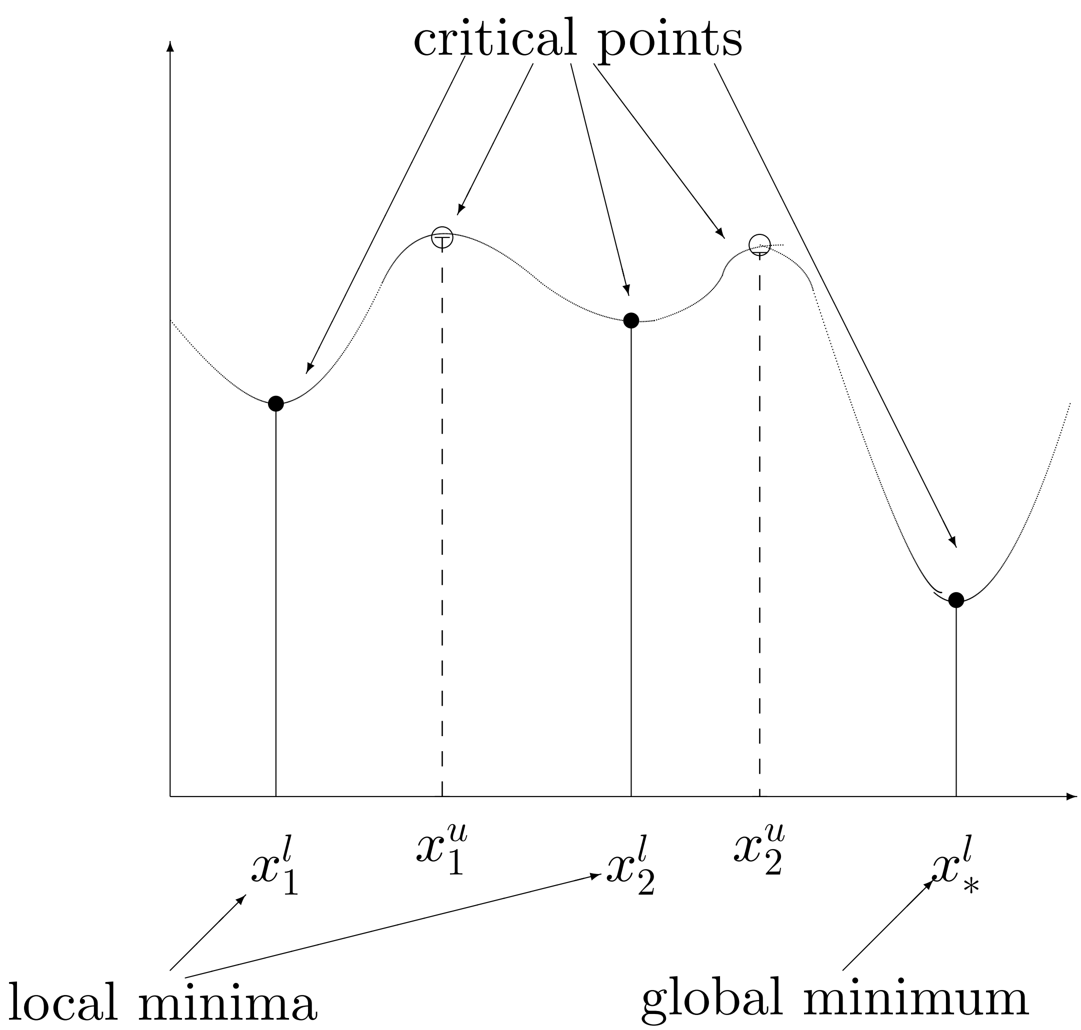
A Brief Introduction to Global Optimization and a Preview of INTOPT_90

by R. Baker Kearfott
Department of Mathematics, University of Southwestern Louisiana
U.S.L. Box 4-1010, Lafayette, LA 70504-1010
U.S.A.
rbk@usl.edu

This talk will:

- compare local and global optimization;
- discuss heuristic, deterministic, and automatically verified algorithms;
- briefly review interval algorithms for global optimization;
- touch on the state of the art.
- time permitting, preview the Fortran 90 software INTOPT_90.

