Validated Constraint Solving – Practicalities, Pitfalls, and New Developments

by
R. Baker Kearfott
Department of Mathematics
University of Louisiana at Lafayette
`rbk@louisiana.edu`

This talk will:

- Review three filtering schemes.
- Explain rigor in linear relaxations.
- Give a sequence of examples where
  1. basic constraint propagation fails but not interval Newton narrowing;
  2. interval Newton narrowing fails but linear relaxations do not.
- Describe recent implementation and numerical experiments with our GlobSol system.

General Problem

minimize $\varphi(x)$
subject to:

$$c_i(x) = 0,\ i = 1, \ldots, m_1,$$
$$g_i(x) \le 0,\ i = 1, \ldots, m_2,$$

where $\varphi : \boldsymbol{x} \rightarrow \mathbb{R}$ and $c_i, g_i : \boldsymbol{x} \rightarrow \mathbb{R}$, and where $\boldsymbol{x} \subset \mathbb{R}^n$ is the hyperrectangle (box) defined by

$$\underline{x}_{i_j} \le x_{i_j} \le \overline{x}_{i_j},\ 1 \le j \le m_3,$$

i_j between 1 and n, where the $\underline{x}_{i_j}$ and $\overline{x}_{i_j}$ are constant bounds.

If φ is constant or absent, this problem becomes a general constraint problem; if, in addition $m_2 = m_3 = 0$, this problem becomes a nonlinear system of equations.

Basic Constraint Propagation

Our view:

- We begin with bounds on the variables.
- We solve a constraint for a variable x_i, obtaining $x_i = g(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots x_n)$.
- We use the bounds on x_j, $j \neq i$ to obtain (hopefully) narrower bounds on x_i (say by evaluating g with interval arithmetic).
- We "propagate" the new bounds on x_i, that is, we use the new bounds on x_i in the constraints in which it occurs to obtain new bounds on the other variables.

Basic Constraint Propagation

An Example

Take the constraint system

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

1. Solve for x_2 in c_1, to obtain $x_2 = x_1^2/2$, then plug $x_1 = [-1, 1]$ into $x_1^2/2$, to obtain $x_2 \in [0, 0.5]$.
2. Solve c_2 for x_1 to obtain $x_1 = x_2^2/2$, then plug the narrower value of x_2 into $x_2^2/2$, to obtain $x_1 \in [0, 0.125]$.
3. Use c_1 again to obtain an even narrower value for x_2.
4. This process can be continued to convergence to $x_1 = 0$, $x_2 = 0$.

Basic Constraint Propagation

When it does and does not work

- Basic constraint propagation only works for linear systems when a permutation of the rows and columns leads to a diagonally dominant system.
- For nonlinear systems, the Jacobi matrix should be permutable to a diagonally dominant system, or so preconditioned.
- Possible research direction: try symbolic preconditioning. (See me for references on how.)

Basic Constraint Propagation

Example when it does not work

Take the constraint system

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

- There is unique solution $c_1 = 0$, $c_2 = 0$ at $x_1 = 0$, $x_2 = 0$.
- Solving c_1 for x_2 as in the previous example gives $x_2 = (x_1^3 + x_1)$.
- Solving $c_1 = 0$ for x_2 and using the exact range of $(x_1^3 + x_1)$ for $x_1 \in [-.5, .5]$ gives $x_2 \in [-.625, .625]$, no improvement.
- Solving for x_2 in the second equation gives the range of $-2x_1$ over $x_1 \in [-.5, .5]$ is $x_2 \in [-1, 1]$, also not an improvement.
- The only remaining alternatives 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 and plugging in $x_2 \in [-.25, .25]$ gives $x_1 \in [-.237, .237]$, an improvement.
- Additional applications of the process give no additional narrowing.

Interval Newton Narrowing

In the above example,

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

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

$$\begin{pmatrix} [0.8888, 1.1112] & [0, 0] \\ [-0.2223, 0.2223] & [1, 1] \end{pmatrix} \begin{pmatrix} v_1 \\ v_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \end{pmatrix},$$

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.

When Interval Newton Narrowing Fails

Take the constraint system

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

- One easily checks that basic constraint propagation fails for this problem.
- 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 $\max x_2$, subject to $-x_1 \le -.5$, $x_1 \le .5$, $-x_2 \le -.5$, $x_2 \le .5$.

When Interval Newton Narrowing Fails

(continued)

- The Fritz–John equations for the problem with min can be written as

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

- Using $u_i \in [0, 1]$, $v_1 \in [-1, 1]$ for the Lagrange multipliers, the interval Jacobi matrix for this system contains many singular matrices, and the corresponding interval Newton method thus cannot succeed.

Linear Relaxations

The basic idea

- If the objective φ is replaced by linear function $\varphi^{(\ell)}$ such that $\varphi^{(\ell)}(x) \le \varphi(x)$ for $x \in \boldsymbol{x}$, then the resulting problem has global optimum less than or equal to the global optimum of the original problem.
- If each inequality constraint g_i replaced by a linear function $g_i^{(\ell)}$ such that $g_i^{(\ell)}(x) \le g_i(x)$ for $x \in \boldsymbol{x}$, then the resulting problem, has optimum that is less than or equal to the optimum of the original problem.
- If there are equality constraints, then each equality constraint can be replaced by two linear inequality constraints, and these inequality constraints can be replaced as above by linear inequality constraints.
- The resulting linear program is termed a *linear relaxation.*

Linear Relaxations

Our Previous Example

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

- Lower bounds of a convex function are tangent lines and upper bounds are secant lines.
- A corresponding linear program for computing an upper bound on x_2, using two underestimators for the convex function $x_2 = x_1^2$, is:

  minimize $-x_2$
  subject to
  $x_2 \le x_1$ (the overestimator),
  $x_2 \ge .125 + .5(x_1 - .25)$,
  $x_2 \le x_1$ (the original constraint),
  $x_1 \in [0,1]$, $x_2 \in [0,1]$.

Linear Relaxation Example

(continued)

- The exact minimum to this linear program is $\varphi = -.5$, corresponding to $x_2 \le 0.5$.
- Thus, we have narrowed x_2 to $x_2 \in [0, 0.5] \subset [0, 1]$.
- Basic constraint propagation now converges.

Rigor in Linear Relaxations

1. Typical procedures have been to compute the coefficients of the linear relaxation with floating point arithmetic, then to solve the relaxation with a state-of-the-art LP solver.
2. With carefully considered directed rounding and interval arithmetic, we can form a machine-representable LP that is an actual relaxation of the original problem.
3. Neumaier and Shcherbina, as well as Jansson, have presented a simple technique to utilize the duality gap to obtain a rigorous lower bound on the solution to an LP, given approximate values of the dual variables.
4. Combining (2) and (3) gives a procedure for rigorous computations of lower bounds on the solution to the original problem.

Implementation in GlobSol

- We have implemented linear relaxations in GlobSol.
- Initial experiments indicate the technique makes possible solution of problems that were previously intractable within GlobSol.
- A preprint of experimental results is available.
- GlobSol still is not fully competitive with other packages using relaxations in a non-validated way (e.g. BARON).
- One possibility for improvement: Use a better LP solver. (GlobSol presently is using a free one from the SLATEC library.)