GLOBAL OPTIMIZATION OF THE CRS OPERATOR USING A GENETIC ALGORITHM

J. Walda and D. Gajewski

email: *jan.walda@uni-hamburg.de*
keywords: *CRS, optimization, genetic algorithm*

ABSTRACT

The CRS operator improves the signal to noise ratio significantly due to the consideration of neighboring midpoints as well as the offset. The determination of the required attributes for the CRS operator is often done by the pragmatic approach to get initial values that are refined by a local optimization. This works reasonable for most parts, however in more complex structures like salt bodies the result is not reliable anymore. Additionally the pragmatic approach does not perform particularly well in the presence of conflicting dips. Therefore we propose to use a genetic algorithm based optimization and show that the stack and especially the determined attributes are significantly better.

INTRODUCTION

The pragmatic approach (Müller, 1999) is a fast and efficient method to get initial stacking parameters for the Common Reflection Surface (CRS) stack (Hubral, 1983; Müller, 1999; Mann, 2002). The estimated initial set of parameters can be optimized locally afterwards by a multidimensional search. This approach has been used for a decade and delivered decent results in a manageable amount of time. However, it is not well suited for the handling of conflicting dips, since multiple operators are required and the pragmatic approach mostly works for the most dominant event.

Common strategies in other scientific fields to solve complex optimization problems are evolutionary and swarm intelligence based algorithms (Kitano, 1990; Morris et al., 1998). In this work we focus on a family of evolutionary algorithms called genetic algorithms originally introduced by Holland (1975). There are various kinds of implementations suited for particular problems (Whitley, 1994). We introduce the general scheme of an evolutionary and genetic algorithm and describe the variation we use in our examples. Finally we compare the conventional simplex based optimization approach with the proposed one.

THE PRAGMATIC APPROACH

The pragmatic approach introduced by Müller (1999) is an efficient estimation of a set of starting parameters for the CRS method that can be used as initial values for a further optimization method in 2-D. It performs three individual searches in different sub-domains of the data. This approach does not use the advantage to obtain a higher coherence due to an increased number of involved traces. The obtained parameters can be optimized further, usually by local optimization schemes. However the initial values determined via the pragmatic approach are not always reliable in complex structures. Therefore we consider global optimization schemes based on genetic algorithms which are part of the evolutionary algorithms. The next section gives a rough overview.

EVOLUTIONARY ALGORITHMS

Evolutionary algorithms (EA) are meta-heuristics that use lower-level heuristics to provide a good solution to a given problem. Heuristics are experience based strategies for problem solving that are able to learn and therefore adapt to a problem. The solution is usually not guaranteed to be optimal but is mostly sufficient for a certain task.

Evolutionary algorithms are inspired by nature, e. g. biological evolution (Eigen, 1973). In contrast to the pragmatic approach evolutionary algorithms use a starting population of potential solutions rather than one initial solution. The Population is usually randomly generated. After the initialization the fitness (here coherence) of each individual is evaluated. It follows a loop until a certain criterion is fulfilled like number of iterations, calculation time and/or accuracy. Within this loop the fittest parents are used for reproduction. The reproduction is done by mutation and crossover of parent genes (or chromosomes) to produce the next generation. The fitness of the new generation is evaluated again and a substitution of the old generation with the new generation is performed. There are several ways to do that but the fittest individuals should always be kept to avoid losing the current best solutions and achieve a better convergence. The genetic algorithms (GA) are a part of the evolutionary algorithm family that mimic the process of natural selection.

GENETIC ALGORITHM

Since genetic algorithms are evolutionary algorithms the overall strategy is very similar to other algorithm of this kind. The main differences origin in the way how the selection of parents, the crossover of their genes and mutations are done. For each of them there are various variations suited for different tasks and shapes of the objective function as well as encoding dependent. The encoding defines the way chromosomes, the attributes of the operator, are parametrized. The most common is a binary encoding where the attributes are encoded in a bit string containing ones and zeros. In our work we use value encoding where the attributes are represented as real numbers. Janikow and Michalewicz (1991) found that real encoding usually is more stable and converges faster then binary encoding for real number problems.

The selection of the parents in this work is done by the roulette wheel scheme, where the individuals are sorted by their fitness and their probability to be chosen as a parent is determined by their fitness value (Bäck, 1996). Another approach is the rank based selection. The individuals are again sorted but instead of a probability based on their fitness, a probability dependent on their rank is used. This increases the chance of less fit individuals to become parents while the fittest are less likely to become a parent (Baker, 1985). This is used to avoid strong bias in the optimization. For a parallel implementation tournament selection (Goldberg, 1990) is useful. In this method two or three random individuals are chosen where the fittest individual becomes a parent for the new offspring.

The crossover determines how chosen parents pass their genes to the next generation. We use a weighted addition of the parameters from the parents where the weight w is chosen randomly between -0.25 and 1.25

$$C_i = w \cdot P_i^1 + (1 - w) \cdot P_i^2. \tag{1}$$

The parameter C_i is the i-th attribute of the offspring, P_i^1 the i-th attribute of the first parent and the P_i^2 the i-th attribute of the second parent. If $w = 1$, the child takes the value of the first parent. In case $w = 0$ the child gets the same value as the second parent. The weight itself ranges from -0.25 to 1.25 to allow for values outside of the boundaries determined by the parents. There are various other crossover methods that can be used. However simple arithmetics are easy to implement and work well.

Mutation is often considered to be the main reason (or only reason) why a genetic algorithm converges. To provide a simple and additional mutation method we use two different kind of mutations. The first is a random mutation where a value gets replaced with a new random value. The probability decreases with iterations. In the second mutation the new value is replaced with a Gaussian distributed value rather than a random value. This additional mutation can improve the convergence.

