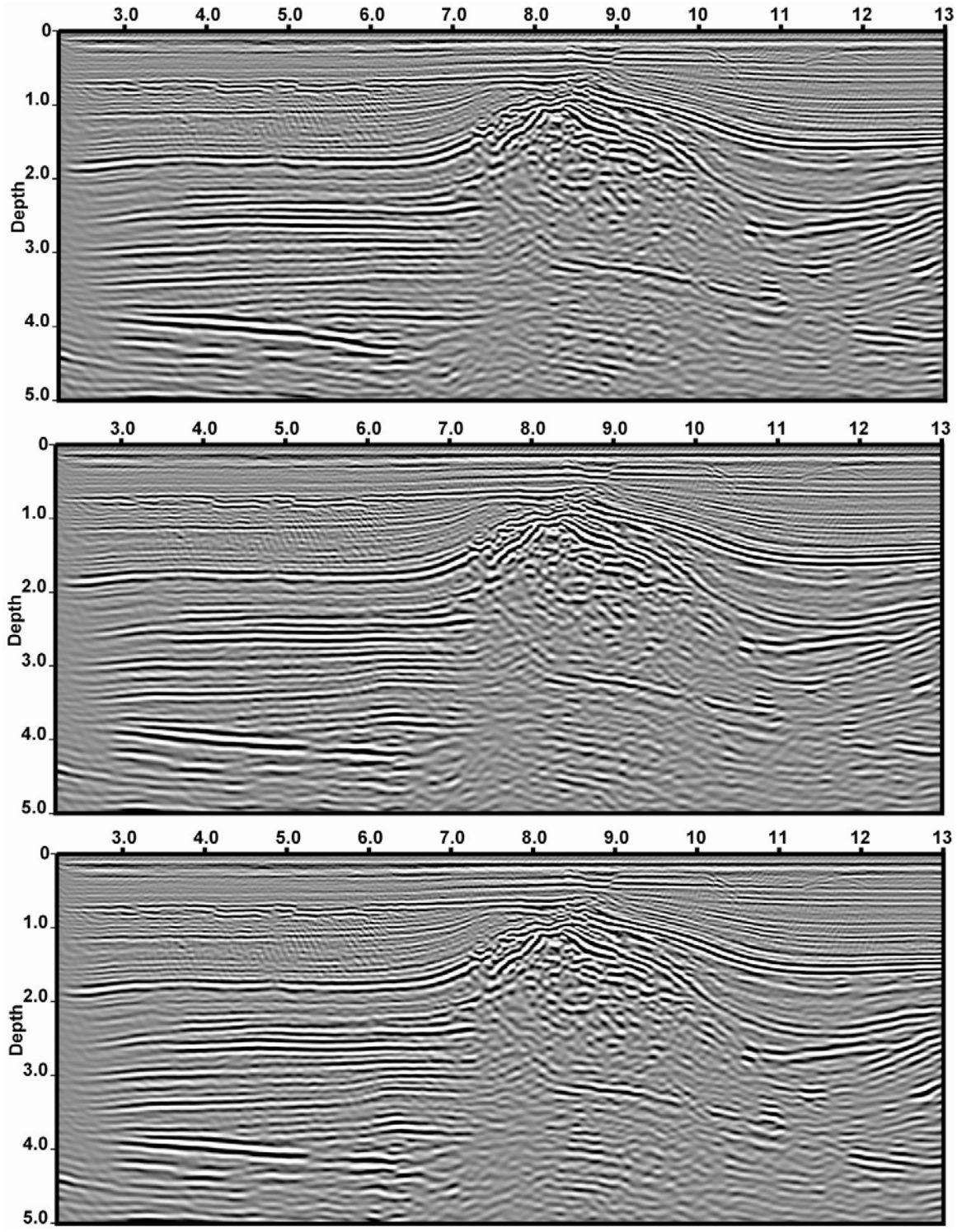


Figure 6.22: *Migrated image obtained from the optimised velocity models of L7 data set.*

