Interval-based validation of a nonlinear estimator

Maël Godard, Luc Jaulin, Damien Massé

Interval-based validation of a nonlinear estimator

Maël Godard [1], Luc Jaulin [1], Damien Massé [2]

[1] *ENSTA Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France.*
mael.godard@ensta-bretagne.org, luc.jaulin@ensta-bretagne.fr
[2] *UBO - Université européenne de Bretagne, Lab-STICC, UMR CNRS 6285, Brest, France.*
damien.masse@univ-brest.fr

Abstract. In engineering, models are often used to represent the behavior of a system. Estimators are then needed to approximate the values of the model's parameters based on observations. This approximation implies a difference between the values predicted by the model and the observations that have been made. It creates an uncertainty that can lead to dangerous decision making. Interval analysis tools can be used to guarantee some properties of an estimator, even when the estimator itself doesn't rely on interval analysis (Adam, 2019) (Adam, 2015). This paper contributes to this dynamic by proposing an interval-based and guaranteed method to validate a nonlinear estimator. It is based on the Moore-Skelboe algorithm (van Emden, 2004). This method returns a guaranteed maximum error that the estimator will never exceed. We will show that we can guarantee properties even when working with non-guaranteed estimators such as neural networks.

Keywords: Parameter estimation, Error maximisation, Interval analysis

1. Introduction

Before using any system it is important to have a model of it to predict it's behaviour. Once the modelization has been done, it is necessary to fit the model's parameter to the studied system. Estimators are used to approximate the value of these parameters based on observations made on the system. This approximation results in an error that can have a significant impact on the model if it gets too important. It can then be useful to characterise this error to validate if the estimator can be used or not.

One way to characterise it is to maximise it, i.e. to give a maximum error made by the estimator in a given context. In the case of Linear Parameter Estimation problems (LPE), many methods such as least squares have proven to be both efficient and easy to use (Lakshmi, 2021). These methods can be extended to solve NonLinear Parameter Estimation problems (NLPE) (Chavent, 2010) (Hartley, 1965), but they tend to be both less efficient and non-guaranteed, or guaranteed only in a specific context.

To overcome this problem, estimators based on Interval Analysis (Jaulin, 1993) have been developed to solve NLPE problems. They provide a set that is guaranteed to contain the true value of the parameters, rather than trying to provide their exact value. However, their use requires knowledge of Interval Analysis tools, which is not always relevant depending on the application. It is therefore important to be able to validate non-interval based estimators also in NLPE problems.

To do so, Interval Analysis tools can be applied to nonlinear and non-interval based estimators to guarantee some of their properties. For example, neural networks are often used as estimators but they fail to give a guaranteed result. Some works have already presented applications of the Interval Analysis tools to neural networks to guarantee some of their properties, for example to evaluate their generalization (Adam, 2019) and domain of validity (Adam, 2015).

This paper contributes to this dynamic by presenting a method to provide the maximum error made by any nonlinear estimator, including neural networks. This method is able to give a guaranteed result even when the estimator itself is not guaranteed. It relies on the Moore-Skelboe algorithm (van Emden, 2004) to provide a guaranteed result. Section 2 describes the notations that will later be used to formalise the problem and the method in Section 3. Finally, Section 4 presents an application with the validation of a gradient descent based estimator and a neural network estimator.

2. Notations and definitions

2.1. ESTIMATOR

An estimator is a function that approximates a quantity, or parameter, based on observed data. An estimate of a quantity $\mathbf{q}$ is commonly denoted $\hat{\mathbf{q}}$.

Let $\mathbb{X}_0 \subseteq \mathbb{R}^n$ be the set of possible parameters and $\mathbb{E} \subseteq \mathbb{R}^m$ the noise set. If $\mathbf{g} : \mathbb{X}_0 \to \mathbb{R}^m$ is the observation function and $\psi : \mathbb{R}^m \to \mathbb{X}_0$ is the estimator to validate, the estimation problem can be defined for all $\mathbf{x} \in \mathbb{X}_0$ and $\mathbf{e} \in \mathbb{E}$ by :

$$\mathbf{y} = \mathbf{g(x)} + \mathbf{e} \tag{1}$$
$$\hat{\mathbf{x}} = \psi\left(\mathbf{y}\right) \tag{2}$$

Where (1) is the Noisy observation and (2) is the estimation.

