A Current Assessment of Interval Techniques in Global Optimization

R. Baker Kearfott

Department of Mathematics
University of Louisiana at Lafayette

October 3, 2006

Outline

1. Introduction
   - Local versus Global Optimization
   - Theoretical and Practical Considerations
   - Alternate Approaches to Global Optimization
   - General Branch and Bound Algorithms
2. Interval Techniques
   - Bounding Ranges
   - Proving Existence and Uniqueness
   - Constraint Propagation and Relaxations
3. Current Software and Prospects

The General Global Optimization Problem

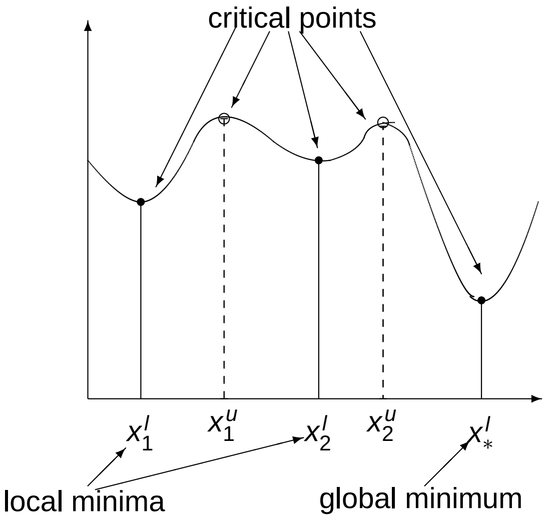
Mathematical Description

minimize $\varphi(x)$
subject to $c_i(x) = 0, i = 1, \ldots, m_1,$
$g_i(x) \leq 0, i = 1, \ldots, m_2,$
where $\varphi : \mathbb{R}^n \to \mathbb{R}$ and $c_i, g_i : \mathbb{R}^n \to \mathbb{R}$.

- We refer to the region defined by the constraints as $\boldsymbol{D}$.
- Often bounds on the search region are given by $\boldsymbol{x} = ([\underline{x}_1, \overline{x}_1], \ldots [\underline{x}_n, \overline{x}_n])$.

Outline

Local Versus Global Optimization



Local Versus Global Optimization

Local Optimization

- The model is steepest descent and locally convergent methods (such as Newton's method) 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.

Outline

Do We Want Local or Global Optima?

Examples

- φ is the cost of running a (nominally) $50,000,000 per month plant:
  The plant manager would like the smallest possible operating cost, but would be happy with a 5% lower cost than before.
- φ represents the potential energy of a particular conformation of a molecule:
  The globally lowest value for φ gives the most information, but local minima give some information, and finding the global minimum may not be practical.
- A mathematician has reduced a proof to showing that $\varphi \geq 1$ everywhere:
  The global minimum must not only be found, but also must be rigorously proven to be so.

Do We Want Local or Global Optima?

Examples

- φ is the cost of running a (nominally) \$50,000,000 per month plant:
  The plant manager would like the smallest possible operating cost, but would be happy with a 5% lower cost than before.
- φ represents the potential energy of a particular conformation of a molecule:
  The globally lowest value for φ gives the most information, but local minima give some information, and finding the global minimum may not be practical.
- A mathematician has reduced a proof to showing that $\varphi \geq 1$ everywhere:
  The global minimum must not only be found, but also must be rigorously proven to be so.

Do We Want Local or Global Optima?

Examples

- φ is the cost of running a (nominally) $50,000,000 per month plant:
  The plant manager would like the smallest possible operating cost, but would be happy with a 5% lower cost than before.
- φ represents the potential energy of a particular conformation of a molecule:
  The globally lowest value for φ gives the most information, but local minima give some information, and finding the global minimum may not be practical.
- A mathematician has reduced a proof to showing that $\varphi \geq 1$ everywhere:
  The global minimum must not only be found, but also must be rigorously proven to be so.

Local Versus Global Optimization

Global Algorithms

- Global optimization is NP hard, meaning that no general algorithm is known that solves all large problems efficiently.
- Thus, barring monumental discoveries, any *general* algorithm will fail for some high-dimensional problems.
- Some low-dimensional problems are also difficult, but many can be solved.
- Successes have occurred for large problems of practical interest by taking advantage of structure.
- Useful results have also been obtained by replacing guarantees by heuristics.
- Advances in computer speed and algorithm construction allow ever more practical problems to be solved.