The reason why genetic algorithms work for most problems is still not solved, Holland (1975) introduced the schema theorem. However Grefenstette and Baker (1989) found issues applying the theorem. Mühlenbein (1992) even showed that an evolutionary algorithm solely based on mutations works well for simple tests. Furthermore hybrid approaches (usually genetic algorithms combined with hill-climbing methods) are better suited for optimization (Davis, 1991). However, the building block hypothesis (Goldberg, 1989) is still an easy to grasp explanation.

Independent of the chosen implementations of selection, crossover and mutation it is highly recommended to use elitism. It means passing the best individual(s) to the next generation in order to preserve the already best obtained solution(s) and further improve the convergence. The performance of genetic algorithms can be enhanced by a combination with other optimization methods.

HYBRID ALGORITHMS

A pure genetic algorithm is very general and uses very few assumptions and therefore is called a 'weak' algorithm (Whitley, 1994). It can be applied to almost any problem but is usually less efficient than more specific optimization schemes. A common way to improve the performance with respect to convergence as well as quality of the obtained solution is to combine it with local hill-climbing methods like Nelder and Mead (1965). This allows for good solutions after few iterations. In the next section we apply this method to a synthetic data-set.

SIGSBEE2A

In the synthetic examples we use the hybrid genetic algorithm with only 10 iterations where GA and Nelder Mead alternate and 40 individuals to show the advantages and accuracy achievable in a comparable short time. The main improvement of the CRS stack obtained with a hybrid genetic algorithm (Fig 1) compared to the pragmatic approach, improved with a local hill climb method, in figure 2 is best visible in the salt and sub-salt area. Some diffractions become more prominent and the events in general are more continuous. This is due to an improved operator which is also displayed by the coherence section for the genetic algorithm in figure 3 compared to the conventional coherence section in figure 4. Again, events are more continuous and more coherent energy is found in salt and sub-salt areas especially for diffractions. While an improvement is easily visible in the stack and the coherence section, the best improvements are visible in the parameter sections of the optimized operator.

The angle of emergence α is the most stable parameter and is not problematic even in the pragmatic approach, therefore this parameter is left out. Figure 5 shows the R_{NIP} section of the hybrid genetic algorithm, figure 6 of the pragmatic approach. The layered parts of the model are very similar. The differences in this areas are mainly regarding consistency and noise. The big differences are within and below the salt body where the pragmatic approach with the local optimization is very noisy and rarely detects events that are also disrupted. The hybrid genetic algorithm is able to show a lot more details in the parameter section with a mainly smooth behavior along events. If done with more iterations it might even not be necessary anymore to apply event consistent smoothing. Hard to image areas however are still noisy. Considering that we only used 10 iterations, there is a lot of room for improvement. During the optimization the last CRS parameter R_N is usually the most difficult parameter to obtain since it can range from $-\infty$ to $+\infty$ and depends on the midpoint aperture which is usually rather small. Therefore the operator itself is not very sensitive with respect to R_N. This shows in the R_N section of the pragmatic approach with the local optimization in figure 8 since most parts are close to 0 which seems not plausible for the radius of curvature of a exploding reflector element, especially in the deeper parts of the model. Using the hybrid genetic algorithm the R_N section in figure 7 becomes physical plausible in most parts of the data. Furthermore a lot of events that are not recognizable before can be seen clearly especially sub-salt and salt diffractions. The significant improvements in the parameter sections are crucial for better results in further CRS attributes based applications like prestack data enhancement (Baykulov and Gajewski, 2009) and diffraction separation (Dell and Gajewski, 2011). After the successful application to synthetic data we test this method on field data.

FIELD DATA

The field data was acquired in the Levantine Basin in the Mediterranean Sea. It contains salt roller and various fault systems causing diffractions. For this data we use a population size of 50 and 40 iterations to achieve a stable optimization result in a reasonable amount of time. The stack is shown in figure 9 where we observe a high number of diffractions in the salt area at 2.5 s. The stack in general shows a high signal to noise ratio and continuous events due to a good fit of the operator. In the deeper part of the stack we observe multiples. The coherence section (figure 10) shows a lot of energy for diffractions which can be