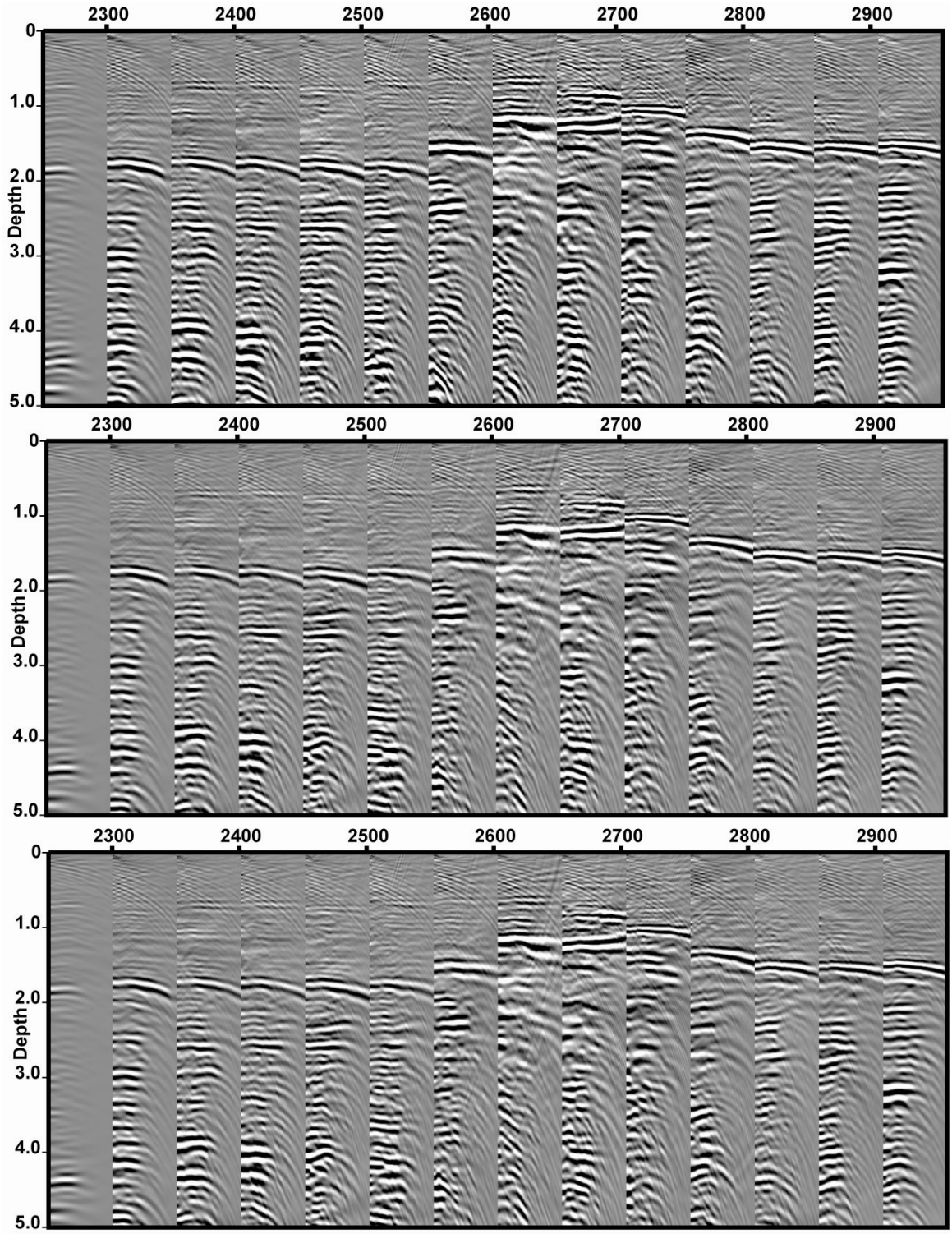


Figure 6.23: *Gathers obtained from the migration of optimised L7 velocity models*

6.8 Discussion and Conclusion

We have presented an automatic velocity optimization technique, one that does not require picking of the prestack migrated image gathers. This is the first time to our knowledge that MOEA has been used to estimate seismic velocity. We have modified the MOEA and introduced some exploitation techniques in such a way that it will not affect the global exploration property. An effort is also made to utilize the good properties of both S and DS function. This proposed techniques use both RMO error and iterative methods. It also requires a smaller number of evaluations of models though the number of unknown parameters is large. This technique resolves the major issue of computation cost for global optimisation methods. The computational cost of this technique is equivalent to that of the gradient methods.