We also define the error function $\epsilon : \mathbb{X}_0 \times \mathbb{E} \to \mathbb{R}$ which gives the error of the estimator for a given parameter vector $\mathbf{x}$ and a given noise $\mathbf{e}$.

$$\epsilon(\mathbf{x}, \mathbf{e}) = \|\mathbf{x} - \hat{\mathbf{x}}\| = \|\mathbf{x} - \psi\left(\mathbf{g}\left(\mathbf{x}\right) + \mathbf{e}\right)\| \tag{3}$$

Figure 1 represents these notions in two different spaces: the parameter space and the observation space. Starting from the real value $\mathbf{x}$, an observation $\mathbf{y}$ is made. It can be decomposed into a perfect observation $\mathbf{g(x)}$ and a noise $\mathbf{e}$. The estimator finally relies on this observation to provide the estimated value $\hat{\mathbf{x}}$. The error ϵ represents the distance between $\mathbf{x}$ and its estimate $\hat{\mathbf{x}}$.
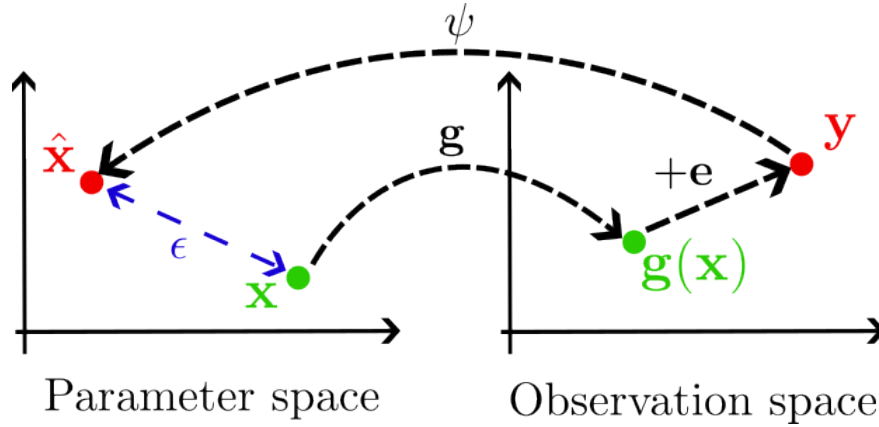
Figure 1. Estimation problem

To validate an estimator we want to evaluate the maximum of its error function on $\mathbb{X}_0 \times \mathbb{E}$, noted $\bar{\epsilon}$. The problem can written as below:

$$\bar{\epsilon} = \max \epsilon(\mathbf{x}, \mathbf{e}) \tag{4}$$

$$\mathbf{x} \in \mathbb{X}_0 \tag{5}$$

$$\mathbf{e} \in \mathbb{E} \tag{6}$$

2.2. INTERVALS AND BOXES

Intervals are used to represent a range of value instead of a single one. An interval can be defined by two of the following:

— Its lower bound, or left bound, noted lb

— Its upper bound, or right bound, noted ub or rb

— Its width, noted w

For a given interval I, the parameters described above are linked by the relation :

$$rb\,(I) = lb\,(I) + w\,(I) \tag{7}$$

The cartesian product of two or more intervals is called a box, and the set of n-dimensionnal boxes is denoted $\mathbb{IR}^n$ for $n \geqslant 1$. For any function $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$, its definition can be extended to $\mathbb{IR}^n$ and we also call $\mathbf{f} : \mathbb{IR}^n \to \mathbb{IR}^m$ its natural inclusion function.

These boxes can for example be used to form a cover of a subset of $\mathbb{R}^n$. A cover of a set $\mathbb{X}$ is a family of subsets of $\mathbb{X}$ whose union is all of $\mathbb{X}$. If $C = \{U_\alpha : \alpha \in A\}$ is an indexed family of subsets $U_\alpha \subset \mathbb{X}$, then C is a cover of $\mathbb{X}$ if $\bigcup_{\alpha \in A} U_\alpha = \mathbb{X}$ (Gamelin, 1999).