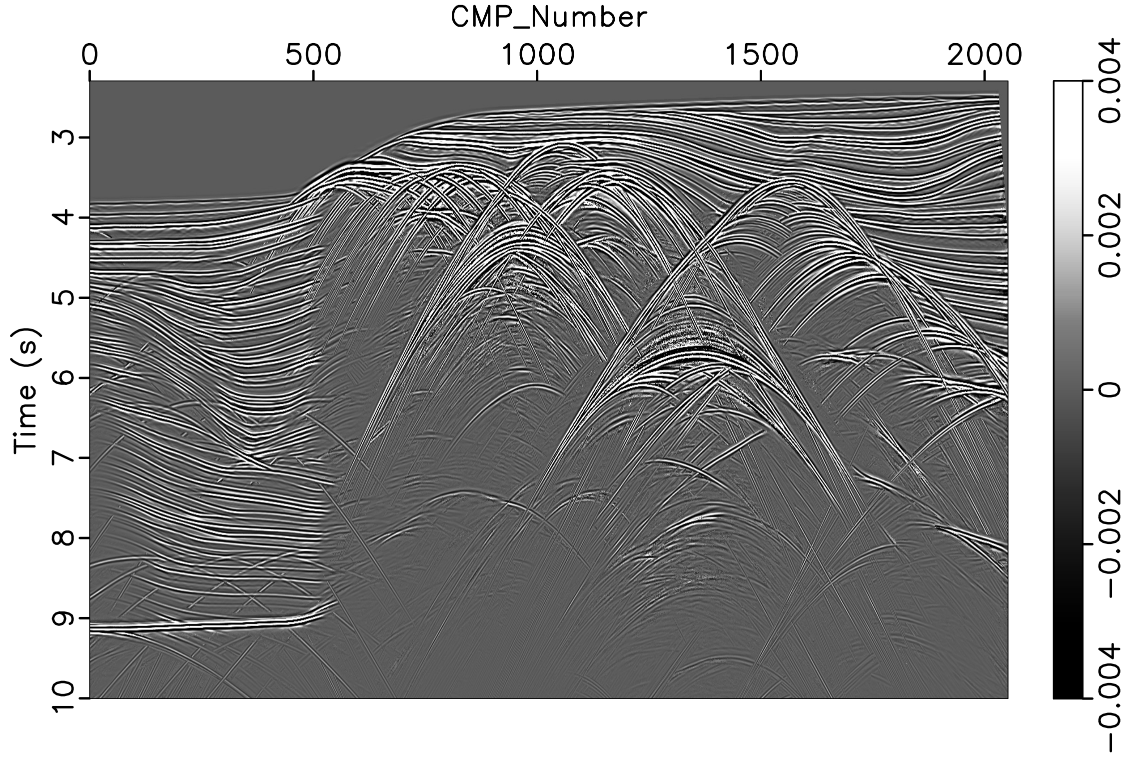

Figure 1: Sigsbee2a stack obtained with the CRS operator using a hybrid genetic algorithm optimization.

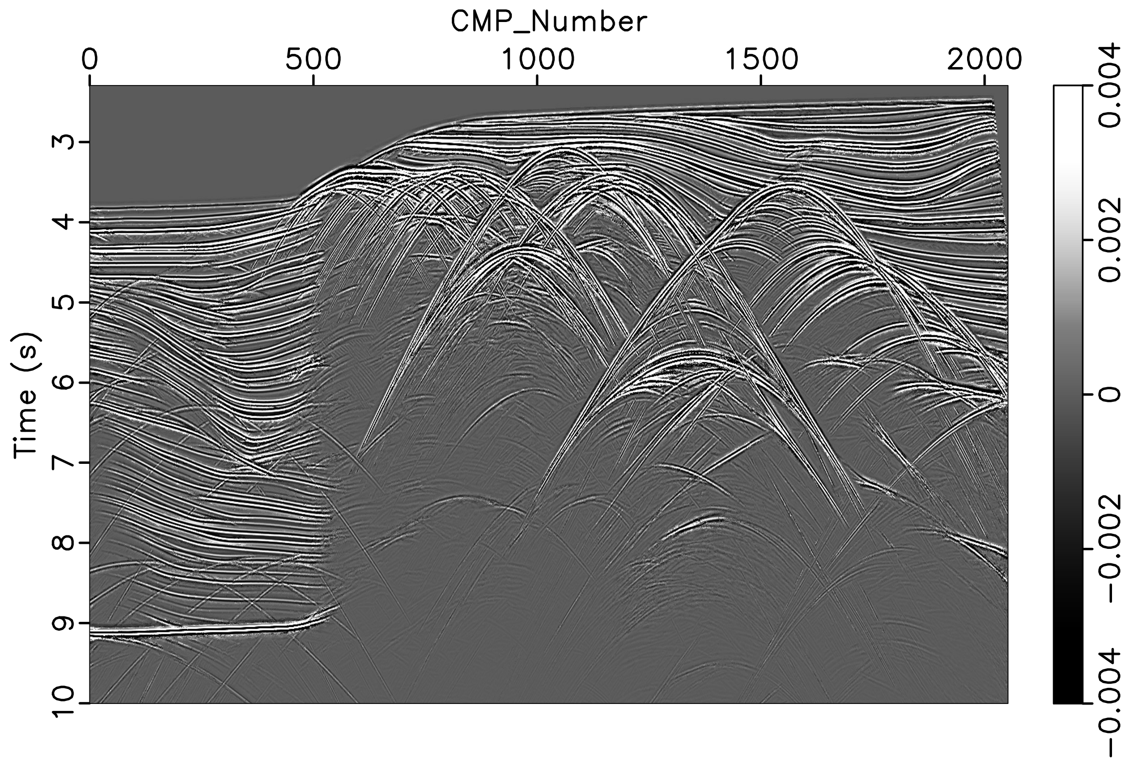

Figure 2: Sigsbee2a stack obtained with the CRS operator using the pragmatic approach and a local optimization.

