# Validated Constraint Solving – Practicalities, Pitfalls, and New Developments

R. Baker Kearfott

University of Louisiana, Lafayette, Louisiana 70504-1010 USA,
rbk@louisiana.edu
URL: http://interval.louisiana.edu/kearfott.html

**Abstract.** Many constraint propagation techniques iterate through the constraints in a straightforward manner, but can fail because they do not take account of the coupling between the constraints. However, some methods of taking account of this coupling are local in nature, and fail if the initial search region is too large. We put into perspective newer methods, based on linear relaxations, that can often replace brute-force search by solution of a large, sparse linear program.
Robustness has been recognized as important in geometric computations and elsewhere for at least a decade, and more and more developers are including validation in the design of their systems. We provide citations to our work to-date in developing validated versions of linear relaxations. This work is in the form of a brief review and prospectus for future development. We give various simple examples to illustrate our points.

**Keywords:** constraint propagation, global optimization, linear relaxations, GlobSol

## 1 Introduction

A very general problem, specific instances of which are important throughout CAD, as well as computing in general and operations research, is the general global optimization problem

$$\begin{array}{l} \text{minimize } \varphi(x) \\ \text{subject to } c_i(x) = 0,\ i = 1, \ldots, m_1, \\ \qquad\qquad g_i(x) \le 0,\ i = 1, \ldots, m_2, \\ \text{where } \varphi : \boldsymbol{x} \to \mathbb{R} \text{ and } c_i, g_i : \boldsymbol{x} \to \mathbb{R}, \text{ and where } \boldsymbol{x} \subset \mathbb{R}^n \text{ is} \\ \text{the hyperrectangle (box) defined by} \\ \qquad\qquad \underline{x}_{i_j} \le x_{i_j} \le \overline{x}_{i_j},\ 1 \le j \le m_3,\ i_j \text{ between 1 and } n, \\ \text{where the } \underline{x}_{i_j} \text{ and } \overline{x}_{i_j} \text{ are constant bounds.} \end{array} \tag{1}$$

If $\varphi$ is constant or absent, problem (1) becomes a general constraint problem; if, in addition $m_2 = m_3 = 0$, problem (1) becomes a nonlinear system of equations.

In general, and indeed, for many specific instances, it is theoretically impossible to avoid exponential time when solving problem 1; see [16]. However, a plethora of techniques that work well for a number of instances of interest have been developed (although none of these are universally practical). Particularly prominent is a set of techniques where the variables and, possibly, some of the parameters defining the objective $\varphi$ and the constraints $c_i$ and $g_i$ are only approximately known initially; in such cases, we can use the relationships among the constraints and the objective to compute tighter bounds on the variables $x_i$. Calling this general idea *subdefinite computations*, Narin'yani and his group developed it starting in the early 1980's [17]; an example implementation of these ideas is UNICALC [1] of Semënov et al. This paradigm also lends itself naturally, with directed roundings, to validated computation, that is, to computation in which the results are valid not just approximately, but with the certainty of a mathematical proof.

Large numbers of researchers have developed various tools and corresponding theories for filtering the bounds on the $x_i$ through the constraints to obtain narrower bounds, a process often described as *constraint propagation*, a burgeoning field of artificial intelligence. A succinct description of the general constraint propagation framework appears in [2]. Benhamou (ibid.) attributes the introduction of interval constraints (equivalent to subdefinite computations) within this constraint propagation framework to Cleary [4]. In the notation of [2], such interval constraint propagation depends upon *narrowing operators* $N_{\mathcal{C}}(\boldsymbol{x})$, where $\boldsymbol{x}$ is a box and $\mathcal{C}$ is the set of constraints[1]. Application of $N_{\mathcal{C}}$ returns a narrower box $\tilde{\boldsymbol{x}}$, where points in $\boldsymbol{x}$ that are infeasible with respect to the constraints $\mathcal{C}$ have been eliminated[2] The constraint propagation community has defined various *consistency conditions*, based on how $N$ eliminates portions of $\boldsymbol{x}$, and individual advances in the field often are in the form of new consistency conditions that can eliminate portions of $\boldsymbol{x}$ that other conditions cannot.

