Limits to Nonlinear Inversion

Klaus Mosegaard

Department of Informatics and Mathematical Modeling
and Center for Energy Resources Engineering
Technical University of Denmark
Richard Petersens Plads, 2800 Lyngby, Denmark
http://www.imm.dtu.dk/~kmos

Abstract. For non-linear inverse problems, the mathematical structure of the mapping from model parameters to data is usually unknown or partly unknown. Absence of information about the mathematical structure of this function prevents us from presenting an analytical solution, so our solution depends on our ability to produce efficient search algorithms. Such algorithms may be completely problem-independent (which is the case for the so-called 'meta-heuristics' or 'blind-search' algorithms), or they may be designed with the structure of the concrete problem in mind.

We show that pure meta-heuristics are inefficient for large-scale, nonlinear inverse problems, and that the 'no-free-lunch' theorem holds. We discuss typical objections to the relevance of this theorem.

A consequence of the no-free-lunch theorem is that algorithms adapted to the mathematical structure of the problem perform more efficiently than pure meta-heuristics. We study problem-adapted inversion algorithms that exploit the knowledge of the smoothness of the misfit function of the problem. Optimal sampling strategies exist for such problems, but many of these problems remain hard.

1 Introduction

Nonlinear inverse problems occur frequently in analysis of physical data, and a variety of algorithms are used to produce acceptable solutions and to analyze their properties. Some problems are only weakly nonlinear and can be locally approximated by linear problems, but others are strongly nonlinear and require special treatment. Modern digital computers have greatly improved our ability to perform nonlinear data inversion, but still the limitations of current techniques are strongly felt.

In this paper we intend to review and analyze some fundamental computational limitations to the solution of nonlinear inverse problems. We will put special emphasis on the interplay between the solution algorithm and the structure of the problem to be solved. Our exposition will, in principle, be relevant for the solution of inverse problems in general, but given the fact that nonlinear inverse theory relies much more on the theory of search- and sampling algorithms than linear theory does, our considerations will be most relevant for the nonlinear case.

K. Jónasson (Ed.): PARA 2010, Part I, LNCS 7133, pp. 11–21, 2012.

2 The Blind Inversion Problem

It is often reported in the literature that solutions to nonlinear inverse problems were obtained by problem-independent algorithms, the so-called *meta-heuristics*. This type of algorithms is claimed to work efficiently because of some general, external (problem independent) principle. For example, simulated annealing [5] inherits its efficiency from thermodynamic principles, genetic algorithms [6,7] exploit evolutionary principles, and taboo-search [8] uses some 'common sense' strategy. In the following, we will call these algorithms *blind inversion algorithms*, and we will investigate their efficiency in some detail.

Blind inversion schemes may be very different in character. They include deterministic as well as Monte Carlo algorithms, and they all share the basic property that they operate in a way that is independent of the particular inverse problem. A blind inversion scheme operates - sequentially or in parallel - by evaluating a misfit function (or fit function) in points in the parameter space, and a particular scheme is solely characterized by the strategy by which it selects new evaluation points from earlier selected points and their misfits. Some algorithms estimate gradients of the misfit (e.g., steepest descent), some use a proposal distribution combined with an acceptance probability (e.g., simulated annealing), others compare misfits at several points and use a selection strategy (genetic algorithms and the neighborhood algorithm [4,9,10].

In the following we shall analyze the performance of blind algorithms, but first we will briefly review some general topological properties of general inverse problems that impede algorithm efficiency when searching high-dimensional parameter spaces.

3 Basic Limitations in Blind Inversion Arising from the Structure of the Problem

Consider a discrete, nonlinear inverse problem

$$\mathbf{d} = \mathbf{g}(\mathbf{m}) \tag{1}$$

where $\mathbf{d}$ is a vector with $N + P$ components, the first N equations relate data $(d_1, \ldots, d_N)$ to M model parameters $\mathbf{m}$, and the remaining P equations express our prior information through $P \leq M - N$ equality constraints. If we assume that $\mathbf{g}$ is a $\mathcal{C}^1$-function, and that we have a solution $\hat{\mathbf{m}}$ to equation (1) for which

