Mixed-Integer Expensive Global Optimization with TOMLAB

Professor Kenneth Holmström

Department of Mathematics and Physics
Mälardalen University
P.O. Box 883, SE-721 23 Västerås, Sweden

Tomlab Optimization AB
Västerås Technology Park, Sweden
{E-mail: hkh@tomlab.biz}

Talk at the Informs Annual Meeting, Denver, October 25, 2004

Abstract

A mixed-integer constrained extension of the RBF algorithm for computationally costly global optimization is presented. Implementation in TOMLAB (http://tomlab.biz) solver rbfSolve is discussed. The algorithm relies on MINLP sub solvers in TOMLAB, currently OQNLP or DIRECT (glcFast or glcSolve). Results for real-life and artificial test problems are presented.

Outline of the talk:

- Introduction
- The Mixed-Integer Expensive (Costly) Global Problem
- Solvers and software for Costly MINLP
- Algorithms for Costly MINLP
- The Radial Basis Function (RBF) Algorithm in more detail
- Test Results
- Conclusions and Further Work
- Tests on a very costly real-life problem

The Mixed-Integer Expensive (Costly) Global Black-Box Non-Convex Problem

$$
\begin{array}{llcccl}
\min\limits_{x} & f(x) & & & & \\
 & -\infty < & x_L & \leq & x & \leq & x_U & < \infty \\
s/t & & b_L & \leq & Ax & \leq & b_U & \\
 & & c_L & \leq & c(x) & \leq & c_U, & x_j \in \mathbb{N} \quad \forall j \in \mathrm{I},
\end{array}
\tag{1}
$$

where $x, x_L, x_U \in \mathbb{R}^d, \quad f(x) \in \mathbb{R}, \quad A \in \mathbb{R}^{m_1 \times d}$,
$b_L, b_U \in \mathbb{R}^{m_1}$ and $\quad c_L, c(x), c_U \in \mathbb{R}^{m_2}$.
The variables x_I, $x \in I$, an index subset of $1, ..., d$,
are restricted to be integers.

Expensive (Costly),Black-Box, Non-Convex Global Optimization

- $f(x)$ takes considerable CPU time, e.g. 30 minutes to compute.
- $f(x)$ is often a complex computer program,
  or the result of an advanced simulation,
  e.g. CFD, tuning of trading strategies, design optimization.
- Derivatives are most often hard to obtain. $f(x)$ is often noisy or nonsmooth.
- The constraints are considered cheap.
- Any costly constraints are included in the objective with a penalty.

Define a penalty type of objective e.g. as

$$
\min_{x} \quad p(x) = f(x) + \sum_{i} w_i \max\left(0,\ c_i(x) - b_{Ui} \quad b_{Li} - c_i(x)\right) \tag{2}
$$

- Costly constraints could be treated similar as the objective, building interpolating surfaces. Some tests made, seems to work.

Basic Algorithm

- Find initial set of $n > d + 1$ sample points x.
  Experimental design problem.
- Compute costly f(x) for initial set of n points.
- Iteration until target f(x) achieved, $n > n_{nmax}$ or no time left.
  - Use the n sampled points to build a smooth interpolation model (surrogate model, response surface model) as an approximation of f(x) surface.
  - Optimize a cheap function of the approximating surface (relative to the costly $f(x)$) to obtain a new trial point to compute the costly f(x) for.
  - Compute and validate new $(x, f(x))$, increase n.
- Save and return all information to enable warm start, after user evaluation.

Software in the TOMLAB Optimization Environment

TOMLAB /CGO for Mixed-Integer Constrained Expensive Global Optimization

- *ego* implements the Efficient Global Optimization (EGO) algorithm by Jones, Schonlau, Welch (1998).
- *rbfSolve* implements the Radial Basis Function (RBF) algorithm by Gutmann and Powell (1998-2001), and Björkman and K. Holmström [2000]. Efficient implementation in MATLAB / FORTRAN using MEX file interfaces.

Time limited evaluation available at http://tomlab.biz

The Radial Basis Algoritm (RBF) in rbfSolve

- Interpolation of all sampled point by radial basis function interpolation.
- Cycle of target values on surface gives trade off between local and global search.
- Target value achievement balanced against size of new interpolation coefficient - gives well-conditioned interpolation matrix as size increases.
- Efficient numerical implementation for cubic and thin plate spline RBFs.
- Global convergence proof (Gutmann 2001).