In this paper, we focus on three consistency conditions in common use, to which we refer as follows:

1. basic constraint propagation
2. interval Newton narrowing
3. linear relaxations

Although there are other conditions (such as those found in [22] or many other places), these three represent what is included in our own work, and also are representative of three schools which have only recently begun to merge. Basic constraint propagation is a fundamental narrowing process in the constraint propagation community, while interval Newton narrowing has been the predominant process used by those in the interval analysis community doing global optimiza-

[1] Here, $\mathcal{C}$ represents the constraints $c$, $g$, bound constraints of problem (1), and possibly constraints based on setting the gradient of the Lagrange function to zero, in the case that $\varphi$ is present and non-constant.

[2] In the global optimization context where $\varphi$ is present, in addition to eliminating portions of $\boldsymbol{x}$ based on feasibility, we may extend the interpretation of $N$ to include elimination of portions of $\boldsymbol{x}$ if a lower bound for $\varphi$ over the feasible portion of $\boldsymbol{x}$ is less than a known upper bound on the global optimum.

tion[3]. Linear relaxations, developed within the global optimization community at large, although very successful at solving practical problems such as those in [21] and [5], have only recently been recognized within the interval analysis community.

In the remainder of this paper, we will give illustrative examples of these three techniques, highlighting their deficiencies and citing literature where appropriate.

## 2 Basic Constraint Propagation

Our own first examination of basic constraint propagation appears in [8], where we call it "substitution-iteration." Following our ideas in [8], basic constraint propagation can be thought of as a nonlinear version of Gauss–Seidel iteration. That is, we are given initial bounds for the variables, and solve for a variable within a constraint. Plugging in the bounds for the other variables may then give narrower bounds on that variable. If so, then we solve for the other variables in constraints in which the narrowed variable occurs, and repeat the process with these other variables. We continue this until no more variables can be significantly narrowed.

*Example 1.* Take the constraint system

$$c_1(x) = x_1^2 - 2x_2, \quad c_2(x) = x_2^2 - 2x_1, \quad x_1 \in [-1, 1], \quad x_2 \in [-1, 1].$$

Selecting $x_2$ in $c_1$, we obtain $x_2 = x_1^2/2$. Plugging $x_1 = [-1, 1]$ into $x_1^2/2$, we obtain $x_2 \in [0, 0.5]$, a significant narrowing. We now solve $c_2$ for $x_1$ to obtain $x_1 = x_2^2/2$. Plugging the narrower value of $x_2$ into $x_2^2/2$, we obtain $x_1 \in [0, 0.125]$, obtaining a narrower value of $x_1$. We can now use the first equation again to obtain an even narrower value for $x_2$. After only for iterations of this process (using INTLAB [20]), we obtain intervals for $x_1$ and $x_2$ containing $x_1 = 0$, $x_2 = 0$ and with widths less than $10^{-16}$.

Just as in the classical linear Gauss–Seidel method, this process will only converge if an ordering of the constraints and variables can be found such that the corresponding Jacobi matrix is diagonally dominant.

*Example 2.* Take the constraint system

$$c_1(x) = x_1^3 + x_1 - x_2, \quad c_2(x) = -2x_1 - x_2, \quad x_1 \in [-.5, .5], \quad x_2 \in [-.25, .25].$$

In Example 2, there is a unique feasible point (i.e. a point at which the constraints are consistent) within the initial bounds at $x_1 = 0$, $x_2 = 0$. However, solving $c_1$ for $x_2$ as in Example 1 gives $x_2 = (x_1^3 + x_1)$. There is interval overestimation when we plug $x_1 = [-.5, .5]$ into $x_1^2 + x_1$ and use naive interval arithmetic, but that is

[3] Bounds on the objective $\varphi$ are also used extensively within this school, but mainly in a rudimentary way to reject boxes, and not to narrow boxes in the sense of a narrowing operator.