$$\mathbf{g}'(\hat{\mathbf{m}}) = \left[\frac{\partial g_i}{\partial m_j} \right]_{\mathbf{m}=\hat{\mathbf{m}}} \tag{2}$$

has full row rank, the implicit function theorem [11] implies that eq. (1) defines a solution-submanifold of dimension $M-(N+P)$ in the neighborhood around $\hat{\mathbf{m}}$ in the parameter space. If data uncertainties and 'softness' of the prior constraints

are described by a neighborhood around $\mathbf{d}$, acceptable solutions in the parameter space can be found in a neighborhood around the solution manifold intersecting $\hat{\mathbf{m}}$.

In inversion aimed at locating only one acceptable solution, one will generally try to supply a sufficient number of tight a priori constraints to ensure that $M = N + P$. In this case, the solution space consists of neighborhoods around isolated points in the parameter space.

Of course, the above description is based on a number of simplifications. For instance: (1) We assumed that the matrix in equation (2) has full row rank. If this is not the case, the mapping g is not surjective, meaning that the dimension of the solution space is larger than in the full-rank case. (2) We have only considered prior information defined as (possibly softened) equality constraints. Introducing constraints that allow the solution to exist only inside a bounded, possibly non-convex, region may render the inverse problem even harder. (3) Strictly speaking, in our theory M should not be the number of model parameters, but the number of degrees of freedom in the model space. The arguments above can easily be modified to take this into account.

4 Basic Algorithmic Limitations in Blind Inversion

4.1 The Performance of Blind Inversion Algorithms

We now turn to the question of assessing the relative merits of blind inversion algorithms. The usual two ways of evaluating relative performances have been (1) to argue for or against algorithms using common-sense, physical or other arguments to discuss their ability to, e.g., locate acceptable misfit regions, or (2) to select suitable test problems and arrange numerical contests between selected algorithms. The first approach does not provide quantitative measures of relative performance, and the latter method is so sensitive to the selected problems and the 'tuning' of each of the considered algorithms that general conclusions are tentative.

Here, we will follow a different path leading to a quantitative comparison. Our reasoning will be similar to the one behind the so-called *No-Free-Lunch Theorem* [14]. It is based on a discretization where a finite number of parameters and data are only allowed to attain a finite set of values. This *double discretization* is actively used in genetic algorithms where parameters are often assumed to be binary numbers with only a few bits. All other numerical methods are, of course, also doubly discretized, because they run on digital computers with a finite precision.

The doubly discretized inverse problem and the information collected by an arbitrary, blind inversion algorithm will be described through the following notation. We consider:

- A *finite set* $\mathcal{M}$ of models. This set consists of all combinations of a finite number of values attained by a finite set of parameters.
- A *finite set* set $\mathcal{S}$ of real numbers. These numbers are the possible fit or misfit values that can be generated by models in $\mathcal{M}$.

- The set $\mathcal{F}_{\mathcal{M}}$ of all fit functions $f : \mathcal{M} \to \mathcal{S}$.
- A sample of size $m < |\mathcal{M}|$ generated by an algorithm having sampled $f \in \mathcal{F}$ in m distinct points:

$$\{(\mathbf{m}_1, s_1), \ldots, (\mathbf{m}_m, s_m)\}. \tag{3}$$

  Note, that points resampled by an algorithm will only count once. On the other hand, auxiliary sample points used by the algorithm, e.g. points sampled with the sole purpose of calculating an approximate 'gradient', count on equal footing with other sample points.
- The (time) ordered set of sample points (arguments indicate time-ordering):

$$C = \{\mathbf{m}_1, \ldots, \mathbf{m}_m\}. \tag{4}$$

- The (time) ordered set of corresponding values of f:

$$s_1, \ldots, s_m. \tag{5}$$

- The set $\mathcal{F}_{\mathcal{M}|C}$ of all fit functions/probability distributions defined on $\mathcal{M}$, but with fixed values in C.