Outline

Classes of Global Optimization Algorithms

- There are two types of algorithms: stochastic and deterministic.
- Deterministic algorithms can be either rigorous or heuristic.

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.

Deterministic Algorithms

- These either
  - involve using problem-specific structure to be assured that computed points are global optimizers, or else
  - involve some kind of systematic global search over the domain, generally *branch and bound* processes.
- The various branch and bound algorithms rely on estimates of ranges of the objective and constraints over subdomains.
- The estimates on the range can be either mathematically exact or heuristic.
- The mathematically exact estimates can be either rigorous (taking account of roundoff error) or non-rigorous (using floating point arithmetic and other approximation algorithms without taking account of numerical errors).

Deterministic Algorithms

- These either
  - involve using problem-specific structure to be assured that computed points are global optimizers, or else
  - involve some kind of systematic global search over the domain, generally *branch and bound* processes.
- The various branch and bound algorithms rely on estimates of ranges of the objective and constraints over subdomains.
- The estimates on the range can be either mathematically exact or heuristic.
- The mathematically exact estimates can be either rigorous (taking account of roundoff error) or non-rigorous (using floating point arithmetic and other approximation algorithms without taking account of numerical errors).

Deterministic Algorithms

- These either
  - involve using problem-specific structure to be assured that computed points are global optimizers, or else
  - involve some kind of systematic global search over the domain, generally *branch and bound* processes.
- The various branch and bound algorithms rely on estimates of ranges of the objective and constraints over subdomains.
- The estimates on the range can be either mathematically exact or heuristic.
- The mathematically exact estimates can be either rigorous (taking account of roundoff error) or non-rigorous (using floating point arithmetic and other approximation algorithms without taking account of numerical errors).

Deterministic Algorithms

- These either
  - involve using problem-specific structure to be assured that computed points are global optimizers, or else
  - involve some kind of systematic global search over the domain, generally *branch and bound* processes.
- The various branch and bound algorithms rely on estimates of ranges of the objective and constraints over subdomains.
- The estimates on the range can be either mathematically exact or heuristic.
- The mathematically exact estimates can be either rigorous (taking account of roundoff error) or non-rigorous (using floating point arithmetic and other approximation algorithms without taking account of numerical errors).

Deterministic Algorithms

- These either
  - involve using problem-specific structure to be assured that computed points are global optimizers, or else
  - involve some kind of systematic global search over the domain, generally *branch and bound* processes.
- The various branch and bound algorithms rely on estimates of ranges of the objective and constraints over subdomains.
- The estimates on the range can be either mathematically exact or heuristic.
- The mathematically exact estimates can be either rigorous (taking account of roundoff error) or non-rigorous (using floating point arithmetic and other approximation algorithms without taking account of numerical errors).

Outline

The Abstract Algorithm

Initialization: Establish an upper bound $\overline{\varphi}$ on the global optimum over the feasible set.

Branching: Subdivide the initial region $\boldsymbol{D}$ into two or more subregions $\tilde{\boldsymbol{D}}$. (*branching* step, where we branch into subregions.)

Bounding: Bound the range of φ below over each $\tilde{\boldsymbol{D}}$, to obtain

$$\underline{\varphi}(\tilde{\boldsymbol{D}}) \leq \left\{\varphi(x) \mid x \in \tilde{\boldsymbol{D}}, c(x) = 0, g(x) \leq 0.\right\}.$$

Fathoming: *IF* $\underline{\varphi} > \overline{\varphi}$ *THEN* discard $\tilde{\boldsymbol{D}}$,
ELSE IF the diameter of $\tilde{\boldsymbol{D}}$ is smaller than a specified tolerance *THEN* put $\tilde{\boldsymbol{D}}$ onto a list of boxes containing possible global optimizers,
ELSE Put $\tilde{\boldsymbol{D}}$ is put onto a list for further branching and bounding.
END IF

Obtaining $\underline{\varphi}$

- Some algorithms rely on Lipschitz constants to obtain $\underline{\varphi}$. These constants can be determined rigorously or heuristically.
- $\underline{\varphi}$ may also be obtained with outwardly rounded interval arithmetic or non-rigorous interval arithmetic, explained below.
- Recent algorithms (especially during the past decade) involve *relaxations* of the global optimization problem to obtain $\underline{\varphi}$. For example, the relaxation can be a linear program, solvable by commercial technology.