The EGO Algorithm

- Builds stochastic process model DACE (Design and Analysis of Computer Experiments).
- Parameters are chosen to maximize the likelihood of the sample.
- New point maximize expected improvement (supposed to be the best linear unbiased predictor).
- Weakness: The predictor and its mean squared error are derived under the assumption that the parameters are known, but they are only estimated.
- Problem: The correlation matrix gets ill-conditioned as the number of points grow.

TOMLAB solver *ego* uses same file format and similar input format as $rbfSolve$.
The solvers can be do a warm start and continue where the other stopped.

Initial Point strategies in TOMLAB /CGO

1. Latin Hybercube for constrained problems (CLH).
   Generate M (large number) of trial points.
   Values for integer variables are generated randomly.
   Evaluate up to M to generate d+1 feasible points (d=dim(x))

2. Standard Latin Hybercube (LH).
   Generate M initial points, dependent on dimension d.
   Values for integer variables are generated randomly.

3. Gutmann.
   Selection of d+1 corner points, optionally adding middle point.

4. Corner points.
   Generation of 2^d corner points, optionally adding middle point.

5. Ellipsoid strategy, dependent on percentage input p
   Random strategy, the percentage value p gives the percentage size of an ellipsoid around the so far sampled points that the new points are not allowed in.
   Range $1\% - 50\%$. Recommended values $10\% - 20\%$.

6. User given set of initial points

7. Warm start with previous run with either *ego* or *rbfSolve*.

Convergence, stopping criteria

- Number of function evaluations
- Maximal CPU time exceeded
- fGoal reached within relative accuracy
- MaxCycle global-local cycles without progress (rbfSolve)

Relies on sub solvers that can solve MINLP problems

Current MINLP sub solver techniques

- glcFast (constrained black-box MINLP algorithm DIRECT).
  FORTRAN, with MATLAB MEX.
  Local search with SNOPT/NPSOL with fixed integer values
- glcSolve (constrained black-box MINLP algorithm DIRECT),
  MATLAB code.
  Local search with SNOPT/NPSOL with fixed integer values
- glcCluster (glcFast, SNOPT/NPSOL, and clustering algorithm)
- OQNLP in TOMLAB /OQNLP
- MINLPBB in TOMLAB /MINLP

glcCluster - clustering algorithm for global nonconvex (MIP) constrained programming

1. Run one pass with m_1 function evaluations for the DIRECT algorithm in glcFast.
   If not feasible, run additional passes with m_1 function evaluations until in total m_2 function evaluations are tried.
2. Analyze the sampled points in glcFast with a clustering algorithm.
   Generate k point clusters.
3. Start with the best point in each of the k clusters.
   Use a local solver (e.g. NPSOL or SNOPT) to solve the problem to optimality, finding local minima.
   Any integer variables are fixed during the local search.
4. Generate new input to glcFast, with the new best point found.
5. Run one pass with m_2 function evaluations in glcFast,
   using a warm start.
6. If glcFast improved the best point found:
   Start a local search with e.g. NPSOL or SNOPT.
   Use the new best point(s) as starting point(s).

glcCluster shows good results on higher-dimensional problems,
up to 20 variables.

