Implementing a Global Solver

in

a General Purpose Callable Library

by

Tony Gau

Linus Schrage

LINDO Systems

http://www.lindo.com

at

Argonne Global Optimization Theory Institute

10 Sept 2003

LINDO SYSTEMS INC.

Global Solver Overview

LINDO API library is an LP, NLP, IP solver used by LINGO and What'sBest spreadsheet add-in.

LINDO API contains a global solver that finds a guaranteed (disclaimer: assuming infinite precision) global optimum to an arbitrary optimization problem;

Fully supports all common math functions:

$x*y$, x/y, x^y, log(x), exp(x), sqrt(x),

sin(x), cos(x), tan(x),

floor(x),

abs(x), max(x,y), min(x,y),

if(x,y,z), AND, OR, [where x is a logical expression]

psn(z), psl(z) [Normal distribution]

LINDO SYSTEMS INC.

Global Solver in LINDO API: Outline

1) Getting a good solution quickly, multistart and other ideas;
2) Guaranteed solutions: a) convex relaxation,   b) split/branch;
3) Constraint propagation, bound tightening, interval arithmetic;
4) Constructing convex relaxations for wide range of functions:
   continuous and smooth: *x+y, x-y, x*y, sin*(*x*), *cos*(*x*), etc.
   continuous, nonsmooth: *abs*(*x*), *max*(*x,y*), *min*(*x,y*),
   smooth not quite continuous, *x/y, x^y, tan*(*x*), *floor*(*x*),
   logical functions: *if*( ), *and*, or, *not* >=, <=, = =, !=,
   application specific functions: Normal cdf & linear loss function;
5) Using linearization + linear MIP only for functions such as:
   abs(), *min*( ), *if*(), special cases of *x*y*;
6) Choosing an algebraic representation, reformulation,
   e.g., *x*(y-x)* vs. *x*y –x^2*;
7) Choosing a machine representation with some vector functions,
8) Choosing a good branching;
9) Numerical stability issues in cut management, branch selection;
10) Computational testing. .

LINDO SYSTEMS INC.

Roots

McCormick(1976): Convex relaxations and branching.

Sahinidis(1996): first general implementation of
Relax, and Branch-if-necessary.

Brearly & Mitra(1975): IP preprocessing literature:
Linear case of interval analysis and constraint propagation.

Kearfott(1998): Interval analysis in nonlinear case.

Ugray, Lasdon, et. al.(2002) Multi-start to find good solution.

Gau: Implementation in LINDO API

Atlihan: Multi-start in LINDO API

LINDO SYSTEMS INC.

Getting a Good Initial Solution, Multi-start

Why? a) User wants a good solution quickly,
b) Do not waste time adding cuts far from optimum,
c) B&B has minimum number of nodes.

Basic Reference for multi-start: Ugray, Lasdon et. al.

For $i = 1$ to *ntrials*:

Randomly select a point, s_i, in n-space so that it is not in the neighborhood of any of preceding points.

Call conventional hill-climbing solver with point s_i as initial solution, giving a final solution f_i.

If solution f_i is best yet, store it.

Set the neighborhood of point f_i big enough to include s_i .

General Global Optimization Methodology

Uses the branch and bound approach popularized by McCormick, Sahinidis.

Two ideas:

1) For each(arbitrary) nonlinear function, given current bounds on variables, automatically generate a convex relaxation of the function. Solve the relaxed convexified model.

2) If solution to the relaxed problem is not feasible to the original model, then branch, i.e., partition the feasible region into two subregions. Calculate new implied bounds on the variables for each subproblem. Go back to (1).

Bound tightening, preprocessing, interval arithmetic, etc.

Why? Relaxations are tighter if bounds on variables are tighter.

Example for operators + and -:

Round 0: Given:

$$2x - y \geq 3; \quad -x + 2y \geq 3; \quad x, y \geq 0;$$

Round 1: Implies:

$$x \geq (3+0)/2 = 1.5; \quad y \geq (3+0)/2 = 1.5;$$

Round 2:

$$x \geq (3+1.5)/2 = 2.25; \quad y \geq (3+1.5)/2 = 2.25;$$

etc.