The real and synthetic data example demonstrates that global optimization methods can be successfully applied to realistic scale seismic problems, at least in two dimensions. However it does not work well in the salt body. Therefore it requires a special attention for salt body. Our example shows that the technique presented here is robust and can be applied to noisy data as well. The little sensitivity of objective function for the data and noise adds extra robustness in optimization process and also boosts the optimization speed.

The technique presented here also differs from other global approaches (Jervis et al., 1996; Docherty et al., 1997; Mansane and Schoenauer, 2000) with respects to the representation of models, evaluation process, exploitation property, and reproduction technique. Jervis et al. (1996) and Docherty et al., (1997) used binary coded spline representation and Mansane and Schoenauer (2000) used real coded Voronoi while we are using real coded regular grid representation. We are evaluating two objective functions simultaneously for a model whereas others evaluate one objective function at a time. We have added exploitation property by reference model, RMO correction and directional smoothing where other approaches have used the blind traditional algorithms. In our approach we are using crossover without mutation where as in other approaches both operator were used. Before the crossover, the parent model is corrected (RMO) and smoothed then only a crossover operator is used. This goes with the philosophy of evolutionary algorithms which says that one should add as much as possible domain

specific information without perturbation the stochastic module of the algorithms. In the Table 6.2 a comparison is made with respect to number of unknown velocity parameter and required number of PSDM to convergence. We can note that though we have 50 to 100 times more unknown velocity parameters, our convergence is almost 100 times faster than that of other methods.

Table 6.2: *Performance of different approach is shown with respect to number of unknown parameters and required number of PSDM to converge.*

Authors	Number of Parameters	Required Number of PSDM
Jervis et al. (1996)	50	2000
Docherty et al.(1997)	68	12994
Mansanne & Schoenauer (2000)	100-200	20000
250m Grid	**457**	**200-400**
100m Grid	**2700**	**300-600**
100m Grid	**2700**	**150-300**

In the future we would like to extend our approach from coarse to fine modeling. There are two different aspects of coarse to fine modeling, one related to the migration and other related to the representation of models. In the migration, we can shift from small frequency bandwidth to large frequency band, increase the number of shots, and reduce the propagation sampling depth. This shifting will increase the computational cost of a model but at the same time it will increase the accuracy of the results. A model can be initially represented on a coarse grid and later on a refined grid. In this process the number of unknown parameters will increase while the model representation will shift from a smooth representation to a fine blocky representation. Both aspects will increase the computational cost but hopefully produce refined models and accurate results. This approach may be helpful in extending our optimization processes from 2D to 3D velocity optimization.

Chapter 7

Conclusion and Perspective

Velocity estimation in geologically complex regions is one of the most challenging problems in petroleum prospection because of strong lateral and vertical velocity variation, multipathing, uneven illumination, and irregular data coverage. Migration velocity analysis methods based on wavefield-continuation methods are robust for such region. To tackle the nonlinear relationship between seismic data and velocity model, a customized hybrid multiobjective evolutionary algorithm was designed, and efficiently used for velocity estimation of both a synthetic and a real data.

The main contributions of this thesis are:

- We have developed an automatic cross-section balancing algorithms for foothill structure. This can be easily extended for other geological structure, using proper forward modeling algorithms (Chapter 4).
- This geologically balanced cross-section was used to design a geologically-sound representation, that was successful in solving the geological structure inversion problem from dip measures. Furthermore, when the geological structure was correctly inversed, we were able to solve the velocity inversion problem and to get a good velocity model. However, our attemps to invert both geological structure and velocity model together only encountered partial success (Chapter 4).
- We hence turned back to representing the velocity using a regular grid. We noticed that coarse grid representation is suitable for simple part of geological structure, whereas fine grid representation is necessary for the complex parts of geological structure. This part of our work also emphasized the importance of the representation issue (Chapter 4).
- Previous works had given hints that the Differential Semblance function could be misleading around the exact values of the velocities: we slightly modified the DS function to make it ”more convex” around the exact values, and sensitive even

for quite large velocity error (Chapter 5). This modified Differential Semblance, together with the standard Semblance function, will then be our two objective functions.