EGO on Shekel's Foxholes

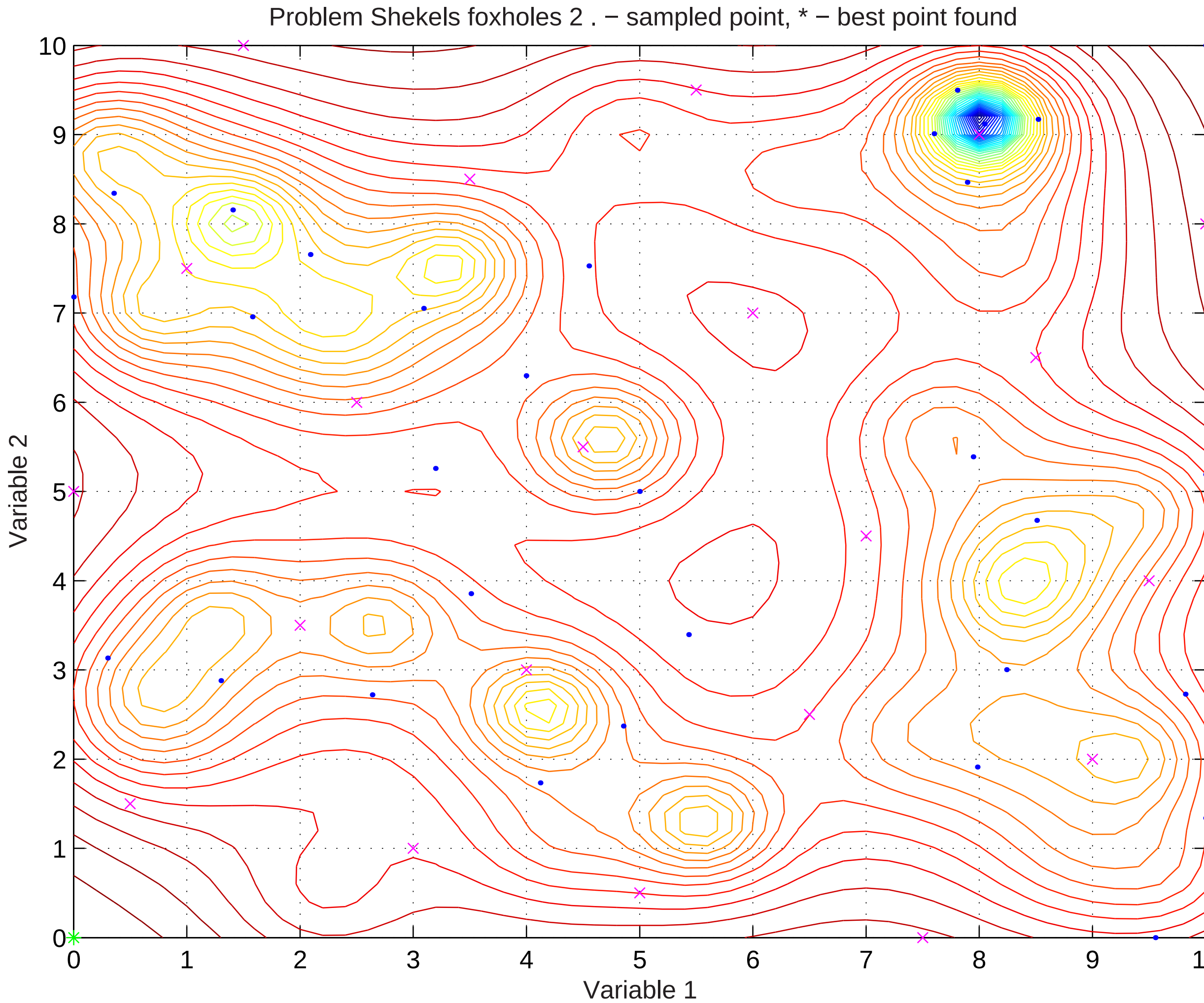

Figure 1: Contour and sampled point for EGO on Shekel's foxholes. Initial set 21 points, marked with x. In total 52 function evaluations.