Obtaining $\underline{\varphi}$

One of János Pintér's Techniques

- For a general algorithm, János uses a statistical model to estimate approximations to local Lipschitz constants.
- This is used in the structure of a deterministic algorithm, but is only heuristic.
- The technique nonetheless leads to success at obtaining useful results for many very large practical problems, in the successful GAMS / LGO software.
- It may be possible to make this process rigorous or more efficient for particular problems by taking advantage of structure.

Obtaining $\overline{\varphi}$

- In principle, $\overline{\varphi}$ can be set to the smallest value of $\varphi(x)$ found, where x is any feasible point.
- Appropriate x can be found by running a constrained local optimizer.
- In practice, this is a difficult part of the algorithm to make rigorous:
  - It must be proven that x is feasible, not possible if equality constraints or active inequality constraints are present.
  - $\overline{\varphi}$ may be rigorously obtained by first proving a feasible point exists within small bounds $\check{\boldsymbol{x}}$ centered at x, then computing a rigorous upper bound on φ over $\check{\boldsymbol{x}}$.

Acceleration Procedures

Various other procedures are used (and indeed are necessary in many cases to make the algorithm practical) to reduce the size of the region $\boldsymbol{D}$ and to possibly reject $\boldsymbol{D}$ before the branching step. To a large extent, this is what makes branch and bound codes unique, and determines overall efficiency. Some such procedures include

- various forms of constraint propagation;
- interval Newton methods;
- proofs of infeasibility of linear programs in combination with linear relaxations (to be briefly explained).

The practicality of a particular code depends on implementation details of these acceleration procedures.

Outline

1 Introduction
- Local versus Global Optimization
- Theoretical and Practical Considerations
- Alternate Approaches to Global Optimization
- General Branch and Bound Algorithms

2 Interval Techniques
- Bounding Ranges
- Proving Existence and Uniqueness
- Constraint Propagation and Relaxations

3 Current Software and Prospects

Bounding Ranges with Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

Bounding Ranges with Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

Bounding Ranges with Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

Bounding Ranges with Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

Bounding Ranges with Interval Arithmetic

A Pitfall: Dependency and Overestimation

- The interval values contain the actual ranges, but are possibly significantly larger. For example, if $f(x) = (x+1)(x-1)$, then

$$\begin{aligned}\boldsymbol{f}([-2,2]) &= \big([-2,2]+1\big)\big([-2,2]-1\big)\\ &= [-1,3][-3,1] = [-9,3],\end{aligned}$$

  whereas the exact range is $[-1,3]$.

- However, if we write f equivalently as $f(x) = x^2 - 1$, and we suppose we compute the range of x^2 exactly, we obtain

$$\boldsymbol{f}([-2,2]) = [-2,2]^2 - 1 = [0,4] - 1 = [-1,3],$$

  the exact range.

Bounding Ranges with Interval Arithmetic

A Pitfall: Dependency and Overestimation

- The interval values contain the actual ranges, but are possibly significantly larger. For example, if $f(x) = (x+1)(x-1)$, then

$$\begin{aligned} \boldsymbol{f}([-2,2]) &= \big([-2,2]+1\big)\big([-2,2]-1\big) \\ &= [-1,3][-3,1] = [-9,3], \end{aligned}$$

  whereas the exact range is $[-1,3]$.
- However, if we write f equivalently as $f(x) = x^2 - 1$, and we suppose we compute the range of x^2 exactly, we obtain

$$\boldsymbol{f}([-2,2]) = [-2,2]^2 - 1 = [0,4] - 1 = [-1,3],$$

  the exact range.

Dependency and Overestimation

Consequences

- The range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- However, there are some notable successes.

Dependency and Overestimation

Consequences

- The range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- However, there are some notable successes.

(slide 35)

Dependency and Overestimation

Consequences

- The range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- However, there are some notable successes.

Dependency and Overestimation
Consequences

- The range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- However, there are some notable successes.