2.3. OPTIMIZATION

An Optimization problem, also called mimization problem, aims to find the minimum of a given function. It is defined by :

– An objective function to minimize $f : \mathbb{R}^n \to \mathbb{R}$

– A domain $\mathbb{X}_0 \subseteq \mathbb{R}^n$

– In the case of a constraint optimization problem, a set of conditions $g_i \{x_1, \ldots, x_n\} \leq 0$ for $i \in \{1, \ldots, m\}$ where g_i are functions of type $\mathbb{R}^n \to \mathbb{R}$.

Algorithms to solve this kind of problem are called Optimization algorithms. Among them, Global Optimization algorithms focus on finding the global minimum of the objective function, noted μ below.

2.4. MOORE-SKELBOE ALGORITHM

A classical Global Optimization algorithm is the Moore-Skelboe algorithm. This algorithm takes as inputs the objective function f, its domain $\mathbb{X}_0$ and a stopping criterion δ. It returns an interval containing the global minimum of the objective function, which bounds are an optimistic and a pessimistic approximation of the global minimum. Multiple variations of the algorithm exist (van Emden, 2004), Algorithm 1 displays its basic implementation.

Algorithm 1 Moore-Skelboe algorithm (MS_0)

Let $\{B_0\}$ be a cover of $\mathbb{X}_0$
while $w(f(B_0)) > \delta$ **do**   ▷ stopping criterion to choose
    Remove B_0 from the cover
    Split B_0
    Insert the result into the cover in increasing order
    of $lb(f(B_i))$, for $i = \{0, \ldots, N-1\}$
end while
return $f(B_0)$ ▷ $\mu \in f(B_0)$

When implementing this algorithm it is also possible to return the last B_0 to obtain a box containing an antecedent of a value between μ and $\mu + \delta$.

The logic behind this algorithm is that at each iteration the box B_0 is chosen so that the interval $f(B_0)$ is the $f(B_i)_{i \geq 0}$ interval with the lower left bound. As $\{B_0, B_1, ..., B_{N-1}\}$ covers the domain $\mathbb{X}_0$:

$$\forall x \in \mathbb{X}_0, lb(f(B_0)) \leq f(x) \tag{8}$$

From this relation we deduce:

$$lb(f(B_0)) \leq \mu \tag{9}$$

As B_0 is not an empty box, $f(B_0)$ always contains at least one value. By definition of the global minimum of a function:

$$\mu \leq rb(f(B_0)) \tag{10}$$

Finally from 9 and 10:

$$\mu \in f(B_0) \tag{11}$$
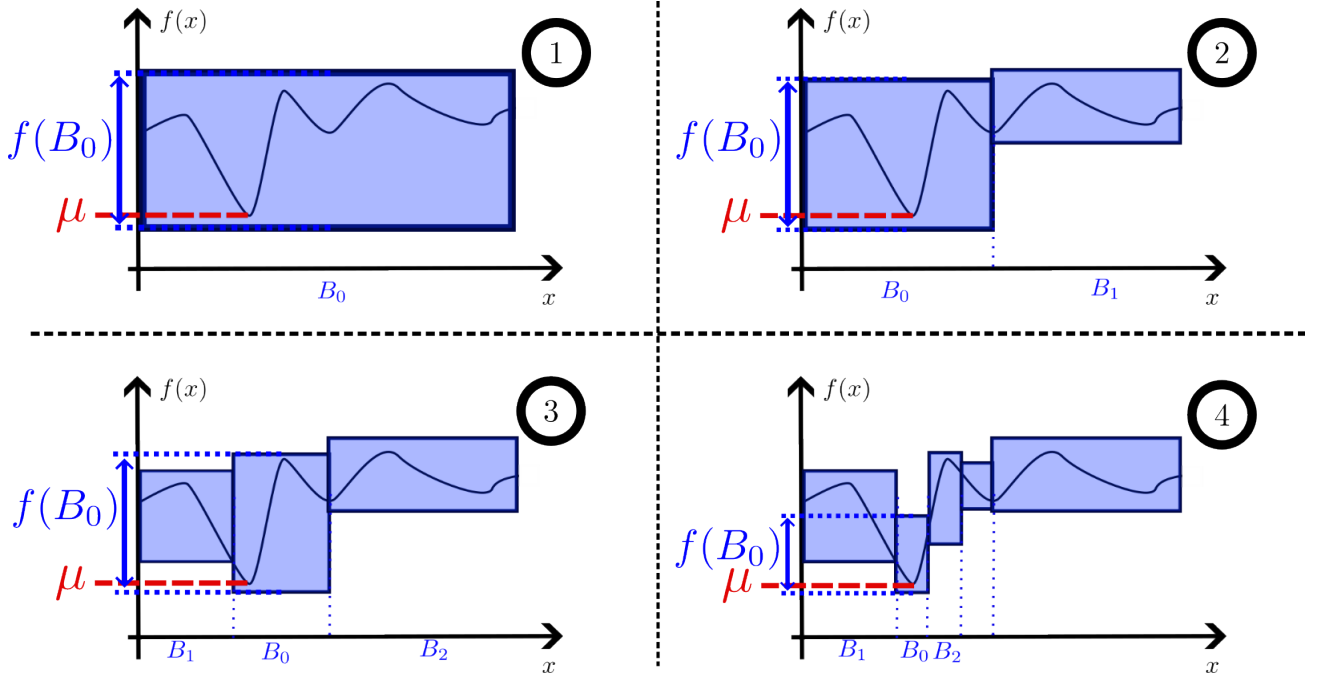
This result is illustrated Figure 2.