- We developed automatic RMO picking algorithms and some dip extraction technique from migrated image and gather (Chapter 5). Both will be used to locally improve the velocity models during the evolutionary optimization in Chapter 6.
- We designed a complete algorithms, based on the results of Chapter 5, using: (1) Both semblance and modified differential semblance objectives; (2) A modified ε-MOEA with very small population size; (3) Reference models that were used as some local archive of good parts of the previously encountered models (4) Guided crossover using local improvements based on: (5) RMO information and (6) Dip information (Chapter 6).
- This hybrid algorithm was then intensively applied the artificial Marmousi problem, and, to a lesser extend, to the North-Sea L7 data. On the one hand, the results are rather satisfactory with respect to the accuracy of the identified velocities, and on the other hand they demonstrate the efficiency of the proposed customization with respect to the overall computing time of the inversion process, as its cost is equivalent or faster then that of gradient methods (Chapter 6).

Perspectives

Two Dimensional velocity inversion

There is still much room for improvement for the performance of the proposed velocity inversion algorithms.

- To improve the efficiency, we have developed some layer striping approach, though it has not been sufficiently experimented with, and hence was not presented her.

- Though we have used ”vertical” reference models for exploiting the information, horizontal reference models can also be used. Moreover, mixing both approaches could be helpful in order to increase the efficiency of exploitation.
- The user (expert geophysicist) usually plays an important role in the process of velocity inversion. Hence there is a need here to add some user reference model, in order to be able to exploit the information provided by users during the optimisation. For example, for velocity estimation in salt structure, user information can be very helpful in identifying the shape of salt structure. Evolutionary algorithms are flexible enough to allow such interactive optimisation.
- Finally, and especially for estimating velocity from a salt-body embedded structure, a mixed domain representation using both the grid representation used in chapter 6 and the Voronoi representation (proposed by Mansanne et al., and briefly described in Chapter 4) could be used.

Three Dimensional

Extension to three dimensional (3D) is itself a big challenge. However a clever selection of migration algorithms, proper extraction and utilization of information obtained from the gathers and development of good exploitation operator can make it possible to extend this approach for 3D.

For example (1)use of Common Azimuth Migration (CAM) with layer stripping approaches. (2) Generation of approximately good initial velocity model (3) use of Prestack residual migration for velocity error estimation from gathers (4) concise representation of velocity model (5) development of user reference model and some new crossover strategy.

Indeed, we do think that this technique is now ready for 3D extension. Tremendous reduction in the computation cost and strong increase in the computational efficiency in recent years make it likely that such approach will become routine use in industrial context.

References

- Allmendinger, R.W. (1998) "Inverse and forward numerical modeling of trishear fault-propagation folds". *Tectonics* 17 (1998), 640–656.
- Allon Bartana, Dan Kosloff, and Igor Ravve, 2006, Angle-domain common-image gathers by wavefield continuation methods" (Paul C. Sava and Sergey Fomel, 2003, GEOPHYSICS, 68, 1065–1074), Geophysics,71,pp x1-x3.
- Al-Yahya, K. M., 1989, Velocity analysis by iterative profile migration: Geophysics, 54, 718–729.
- Barut, M., de Bazelaire, E., Ravat, J., and Taveau,G., 1991, A methodology to image tectonically complex areas using polystack and prestack depth migration, *in* Versteeg, R., and Grau, G., Eds., The Marmousi experience: Proc. 1990 EAEGworkshop on Practical Aspects of Seismic Data Inversion: Eur. Asson. Expl. Geophys., 159–167.
- Beyer Hans-Georg, Deb K (2001) "On self-adaptive features in real-parameter evolutionary algorithms" IEEE Trans. Evolutionary Computation 5(3): 250-270
- Beyer, H.G. and Deb, K. On self-adaptive features in real-parameter evolutionary algorithms. IEEE Transactions on Evolutionary Computation, 5(3). 250-270. (2001).
- Biondi, B. L., 2003, 3-D seismic imaging
- Biondi, B., and P. Sava, 1999, Wave-equation migration velocity analysis: 69th Annual International Meeting, SEG, Expanded Abstracts, 1723–1726
- Biondi, B., and W. Symes, 2004, Angle-domain common-image gathers for migration velocity analysis by wavefield-continuation methods: Geophysics, 69, 1283–1298.
- Biondi, B., and W.W. Symes: Angle-domain common-image gathers for migration velocity analysis by wavefield-continuation imaging. Geophysics, v. 69, pp. 1283-1298
- Boschetti, F., Dentith, M.C. & List, R.D. (1996) "Inversion of seismic refraction data using genetic algorithms", *Geophysics*, 61, 1715–1727.
- Boyer, S.E., and Elliott, D. (1982) "Thrust systems": American Association of Petroleum Geologists Bulletin, v. 66, 1196-1230.
- Brandsberg-Dahl, S., De Hoop, M. V., and Ursin, B., 1999, Velocity analysis in the common scattering-angle/azimuth domain: Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1222–1223.
- Carlos A. Coello Coello and Gregorio Toscano Pulido. Multi-objective Optimization using a Micro-Genetic Algorithm. In Lee Spector et al., editor, *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'2001)*, pages 274–282, San Francisco, California, 2001. Morgan Kaufmann Publishers.
- Chauris, H., M. Noble, G. Lambaré, and P. Podvin, 2002a, Migration velocity analysis from locally coherent events in 2-D laterally heterogeneous media, Part I: Theoretical aspects: Geophysics, 67, 1202–1212.