rbfSolve on Shekel's Foxholes

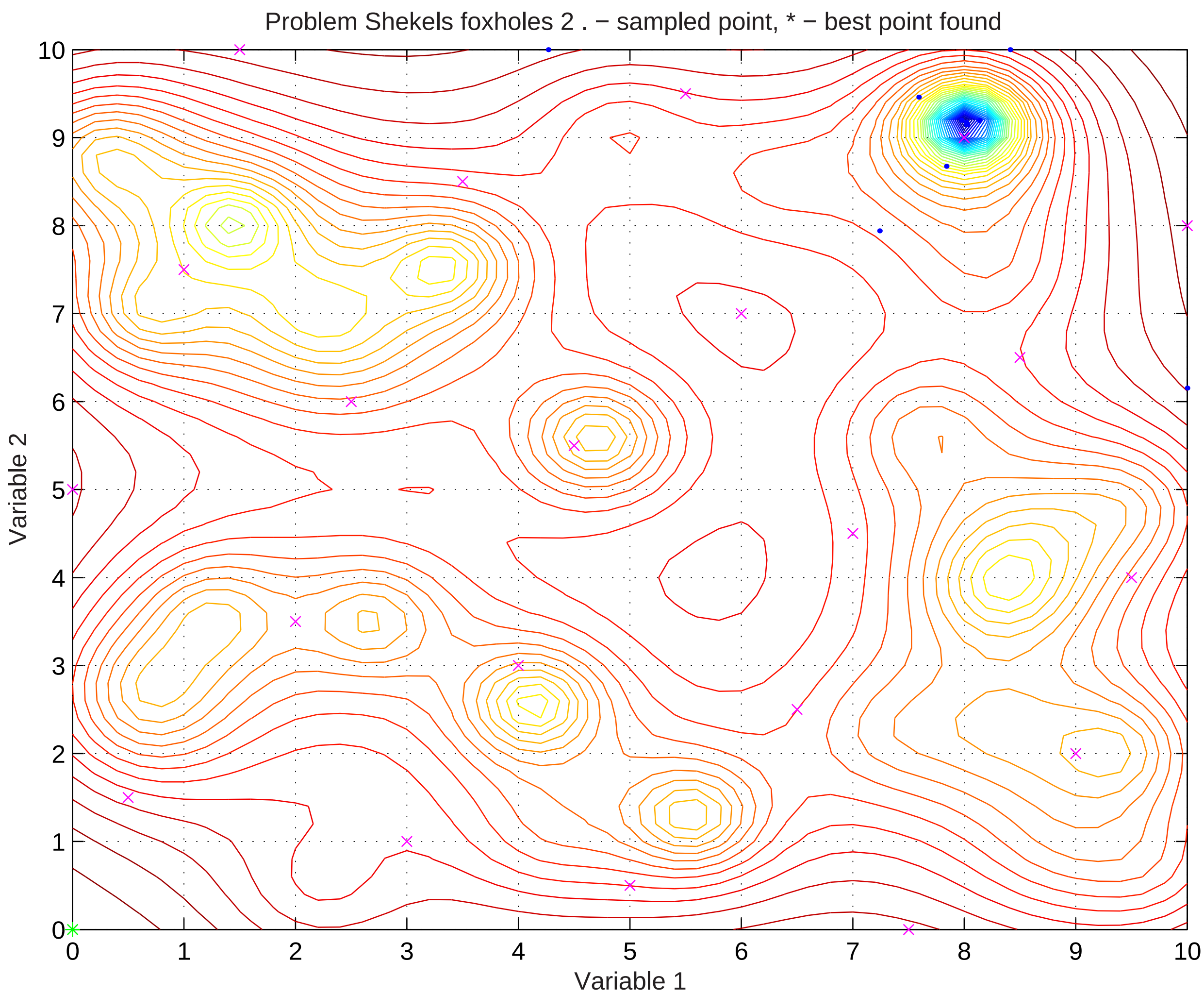

Figure 2: Contour and sampled point for rbfSolve on Shekel's foxholes. Initial set 21 points, marked with x. In total 52 function evaluations.