Figure 2. Moore-Skelboe algorithm example

3. Problem Formulation and method

To validate an estimator, we want to estimate the maximum error that it commits on a given set of possible parameters. This maximisation problem on the function ϵ can also be interpreted as a Optimization problem on the function $-\epsilon$.

The Moore-Skelboe algorithm can then be applied with the objective function $f = -\epsilon$ and with $\{B_0 = \mathbb{X}_0 \times \mathbb{E}\}$ as the initial cover.

Upon splitting B_0, only its component belonging to $\mathbb{X}_0$ will be splitted. By doing so the algorithm can also give the information of which parameter value is the most sensitive to noise when performing the estimation.

Given the context, we want to ensure that our estimation of the maximum error is pessimistic if not exact, so the focus will be on $-lb(f(B_0))$.

4. Application

4.1. APPLICATION CASE

To illustrate an application of this method the case described Figure 3 will be considered. In this case the parameter to estimate is the 2-Dimensional position noted $\mathbf{x}$, and the observation is the distances between x and three landmarks noted $\mathbf{a}$, $\mathbf{b}$ and $\mathbf{c}$. The observation function is then $\mathbf{g} : \mathbb{R}^2 \to \mathbb{R}^3$, defined by :

$$\mathbf{g}\left(\mathbf{x}\right) = \begin{pmatrix} d_a \\ d_b \\ d_c \end{pmatrix} = \begin{pmatrix} \|\mathbf{a} - \mathbf{x}\| \\ \|\mathbf{b} - \mathbf{x}\| \\ \|\mathbf{c} - \mathbf{x}\| \end{pmatrix} \tag{12}$$
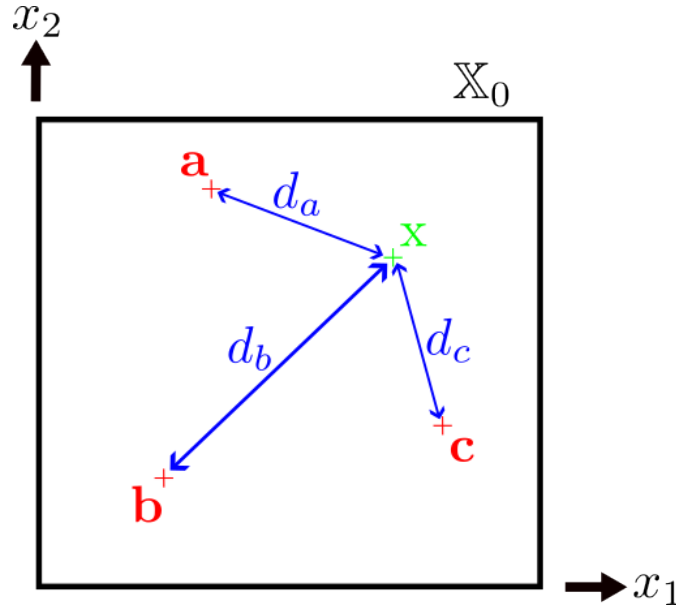

Figure 3. Application example

The norm used below to calculate the distances is the Euclidian norm. For numerical applications, we will take :

- $\mathbb{X}_0 = [5, 25]^2$

- $\mathbb{E} = [-0.2, 0.2]^3$

- $a = (10, -9)\,; b = (5, 12)\,; c = (-15, 0)$

4.2. NEURAL NETWORK ESTIMATOR

Neural networks are a common example of non-guaranteed estimator. To prove the efficiency of our method, we created a neural network estimator represented Figure 4. It is composed of :

An Input layer of 3 neurones for the three noisy distances.