Consider a blind inversion problem where we have no knowledge of the actual fit function, and we search for at least one acceptable solution to the problem. From the outset, the total number of possible fit functions is equal to

$$|\mathcal{F}_{\mathcal{M}}| = |\mathcal{S}|^{|\mathcal{M}|}. \tag{6}$$

We can now ask: What is the probability that an algorithm, when sampling $\mathcal{M}$ in m distinct points, sees the function values $s_1, \ldots, s_m$? To compute this we observe that, when fixing the function values at the m points of subset C (see Figure 1), the number of remaining, possible fit functions is narrowed in and is equal to

$$|\mathcal{F}_{\mathcal{M}|C}| = |\mathcal{S}|^{|\mathcal{M}|-m}. \tag{7}$$

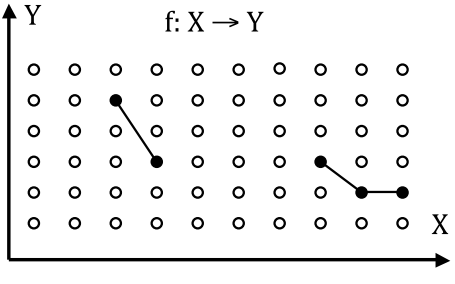

Fig. 1. Knowledge of an unknown fit function f after evaluation of the function in 5 points

This means that the probability that an algorithm in m function evaluations sees the particular fit function values $s_1, \ldots, s_m$ is

$$P(s_1, \ldots, s_m | a, m) = |\mathcal{F}_{\mathcal{M}|C}| |\mathcal{F}_{\mathcal{M}}|^{-1} = |\mathcal{S}|^{-m} \tag{8}$$

This number is independent of the location of the sample points and hence the algorithm used. Any algorithm a performing m iterations where it visits (algorithm dependent) points $\mathbf{m}_1^{(a)}, \ldots, \mathbf{m}_m^{(a)}$ can obtain functional values $s_1, \ldots, s_m$ with exactly $|\mathcal{F}_{\mathcal{M}|C}|$ different fit functions $f \in \mathcal{F}_{\mathcal{M}}$, and $|\mathcal{F}_{\mathcal{M}|C}|$ is independent of a, so for any pair of algorithms a_1 and a_2 we have

$$P(s_1, \ldots, s_m | \, m, a_1) = P(s_1, \ldots, s_m | \, m, a_2) \tag{9}$$

where $P(\cdot \,|\, \cdot)$ denotes conditional probability.

Any performance measure for inversion algorithms searching for near-optimal data fits is of the form $\Phi : \mathcal{S}^m \to \mathbf{R}$, for instance:

$$\Phi(s_1, \ldots, s_m) = \max\{s_1, \ldots, s_m\}, \tag{10}$$

which must be large for good performance. Even Monte Carlo algorithms, aimed at importance-sampling of probability distributions over the parameter space, operate as near-optimization algorithms in the computer-intensitive, initial phase (the burn-in phase) where the first acceptable solution is sought.

The probability distribution of $\Phi(s_1, \ldots, s_m)$ depends only on $P(s_1, \ldots, s_m | \, m, a)$, and as a consequence of equation (9) it is therefore independent of the algorithm a. We have now shown

Theorem 1. *(Similar to the* **No-Free-Lunch Theorem** *by Wolpert and Macready [14]). The distribution of any performance measure for inversion, when all fit functions are equally probable (blind inversion), is exactly the same for all inversion algorithms.*

4.2 Critique of the No-Free-Lunch Theorem

One obvious critique of the usefulness of the No-Free-Lunch Theorem is that, in typical fields of application, the fit functions belong to a *narrow subfamily* of functions (e.g., smooth functions), and some algorithms work better than others on such families. This objection is based on the observation that for each function sub-family $\mathcal{G}_{\mathcal{M}} \subseteq \mathcal{F}_{\mathcal{M}}$, the total number of ways a particular set of fit values $s_1, \ldots, s_m$ can be obtained in a set of m sample points $\mathbf{m}_1, \ldots, \mathbf{m}_m$ from fit functions $f \in \mathcal{G}_{\mathcal{M}}$ will in general depend on $\mathbf{m}_1, \ldots, \mathbf{m}_m$ (and hence on the algorithm a that is choosing the sample points). However, this objection is, as we shall see now, invalid in the blind inversion case.