not the only issue here, as we'll see as we follow through the computations; we'll use the exact range of $(x_1^3+x_1)$ for $x_1 \in [-.5, .5]$, that is, $x_2 \in [-.625, .625]$, which is no improvement. Similarly, if we solve for $x_2$ in the second equation, we obtain that the range of $-2x_1$ over $x_1 \in [-.5, .5]$ is $x_2 \in [-1, 1]$, not an improvement. The only remaining alternatives for improvement are to solve for $x_1$ in $c_1$ or $c_2$. Solving for $x_1$ in $c_2$ gives no improvement, but solving for $x_1$ in $c_1$ (using the cubic equation) and plugging in $x_2 \in [-.25, .25]$ gives $x_1 \in [-.237, .237]$, an improvement. Now, solving for $x_2$ in $c_2$ gives $x_2 = -2x_1 \in [-.474, .474]$, not a narrowing. Thus, the process becomes stationary at $x_1 \in [-.237, .237]$, $x_2 \in [-.25, .25]$. At this point, a constraint propagation procedure employing only this basic technique would bisect $\boldsymbol{x}$, to increase the number of boxes, a potentially expensive procedure.

The difficulty in example 2 is that there is no ordering of the variables and constraints for which the resulting Jacobi matrix at the solution $x = (0, 0)$ is diagonally dominant. In such cases, a process that somehow decouples the system would be superior to basic constraint propagation. One such procedure is interval Newton methods.

## 3 Interval Newton Narrowing

Interval Newton methods are ubiquitous throughout the interval analysis literature. An good reference for the theory is [18]; in our own work, we have introduced and summarized interval Newton methods in [10]. At their most basic, interval Newton methods work on systems of $n$ equations in $n$ unkowns[4]. Systems with both equality and inequality constraints and optimization problems can be handled by forming the Lagrange multiplier system (or more generally, the Fritz–John equations).

An interval Newton operator is an operator of the form $\boldsymbol{v} = \boldsymbol{N}(F, \boldsymbol{x}, \check{x})$, where $\boldsymbol{v}$ is an interval vector that bounds the solution set to

$$\boldsymbol{A}v = -F(\check{x}), \tag{2}$$

where $\boldsymbol{A}$ is some Lipschitz matrix for $F$ or a slope matrix[5] for $F$ centered at $\check{x}$. (For example $\boldsymbol{A}$ can consist of element-wise interval extensions of the Jacobi matrix of $F$ over $\boldsymbol{x}$.) Generally, the bounds $\boldsymbol{v}$ are obtained by first preconditioning (2) by multiplying $\boldsymbol{A}$ and $-F$ by a preconditioner matrix $Y$, where $Y$ serves to partially decouple the equations (and make the resulting matrix $Y\boldsymbol{A}$ more like the identity matrix).

In Example 2 above,

$$F(x) = \begin{pmatrix} c_1(x) \\ c_2(x) \end{pmatrix} = \begin{pmatrix} x_1^3 + x_1 - x_2 \\ -2x_1 - x_2 \end{pmatrix},$$

[4] However, using appropriate preconditioning, interval with practical use can be devised for rectangular systems (both overdetermined and underdetermined). See, for example [9, Ch. 3] for an example of such preconditioners.

[5] See, for example, [18] for the definitions of Lipschitz matrix and slope matrix.

and an element-wise interval extension of the Jacobi matrix of $F$ over the initial $\boldsymbol{x}$ is

$$\boldsymbol{F}'(\boldsymbol{x}) \in \begin{pmatrix} [1, 1.75] & -1 \\ -2 & -1 \end{pmatrix}.$$

If the inverse of the midpoint matrix for $\boldsymbol{F}'(\boldsymbol{x})$ is used as a preconditioner matrix, then, if $\check{x} = (0, 0)^T$, the preconditioned system becomes

$$\begin{pmatrix} [0.8888, 1.1112] & [0.0000, 0.0000] \\ [-0.2223, 0.2223] & [0.9999, 1.0001] \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

and, using the interval Gauss–Seidel method, new bounds for $v$ are

$$v \in \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

that is, we obtain the solution sharply.

However, for some systems, multiplying by $Y$ can result in significant overestimation due to interval dependency. We give a simple example of such a system in [15]. There, we show that such problems can sometimes be handled by preconditioning the system *symbolically*[6].