Two Denses Hidden layers of 32 neurones with relu activation.

A Dense Ouput layer of 2 neurones with relu activation for the two components of $\hat{\mathbf{x}}$

The relu function can be defined on $\mathbb{R}$ by $relu(x) = max(0, x), x \in \mathbb{R}$. The relu activation means that in each layer the relu function will be applied on the output, making the neural network estimator nonlinear.
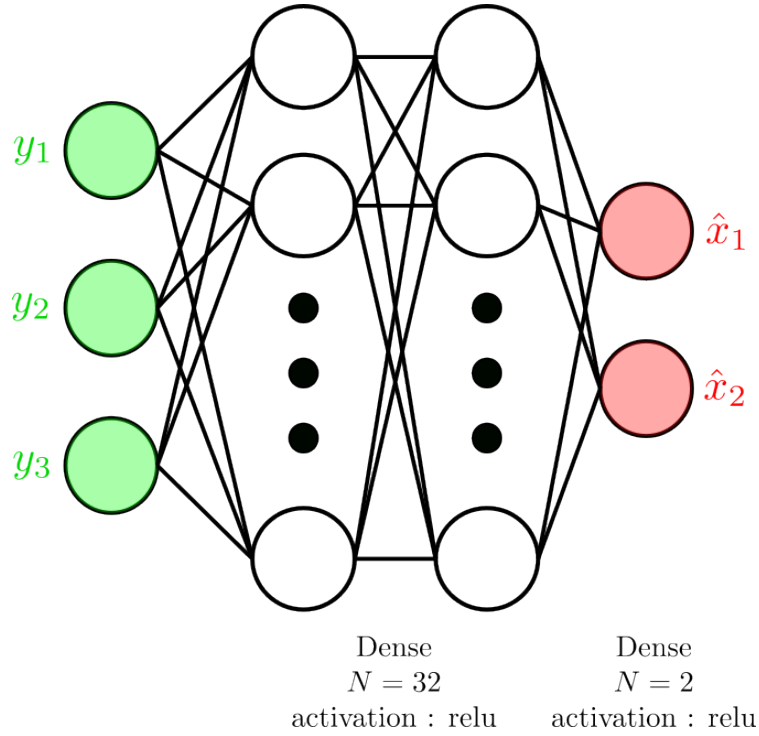

Figure 4. Neural network estimator

This neural network has then been trained on noisy data. The estimation is finally obtained by making a prediction with the resulting trained neural network.

4.3. NUMERICAL APPLICATION

The maximum error given by the Moore-Skelboe algorithm in this case is $\bar{\epsilon} = 1.7$. A small simulation, shown in Figure 5, allows us to visualise this $\bar{\epsilon}$.

