The GlobSol Project: Rigorous Global Solutions (Overview and Recent Developments)

by

George F. Corliss, georgec@mscs.mu.edu
Department of Mathematics, Statistics, and Computer Science
Marquette University

and

R. Baker Kearfott, rbk@usl.edu
Department of Mathematics, University of Southwestern Louisiana

This talk will

- Highlight the nature of deterministic global optimization
- Review capabilities of the GlobSol software package
- Review an example of how to use GlobSol
- Outline improvements to GlobSol in 1998–1999
- Give an example of GlobSol's use on imprecisely known data
- Outline GlobSol's new installation procedure

Deterministic Global Optimization

- involves some kind of systematic global search over the domain.
- The various algorithms rely on estimates of the range of the objective function over subdomains.
- Some algorithms (due to Mladineo, Schubert, Wood, etc.) rely on Lipschitz constants to obtain estimates of ranges.
- Bounds on ranges or approximate bounds on ranges are also obtained with outwardly rounded interval arithmetic or non-rigorous interval arithmetic, respectively.

Deterministic Global Optimization

Interval Methods

- Evaluation of a an objective function $\phi(X)$ at an interval vector $\boldsymbol{X}$ gives bounds on the actual range of ϕ over $\boldsymbol{X}$.
  - If directed rounding is used, the bounds rigorously contain the mathematical range.
  - The bounds, in general, are overestimates.
- If the lower bound of $\phi(\boldsymbol{X})$ is greater than a previously computed objective value $\phi(X)$, then $\boldsymbol{X}$ can be discarded.
- Interval Newton Methods, combined with directed rounding, can *prove* existence and uniqueness of critical points, as well as reduce the size of regions $\boldsymbol{X}$.

On the State of the Art

- Minimizing a function over a compact set in $\mathbb{R}^n$ is an NP-complete problem.
- Thus, barring monumental discoveries, any *general* algorithm will fail for some high-dimensional problems.
- There are many practical problems that can be solved in low-dimensional spaces.
- Some low-dimensional problems are difficult.
- Advances in computer speed and algorithm construction have allowed many more practical problems to be solved, including high-dimensional ones.

What is GlobSol?

- A Fortran 90 package
  - well-tested.
  - self-contained.
- Solves constrained and unconstrained global optimization problems
- Separate program solves square algebraic systems of equations.
- Utility programs for interval and point evaluation, etc.
- Subroutine / module libraries for interval arithmetic, automatic differentiation, etc.
- Publicly available free of charge

  `http://interval.usl.edu/GLOBSOL/GlobSol.tar.Z`

GlobSol

Special Features

- Objective function and constraints are coded as Fortran 90 programs
- Can use constraint propagation (substitution/iteration) on the intermediate quantities in objective function, equality, and inequality constraint evaluation.
- Can use an overestimation-reducing "peeling" process for bound-constraints.
- Uses an effective point method to find approximate feasible points.
- Has a special augmented system mode for least squares problems.

GlobSol

Special Features, continued

- Uses epsilon-inflation and set-complementation, with carefully controlled tolerances,
  - to avoid singularity problems.
  - to facilitate verification.

GlobSol Features

(continued)

- Has extensive error-checking (user input, internal errors, etc.)
- Has on-line web page documentation.
- The algorithm is configurable.
- Has various levels of printing, for various algorithm aspects.
- Source code and libraries for components are available.
  - Automatic differentiation access.
  - Interval arithmetic access.
  - User-modifiable, with adequate study.
- Gives performance statistics, both in report form and for input to spreadsheets.

Use of GlobSol

An Example

The following Fortran 90 program defines the objective function

$$\text{minimize} \quad \phi(X) = -2 * x_1^2 - x_2^2$$

subject to constraints

$$\begin{aligned} x_1^2 + x_2^2 - 1 &\le 0 \\ x_1^2 - x_2 &\le 0 \\ x_1^2 - x_2^2 &= 0 \end{aligned}$$

Use of GlobSol

An Example, continued

```
PROGRAM SIMPLE_MIXED_CONSTRAINTS
  USE CODELIST_CREATION
      PARAMETER (NN = 2)
      TYPE(CDLVAR), DIMENSION(NN) :: X
      TYPE(CDLLHS), DIMENSION(1  ):: PHI
      TYPE(CDLINEQ), DIMENSION(2) :: G
      TYPE(CDLEQ), DIMENSION(1)   :: C

      OUTPUT_FILE_NAME = 'MIXED.CDL'
      CALL INITIALIZE_CODELIST(X)

  PHI(1) = -2*X(1)**2 - X(2)**2
  G(1) = X(1)**2 + X(2)**2 - 1
  G(2) = X(1)**2 - X(2)
  C(1) = X(1)**2 - X(2)**2

      CALL FINISH_CODELIST
END PROGRAM SIMPLE_MIXED_CONSTRAINTS
```

GlobSol Example

(continued)

1. Running the above program produces an internal representation, or code list.
2. The optimization code interprets the code list at run time to produce floating point and interval evaluations of the objective function, gradient, and Hessian matrix.
3. A separate data file defines the initial search box, the bound constraints, and the initial guess, if any.
4. Separate data files supply algorithm options, such as which interval Newton method to use and how to precondition the linear systems.

GlobSol Example

The Data File

```
1D-5          ! General domain tolerance
 0   1        ! Bounds on the first variable
 0   1        ! Bounds on the second variable
F F           ! X(1) has no bound constraints
F F           ! X(2) has no bound constraints
```

Subsequent optional lines can give an initial guess point.