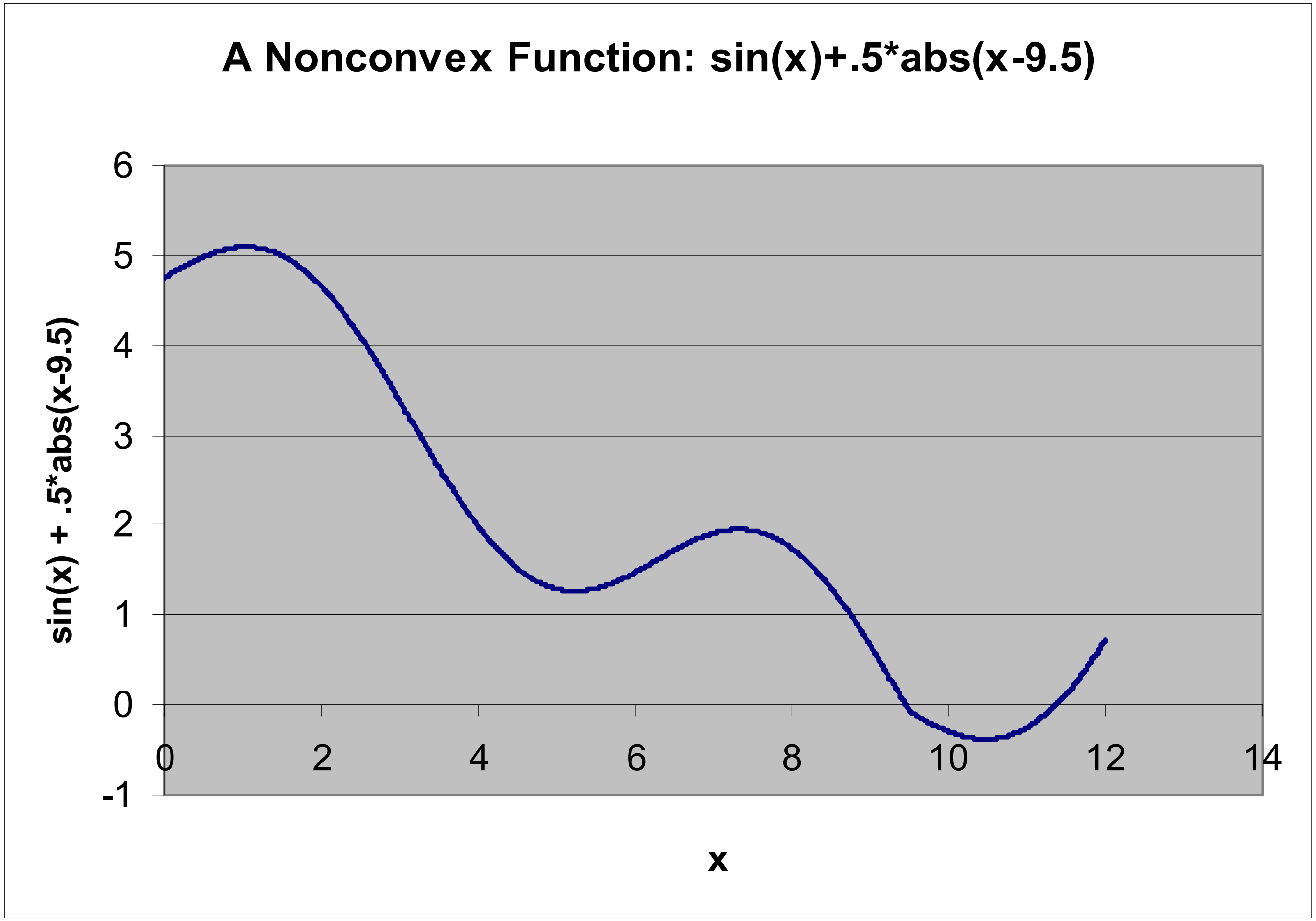
Need rules for stopping,

generalize for every operator supported.