Consider all subfamilies $\mathcal{G}_{\mathcal{M}}$ of functions in $\mathcal{F}_{\mathcal{M}}$. Functions in subfamily $\mathcal{G}_{\mathcal{M}}$ with fixed values on the subset C form the set $\mathcal{G}_{\mathcal{M}} \cap \mathcal{F}_{\mathcal{M}|C}$. The average number of functions in such a subfamily is proportional to

$$\sum_{\forall \mathcal{G}_{\mathcal{M}} \subseteq \mathcal{F}_{\mathcal{M}}} |\mathcal{G}_{\mathcal{M}} \cap \mathcal{F}_{\mathcal{M}|C}| = \sum_{\forall \mathcal{G}_{\mathcal{M}\setminus C}} |\mathcal{G}_{\mathcal{M}\setminus C}| \tag{11}$$

The right-hand side of (11) depends only on $|\mathcal{M}| - |C|$ and hence only on the number m. Hence, the actual elements of C, which are chosen as a result of a's search strategy, does not influence the average number of functions in $\mathcal{G}_{\mathcal{M}} \cap \mathcal{F}_{\mathcal{M}|C}$. We have demonstrated

Theorem 2. *The efficiency of blind inversion algorithms, averaged over all subfamilies of fit functions, is the same for all inversion algorithms.*

We have found that the efficiency of all blind inversion schemes, e.g., Steepest ascent/descent, Simulated Annealing, Markov-Chain Monte Carlo, Random Search, Genetic Algorithm, Neighborhood Algorithm, Taboo Search, Line Search and Exhaustive Search are exactly the same, when averaged over alle possible inverse problems or subsets of problems.

How can we accept this surprising result when all experience shows that, for instance, Random Search is much less efficient than Genetic Algorithms? The only explanation for the apparent paradox is that the above mentioned blind algorithms are, in practice, supplemented with procedures that inform the algorithm about the structure of the problem. As we shall discuss in the next section, this is indeed the case.

5 The Informed Inversion Problem

5.1 The Necessity of Algorithm Tuning

We have demonstrated above that the performance of all blind inversion schemes are exactly the same, when averaged over all conceivable fit functions, or subsets hereof. This means that if, for example, a genetic algorithm performs better than a crude random search on certain problems, it will perform worse on other problems. Apparently, this result is contradicted by the experience of a vast number of researchers who have seen popular algorithms outperforming crude random search by several orders of magnitude.

The only way we can resolve this paradox is to point at the *tuning* of inversion algorithms. Most expositions explaining the functioning of inversion algorithms emphasize the external ideas behind their design, and attribute the algorithm's efficiency to these ideas. Simulated Annealing is, for example, relying on an idea taken from natural minimization of the internal energy seen in thermodynamical systems under slow cooling, Genetic algorithms use ideas from natural biological selection to generate near-optimal solutions, etc. The consequence of Theorem 1 above is, however, that none of these external design ideas can provide any degree of success. Instead, we must turn to the tuning of algorithms to obtain the desired efficiency.

Unfortunately, there is a vast number of very different ways of tuning inversion algorithms. For this reason we shall only discuss one of the most important ideas, namely tuning to a known smoothness of the fit function through the choice of distance between sample points, for some algorithms termed the 'step length'. An algorithm that is tuned to the problem is no longer a blind algorithm (a metaheuristic). It has become a so-called *heuristic* – an informed inversion algorithm.

5.2 A Lower Bound on the Required Number of Iterations by a Smoothness-Tuned Algorithm

Assume that the parameter space $\mathcal{M} \in \mathbf{R}^M$ is an M-dimensional cube of edge length L, and consider a class $\mathcal{F}$ of fit functions defined over $\mathcal{M}$. Assume that fit functions $f \in \mathcal{F}$ can be approximated to any precision by

$$f(\mathbf{m}) \approx \sum_{j=1}^{J} u_j \ \phi_j(\mathbf{m}) \tag{12}$$

where $\phi_1, \phi_2, \ldots, \phi_J$ is a set of linearly independent basis functions, and where J will depend on the required precision. For instance, according to a theorem by Brown [15], any *continuous* fit function $f \in \mathcal{F}$ can be approximated to any precision by (12) if $\phi_1, \phi_2, \ldots, \phi_J$ are *radial basis functions* of the form