- Chauris, H., Noble, M., and Podvin, P., [1998]. Testing the behaviors of differential semblance for velocity estimation: 68th Annual International Meeting and Exposition, Society of Exploration Geophysicists, 1305-1308.
- Chester J.S. (2003) "Mechanical stratigraphy and fault–fold interaction, Absaroka thrust sheet, Salt River Range" Wyoming, *Journal of Structural Geology* 25 (2003), 1171–1192.
- Claerbout, J. F., 1971, Toward a unified theory of reflector mapping: Geophysics, **36**,
- Clapp, R. G., B. Biondi, and J. F. Claerbout, 2004, Incorporating geologic information into reflection tomography: Geophysics, 69, 533–546.
- Coello C.C.A. (1999) "A comprehensive survey of evolutionary-based multi-objective optimization techniques" *Knowledge and Information Systems An International Journal* 1 (3), 269–308.
- Coello, C. A. C, VanVeldhuizen, D. A. and Lamont, G. (2002). *Evolutionary Algorithms for Solving Multi-objective Problems*. Boston, MA: Kluwer.
- Contreras J. (1991) "Kinematic modeling of cross-sectional deformation sequences by computer simulation: coding and implementation of the algorithm", Computers & Geosciences, v.17 n.9, 1197-1217.
- Contreras, J., Suiter, M. (1990) "Kinematic modeling of cross-sectional deformation sequences by computer simulation". Journal of Geophysical Research 95, 21913-21929.
- Contreras, J., Suter, M. (1997) "A kinematic model for the formation of duplex systems with a perfectly planar roof thrust". Journal of Structural Geology 19, 269-278.
- Dahlen, F., Hung, S.-H. & Nolet, G., 2000. Fr´echet kernels for finite-frequency traveltimes–I. Theory, *Geophys. J. Int.*, 141, 157– 174.
- Dahlstrom C.D.A. (1990) "Geometric constraints derived from the law of conservation of volume and applied to evolutionary models for detachment folding" *American Association of Petroleum Geologists Bulletin* 74 (1990), pp. 336–344.
- Deb K. (2001) "Multi-Objective Optimization using Evolutionary Algorithms" Wiley, Chichester, UK.
- Deb K. and Agrawal R. (1995). "Simulated binary crossover for continuous search space" *Complex Systems*, 9(2):115–148.
- Deb K. and Agrawal R. B. Simulated binary crossover for continuous search space. *Complex Systems*, 9:115-148, 1995.
- Deb, K, Mohan, M. and Mishra, S. (2003). "A Fast Multi-objective Evolutionary Algorithm for Finding Well-Spread Pareto-Optimal Solutions" *KanGAL Report No. 2003002*.
- Deb, K. (2001). *Multi-objective Optimization Using Evolutionary Algorithms.* Chichester, UK: Wiley.
- Deb, K. and Agrawal, R.B. Simulated Binary Crossover for Continuous Search Space, Complex Systems, 9, 115-148 (1995)