rbfSolve on Shekel's Foxholes

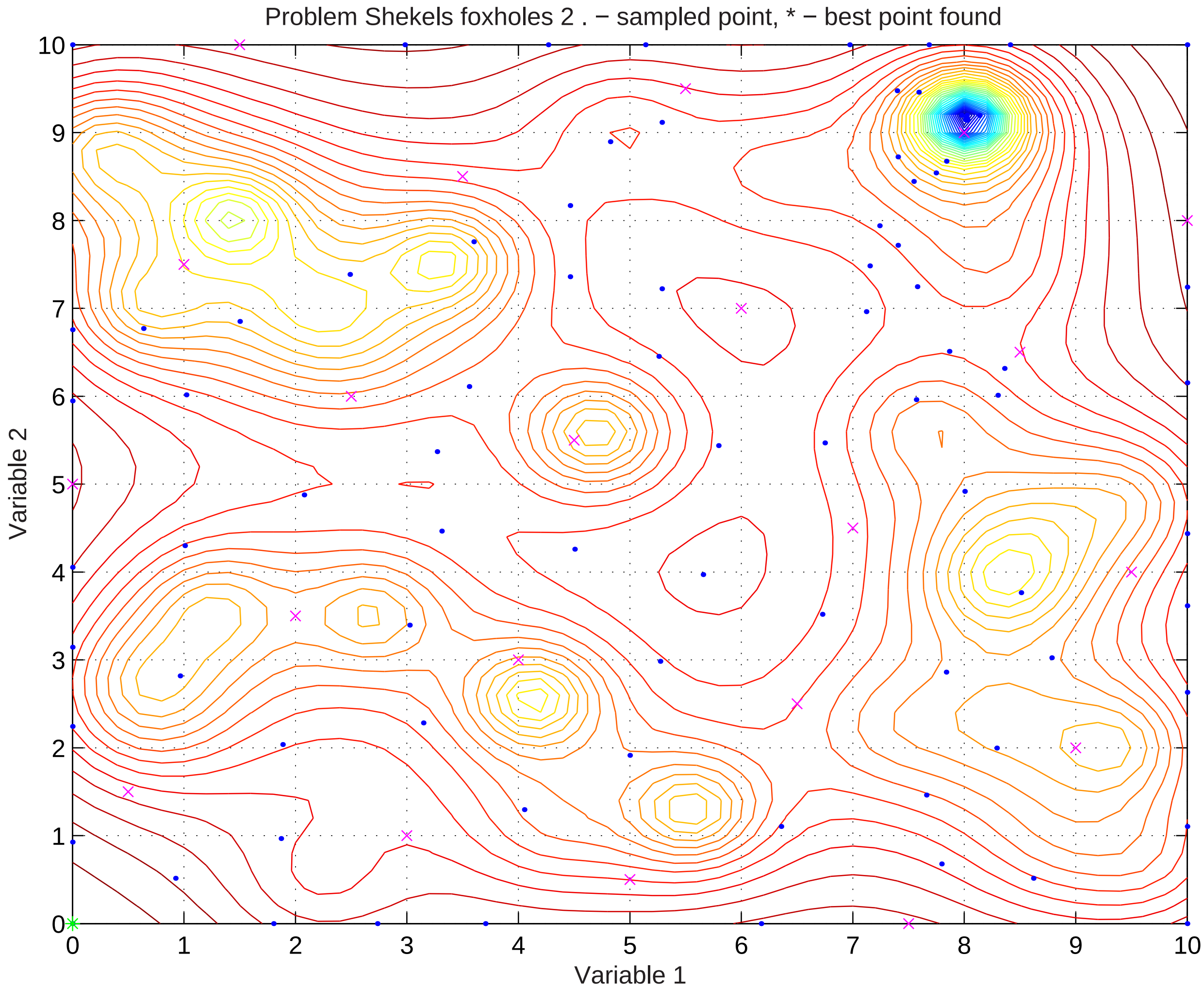

Figure 3: Contour and sampled point for rbfSolve on Shekel's foxholes. Initial set 21 points, marked with x. In total 200 function evaluations.