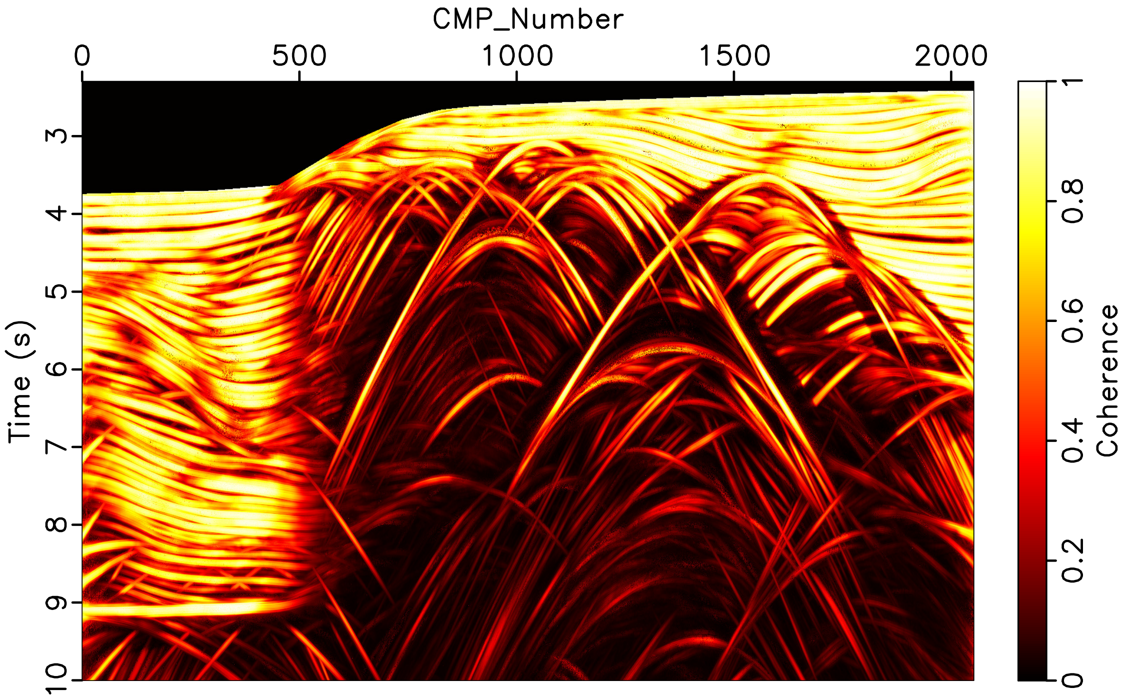

Figure 3: Sigsbee2a coherence section obtained with the CRS operator using a hybrid genetic algorithm optimization.

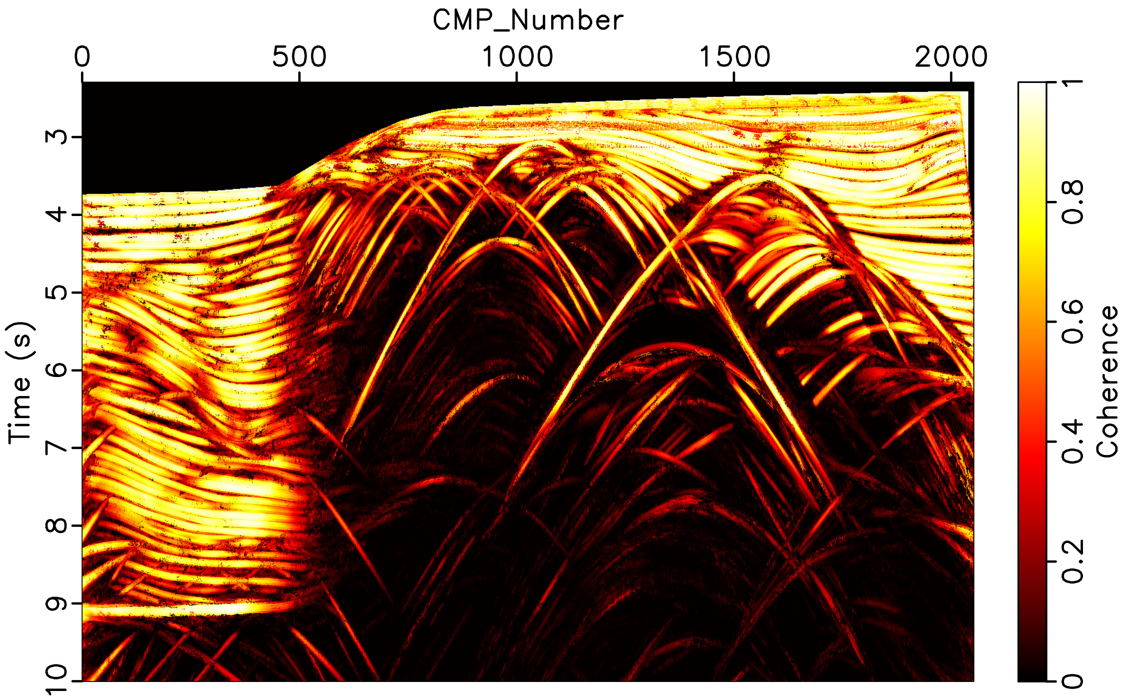

Figure 4: Sigsbee2a coherence section obtained with the CRS operator using the pragmatic approach and a local optimization.

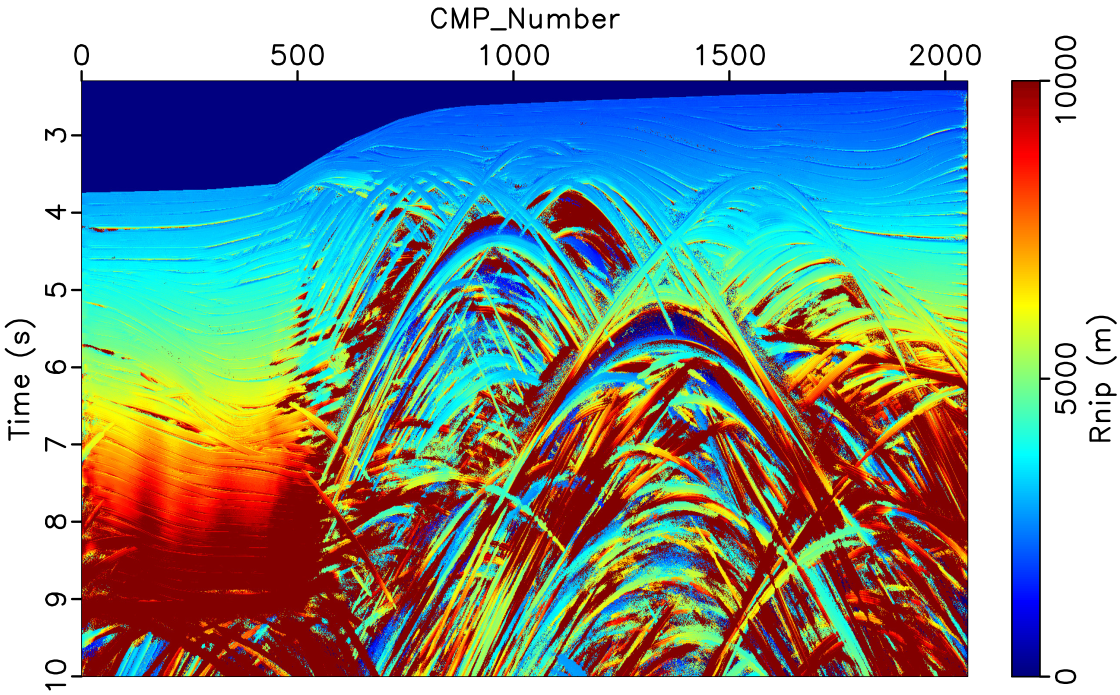

Figure 5: Sigsbee2a R_{NIP} obtained with the CRS operator using a hybrid genetic algorithm optimization.