- Deb, K. and Beyer, H.-G. Self-Adaptation in Real-Parameter Genetic Algorithms with Simulated Binary Crossover, Proc. Genetic and Evolutionary Computation Conf. 1999 (GECCO-99), 172-179 (1999)
- Deb, K. and Beyer, H.-G. Self-adaptive genetic algorithms with simulated binary crossover. Evolutionary Computation Journal, 9(2), 197-221. (2001).
- Deb, K., Pratap. A, Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transaction on Evolutionary Computation, 6*(2), 181-197.
- Delprat-Jannaud, F., and P. Lailly, 1992, What information on the earth model do reflection travel time provide?: Journal of Geophysical Research, 97, 19 827–19 844.
- Duquet, B., and Marfurt, K. J., 1999, Filtering coherent noise during prestack depth migration: Geophysics, 64, 1054-1066.
- E. Zitzler, M. Laumanns, and L. Thiele. *SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm.* Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich, Gloriastrasse 35, CH-8092 Zurich, May 2001,
- Endignoux L, Moretti I., and Roure F., 1989, "Forward modeling of the southern Apennines", Tectonics V8, 5, 1095-1104.
- Erslev E.A. (1991) "Trishear fault-propagation folding" *Geology* 19 pp. 617–620.
- F. Herrera, M. Lozano, "Tackling real-coded genetic algorithms: Operators and tools for the behavioural analysis," Artificial Intelligence Review (1998).
- Fischer, M.P., Jackson, P.B., (1999) "Stratigraphic controls on deformation patterns in fault-related folds: a detachment fold example from the Sierra Madre Oriental, northeast Mexico" Journal of Structural Geology 21, 613–633.
- for shot-profi le migration: Geophysics, **67**, 883–889.
- G. E. Liepins and Michael D. Vose. Representational Issues in Genetic Optimization. Journal of Experimental and Theoretical Computer Science, 2(2):4-30, 1990.
- H.-M. Voigt, H. Muhlenbein and D. Cvetkovic, Fuzzy recombination for the breeder genetic algorithm. To appear in Sixth Int. Conf. on Genetic Algorithms Morgan Kaufmann,(1995)
- Hardy, S. (1995) "A method for quantifying the kinematics of fault-bend folding"*Journal of Structural Geology 17, 1785-1788.*

  http://sepwww.stanford.edu/sep/biondo/Lectures/index.html
- J. Devaney and M. L. Oristaglio, 1984, Geophysical diffraction tomography, SEG Technical Program Expanded Abstracts pp 330-333.
- Jamison W.R. (1992)" Stress controls of fold thrust style" *Thrust Tectonics*, Chapman and Hall, London (1992), 155–164.
- Jamison W.R. (1997) "Quantitative evaluation of fractures on Monkshood anticline, a detachment fold in the Foothills of Western Canada" *The American Association of Petroleum Geologist Bulletin* 81 (1997), 1110–1132.

- Jervis, M., Sen, M. K., and Stoffa, P. L., 1996, Prestack migration velocity estimation using nonlinear methods: Geophysics, 61, 138-150.
- Jiao, J., Stoffa, P. L., Sen, M. K., and Seifoullaev, R. K., 2002, Residual migration-velocity analysis in the plane-wave domain: Geophysics, 67, 1252–1269.
- Jin, S., and Madariaga, R., 1994, Nonlinear velocity inversion by a two-step Monte Carlo: Geophysics, 59, 577–590.
- L. J. Eshelman and J. D. Schaffer, "Real-coded genetic algorithms and interval-schemata," in Foundations of Genetic Algorithms-2. San Mateo, CA: Morgan Kaufman, pp.187-202, (1993)
- Lafond, C. F., and A. R. Levander, 1993, Migration moveout analysis and depth focusing: Geophysics, 58, 91–100.
- Léger, M., Interpolation from Lagrange to Holberg, Curve and Surface Fitting, A. Cohen, C. Rabut and L. Schumaker (eds), Vanderbilt University Press, Nashville, 2000, 281-290.,
- Liu, Z., and N. Bleistein, 1995, Migration velocity analysis: Theory and an iterative algorithm: Geophysics, 60, 142–153.
- Mansanné, F., and Schoenauer, M., (2002), "An automatic geophysical inversion procedure using a genetic algorithm" Soft computing for reservoir characterization and modeling : Physica-Verlag, Heidelberg, 331–335
- Meeting, SEG, Expanded Abstracts, 1850–1853.
- Mitra, S. (1988) "Three-dimensional geometry and kinematic evolution of the Pine Mountain thrust system, Southern Appalachians" Geological Society of America Bulletin, v. 100, 72-95.
- Mitra, S. (1992) "Balanced structural interpretations in fold and thrust belts" Structural geology of fold and thrust belts, Johns Hopkins University Press, Baltimore, p. 53-77.
- Moretti I. and Larrère M., 1989, "LOCACE: Computer-Aided Construction of Balanced Geological Cross-sections", Geobyte, Oct. 1-24.21.
- Morgan J.K.and Karig D.E. (1995) "Kinematics and balanced cross-section across the toe of the eastern Nankai accretionary prism" *Journal of Structural Geology* 17 , 31–45.
- Mulder, W., and A.ten Kroode, 2002, Automatic velocity analysis by differential semblance optimization: Geophysics, 67, 1184–1191.
- Ono, I. and Kobayashi, S. (1997) A Real-coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distribution Crossover, Proc. 7th Int'l Conf. on Genetic Algorithms, 246-253
- Paul C. Sava and Sergey Fomel Reply to the discussion "Allon Bartana, Dan Kosloff, and Igor Ravve" Geophysics,71,pp x3-x4.
- Paul C. Sava, Biondo Biondi, and John Etgen, 2005, Wave-equation migration velocity analysis by focusing diffractions and reflections Geophysics 70, pp U19.