$$\phi_j(\mathbf{m}) = g(\|\mathbf{m} - \mathbf{m}_j\|^2), \quad \mathbf{m}_j \in Q \tag{13}$$

where g is a non-constant, completely monotone[1] function defined on $[0, \infty[$, and Q is a compact subset of $\mathbf{R}^M$ containing more than one point. The radial basis functions $\phi_k(\mathbf{m})$ have 'spherical symmetry', and they are translates of each other (have different $\mathbf{m}_j$). The class of radial basis functions is very wide, containing members as, for instance, $\exp(-\|\mathbf{m}-\mathbf{m}_j\|^2)$, $\|\mathbf{m}-\mathbf{m}_j\|$ and $\|\mathbf{m}-\mathbf{m}_j\|^2 \ln(\|\mathbf{m}-\mathbf{m}_j\|)$.

Assume first that the only information we have about a fit function f is that it is 'band limited' in the sense that it is given exactly by the finite linear combination

$$f(\mathbf{m}) = \sum_{j=1}^{J} u_j \ \phi_j(\mathbf{m}), \tag{14}$$

and that we search for acceptable solutions to the inverse problem. Given that an algorithm has sampled f in k distinct points $\mathbf{m}_1, \ldots, \mathbf{m}_K$ and recorded the corresponding function values $s_1, \ldots, s_K$, what is the chance that we have located the neighborhood of the global maximum/minimum for f? To answer this question, let us express our knowledge after K iterations through the equations:

$$s_k = \sum_{j=1}^{J} u_j \ g(\|\mathbf{m}_k - \mathbf{m}_j\|^2) \qquad k = 1, \ldots, K \tag{15}$$

If we define the vectors $\mathbf{s} = (s_1, \ldots, s_K)^T$, $\mathbf{u} = (u_1, \ldots, u_J)^T$, and the matrix $\{\mathbf{G}\}_{kj} = g(\|\mathbf{m}_k - \mathbf{m}_j\|^2)$, equation (15) can be written

$$\mathbf{s} = \mathbf{G}\mathbf{u} \ . \tag{16}$$

[1] A function g defined on $[0, \infty[$ is completely monotone if g is continuous on $[0, \infty[$, infinitely often differentiable on $]0, \infty[$), and $(-1)^k g^{(k)}(t) \geq 0$ for $t > 0$ and $k = 1, 2, 3, \ldots$.

If the solution $\mathbf{u}$ to this equation is non-unique, at least one of its components $u_1, \ldots, u_J$ is unbounded (that is, can take any value without violating equation (16)). If u_l is such an unbounded component, f is (at best) only known modulo an unbounded additional term $u_l \; g(\|\mathbf{m} - \mathbf{m}_l\|^2)$. It is clear that the unknown u_l renders f's value in $\mathbf{m}_l$, and hence f's global maximum/minimum, undetermined. In other words, unless we can uniquely determine $\mathbf{u}$, we will be unable to locate the global maximum for f. $\mathbf{u}$ is uniquely determined only if $\mathbf{G}^T\mathbf{G}$ is non-singular, and a necessary condition for this to be satisfied is that $K \geq J$. We have shown

Theorem 3. *An inverse problem, whose fit function is known to be a linear combination of a linear independent set of J basis functions, cannot be solved through less than J distinct function evaluations.*

In the general case, we cannot expect to express f exactly with a finite number of basis functions. Then f is only expressed with a certain accuracy $\epsilon > 0$, in the sense that the discrepancy

$$n(\mathbf{m}) = f(\mathbf{m}) - \sum_{j=1}^{J} u_j \; \phi_j(\mathbf{m}) \tag{17}$$

is constrained by

$$max_{\mathbf{m}} \, |n(\mathbf{m})| \leq \epsilon \quad . \tag{18}$$

In this case the problem is that equation (16) does not reliably determine the coefficient vector $\mathbf{u}$ when $\mathbf{G}^T\mathbf{G}$ is ill-conditioned. The discrepancy propagates into the coefficient vector, creating large errors, and the location of the global maximum of f remains unknown.