rbfSolve on Goldstein and Price - 40 points

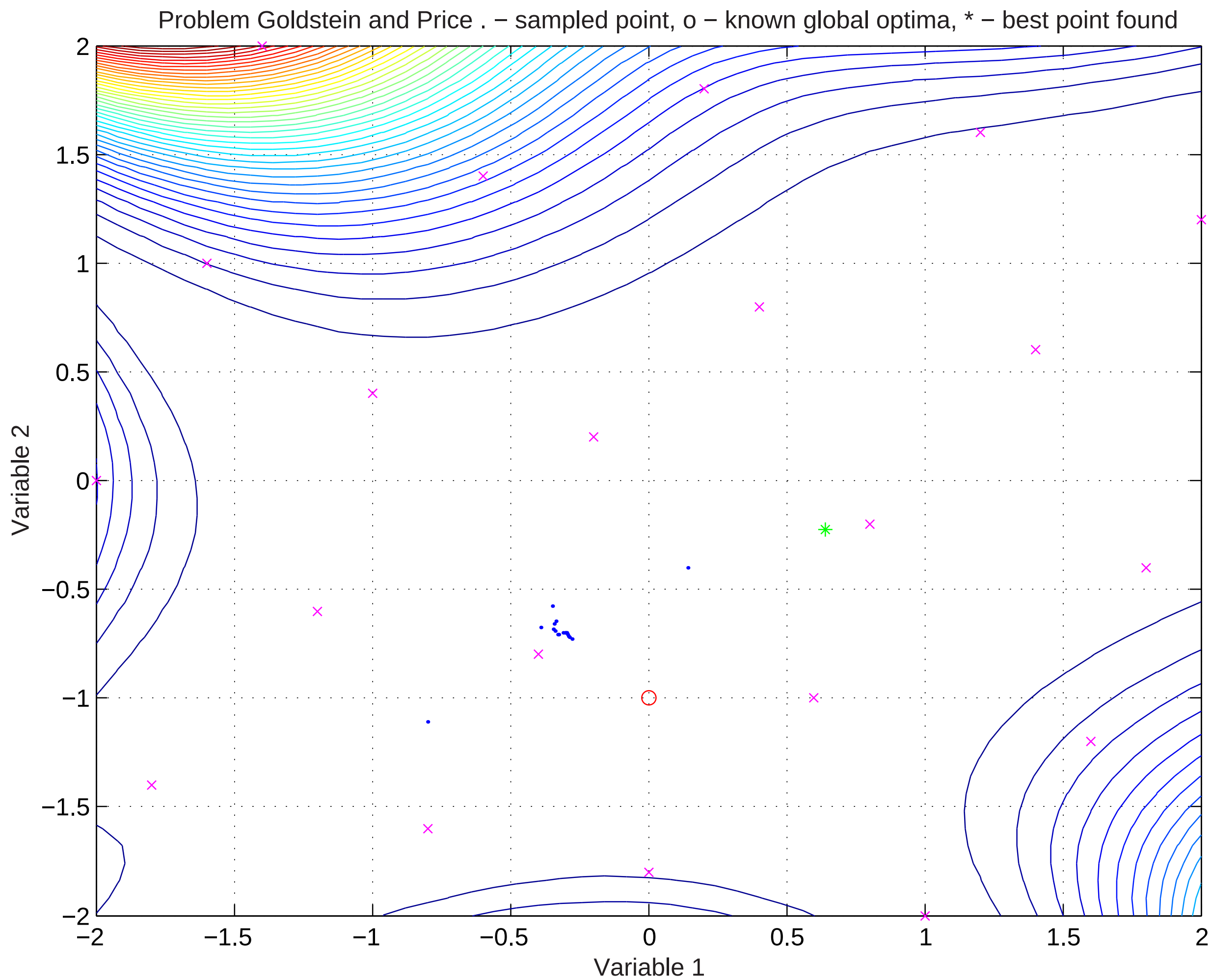

Figure 4: Contour and sampled point for rbfSolve on Goldstein and Price. Initial set 21 points, marked with x. In total 40 function evaluations.