GlobSol Example

Output File – abridged first part

```
Output from FIND_GLOBAL_MIN on  04/06/1999  at  08:03:52.
Version for the system is: March 20, 1999

Codelist file name is: MIXEDG.CDL
Box data file name is: MIXED.DT1

Initial box:
[    0.0000E+00,   0.1000E+01 ] [    0.0000E+00,   0.1000E+01 ]

BOUND_CONSTRAINT:
  F F   F F

---------------------------------------
CONFIGURATION VALUES:

EPS_DOMAIN:   0.1000E-04   MAXITR:   60000
DO_INTERVAL_NEWTON: T  QUADRATIC: T  FULL_SPACE: F
VERY_GOOD_INITIAL_GUESS:F
USE_SUBSIT:T
OUTPUT UNIT:7 PRINT_LENGTH:1
Default point optimizer was used.
```

GlobSol Example

Output File – abridged second part

```
 THERE WERE NO BOXES IN COMPLETED_LIST.

 LIST OF BOXES CONTAINING VERIFIED FEASIBLE POINTS:

 Box no.:1
Box coordinates:
 [    0.7071E+00,   0.7071E+00 ] [    0.7071E+00,   0.7071E+00 ]
PHI:
 [   -0.1500E+01,  -0.1500E+01 ]
 Level: 3
 Box contains the following approximate root:
    0.7071E+00   0.7071E+00
 OBJECTIVE ENCLOSURE AT APPROXIMATE ROOT:
 [   -0.1500E+01,  -0.1500E+01 ]
U0:
 [    0.3852E+00,   0.3852E+00 ]
U:
 [    0.5777E+00,   0.5777E+00 ] [    0.0000E+00,   0.1000E+01 ]
V:
 [    0.1926E+00,   0.1926E+00 ]
INEQ_CERT_FEASIBLE:
  F T
NIN_POSS_BINDING:1

Number of bisections: 1
BEST_ESTIMATE:   -0.1500E+01
 Total number of boxes processed in loop:            4
 Overall CPU time:    0.5000D-01
```

Recent Improvements to GlobSol

- Simplified installation
- Provided makefiles for several compiler / operating system combinations
- Eliminated many bugs
- Incorporated numerical techniques for interval constants in objective and constraint definitions
- Enabled constraint propagation for equality and inequality constraints

Simple Example of "Thick" Constants

$$\text{minimize} \ \ (x_1 - [1,2])^2 + (x_2 - [3,4])^2$$

```
PROGRAM THICK_PARAMETER_EXAMPLE
 USE CODELIST_CREATION
 IMPLICIT NONE

 INTEGER, PARAMETER:: NN=2
 TYPE(CDLVAR), DIMENSION(NN):: X
 TYPE(CDLLHS) :: PHI

 OUTPUT_FILE_NAME='thick_parameter_example.CDL'
 CALL INITIALIZE_CODELIST(X)

 PHI = (X(1) - INTERVAL(1,2))**2 &
       + (X(2)-INTERVAL(3,4))**2

 CALL FINISH_CODELIST
END PROGRAM THICK_PARAMETER_EXAMPLE
```

Example with Thick Constants

(continued)

The "solution" is the set of all possible minima with constants in $[1, 2]$ and $[3, 4]$. Thus, a minimum minimum and maximum minimum are obtained. The solution is

$$x_1 \in [1, 2], \quad x_2 \in [3, 4], \quad \text{and } \phi = 0.$$

With initial box ([-10,10], [-10,10]), GlobSol gives

```
 Box no.:           1
 Box coordinates:
 [ 0.1000D+01, 0.1500D+01 ] [ 0.3000D+01, 0.4000D+01 ]
PHI:
 [ 0.0000D+00, 0.2000D+01 ]

 Box no.:           2
 Box coordinates:
 [ 0.1500D+01, 0.2000D+01 ] [ 0.3000D+01, 0.4000D+01 ]
PHI:
 [ 0.0000D+00, 0.2000D+01 ]

 BEST_ESTIMATE:    0.5000D+00
 Total number of boxes processed in loop:           4
 Overall CPU time:    0.0000D+00
```

Thick Constants

The basic idea

- The algorithm is made practical by stopping bisection when

  $$\text{w(value at center)} > \beta\text{w(interval image)}$$

  for adjustable parameter $\beta \in [0, 1]$.
- Smaller β leads to less work, but may lead to additional overestimation.
- Detailed experimental results will be given elsewhere.

A Practical Application

- This is being applied to a non-trivial maintenance interval optimization problem.
- This is ongoing work with Claudio Rocco.

GlobSol Installation and Use

Installation and use has been greatly simplified.

1. Create a root directory for GlobSol.
2. Obtain the GlobSol file from `http://interval.usl.edu/GLOBSOL/GlobSol.tar.Z` and extract it into the GlobSol root directory.
3. Select an appropriate makefile and system-dependent file set (e.g. Sun with Sun compiler, Sun with NAG compiler), and extract these.
4. Edit a line in the makefile to specify the GlobSol root directory.
5. Edit a line in the macro (e.g. `unix_fmake`) that creates internal representations from user-supplied objective function programs.
6. Run “make”.
7. All executable, macro, and library files are now either in the GlobSol root directory or the subdirectory executables.

GlobSol References

- For the source, installation instructions, user guide, etc.:
  `http://www.mscs.mu.edu/~globsol/`
- *Rigorous Global Search: Continuous Problems*, R. B. Kearfott, Kluwer Academic Publishers, 1996. Contains
  - Most of the basic ideas underlying GlobSol.
  - Structure of the research code that eventually became GlobSo.
- For these transparencies:
  `http://interval.usl.edu/preprints/1999_SIAM.ps`
  (Postscript)

  `http://interval.usl.edu/preprints/1999_SIAM.dvi`
  (TeX DVI)