Local Versus Global Optimization



Local Optimization Versus Global Optimization

Local Optimization

- The model is steepest descent with univariate line searches (for monotone decrease of the objective function). (Start a ball on a hill and let it roll to the bottom of the nearest valley.)
- Algorithm developers speak of "globalization," but mean only the design of algorithm variants that increase the domain of convergence. (See J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Least Squares*, Prentice–Hall, 1983.)
- Algorithms contain many heuristics, and do not always work. However, many useful implementations exist.

Local Optimization Versus Global Optimization

Global Optimization

- is a much harder problem. Progress has accelerated with increases in computing power.
- Early milestones are L. C. W. Dixon and G. P. Szego, *Towards Global Optimization* (North–Holland, 1975), and *Towards Global Optimization 2* (North–Holland, 1977).
- Two types of algorithms: <u>stochastic</u> and <u>deterministic</u>.
- Deterministic algorithms can be either <u>rigorous</u> or <u>heuristic</u>.

Global Optimization

Stochastic Algorithms

Monte-Carlo search: Random points are generated in the search space. The point with lowest objective value is taken to be the global optimum.

Simulated annealing: is similar to a local optimization method, although larger objective values are accepted with a probability that decreases as the algorithm progresses.

Genetic algorithms: Attributes, such as values of a particular coordinate, correspond to particular "genes." "Chromosomes" of these genes are recombined randomly, and only the best results are kept. Random "mutations" are introduced.

Global Optimization

Deterministic 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 $\mathbf{X}$ gives bounds on the actual range of ϕ over $\mathbf{X}$.
  - If directed rounding is used, the bounds rigorously contain the mathematical range.
  - The bounds, in general, are overestimates.
- If the lower bound of $\phi(\mathbf{X})$ is greater than a previously computed objective value $\phi(X)$, then $\mathbf{X}$ 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 $\mathbf{X}$.

Global Optimization

Hybrid Stochastic / Deterministic Algorithms

Recently, several people have combined statistical methods with deterministic methods.

- Janos Pinter uses a statistical model to estimate local approximations to Lipschitz constants for a global search.
- Donald Jones constructs a cumulative statistical model of of the objective, and uses deterministic global optimization with a simpler objective to determine optimal placement of the next sample point.
- Kaj Madsen uses multiple starts of a local optimizer to simulate a rigorous global search with interval methods.

Hybrid Stochastic / Deterministic Algorithms

Janos Pinter's Approach

- Has the structure of a rigorous global search.
- Is only heuristic, but is nonetheless successful at solving many practical problems.
- Has solved problems in hundreds of variables, including some long-standing open mathematical (geometrical) questions (but not rigorously).
- Is a mainstay of Janos' consulting business (his primary employment).

Hybrid Stochastic / Deterministic Algorithms

Donald Jones' Work

- Begins with a statistical model with a few points.
- Optimal placement of the next sample point is the global optimum of a simple quadratic.
- The quadratic is optimized with interval techniques.
- Graphical renderings of the fitted objective are much closer to the actual graphs than alternate techniques.

Hybrid Stochastic / Deterministic Algorithms

Kaj Madsen's Technique

- Kaj starts with an interval-based algorithm for global optimization.
- Assuming that interval evaluations are not available, lower bounds on the objective function are obtained by repeated local approximate optimization with random starts.
- The new technique in only heuristic (not rigorous).
- The new technique can be used for "black box" optimization.
- Kaj compares the reliability and efficiency of the new technique to interval methods.

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.

Interval Methods

Advantages

easier to use: Obtaining bounds with interval methods involves programming the objective function, while using Lipschitz constant-based methods may require extensive preliminary analysis.

more efficient: Despite interval overestimation of ranges, the overestimation is often less than with a fixed Lipschitz constant. *(But keep in mind the success of hybrid deterministic / stochastic algorithms.)*

more capable: With directed roundings, interval methods cannot lie. Also, interval Newton iteration results in quadratic convergence effects.

On Constraints

- Constrained problems are more difficult, since an objective function value $\phi(X)$ does not represent an upper bound on the minimum unless X is feasible.
- The Fritz–John system (Lagrange multipliers) may be used in the interval Newton methods, but other techniques must also be incorporated for practical algorithms.
- Constraints may be handled heuristically (by solving a perturbed problem) or rigorously.
- Alternate techniques are available for handling bound constraints.