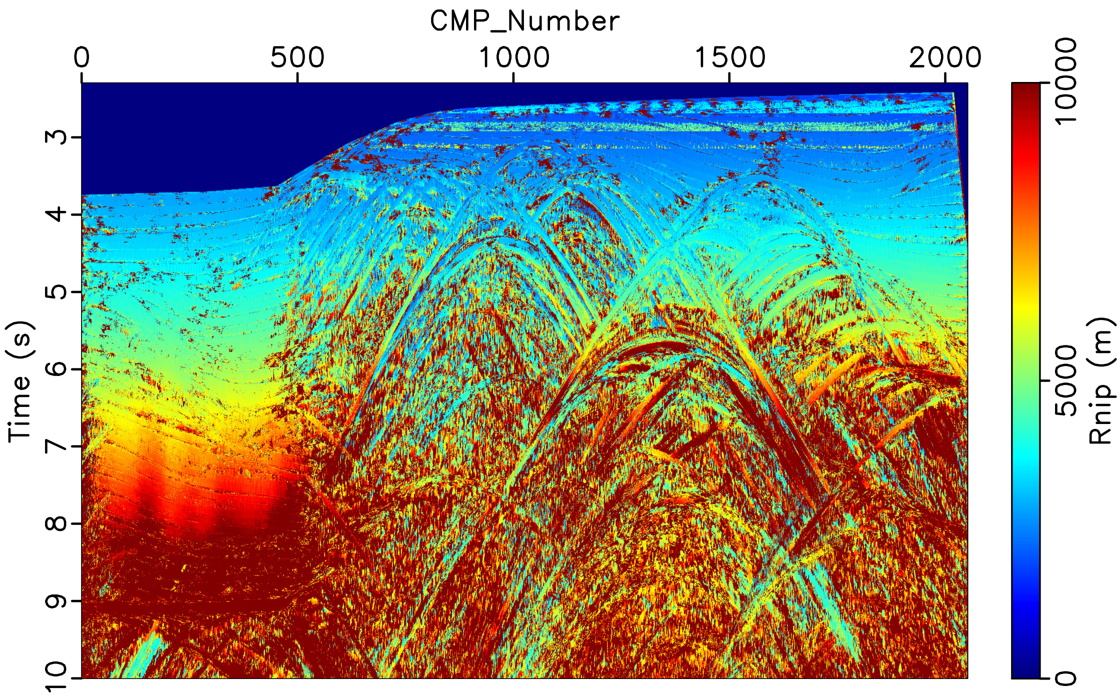

Figure 6: Sigsbee2a R_{NIP} obtained with the CRS operator using the pragmatic approach and a local optimization.

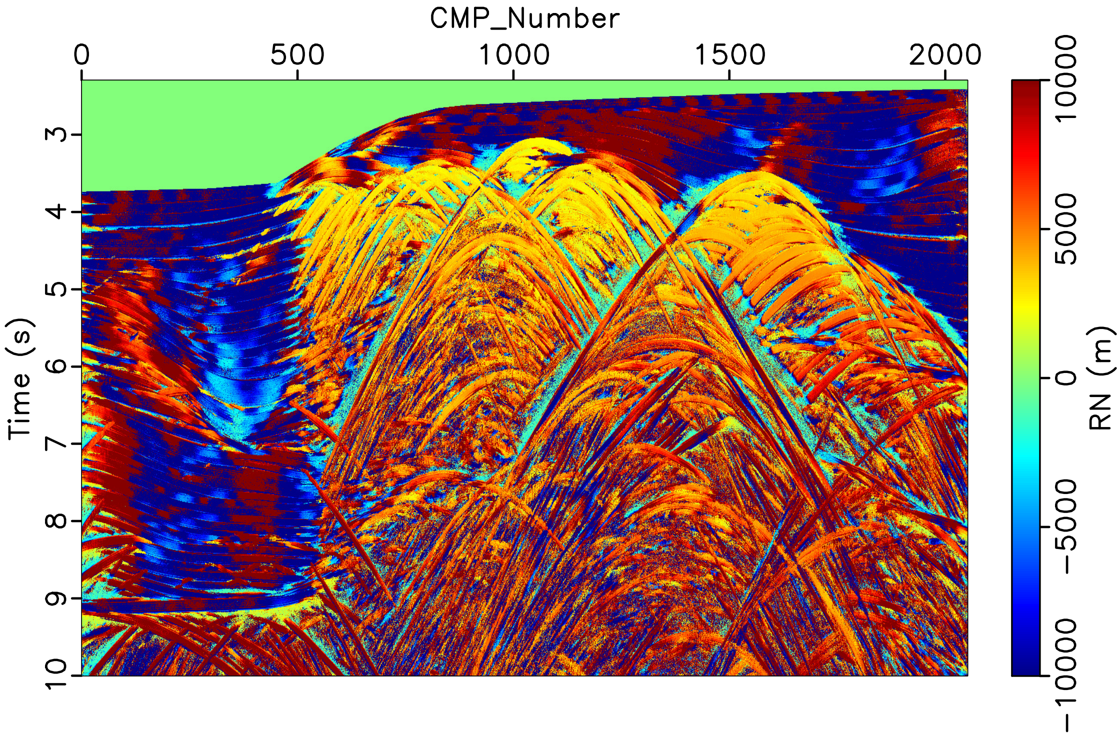

Figure 7: Sigsbee2a R_N obtained with the CRS operator using a hybrid genetic algorithm optimization.