- Plessix, R.-E., Mulder, W.A. and ten Kroode, A.P.E., [2002]. Automatic crosswell tomography by semblance and differential semblance optimization: theory and gradient computation, Geophysical Prospecting 48, 913–935
- Pratt R.G., and Symes, W. W. [2002]. Semblance and differential semblance optimisation for waveform tomography: a frequency domain implementation, Sub-basalt imaging Journal of Conference Abstracts Cambridge Publications Volume 7(2), 183
- Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain — Part 1: Theory and verification in a physical scale model: Geophysics, 64, 888–901.
- Prucha, M., Biondi, B., and Symes, W., 1999, Angle-domain common-image gathers by waveequation migration: 69th Ann. Internat. Meeting, Soc. Expl. Geophys., Expanded Abstracts, 824. 827.
- R. Gerhard Pratt,1999, Seismic waveform inversion in the frequency domain, Part 1: Theory and verification in a physical scale model Geophysics 64, pp 888-901
- R.S. Rosenberg. Simulation of genetic populations with biochemical properties. PhD thesis, University of Michigan, Ann Harbor, Michigan, 1967.
- Ramsay, J.G. (1992) "Some geometric problems of ramp-flat thrust models" Thrust tectonics, Chapman-Hall, London, 191-200.
- Rich, J. L. (1934) "Mechanics of low-angle overthrust faulting as illustrated by Cumberland thrust block, Virginia, Kentucky and Tennessee" Bulletin of the American Association of Petroleum Geologists 18, 1584-l 596.
- Rickett, J. E., and P. Sava, 2002, Offset and angle-domain common image-point gathers
- Rickett, J., and Sava, P., 2002, Offset and angle-domain common image-point gathers for shot-profile migration: Geophysics, 67, 883.889.
- S. Tsutsui, M. Yamamura, and T. Higuchi. Multi-parent Re- combination with Simplex Crossover in Real Coded Genetic Algorithms. In Proc. of the GECCO-99, pages 657-664, 1999.
- Sava P. 2003. Prestack residual migration in the frequency domain. Geophysics 67, 634–640.
- Sava, P., 2003, Prestack residual migration in the frequency domain: Geophysics, 67, 634–640.
- Sava, P., and B. Biondi, 2004a, Wave-equation migration velocity analysis — I: Theory: Geophysical Prospecting, 52, 593–606. 2004b, Wave-equation migration velocity analysis — II: Subsalt imaging examples: Geophysical Prospecting, 52, 607–623.
- Sava, P., and Fomel, S., 2003, Angle-domain common-image gathers by wave_eld continuation methods: Geophysics, 68, 1065.1074.
- Sava, P., and S. Fomel, 2005, Time-shift imaging condition: 75th Annual International
- Schoenauer M.and M. Sebag M, (2002), "Using Domain Knowledge in Evolutionary System Identification" Evolutionary Algorithms in engineering and Computer Science John Wiley