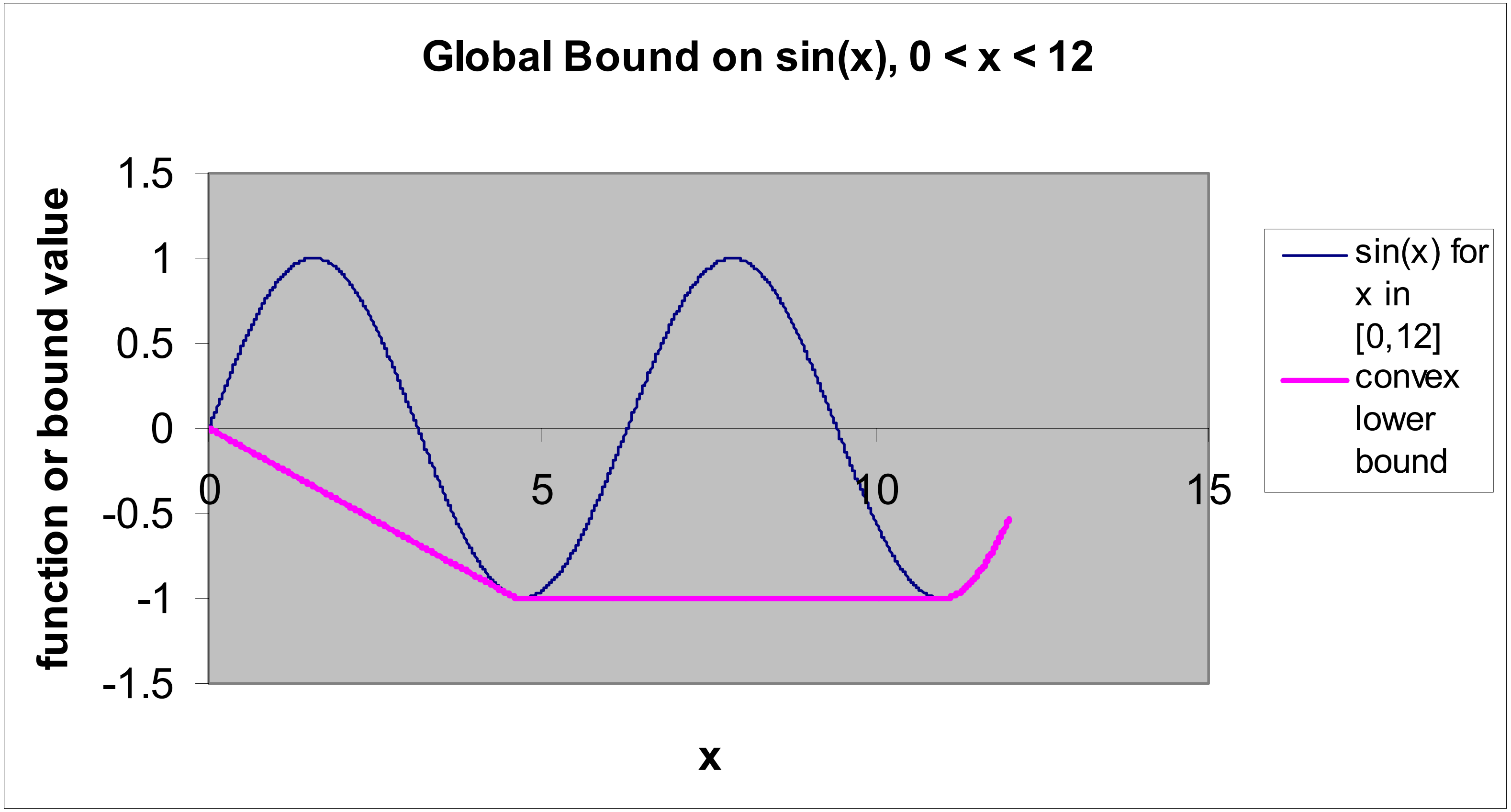
Creating a Convex Relaxation/Bound

Example: $Min = sin(x) + .5*abs(x\text{-}9.5);$

$s.t. \quad 0 \le x \le 12;$