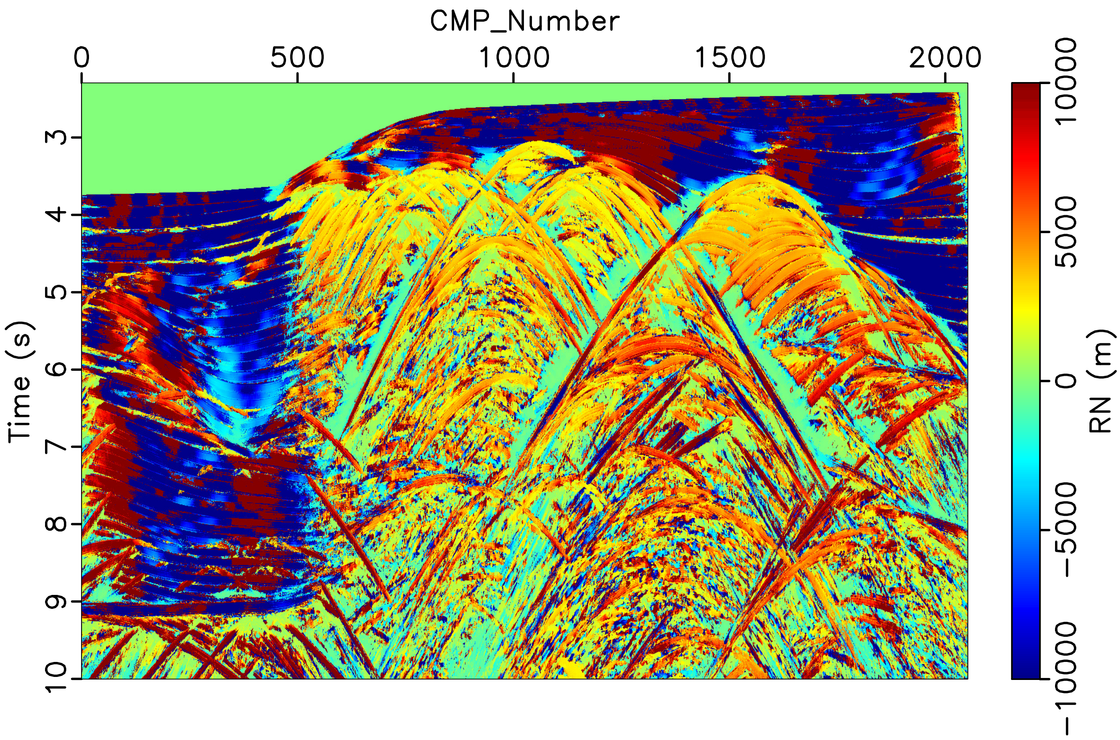

Figure 8: Sigsbee2a R_N obtained with the CRS operator using the pragmatic approach and a local optimization.

observed in all parts of the section. The R_{NIP} section (figure 11) is very smooth up to the time multiples occur. Only very few samples show a bad parameter estimation most notably at the seafloor. The dark blue part at the top (water column) was excluded from the optimization to save computation time. The salt body shows significant higher R_{NIP} values. The R_N section is displayed in figure 12. It is the least stable parameter of the CRS parameters. The section is more noisy compared to the R_{NIP} section, especially when R_N alternates between $-\infty$ and $+\infty$ as in the case of a horizontal reflector.

The global optimization shows very reasonable and smooth parameter sections for the field data. However it is very costly since the calculation of the objective function is expensive and global optimization schemes require more function evaluations than other approaches. The computational effort can be reduced significantly by including results from neighboring samples and previous results into the starting population to improve convergence.

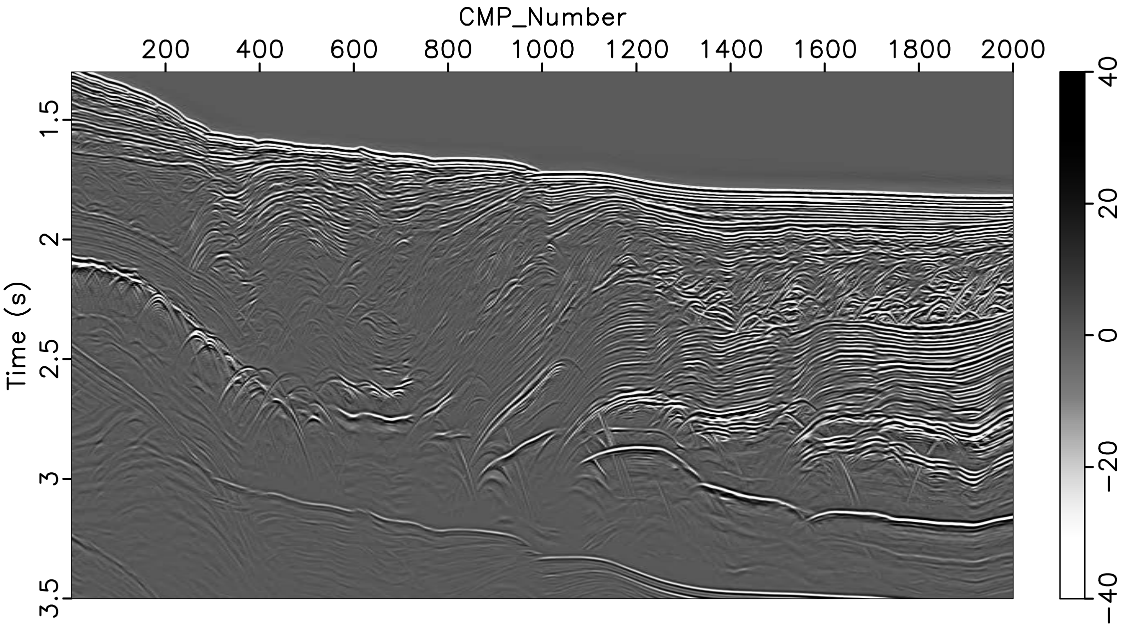

Figure 9: Field data stack obtained with the CRS operator using a hybrid genetic algorithm optimization.