The INTOPT_90 Package

Main Features

- is in portable Fortran 90.
- solves unconstrained and constrained problems, as well as nonlinear algebraic systems.
- can use good initial guesses, and can use any good local optimization routine.
- can use constraint propagation techniques (substitution/iteration) on the intermediate quantities in objective function evaluation.
- Objective function and constraints are input simply as Fortran 90 programs.
- The algorithm is configurable.

Use of INTOPT_90

An Example

The following file defines the objective function

$$\phi(X) = (x_1 - 1)^4 + (x_2 - 1)^4 + (x_3 - 2)^4$$

subject to constraints

$$\begin{aligned} x_1^2 + x_2^2 + x_3^2 - 6 &= 0 \\ x_1^2 + x_2^2 - 3 &= 0 \end{aligned}$$

```
PROGRAM WOLFE3
  USE OVERLOAD
      PARAMETER (NN=3)
      PARAMETER (NSLACK=0)
      TYPE(CDLVAR), DIMENSION(NN+NSLACK):: X
      TYPE(CDLLHS), DIMENSION(1):: PHI
      TYPE(CDLINEQ), DIMENSION(2):: G

      OUTPUT_FILE_NAME='WOLFE3.CDL'
      CALL INITIALIZE_CODELIST(X)

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

  G(1) = X(1)**2 + X(2)**2 + X(3)**2 - 6
  G(2) = X(1)**2 + X(2)**2 - 3

      CALL FINISH_CODELIST
END PROGRAM WOLFE3
```

INTOPT_90 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.
5. Excerpts from the output file follow.

INTOPT_90 **Example**

Excerpts from the Output File

```
 Output from RUN_GLOBAL_OPTIMIZATION on  06/19/1996 at 18:35:44.
 DATA WAS TAKEN FROM DATA FILE: wolfe3.DT1
(lines deleted)

 LIST OF BOXES CONTAINING VERIFIED FEASIBLE POINTS:

 Box no.:          1
 Box coordinates:
     .1225D+01    .1225D+01     .1225D+01    .1225D+01
     .1732D+01    .1732D+01
PHI:
     .1026D-01    .1026D-01
(lines deleted)

 Box contains the following approximate root:
     .1225D+01    .1225D+01    .1732D+01
 OBJECTIVE ENCLOSURE AT APPROXIMATE ROOT:
     .1026D-01    .1026D-01

(lines deleted)

 Total number of dense slope matrix evaluations:        116
 Total number second-order interval evaluations of the
 original function:        29
 Total number dense interval constraint evaluations:        188
(lines deleted)
  Total number of boxes processed in loop:         13
 Overall CPU time:    .3000D+01
```

Strengths of INTOPT_90

- Problems are easy to input.
- Algorithm configuration is flexible (good for algorithm research and for problems with varying properties).
- Provides constrained optimization, unconstrained optimization, and nonlinear algebraic systems within the same framework.
- Provides *rigorous* global search.
- Is portable.
- Has configurable levels of printing.
- Compiles numerous performance statistics.
- A book explains its use and underlying algorithms and theory (*Rigorous Global Search: Continuous Problems*, Kluwer, 1996 or 1997).

Weaknesses of INTOPT_90

- Interpretive nature of function and derivative evaluation is slow. (Floating point is 8 times slower than compiled floating point, and interval is 16 times slower than compiled floating point.)
- Does not take advantage of sparsity structure or other special features of the problem. (However, the user can extend the list of elementary functions, if he can write routines that supply good bounds on ranges for specific subexpressions.)
- Better local optimization routines can be found than those presently bundled with INTOPT_90. (However, it is relatively easy in INTOPT_90 to replace these routines.)

INTOPT_90 Reference

Rigorous Global Search: Continuous Problems, R. B. Kearfott, Kluwer Academic Publishers, 1996.

Contains

- An introduction to interval methods
- An introduction to global search algorithms
- Some specifics for INTOPT_90.

Commericalization of INTOPT_90

Goals

- To establish verified global optimization and, more generally, interval computations in the mainstream of scientific computing practice.
- To make verified global optimization technology and interval computations more widely "available to the masses" than before.

This is done through a Sun Microsystems Cooperative Research contract, with participants with varied backgrounds from different universities.