Outline

Proving Existence and Uniqueness

Interval Newton Methods

Let $f(x) = 0$, $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ represent a system of nonlinear equations. Think of an interval Newton method as an operator sending $\boldsymbol{x}$ to $\tilde{\boldsymbol{x}}$:

$$\tilde{\boldsymbol{x}} = \boldsymbol{N}(f; \boldsymbol{x}, \check{x}) = \check{x} + \boldsymbol{v}, \quad \text{where} \quad \Sigma(\boldsymbol{A}, -f(\check{x})) \subset \boldsymbol{v},$$

where $\boldsymbol{A}$ is a Lipschitz matrix for f over $\boldsymbol{x}$ and $\Sigma(\boldsymbol{A}, -f(\check{x}))$ is that set $\{x \in \mathbb{R}^n\}$ such that there exists an $A \in \boldsymbol{A}$ with $Ax = -f(\check{x})$.

Theorem

Suppose $\tilde{\boldsymbol{x}}$ is the image of $\boldsymbol{x}$ under an interval Newton method. If $\tilde{\boldsymbol{x}} \subseteq \boldsymbol{x}$, it follows that there exists a unique solution of $f(x) = 0$ within $\boldsymbol{x}$.

Interval Newton Methods

A Practical Summary

- $\boldsymbol{N}(f;\boldsymbol{x},\check{x})$ can be computed similarly to a classical point multivariate Newton step, but with interval arithmetic.
- Any solutions of $f(x)=0$ in $\boldsymbol{x}$ must be in $\boldsymbol{N}(f;\boldsymbol{x},\check{x})$. (Hence, an interval Newton can be used to reduce the volume of $\boldsymbol{x}$, an acceleration procedure.)
- $\boldsymbol{N}(f;\boldsymbol{x},\check{x})\subset\boldsymbol{x}$ implies there is a unique solution to $f(x)=0$ in $\boldsymbol{N}(f;\boldsymbol{x},\check{x})$, and hence in $\boldsymbol{x}$.
- Another consequence: $\boldsymbol{N}(f;\boldsymbol{x},\check{x})\cap\boldsymbol{x}=\emptyset$ implies $\boldsymbol{x}$ is fathomed, and $\boldsymbol{x}$ may be discarded.
- Interval Newton methods are locally quadratically convergent.

Interval Newton Methods
Pitfalls

- For a given width of $\boldsymbol{x}$, the width of $\boldsymbol{N}(f;\boldsymbol{x},\check{x})$ is proportional to the condition number of the Jacobi matrix of f.
- For large widths of $\boldsymbol{x}$, overestimation in evaluation of the Jacobi matrix may cause the interval extension of the Jacobi matrix to contain singular matrices.
- Hence, existence / uniqueness verification is sometimes problematical unless a good approximation to an optimum is known; even then, it may not be possible to eliminate a large region.
- Nonetheless, interval Newton methods can sometimes be coaxed to narrow some coordinates, even in the presence of singularities and $\boldsymbol{x}$ large.

Interval Newton Methods

Pitfalls

- For a given width of $\boldsymbol{x}$, the width of $\boldsymbol{N}(f; \boldsymbol{x}, \check{x})$ is proportional to the condition number of the Jacobi matrix of f.
- For large widths of $\boldsymbol{x}$, overestimation in evaluation of the Jacobi matrix may cause the interval extension of the Jacobi matrix to contain singular matrices.
- Hence, existence / uniqueness verification is sometimes problematical unless a good approximation to an optimum is known; even then, it may not be possible to eliminate a large region.
- Nonetheless, interval Newton methods can sometimes be coaxed to narrow some coordinates, even in the presence of singularities and $\boldsymbol{x}$ large.

Interval Newton Methods

Pitfalls

- For a given width of $\boldsymbol{x}$, the width of $\boldsymbol{N}(f;\boldsymbol{x},\check{x})$ is proportional to the condition number of the Jacobi matrix of f.
- For large widths of $\boldsymbol{x}$, overestimation in evaluation of the Jacobi matrix may cause the interval extension of the Jacobi matrix to contain singular matrices.
- Hence, existence / uniqueness verification is sometimes problematical unless a good approximation to an optimum is known; even then, it may not be possible to eliminate a large region.
- Nonetheless, interval Newton methods can sometimes be coaxed to narrow some coordinates, even in the presence of singularities and $\boldsymbol{x}$ large.