CONCLUSION

The determination of high quality attributes is crucial for their use in further processing. The genetic algorithm shows great potential in obtaining good CRS attributes especially when combined with a local hill climbing method. The stack is more consistent and reveals more details in the sub-salt and salt area which is a result of a better obtained coherence by fitting the CRS operator using the full data volume and not just parts of it. The parameter sections are smooth and reasonable.

ACKNOWLEDGMENTS

The work was supported by the WIT consortium. The synthetic seismic data were produced by the Subsalt Multiples Attenuation and Reduction Technology Joint Venture (SMAART JV). The field data were kindly provided by TGS. We would like to thank the Applied Seismics Group Hamburg for continuing discussion.

REFERENCES

Bäck, T. (1996). *Evolutionary Algorithms in Theory and Practice*. Oxford Univ. Press.

Baker, J. (1985). Adaptive selection methods for genetic algorithms. In Grefenstette, J. and Erlbaum,

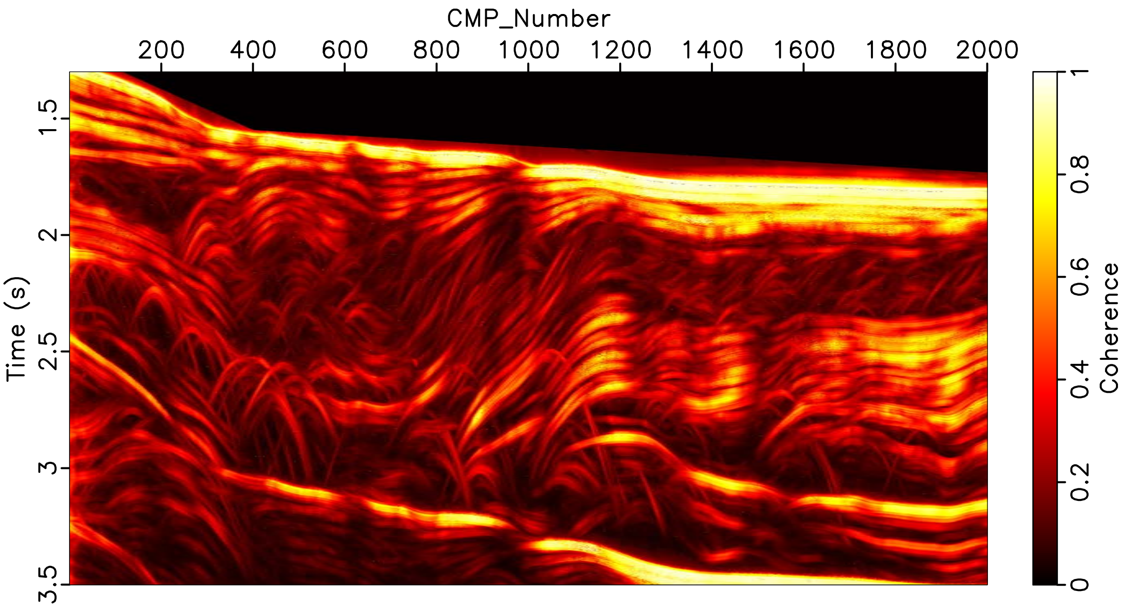

Figure 10: Sigsbee2a coherence section obtained with the CRS operator using a hybrid genetic algorithm optimization.

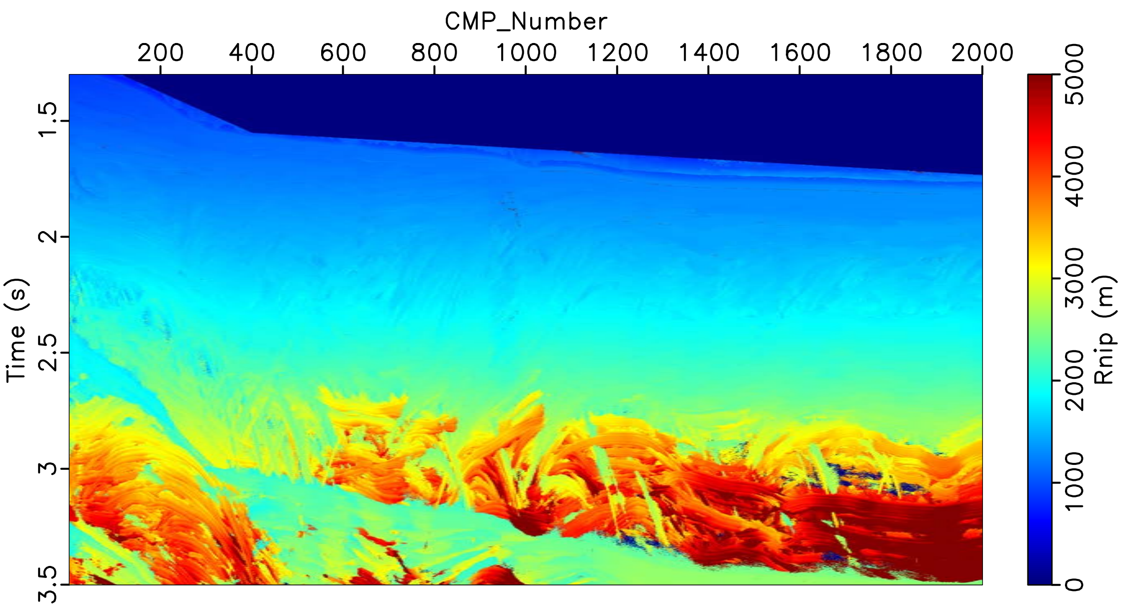

Figure 11: Sigsbee2a R_{NIP} section obtained with the CRS operator using a hybrid genetic algorithm optimization.