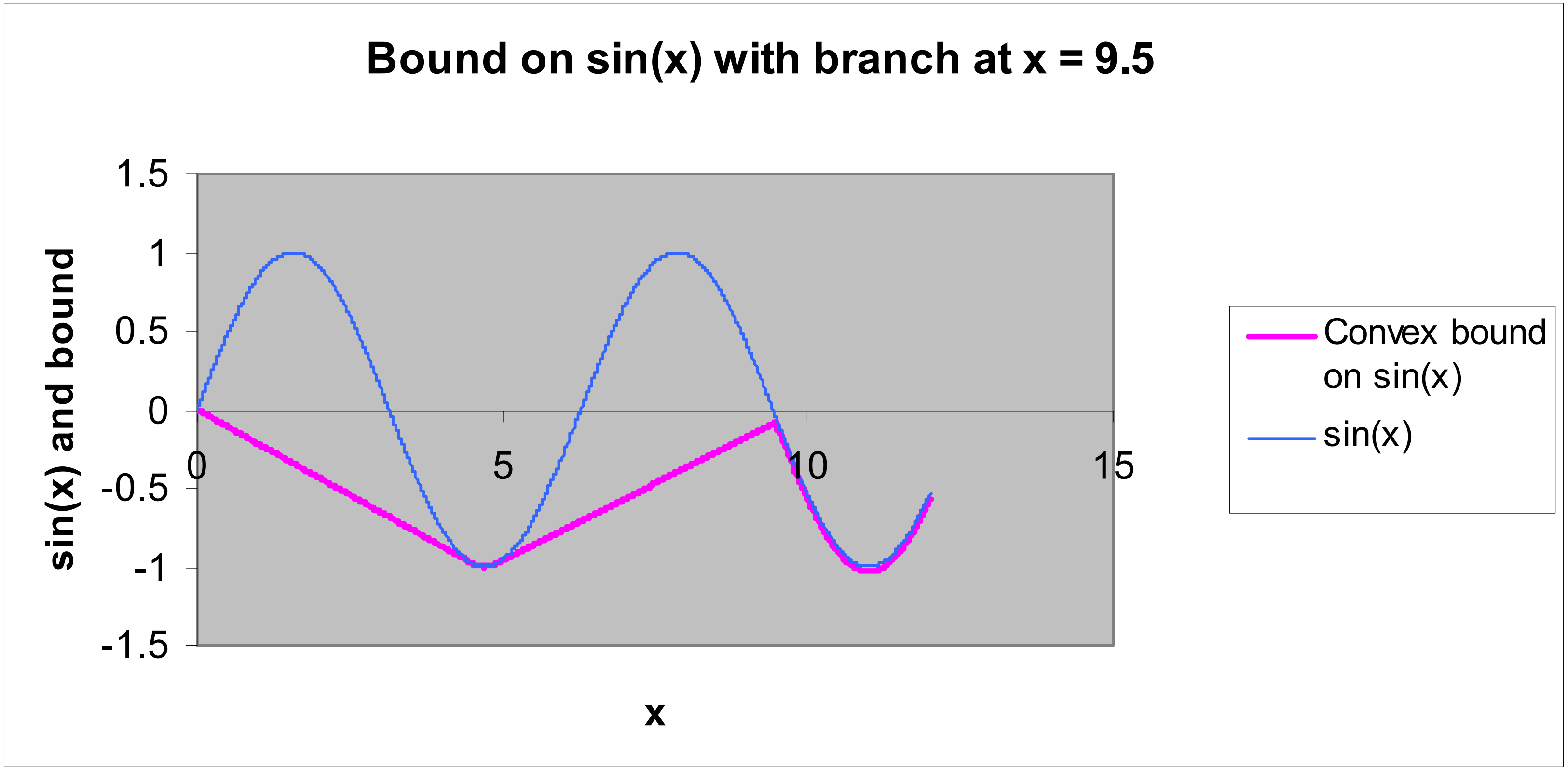
Bounding a Nonconvex Function



We replace sin( ) by its convex bound. Solve, get $x = 9.5$.