Nonetheless, there are significant practical problems that cannot be handled even with interval Newton narrowing with symbolic preconditioning in combination with basic constraint propagation. We have found this to be particularly true in practice for systems of equations arising from applying Lagrange multipliers (or the Fritz–John conditions) to constrained optimization problems; we gave a practical example of this in [14, 13], and have studied it more thoroughly in [12]. Here, we present a simplified example of a pure constraint problem.

*Example 3.* Take the constraint system

$$c_1(x) = x_1^2 - x_2, \quad g_1(x) = x_2 - x_1, \quad x_1 \in [0, 1], \quad x_2 \in [0, 1].$$

One can easily verify that basic constraint propagation is unsuccessful on this problem. Although this particular system has a unique solution at $x_1 = 0$, $x_2 = 0$, since this is an inequality-constrained problem, it is possible to have a solution set with non-empty measure. To obtain lower and upper bounds on this solution set, we may solve the corresponding constrained optimization problems with objective functions $\min x_1$, $\max x_1$, $\min x_2$ and $\max x_2$ and with the additional constraints $-x_1 \le 0$, $x_1 \le .5$, $-x_2 \le 0$, $x_2 \le .5$. Considering the problem with

[6] To do symbolic arithmetic on functions, we compute within a vector space of the coefficients for a basis for the space of functions in question. In [15], we used Taylor polynomials with remainder terms, and we used the COSY system [3] for the actual implementation.

$\min x_1$, the Fritz–John equations (as in [9, (5.7), p. 197]) can be written as

$$\begin{pmatrix} u_0 - u_1 - u_2 + u_3 + 2x_1 v_1 \\ u_1 - u_4 + u_5 - v_1 \\ u_1(x_2 - x_1) \\ -u_2 x_1 \\ u_3 x_1 \\ -u_4 x_2 \\ u_5 x_2 \\ x_1^2 - x_2 \\ u_0 + u_1 + u_2 + u_3 + u_4 + u_5 + v_1^2 - 1 \end{pmatrix} = 0 \qquad (3)$$

The Jacobi matrix for this system is

$$\frac{\partial F}{\partial x\;\partial u\;\partial v} = \begin{pmatrix} 2v_1 & 0 & 1 & -1 & -1 & 1 & 0 & 0 & 2x_1 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & -1 \\ -u_1 & u_1 & 0 & x_2 - x_1 & 0 & 0 & 0 & 0 & 0 \\ -u_2 & 0 & 0 & 0 & -x_1 & 0 & 0 & 0 & 0 \\ u_3 & 0 & 0 & 0 & 0 & x_1 & 0 & 0 & 0 \\ 0 & -u_4 & 0 & 0 & 0 & 0 & -x_2 & 0 & 0 \\ 0 & u_5 & 0 & 0 & 0 & 0 & 0 & x_2 & 0 \\ 2x_1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 2v_1 \end{pmatrix}.$$

Interestingly, since linear combinations of entries in any column are single-use expressions, there is no overestimation in either naive evaluation of the interval Jacobi matrix or in preconditioning the resulting interval matrix. If we do not know bounds on the Lagrange multipliers $u_i$, $0 \le i \le 5$ and $v_1$ beforehand, then we may choose them to be the natural values $u_i \in [0,1]$, $0 \le i \le 5$, $v_1 \in [-1,1]$. The resulting interval Jacobi matrix is

$$\begin{pmatrix} [-2,2] & 0 & 1 & -1 & -1 & 1 & 0 & 0 & [0,2] \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & 1 & -1 \\ [-1,0] & [\;0,1] & 0 & [-1,1] & & 0 & 0 & 0 & 0 \\ [-1,0] & 0 & 0 & 0 & [-1,0] & 0 & 0 & 0 & 0 \\ [\;0,1] & 0 & 0 & 0 & 0 & [0,1] & 0 & 0 & 0 \\ 0 & [-1,0] & 0 & 0 & 0 & 0 & [-1,0] & 0 & 0 \\ 0 & [\;0,1] & 0 & 0 & 0 & 0 & 0 & [0,1] & 0 \\ [0,2] & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & [-1,1] \end{pmatrix},$$

which contains many singular matrices. Thus, an interval Newton method using this system will not, in general succeed[7]. In fact, at the solution $x_1 = 0$, $x_2 = 0$,