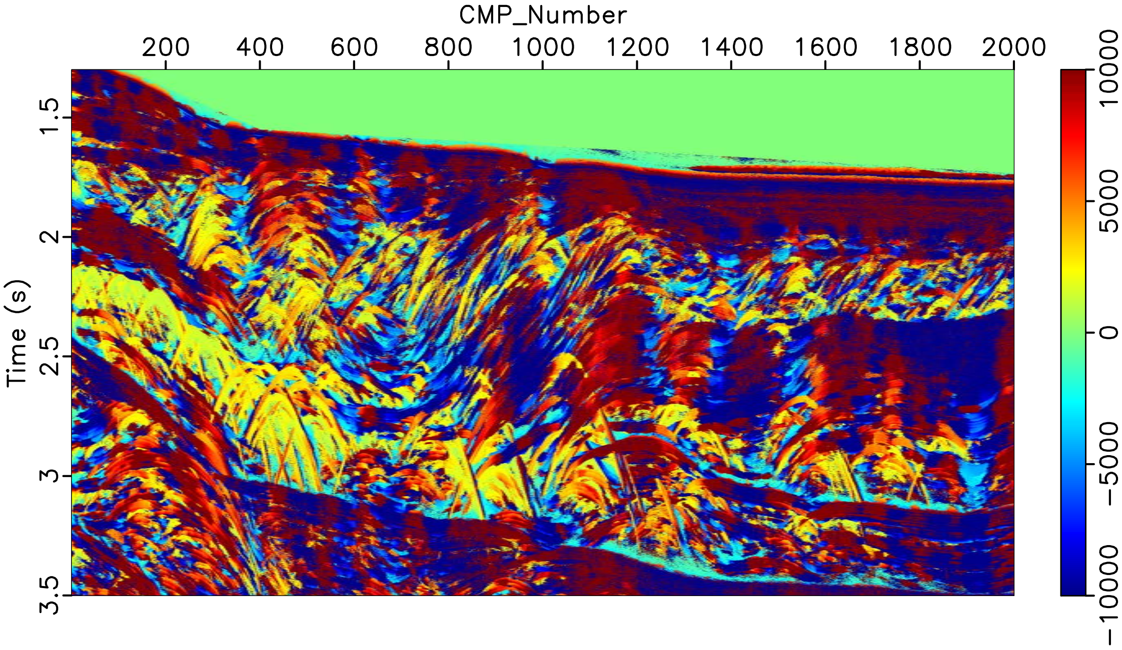

Figure 12: Sigsbee2a R_N section obtained with the CRS operator using a hybrid genetic algorithm optimization.

L., editors, *Proceedings of the International Conference on Genetic Algorithms and Their Applications*, Hillsdale, NJ.

Baykulov, M. and Gajewski, D. (2009). Prestack seismic data enhancement with partial common-reflection-surface (crs) stack. *Geophysics*, 74(3):V49–V58.

Davis, D. (1991). *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York.

Dell, S. and Gajewski, D. (2011). Common-reflection-surface-based workflow for diffraction imaging. *Geophysics*, 76(5):S187.

Eigen, M. (1973). *Ingo Rechenberg: Evolutionsstrategie Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*. Friedrich Frommann Verlag, Stuttgart-Bad Cannstatt. with an afterword by Manfred Eigen.

Goldberg, D. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Professional, 1 edition.

Goldberg, D. (1990). A note on boltzmann tournament selection for genetic algorithms and population-oriented simulated annealing. Technical Report TCGA 90003, Engineering Mechanics, University of Alabama.

Grefenstette, J. J. and Baker, J. (1989). How genetic algorithms work: a critical look at implicit parallelism. In *Proceedings of the Third International Conference on Genetic Algorithms*. Morgan Kaufmann, San Mateo, CA.

Holland, J. (1975). *Adaption In Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Hubral, P. (1983). Computing true amplitude reflections in a laterally inhomogeneous earth. *Geophysics*, 48:1051–1062.

Janikow, C. Z. and Michalewicz, Z. (1991). An experimental comparison of binary and floating point representations in genetic algorithms. pages 31–36. Proceedings of the Fourth International Conference on Genetic Algorithms.

Kitano, H. (1990). Designing neural networks using genetic algorithms with graph generation system. *Complex Systems Journal*, 4:461–476.

Mann, J. (2002). *Extensions and Applications of the Common-Reflection-Surface Stack Method*. PhD thesis, Logos Verlag, Berlin.

Morris, G. M., Goodsell, D. S., Halliday, R. S., Huey, R., Hart, W. E., Belew, R. K., and Olson, A. J. (1998). Automated docking using a lamarckian genetic algorithm and an empirical binding free energy function. *Journal of computational chemistry*, 19(14):1639–1662.

Mühlenbein, H. (1992). How genetic algorithms really work: I. mutation and hillclimbing. In Männer, R. and Manderick, B., editors, *Parallel Problem Solving from Nature*, volume 2, North Holland, Amsterdam.

Müller, T. (1999). *The Common Reflection Surface Stack Method: Seismic Imaging without explicit knowledge of the velocity model*. PhD thesis, University of Karlsruhe.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7:308–313.

Whitley, D. (1994). A genetic algorithm tutorial. *Statistics and computing*, 4.2:65–85.