We branch on $x \le 9.5$ vs. $x \ge 9.5$ and re-bound.

The branch $x \ge 9.5$ is convex with solution $x = 10.47197$.



Bound discards $x \le 9.5$ branch, and we are done.

Linearization, Methodology

Some functions can be recognized and linearized exactly.

Let δ be a 0/1 variable. M = a big number.

Given:

a) $r = \max(x,y)$;

Linearization:

$r \geq x$; $r \geq y$; $r \leq x + \delta M$; $r \leq y + (1- \delta)M$;

b) $r = \text{abs}(x) = \max(x,-x)$;

c) $r = \min(x,y) = -\max(-x,-y)$;

Linearization continued.

d) $r = \mathrm{IF}(\delta, x, y);$

$$x - (1-\delta) M \le r \le x + (1-\delta) M;$$

$$y - \delta M \le r \le y + \delta M;$$

e) $r = \delta y;$

$$y - (1-\delta) M \le r \le y + (1-\delta) M;$$

$$r \le \delta M;$$

f) $xy = 0;$ (Complementarity)

$$-(1-\delta) M \le x \le (1-\delta) M;$$

$$-\delta M \le y \le \delta M;$$

LINDO SYSTEMS INC.

Global Optimization with IF(, ,) Function

A small text book example:

	A	B	C	D
1	EOQ Inventory with Quantity Discount			
2	All Units Case, C and M, Chapter 7			
3	Parameters			
4	120000	= D = demand/year		
5	100	= K = setup cost		
6	0.2	= i = interest charge		
7	Discount schedule			
8	Breakpoint	Cost/unit at or above this level		
9	0	3		
10	5000	2.96		
11	10000	2.92		
12	10000	= Q = amount to order		
13	Total cost/year=	$354,520.00	=(K*D/Q)+(i*Q/2+D)*IF(Q<A10,B9,IF(Q<A11,B10,B11))	

IF(, ,) Function and its Usefulness

IF(, ,) is convenient for representing quantity discount price schedules, using nested IF's.

A customer example:

7 discount levels,

13 suppliers,

361 SKU's

Resulted in model with

4646 rows and 7790 variables.

The model as it came from the user….

```
cost=IF(D3<'Rebate Structure'!$A$3,0,IF('Rebate Calculation'!D3<'Rebate Structure'!$A$4,'Rebate Structure'!D3*'Rebate Calculation'!D3,IF('Rebate Calculation'!D3<'Rebate Structure'!$A$5,'Rebate Structure'!D4*'Rebate Calculation'!D3,IF('Rebate Calculation'!D3<'Rebate Structure'!$A$6,'Rebate Structure'!D5*'Rebate Calculation'!D3,IF('Rebate Calculation'!D3<'Rebate Structure'!$A$7,'Rebate Structure'!D6*'Rebate Calculation'!D3,IF(D3<'Rebate Structure'!$A$8,'Rebate Structure'!D7*'Rebate Calculation'!D3,IF('Rebate Calculation'!D3<'Rebate Structure'!$A$9,'Rebate Structure'!D8*'Rebate Calculation'!D3,IF('Rebate Calculation'!D3<'Rebate Structure'!$A$10,'Rebate Structure'!D9*'Rebate Calculation'!D3)))))))
```

Choosing an Algebraic Representation/Reformulation

1) $x*x$ is converted to x^2 to get tighter convex relaxation;

2) More generally: $f_1(x*y) \geq 0$; $f_2(y*x) \geq 0$;

is converted to: $f_1(w) \geq 0$; $f_2(w) \geq 0$; $w = x*y$;

3) $x*(y-x)$ vs. $x*y - x$^2;

One may be better for tight intervals, the other for a tight relaxation.

Careful Rounding and Preprocessing

“Careful”, though not rigorous rounding is used in LINGO/LINDO API.