[7] However, conceivably, some of the coordinates could possibly be narrowed with the linear programming preconditioners of [9, Chapter 3].

the Fritz–John system (3) becomes

$$\begin{pmatrix} u_0 - u_1 - u_2 + u_3 \\ u_1 - u_4 + u_5 - v_1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ u_0 + u_1 + u_2 + u_3 + u_4 + u_5 v_1^2 - 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix},$$

which is underdetermined; thus, the Jacobi matrix at the solution itself must be singular, and an interval Newton method cannot hope to narrow all coordinates, unless we fix one or more of the multipliers at their lower or upper bounds[8].

## 4 Linear Relaxations

Problems as illustrated above in Example 3 have led us to consider *linear relaxations*, a technique by which a global optimization problem is approximated by a linear program (LP). The solution to this linear program, obtained approximately by a state-of-the-art LP solver, then gives a lower bound to the solution to the original global optimization problem. This lower bound can then be used to advantage in the constraint propagation scheme. For example, optimization problems could be formulated as we described above below Example 3, and sharper lower and upper bounds could then be obtained.

Various researchers use linear relaxations to great advantage in solving practical problems. One example of this is in the commercial global optimization software BARON, whose underlying ideas are explained in [21]. However, such software has not included validation until recently. Two years ago, Neumaier and Shcherbina [19], as well as Jansson [7] exhibited a simple technique, based on the duality gap, by which a rigorous lower bound to the minimum of an LP can be obtained from approximate values of the dual variables. This enables linear relaxations to be used in a validated context.

We explain our implementation of linear relaxations in [13] and [6], while we conduct various numerical experiments within our GlobSol environment (see [11] for a summary of GlobSol) in [12].

For Example 3 above, we may follow loosely our procedure in [13] to produce linear relaxations. A corresponding linear program for computing a lower bound on $x_1$, using a single underestimator for the convex function $x_2 = x_1^2$, can be as

[8] Actually, this is a possibility that can be investigated, although there may be too many possible values of the $u$'s and $v$'s for this to be practical, if there are many constraints.

follows:

$$
\begin{aligned}
&\text{minimize } x_1\\
&\text{subject to } x_2 \le x_1 \text{ (the overestimator)},\\
&\qquad\qquad x_2 \ge .125 + .5(x_1 - .25),\\
&\qquad\qquad x_2 \le x_1 \text{ (the original constraint)},\\
&\qquad\qquad x_1 \in [0,1],\ x_2 \in [0,1].
\end{aligned}
\tag{4}
$$

(Here, we have relaxed the constraints directly, for simplicity. In [13] and [6], we illustrate how the expressions can be parsed and relaxed automatically.) The exact minimum to this linear program is $x_1 = 0$, and the solution point is $x_1 = 0$, $x_2 = 0$, the solution to the original constraint problem. If we replace the objective in (4) by $-x_1$, we obtain an upper bound of 1 for $x_1$, not an improvement If we replace the objective by $x_2$, we obtain 0 for the lower bound on $x_2$, but we similarly obtain 0.5 for the upper bound on $x_2$, an improvement. We thus obtain

$$x \in \begin{pmatrix} [0,1] \\ [0,.5] \end{pmatrix} \subset \begin{pmatrix} [0,1] \\ [0,1] \end{pmatrix},$$

a significant narrowing in the second coordinate. Basic constraint propagation now converges.

This illustrates that linear underestimators are capable of making headway on problems, even pure constraint problems, for which both basic constraint propagation and interval Newton methods alone do not do well. This is especially true if there is much implicit dependency in the constraints, and the constraints are approximately linear. However, this is not always the case. For example, certain problems can be treated with basic constraint propagation or with interval Newton methods, but linear relaxations are inefficient. See [12] for further information and results on various test problems.

## 5 Summary and Conclusions

Through simple examples, we have illustrated deficiencies in two techniques widely used in the past in validated constraint propagation, and we have advocated linear relaxations, a technique widely used in the non-validated global optimization community but not examined seriously until recently for validated constraint solving. Linear relaxations are not a panacea, but complement well other techniques used in the validated constraint solving community.