The above considerations set a fundamental, unavoidable lower limit to the number of iterations required by any inversion algorithm working with a 'band-limited' fit function. We have not described how an ideal (maximum efficiency) algorithm should work, but it is clear from the discussion that, contrary to the 'blind inversion' case, not all algorithms are equally good.

To see this, consider again equation (16). An algorithm choosing its first J distinct sampling points such that $\mathbf{G}^T\mathbf{G}$ is non-singular and well-conditioned has collected sufficient information to locate the global maximum for f (although we have not shown how to do this). On the other hand, an algorithm choosing its first J distinct sampling points such that $\mathbf{G}^T\mathbf{G}$ is singular or ill-conditioned is still missing information about the location of the global maximum of f. For instance, a sub-optimal algorithm may, after J distinct function evaluations, have failed to sample all J basis functions in points sufficiently near their maxima. Such an algorithm will need more than J distinct function evaluations to render equation (16) solvable.

6 The Complexity of an Inverse Problem with Known Smoothness

Consider an inverse problem with a fit function f defined in an interval $\mathcal{M}$ of edge length L in $\mathbf{R}^M$ (an M-dimensional 'box'). Assume that f is smooth in the sense that it can be expanded in a linear independent, finite set of *radial* basis functions, centered in a regular grid in $\mathcal{M}$ with grid spacing l. The required, total number of basis functions is then $(L/l + 1)^M$, and according to Theorem (3) this is the smallest number of distinct function evaluations needed to solve the inverse problem. To reach this minimum number of evaluations, the safest strategy for an algorithm is to sample close to the maxima of the basis functions, and this calls for a sampling distance close to a multiple of l.

It should be noted, however, that in this important case the smallest number of distinct function evaluations needed by any algorithm to solve the inverse problem grows at least exponentially in M. This growth is severe, and shows that the inverse problem is *hard* in the sense that the solution time grows faster than any polynomial function [16]. In practice, this means that even significant improvements in computer speed will only allow the inverse problem to be solved with a few more model parameters. Let us summarize this important observation in the following:

Theorem 4. *Consider an inverse problem for which our only knowledge is that its fit function can be expanded in a set of linearly independent, radial basis functions, and assume that the basis functions are centered in a regular grid covering the model space. Then the computation time for any algorithm aimed at solving the inverse problem will grow at least exponentially with the number of unknown model parameters.*

7 Discussion and Conclusion

Wolpert and Macready [14] showed that, contrary to the belief of many practicians, there is no difference between the performance of the many existing, and popular, meta-heuristics, unless they are tuned to the problem at hand (and therefore no longer problem-independent). This means that all attempts to improve on inversion algorithms must focus on the tuning. One of the most common tuning parameters is the 'step length' (as in, e.g., simulated annealing) or 'sampling density' (as in, e.g., the neighborhood algorithm). This kind of tuning applies to cases where smoothness is the only known property of the fit function.

Usually, the smoothness is determined empirically through experimentation with a range of sampling densities. An example is steepest descent algorithms where step lengths are adjusted in order to avoid 'instability' of the algorithm. Another example is the adjustment of the step length in Markov-chain Monte Carlo methods, until the rate of accepted moves is reasonable [17]. A third example is the neighborhood algorithm where the density of resampling can be adjusted.

In many inverse problems arising in physical sciences, the data, and hence the fit (or misfit) functions are smooth. An example is seismic waveform inversion, where the band-limitation of the seismic source function is inherited by the fit function. If one attempts to solve this problem with an algorithm that is only informed about the smoothness of the fit function (through numerical experimentation or through some spectral information), the solution time will grow exponentially with the number of unknown parameters. This means that large-scale, seismic waveform inversion problems are essentially unsolvable in this way. On the other hand, it is well known that solution of such problems may be feasible with 'well-informed' algorithms, based on the theory of seismic wave propagation.