rbfSolve on Goldstein and Price - 60 points

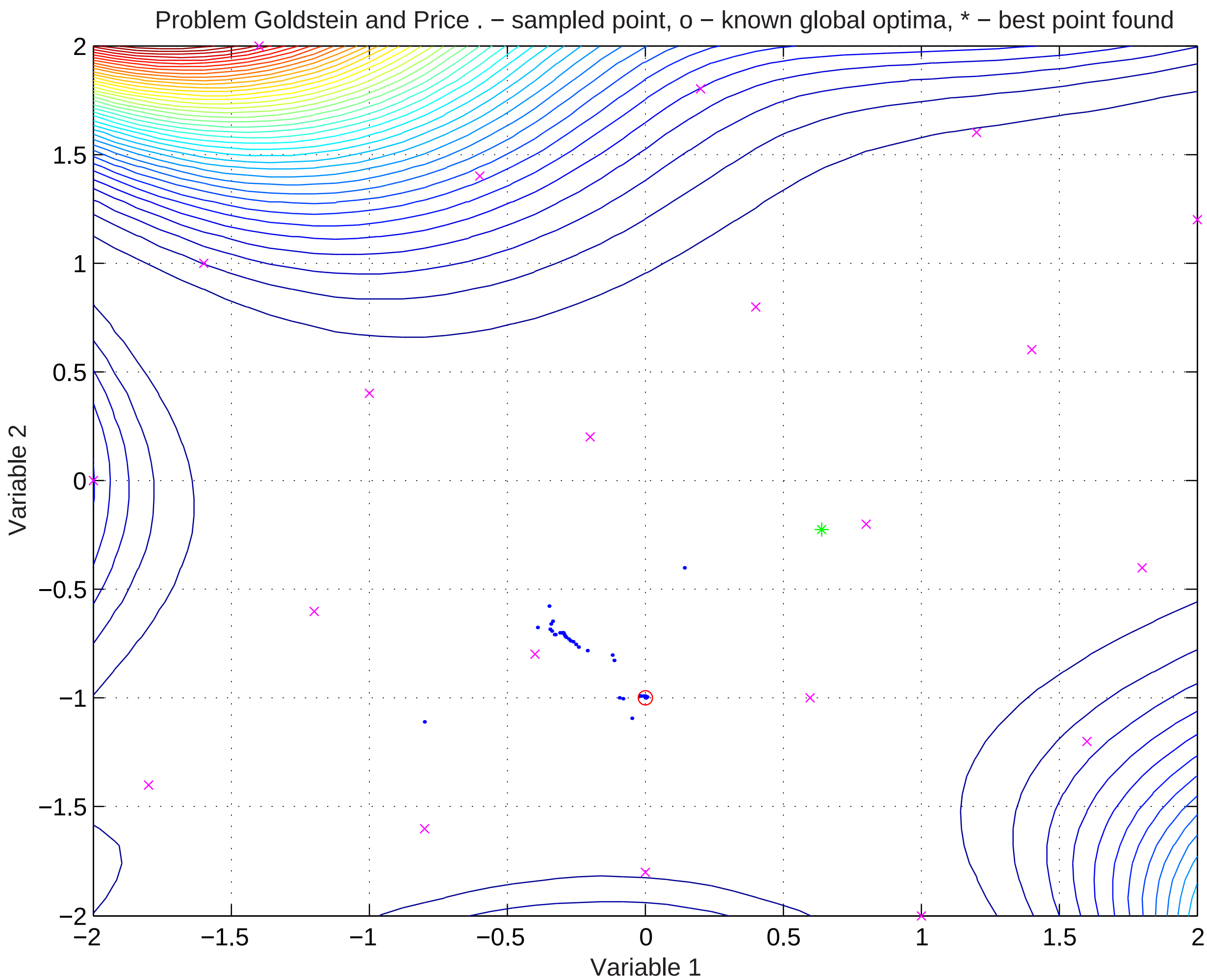

Figure 5: Contour and sampled point for rbfSolve on Goldstein and Price. Initial set 21 points, marked with x. In total 60 function evaluations.

Radial Basis Function (RBF) interpolation

$$s_n(x) = \sum_{i=1}^{n} \lambda_i \phi\left(\|x - x_i\|_2\right) + p(x), \tag{3}$$

with $\lambda_1, ..., \lambda_n \in \mathbb{R}$, $x_i \in \mathbb{R}^d$, and $p(x)$ $m-$degree polynomial.

RBF	$\phi(r) > 0$	$p(x)$	m
cubic	r^3	$a^T \cdot x + b$	1
thin plate spline	$r^2 \log r$	$a^T \cdot x$ + b	1
linear	r	b	0
multiquadric	$\sqrt{(}r^2 + \gamma^2)$		-
Gaussian	$\exp(-\gamma r^2)$		-

We consider *cubic* and *thin plate spline* in the following.

RBF Interpolation

$$\begin{pmatrix} \Phi & P \\ P^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ c \end{pmatrix} = \begin{pmatrix} F \\ 0 \end{pmatrix}, \tag{4}$$

where Φ is the $n \times n$ matrix with $\Phi_{ij} = \phi\left(\|x_i - x_j\|_2\right)$ and

$$P = \begin{pmatrix} x_1^T & 1 \\ x_2^T & 1 \\ . & . \\ . & . \\ x_n^T & 1 \end{pmatrix}, \lambda = \begin{pmatrix} \lambda_1 \\ \lambda_2 \\ . \\ . \\ \lambda_n \end{pmatrix}, c = \begin{pmatrix} b_1 \\ b_2 \\ . \\ . \\ b_d \\ a \end{pmatrix}, F = \begin{pmatrix} f(x_1) \\ f(x_2) \\ . \\ . \\ f(x_n) \end{pmatrix}. \tag{5}$$

If $rank(P) = d + 1$ then the matrix is nonsingular, i.e.
a unique solution exists (Powell 1992).

Powell (1999) shows that it is the "smoothest" function s from a linear space that satisfies the interpolation conditions

$$s(x_i) = f(x_i), \quad i = 1, ..., n. \tag{6}$$

The smoothest radial basis interpolation is obtained by minimizing the semi-norm

$$s_n = arg\min_s \ < s, s > \tag{7}$$

Objective Function (Idea 1)

First compute local solution to

$$\min_{y\in\Omega} s_n(y). \tag{8}$$

Define target value

$$f_n^* \in \left(-\infty, \min_{y\in\Omega} s_n(y)\right]. \tag{9}$$

Perform a cycle $k = 0, ..., 5$ and choose f_n^* as

$$f_n^*(k) = \min_{y\in\Omega} s_n(y) - w_k \cdot \left(\max_i {}^1 f(x_i) - \min_{y\in\Omega} s_n(y)\right), \tag{10}$$

with

$$w_k = (1 - 0.2 \cdot k)^2 \tag{11}$$

Define

$$g_k(y, n, s_n(y)) = \mu_n(y)\left[s_n(y) - f_n^*(k)\right]^2 \tag{12}$$

The coefficient $\mu_n(y)$ is the new λ-coefficient if the trial y would be included in the RBF interpolation (i.e. λ_{n+1}).

[1] $\max_i f(x_i)$ is **not** taken over all points, instead:
Largest $n - n_{max}$ function values are removed, n_{max} :
if $(n - n_{init})\text{mod}(N+1) = 0$, then $n_{max} = n$,
otherwise $n_{max} = \max\{2, n_{max} - \text{floor}((n - n_{init})/N)\}$.