Interval Newton Methods
Pitfalls

- For a given width of $\boldsymbol{x}$, the width of $\boldsymbol{N}(f;\boldsymbol{x},\check{x})$ is proportional to the condition number of the Jacobi matrix of f.
- For large widths of $\boldsymbol{x}$, overestimation in evaluation of the Jacobi matrix may cause the interval extension of the Jacobi matrix to contain singular matrices.
- Hence, existence / uniqueness verification is sometimes problematical unless a good approximation to an optimum is known; even then, it may not be possible to eliminate a large region.
- Nonetheless, interval Newton methods can sometimes be coaxed to narrow some coordinates, even in the presence of singularities and $\boldsymbol{x}$ large.

Outline

Constraint Propagation

In simple terms, constraint propagation is based on

1. solving a relation for a selected variable in terms of the other variables;
2. using the bounds on the other variables (e.g. by evaluating the expression with interval arithmetic) to determine new bounds for the selected variable;
3. iterating the process (in some order and with some stopping criteria) to narrow bounds on as many variables as is practical.

Constraint Propagation

In simple terms, constraint propagation is based on

1. solving a relation for a selected variable in terms of the other variables;
2. using the bounds on the other variables (e.g. by evaluating the expression with interval arithmetic) to determine new bounds for the selected variable;
3. iterating the process (in some order and with some stopping criteria) to narrow bounds on as many variables as is practical.

Constraint Propagation

In simple terms, constraint propagation is based on

1. solving a relation for a selected variable in terms of the other variables;
2. using the bounds on the other variables (e.g. by evaluating the expression with interval arithmetic) to determine new bounds for the selected variable;
3. iterating the process (in some order and with some stopping criteria) to narrow bounds on as many variables as is practical.

Relaxations

- A *relaxation* to a global minimization problem is a related problem whose global minimum is less than or equal to the global minimum of the original problem.
- Relaxations can be formed by
  - replacing the objective φ by a simpler function that is less than or equal to φ on the feasible set;
  - replacing each function g_i corresponding to an inequality constraint by a simpler function that is less than or equal to g_i on the feasible set;
  - replacing each equality constraint by two inequality constraints, then handling these.
- Several successful commercial global optimization packages use *linear relaxations* (i.e. the relaxation is a linear program).

Relaxations

- A *relaxation* to a global minimization problem is a related problem whose global minimum is less than or equal to the global minimum of the original problem.
- Relaxations can be formed by
  - replacing the objective φ by a simpler function that is less than or equal to φ on the feasible set;
  - replacing each function g_i corresponding to an inequality constraint by a simpler function that is less than or equal to g_i on the feasible set;
  - replacing each equality constraint by two inequality constraints, then handling these.
- Several successful commercial global optimization packages use *linear relaxations* (i.e. the relaxation is a linear program).

Relaxations

- A *relaxation* to a global minimization problem is a related problem whose global minimum is less than or equal to the global minimum of the original problem.
- Relaxations can be formed by
  - replacing the objective φ by a simpler function that is less than or equal to φ on the feasible set;
  - replacing each function g_i corresponding to an inequality constraint by a simpler function that is less than or equal to g_i on the feasible set;
  - replacing each equality constraint by two inequality constraints, then handling these.
- Several successful commercial global optimization packages use *linear relaxations* (i.e. the relaxation is a linear program).

Relaxations

Non-Verified versus Verified

- Commercial packages have used approximate solutions to linear relaxations, so, although deterministic, are not rigorous.
- In 2002, Neumaier and Scherbina showed how, given an approximate solution to a linear program, a mathematically rigorous lower bound on the solution can be computed by a technique involving the duality gap.
- Similarly, Neumaier and Scherbina presented a simple computation to rigorously show that a linear program is infeasible. (If a relaxation is infeasible, then the original problem must also be infeasible.)
- This enables several successful techniques for non-verified (non-interval) algorithms to be incorporated into rigorous algorithms.

Relaxations

Non-Verified versus Verified

- Commercial packages have used approximate solutions to linear relaxations, so, although deterministic, are not rigorous.
- In 2002, Neumaier and Scherbina showed how, given an approximate solution to a linear program, a mathematically rigorous lower bound on the solution can be computed by a technique involving the duality gap.
- Similarly, Neumaier and Scherbina presented a simple computation to rigorously show that a linear program is infeasible. (If a relaxation is infeasible, then the original problem must also be infeasible.)
- This enables several successful techniques for non-verified (non-interval) algorithms to be incorporated into rigorous algorithms.