## References

1. A. B. Babichev, O. B. Kadyrova, T. P. Kashevarova, A. S. Leshchenko, and A. L. Semenov. UniCalc, a novel approach to solving systems of algebraic equations. *Interval Computations*, 1993(2):29–47, 1993.
2. F. Benhamou. Interval constraints, interval propagation. In C. Floudas and P. Pardalos, editors, *Encyclopedia of Optimization*, Dordrecht, Netherlands, 2001. Kluwer.

3. M. Berz. Cosy infinity web page, 2000. http://cosy.pa.msu.edu/.
4. J. G. Cleary. Logical arithmetic. *Future Computing Systems*, 2(2):125–149, 1987.
5. C. A. Floudas. *Deterministic Global Optimization: Theory, Algorithms and Applications*. Kluwer, Dordrecht, Netherlands, 2000.
6. S. Hongthong and R. B. Kearfott. Rigorous linear overestimators and underestimators, 2004. preprint, http://interval.louisiana.edu/preprints/estimates_of_powers.pdf.
7. Ch. Jansson. A rigorous lower bound for the optimal value of convex optimization problems. *J. Global Optim.*, 28(1):121–137, January 2004.
8. R. B. Kearfott. Decomposition of arithmetic expressions to improve the behavior of interval iteration for nonlinear systems. *Computing*, 47(2):169–191, 1991.
9. R. B. Kearfott. *Rigorous Global Search: Continuous Problems*. Kluwer, Dordrecht, Netherlands, 1996.
10. R. B. Kearfott. Interval analysis: Interval Newton methods. In *Encyclopedia of Optimization*, volume 3, pages 76–78. Kluwer, 2001.
11. R. B. Kearfott. Globsol: History, composition, and advice on use. In *Global Optimization and Constraint Satisfaction*, Lecture Notes in Computer Science, pages 17–31, New York, 2003. Springer-Verlag.
12. R. B. Kearfott. Empirical comparisons of linear relaxations and alternate techniques in validated deterministic global optimization, 2004. preprint, http://interval.louisiana.edu/preprints/validated_global_optimization_search_comparisons.pdf.
13. R. B. Kearfott and S. Hongthong. A preprocessing heuristic for determining the difficulty of and selecting a solution strategy for nonconvex optimization, 2003. preprint, http://interval.louisiana.edu/preprints/2003_symbolic_analysis_of_GO.pdf.
14. R. B. Kearfott, M. Neher, S. Oishi, and F. Rico. Libraries, tools, and interactive systems for verified computations: Four case studies. In R. Alt, A. Frommer, R. B. Kearfott, and W. Luther, editors, *Numerical Software with Result Verification, Lecture Notes in Computer Science no. 2991*, pages 36–63, New York, 2004. Springer-Verlag.
15. R. B. Kearfott and G. W. Walster. Symbolic preconditioning with Taylor models: Some examples. *Reliable Computing*, 8(6):453–468, 2002.
16. V. Kreinovich, A. Lakeyev, J. Rohn, and P. Kahl. *Computational Complexity and Feasibility of Data Processing and Interval Computations*. Kluwer, Dordrecht, Netherlands, 1998.
17. A. S. Narin'yani. Intelligent software technology for the new decade. *Comm. ACM*, 34(6):60–67, June 1991.
18. A. Neumaier. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, England, 1990.
19. A. Neumaier and O. Shcherbina. Safe bounds in linear and mixed-integer programming. *Math. Prog.*, 99(2):283–296, March 2004. http://www.mat.univie.ac.at/~neum/ms/mip.pdf.
20. S. M. Rump et al. INTLAB home page, 2000. http://www.ti3.tu-harburg.de/~rump/intlab/index.html.
21. M. Tawarmalani and N. V. Sahinidis. *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, and Applications*. Kluwer, Dordrecht, Netherlands, 2002.
22. P. Van Hentenryck, L. Michel, and Y. Deville. *Numerica: A Modeling Language for Global Optimization*. MIT Press, Cambridge, MA, 1997.