For this simulation, noisy measurements are generated and then passed to the estimator. The result, in red, can then be compared to the expected value, in green. As $\bar{\epsilon}$ represents the maximum distance between $\mathbf{x}$ and $\hat{\mathbf{x}}$, the green dot representing $\mathbf{x}$ should always be in the purple circle of radius $\bar{\epsilon}$ centered at $\hat{\mathbf{x}}$.
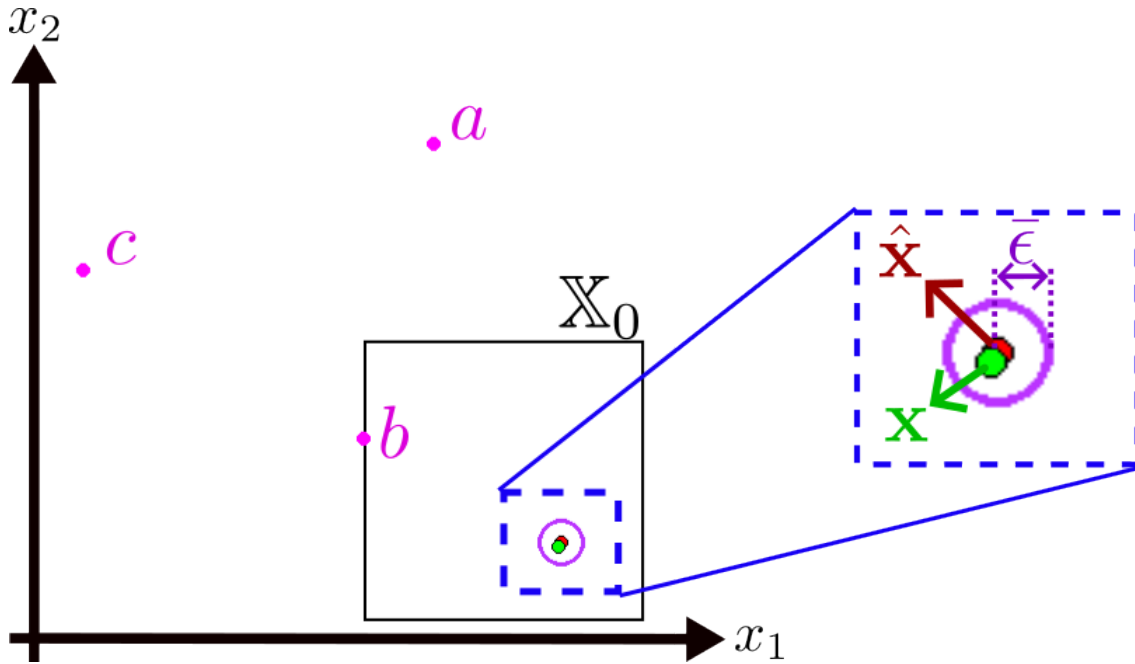
Figure 5. Simulation to visualise $\bar{\epsilon}$

Depending on the accuracy required by the application, we can then either use the neural network as it is or train it again, possibly by modifying its structure.

Using this method, we were able to find a guaranteed property for a neural network estimator that is both non-guaranteed and nonlinear. As the neural network presented here was simple the basic implementation of the Moore-Skelboe algorithm was sufficient. For more complex estimator variants of the algorithm may prove more efficient as the one presented here gives an increasingly pessimistic result as the estimator becomes complex.

5. Conclusion

This paper proposes an interval-based and guaranteed method to validate a nonlinear estimator thanks to the Moore-Skelboe algorithm. It allows us to find a guaranteed property in the form of a maximum error even when working with non-guaranteed estimators such as neural networks. Variants of the Moore-Skelboe algorithm presented in (van Emden, 2004) could be implemented to make this method more efficient in terms of time and computational complexity, but the application presented here shows that the basic implementation of the Moore-Skelboe algorithm is sufficient in simple cases.

Acknowledgements

References

Lakshmi, K., B. Mahaboob, M. Rajaiah, and C. Narayana. Ordinary least squares estimation of parameters of linear model. *Journal of Mathematical and Computational Science*, 2021.

Chavent, G. Nonlinear Least Squares for Inverse Problems: Theoretical Foundations and Step-by-Step Guide for Applications. *Springer*, 2010.

Hartley, H. O. and A. Booker. Nonlinear least squares estimation. *The Annals of Mathematical Statistics*, 1965.

Jaulin, L. and E. Walter. Set inversion via interval analysis for nonlinear bounded-error estimation. *Automatica*, 29(4):1053–1064, 1993.

Adam, S. P., D. A. Karras, G. D. Magoulas, and M. N. Vrahatis. Reliable estimation of a neural network's domain of validity through interval analysis based inversion. In 2015 *International Joint Conference on Neural Networks (IJCNN)*, 1–8, 2015.

Adam, S.P., A. C. Likas and M. N Vrahatis. Evaluating generalization through interval-based neural network inversion. *Neural Comput and Applic*, 31:9241–9260, 2019.

Gamelin, T. W., and R. E. Greene. Evaluating generalization through interval-based neural network inversion. In Dover Publications, *Introduction to Topology, Second Edition*, 1999.

van Emden, M. and B. Moa. Termination criteria in the Moore-Skelboe Algorithm for Global Optimization by Interval Arithmetic. In Floudas, C.A., Pardalos, P. (eds) *Frontiers in Global Optimization. Nonconvex Optimization and Its Applications*, vol 74. Springer, Boston, MA, 2004.