We should mention a couple of objections that could be raised against our exposition. First, does the above theory account for the situation that an algorithm may sometimes, by accident, start its search/sampling close to a solution? The answer to this question is that in large-scale inverse problems with many unknown parameters, there is a negligible probability that an algorithm, only informed about the smoothness of f, would start near an acceptable solution. We have therefore ignored this situation.

A second objection concerns the fact that we have treated deterministic methods (searching only for one feasible solution), and Monte Carlo sampling methods (aiming at finding many feasible solutions) in a unified theory. Clearly, sampling methods start with a 'burn-in phase' which is comparable to deterministic methods in its aim at locating one acceptable model, but this phase is followed by a 'sampling phase' which is apparently the real production phase of the sampling algorithm. To what extent is the sampling phase considered in our theory? The answer is, that our theory only considers the burn-in phase of a Monte Carlo sampling. In this connection it should be remembered that the burn-in process not only concerns the initial search for acceptable solutions. If the fit function for the problem has many isolated islands of acceptable solutions, the burn-in time is also a measure of the time it takes for the algorithm to move from one solution island to the next.

As a final remark we should note that, although we have demonstrated that all blind inversion schemes are equally (in)efficient, and that efficient algorithms can only be obtained through problem-dependent tuning, it is certainly possible that some algorithm designs are more easily tuned than others. This may be responsible for some of the differences that practitioners observe between popular algorithms.

Acknowledgments. The ideas behind this paper grew out of numerous inspiring discussions with Albert Tarantola who died tragically 6. december 2009. The author would also like to thank Thomas Mejer Hansen for constructive discussions and criticism. The work was supported by The Danish Council for Independent Research - Natural Sciences (FNU) under contract no. 272-08-0568, and DONG Energy, Denmark.

References

1. Nolet, G., van Trier, J., Huisman, R.: A formalism for nonlinear inversion of seismic surface waves. Geoph. Res. Lett. 13, 26–29 (1986)
2. Nolet, G.: Partitioned wave-form inversion and 2D structure under the NARS array. J. Geophys. Res. 95, 8499–8512 (1990)
3. Snieder, R.: The role of nonlinearity in inverse problems. Inverse Problems 14, 387–404 (1998)
4. Sambridge, M.: Exploring multi-dimensional landscapes without a map. Inverse Problems 14(3), 427–440 (1998)
5. Kirkpatrick, S.C., Gelatt, D., Vecchi, M.P.: Optimization by simulated annealing. Science 220, 671–680 (1983)
6. Holland, J.H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor (1975)
7. Goldberg, D.E.: Genetic Algorithms in Search, Optimization and Machine Learning. Kluwer Academic Publishers, Boston (1989)
8. Glover, F.: Future Paths for Integer Programming and Links to Artificial Intelligence. Comput. & Ops. Res. 13(5), 533–549 (1986)
9. Sambridge, M.: Geophysical inversion with a Neighbourhood algorithm -I. Searching a parameter space. Geoph. Jour. Int. 138, 479–494 (1999a)
10. Sambridge, M.: Geophysical inversion with a Neighbourhood algorithm -II. Appraising the ensemble. Geoph. Jour. Int. 138, 727–746 (1999b)
11. Cauchy, A.L.: First Turin Memoir (1831)
12. Mosegaard, K., Tarantola, A.: Monte Carlo sampling of solutions to inverse problems. J. Geophys. Res. 100(B7), 12,431–12,447 (1995)
13. Khachiyan, L.G.: The problem of computing the volume of polytopes is np-hard. Uspekhi Mat. Nauk. 44, 199–200 (1989)
14. Wolpert, D.H., Macready, W.G.: No Free Lunch Theorems for Optimization. IEEE Transactions on Evolutionary Computation 1, 67–82 (1997)
15. Brown, A.L.: Uniform approximation by radial basis functions. In: Light, W.A. (ed.) Advances in Numerical Analysis, vol. 2, pp. 203–206. Oxford University Press, Oxford (1992)
16. Papadimitriou, C.H., Steiglitz: Combinatorial Optimization: Algorithms and Complexity. Dover Publications, Inc., Mineola (1998)
17. Hastings, W.K.: Monte Carlo sampling methods using Markov Chain and their applications. Biometrika 57, 97–109 (1970)