Relaxations

Non-Verified versus Verified

- Commercial packages have used approximate solutions to linear relaxations, so, although deterministic, are not rigorous.
- In 2002, Neumaier and Scherbina showed how, given an approximate solution to a linear program, a mathematically rigorous lower bound on the solution can be computed by a technique involving the duality gap.
- Similarly, Neumaier and Scherbina presented a simple computation to rigorously show that a linear program is infeasible. (If a relaxation is infeasible, then the original problem must also be infeasible.)
- This enables several successful techniques for non-verified (non-interval) algorithms to be incorporated into rigorous algorithms.

BARON

The *Branch And Reduce Optimization Navigator,* due to Nick Sahinidis et al, is highly competitive and commercial (available through GAMS, with a version on NEOS).
BARON:

- uses interval arithmetic in selected places to compute ranges;
- is effective at using linear relaxations, constraint propagation, and associated techniques, but in a non-rigorous fashion;
- is successful apparently partly due to the astute combination of various basic techniques, and partly due to state-of-the-art components (such as a good linear program solver);
- is non-rigorous overall, but possibly can be made rigorous without undue loss of efficiency;
- has some limitations.

ICOS

The *Interval COnstraints Solver*, from Yahia Lebbah et al at INRIA Sophia Antipolis, is a fully mathematically rigorous constrained optimizer. ICOS:

- is built upon decades of research in constraint propagation languages;
- uses linear relaxations in conjunction with the commercial ILOG CPLEX solver;
- is totally mathematically rigorous;
- is pointing the way to how mathematically rigorous solvers might be made overall competitive.

GlobSol

GlobSol is our own Fortran-90 based research code for global optimization. GlobSol:

- is self-contained and totally mathematically rigorous;
- grew out of a 1980's code (INTBIS) for rigorous solution of nonlinear systems of equations;
- it heavily-based on interval Newton technology, but also includes a version of constraint propagation;
- has linear relaxations and other experimental techniques in a non-released version.
- can possibly be made more competitive with restructuring and rewriting;
- can possibly benefit from partnerships that supply commercial components (such as a commercial LP solver).

GlobSol
Some History and Successes

- GlobSol was put into its present form with a SunSoft Cooperative Research contract.
- During the term of that contract, collaborations with various persons, especially through George Corliss, led to some notable successes.
- The best successes concerned portfolio analysis, and led one of the graduate students involved to form a consulting firm based on the technology.

GAMS / LGO

- As mentioned, GAMS / LGO utilizes a hybrid combination of branch and bound and statistical techniques.
- János Pintér should best be able to address
  - the practical aspects of LGO and
  - the relationship of LGO to rigorous computations.

(slide 59)

Introduction
Local versus Global Optimization
Theoretical and Practical Considerations
Alternate Approaches to Global Optimization
General Branch and Bound Algorithms

Interval Techniques
Bounding Ranges
Proving Existence and Uniqueness
Constraint Propagation and Relaxations

Current Software and Prospects

Other Packages

There are numerous other packages we have not mentioned.

- We have only mentioned the packages we have perceived to presently be major players in real-world applications. For a slightly more complete listing, see Arnold Neumaier's page at `http://www.mat.univie.ac.at/~neum/glopt/software_g.html`
- We have not attempted to discuss purely stochastically-based packages (simulated annealing, genetic algorithms, etc.)
- For a listing of general interval-based software, see `http://www.cs.utep.edu/interval-comp/intsoft.html`.

Prospects for Rigorous Software
Summary

- Continued study of successful non-rigorous techniques, of their interaction in specific implementations, and of how they can be made rigorous will lead to more competitive totally rigorous codes.
- Emerging theoretical developments (such as newly proposed necessary and sufficient conditions for global optima) may be useful in mathematically rigorous algorithms.
- There is much evidence that successful codes are built upon successful local optimizers and successful linear program solvers. This includes mathematically rigorous codes.

Prospects for Rigorous Software

Possible Limitations

- There is strong evidence that the most practical (or the only practical) codes for challenging applications will be designed specifically for those applications.
- Perhaps total mathematical rigor is out of reach for certain problems (for which heuristics or stochastic elements must be used).

(slide 62)