- Schultz, P. S., and Claerbout, J. F., 1978, Velocity estimation and downward-continuation by wavefront synthesis: Geophysics, 43, no. 4, 691.714.
- Sen, M. K. and P. L. Stoffa, Nonlinear one-dimensional seismic waveform inversion using simulated annealing, *Geophysics*, *56*, 1624-1638, 1991.
- Shen P. 2004. seismic velocity analysis by wave-equation migration. (Thesis) Center for Computational Geophysics, Department of Earth Science, 6100 Main St., Rice University, Houston.
- Stolk, C. C., and W. W.Symes, 2003, Smooth objective functionals for seismic velocity inversion: Inverse Problems, 19, 73–89.
- Stolk, C.C., and W.Symes, 2004, Kinematic aritifacts in prestack depth migration: Geophysics, 69, 562–575.
- Stork, C., 1992, Re_ection tomography in the postmigrated domain: Geophysics, 57, 680.692.
- Stork, C., Kitchenside, P., Yingst, D., Albertin, U., Kostov, C., Wilson, B., Watts, D., Kapoor, J., and Brown, G., 2002, Comparison between angle and offset gathers from wave equation migration and Kirchhoff migration: 72nd Ann. Internat. Meeting, Soc. of Expl. Geophys., Expanded Abstracts,1200.1203.
- Storti F. and Salvini F. (1996) "Progressive rollover fault-propagation folding: a possible kinematic mechanism to generate regional-scale recumbent folds in shallow foreland belts" *American Association of Petroleum Geologists Bulletin* 80, 174–193.
- Suppe, J. (1983) "Geometry and kinematics of fault-bend folding" American Journal of Science, v. 283, p. 684-721.
- Suppe, J. and Medwedeff, D.A. (1984) "Fault-propagation folding" Geological Society of America Abstracts with Programs, v. 16, 670.
- Suppe, J. and Medwedeff, D.A. (1990), "Geometry and kinematics of fault-propagation folding" Eclogae geologicae Helvetiae, v. 83, 409-454.
- Symes, W., and J.Carazzone, 1991, Velocity inversion by differential semblance optimization: Geophysics, 56, 654–663.
- T. Nomura and T. Miyoshi, “Numerical coding and unfair average crossover in GA for fuzzy rule extraction in dynamic environment”, Fuzzy Logic, Neural Networks and Evolutionary Computation, eds., T. Furuhashi and Y. Uchikawa, SpringerVerlag, pp. 55-72, 1996.
- Tanner, P.W.G. (1992) "Morphology and geometry of duplexes formed during flexural-slip folding" Journal of Structural Geology, v. 14, 1173-1192.
- Tarantola, A., 1984a, Inversion of seismic reflection data in the acoustic approximation: Geophysics, 49, 1259–1266. 1984b.
- Thomas Bäck. Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming Genetic Algorithms. Oxford University Press Oxford, UK,(1996).
- Veldhuizen D. and Lamont G. (2000) "Multiobjective evolutionary algorithms: analysing the state-of-the-art" *Evolutionary Computation* 8 (2), 125–147.

- Walter S. Lynn and Jon F. Claerbout, 1982, Velocity estimation in laterally varying media, Geophysics 47, 884
- Wapenaar, C. P. A., and Berkhout, A. J., 1987, Full prestack versus shot record migration: 69th Ann. Internat. Meeting, Soc. of Expl. Geophys., Expanded Abstracts, Session:S15.7.
- Wapenaar, K., Van Wijngaarden, A., van Geloven, W., and van der Leij, T., 1999, Apparent AVA effects of _ne layering: Geophysics, 64, no. 6, 1939.1948.
- Whitmore, N. D., 1983, Iterative depth migration by backward time propagation: 53rd Annual Internat.Mtg., Soc. Expl. Geophys., Expanded Abstracts, Session:S10.1.
- Woodward N.B. (1999) "Competitive macroscopic deformation processes", *Journal of Structural Geology* 21, 1209–1218.
- Woodward, M. J., 1992, Wave-equation tomography: Geophysics, 57, 15–26.
- Xie, X. B., andWu, R. S., 2002, Extracting angle domain information from migrated wave_eld: 72th Ann. Internat. Mtg., Soc. Expl. Geophys., 1360.1363.
- Xu, S., Chauris, H., Lambare, G., and Noble, M. S., 2001, Common-angle migration: A strategy for imaging complex media: Geophysics, 66, 1877.1894.
- Zhu, J., Lines, L. R., and Gray, S. H., Smiles and Frowns in migration/velocity analysis", Geophysics 63, 1200{1209 (1998).
- Zoetemeijer, R. (1993) "Tectonic Modeling of Foreland Basins: thin skinned thrusting, syntectonic sedimentation and lithospheric flexure" Ph.D. thesis, University of Amsterdam.