Example: Arnold Neumaier’s problem, may be difficult to solve accurately for some solvers. LINGO solves to optimality in 0 secs.

```
n = 20;

min = - x(n);

   (s+1)*x(1) - x(2) >= s-1;
  -s*x(n-2) -(3*s-1)*x(n-1) + 3 *x(n) >= -(5*s-7);

 @for( point(i)| i #gt# 1 #and# i #lt# n:
     -s*x(i-1) +(s+1)*x(i) - x(i+1) >=((-1)^i)*(s+1)
      );

  @for( point(i)| i #le# 13:
    @bnd( 0, x(i), 10)
      );
  @for( point(i)| i #gt# 13:
    @bnd( 0, x(i), 1000000)
      );
  @for( point(i):
    @gin(x(i))
      );
```

LINDO SYSTEMS INC.

Some solvers have difficulty finding a correct solution to this problem with 6 variables and 1 constraint;

```
! (bigsum01)  Obj = -540564, LINGO time = .2 secs.;
  MIN = -  81 * X_1 - 221 * X_2 - 219 * X_3
        - 317 * X_4 - 385 * X_5 - 413 * X_6;

    12228 * X_1 + 36679 * X_2 + 36682 * X_3
  + 48908 * X_4 + 61139 * X_5 + 73365 * X_6 = 89716837;

 @GIN( X_1); @GIN( X_2); @GIN( X_3);
 @GIN( X_4); @GIN( X_5); @GIN( X_6);
 @BND( 0, X_1, 99999); @BND( 0, X_2, 99999);
 @BND( 0, X_3, 99999); @BND( 0, X_4, 99999);
 @BND( 0, X_5, 99999); @BND( 0, X_6, 99999);
```

How to Input Nonlinear Programs?

A) Through a file:

1) LINGO Script:

Execute `runlingo` *`scriptfile`*`.lng`

2) Low level RPN notation:

Execute `runlindo` *`modelfile`*`.mpi.`

B) Through memory:

1) LINGO Script:

```
nError= LSexecuteScriptLng( pLINGO, cScript);
```

2) Low level RPN notation:

```
nError= LSloadInstruct( pModel,…,codelist,…);
```

What Does an RPN Codelist(.mpi) Look Like?

```
! minimize =  x*sin(x*pi) + 10
! subject to
!                      x  -  10 <= 0;
BEGINMODEL  XSINXPI
VARIABLES
    X0001  8.0  0.0  10.0  C
OBJECTIVES
  XSINXPI LS_MIN
    EP_PUSH_VAR       X0001
    EP_PUSH_NUM       3.1415926
    EP_MULTIPLY
    EP_SIN
    EP_PUSH_VAR       X0001
    EP_MULTIPLY
    EP_PUSH_NUM      10.0
    EP_PLUS
CONSTRAINTS
  ROW1 L
    EP_PUSH_VAR       X0001
    EP_PUSH_NUM      10.0
    EP_MINUS
ENDMODEL
```

Performance on Continuous NLPs

- A suite of 60 continuous NLPs arising in different applications
  - Nonlinear Least Squares Regression
  - Inventory Management and Network Flows
  - Chemical Processes
  - Engineering Design (constrained polynomials etc…)
- NLP Model Sizes
  - (Min - Max) Constraints: (0 - 576)
  - (Min - Max) Variables: (1 - 518)

Performances on Continuous NLPs (Cont.)

- Server Specs (P4, 1.4 GHz, 2G RAM, NT4)
- Seconds required to solve the entire suite
  - Global solver: 1789 secs
  - Multi-Start solver: 333 secs
  - Single-Start solver: 11 secs
- Proving global optimality takes more time.
- Multi-starts help finding improved solutions
- Single-start is the fastest but solution quality is compromised.

Performance on Continuous NLPs (Cont.)

The Global Solver

found provably optimal solutions for 58 problems
proved infeasibility for 2 problems

The Multi-Start Solver (with 5 multi-starts)

obtained the global optima in 39 out of 58 problems.
failed to find a feasible solution in 4 out of 58 problems.
found better solution than single-start in 11 out of 19 problems.

Single-Start Solver

obtained the global solution in 30 out of 58 problems.
failed to find a feasible solution in 5 out of 58.

Performances on Mixed-Integer NLPs

- A suite of 50 NLPs with integer variables.
- Model Sizes
    - (Min-Max) Constraints: (1 - 113)
    - (Min-Max) Variables: (1 - 131)
- Global Solver
    - found provably global optima for 50 out of 50 problems. (Total time: 2475 secs.)
- Multi-Start Solver
    - performed 2 multi-starts at every node in B&B tree.
    - obtained global optima in 42 out of 50 problems (Total time: 404 secs).
    - no feasible solutions for 3 out of 50 problems.

Some Recent Example Problems:

Problem	Constraints	Vars	NLvars	Intvars
test15-global	959	2492	764	192

*Application: power plant operation. Originally took 15 hours, now takes 2 hours to global optimum. Types of nonlinearities: x*y, x^k, abs(x), IF( )*

References

Adjiman, C.S., S. Dallwig, C.A. Floudas, and A. Neumaier, "A global optimization method, αBB, for general twice-differentiable constrained NLPs-I: theoretical aspects", *Computers and Chemical Engineering*, vol 22, no. 9, pp. 1137-1158, 1998.

Adjiman, C.S., I.P. Androulakis, and C.A. Floudas, "A global optimization method, αBB, for general twice-differentiable constrained NLPs-II: Implementation and computational results", *Computers and Chemical Engineering*, vol 22, no. 9, pp. 1159-1179, 1998.

Babichev, A. B., O. B. Kadyrova, T. P. Kashevarova, A. L. Semenov. : *UniCalc as a Tool for Solving Problems with Inaccurate and Sub-definite Data. Interval Computations*. 3: 13 – 17, 1992.

Floudas, C.A., P.M.Pardalos, C.S. Adjiman, W.R. Esposito, Z.H.GUmus, S.T. Harding, J.L. Klepeis, C.A. Meyer, and C.A. Schweiger, *Handbook of Test Problems in Local and Global Optimization*, Kluwer Academic Publishers, 1999.

Floudas, C.A., "*Determinisitc Global Optimization: Theory, Methods, and Applications*", Kluwer Academic Publishers, 2000.

Gau, C. –Y., and M. A. Stadtherr, "Deterministic Global Optimization for Error-in-Variables Parameter Estimation" *AICHE J*., 48: 1192 – 1197, 2002.

Hansen, E. R.. *Global Optimization Using Interval Analysis*. Marcel Dekker, New York, NY, 1992

Horst, R., and H. Tuy, "*Global Optimization: Deterministic Approaches*," 2nd ed., Springer-Verlag, Berling, 1993.

Kearfott, B.(1998), “On proving existence of feasible points in equality constrained optimization problems”, *Mathematical Programming,* vol. 83, pp. 89-100.

McCormick, G. P. “Computability of Global Solutions to Factorable Nonconvex Probgrams: Part I – Convex Underestimating Problems”, *Mathematical Programming*. Vol. 10, pp. 147–175, 1976 .

Moore. *Interval Analysis*. Prentice-Hall, Engelwood Cliffs, NJ, 1966.

Neumaier, A.. *Interval Methods for Systems of Equations*. Cambridge University Press, Cambridge, 1990.

Rockafellar, R. T. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

Sahinidis, N.(1996), “BARON: A General Purpose Global Optimization Software Package, *Journal of Global Optimizaiton*, vol. 8, pp. 201-205.

Tawarmalani, M. and N. V. Sahinidis, *Convexification and Global Optimization in Continuous and Mixed-Integer Nonlinear Programming: Theory, Algorithms, Software, and Applications*, Kluwer Academic Publishers, Dordrecht, 2002.

Ugray, Z., L. Lasdon, J. Plummer, F. Glover, J. Kelly, and R. Marti(2002), “A Multistart Scatter Search Heuristic for Smooth NLP and MINLP Problems”, Tech. report, U. Texas.