Other Algorithmic Options

- Scale x to unit hypercube.
  Avoid ill-conditioning in linear equation system.

- Reduce influence of large $f(x)$,
  $f_i = \min(median(f(X)), f_i), i = 1, ..., n$

- Choice of RBF function, cubic or thin plate spline.

- Removal of close points, keeping the best.

- Update of interpolation matrix for each new costly $f(x)$:

  1. Update Cholesky and QR factors.
  2. Update LU factorization.
  3. Recompute full LU with pivoting.
     Current choice, gives maximal accuracy.

- Computation of $\mu_n(y)$ for many y

  1. Solution of full system for each y.
  2. Smart update of Cholesky and QR factors, and solve.
  3. Use LU factorization, doing block-LU trick (current choice).

- Optional infStep in global-local cycle.
  The infStep sets $f_n^*(k) = -\infty$, i.e. the objective is

$$g_k(y, n, s_n(y)) = \mu_n(y) s_n(y) \tag{13}$$

Tests on mixed-integer nonlinear problems

Problem	d	x_I	Ax	Ax =	$c(x)$	$c(x)$ =	Domain
Floudas-Pardalos 12.2TP1	5	3	3	0	2	2	$[0]^5 - [1, 1, 1, 1.5, 1.6]$
Floudas-Pardalos 3.4TP3	6	2	3	0	2	0	$[0, 0, 1, 0, 1, 0] - [6, 6, 5, 6, 5, 10]$
Kocis & Grossmann 1998	5	3	3	0	2	2	$[0, 10^{-8}, 0, 0, 0] - [10^8, 10^8, 1, 1, 1]$
Floudas 1995 6.6.5	3	1	2	0	1	0	$[0.2, -2.22554, 0] - [1, -1, 1]$
Kocis & Grossmann 1989	4	2	1	0	4	0	$[0, 0, 0, 0] - [10, 20, 1, 1]$
Pörn et al. 1997	2	2	3	0	1	0	$[1, 1] - [5, 5]$
Kesavan et al. 2004 D	5	3	3	1	1	0	$[0, 0, 0, 1, 1] - [1, 1, 1, 10, 6]$
Floudas-Pardalos 12.2TP3	7	4	5	0	4	0	$[0]^7 - [1.2, 1.8, 2.5, 1, 1, 1, 1]$
Floudas-Pardalos 12.2TP5	2	2	3	0	1	0	$[1, 1] - [5, 5]$
Floudas-Pardalos 12.2TP6	2	2	2	0	1	0	$[1, 1] - [10, 6]$
Floudas-Pardalos 12.2TP4	11	8	4	0	3	3	$[0]^{11} - [1]^{11}$

Number of function evaluations needed to reach: $\frac{f_{best} - f_{global}}{|f_{global}|} \leq 1$ %

Initial Strategy	CLH	CLH	CLH	LH	LH	GUT	CLH	CLH	EGO
Cycle Rule, objective	0	1	2	0	1	1	0	1	
Sub solver	Clust	Clust	Clust	Clust	Clust	Clust	GLC	GLC	oqnlp
Floudas-Pardalos 12.2TP1	7	7	7	52	52	8	7	7	9
Floudas-Pardalos 3.4TP3	77	29	26	139	74	50	69	43	89
Kocis & Grossmann 1998	13	11	13	52	52	8	8	15	
Floudas 1995 6.6.5	12	12	12	53	40	9	7	7	
Kocis & Grossmann 1989	104	50	66	94	72	67	9	11	
Pörn et al. 1997	4	4	4	4	4	5	4	4	
Kesavan et al. 2004 D	23	14	14	52	58	10	9	9	
Floudas-Pardalos 12.2TP3	177	93	83	200	200	10	200	200	18
Floudas-Pardalos 12.2TP5	4	4	4	4	4	5	4	4	
Floudas-Pardalos 12.2TP6	5	4	4	24	26	5	5	4	
Floudas-Pardalos 12.2TP4	154	91	86	142	127	15	-	-	

Conclusions

- The use of radial basis interpolation methods for costly (expensive) mixed-integer nonlinear optimization problems is promising.
- The method relies completely on sub solvers being able to solve a large number of MINLP problems. The sub solvers should not spend too much time on each problem, otherwise the overall performance will be very slow.
- The global black box solvers handle the sub problems well, if there is not too complicated constraints involving special combination of integers to obtain feasiblity. For such cases a branch-and-bound or other approach for the MINLP solution is needed.
- The results are sensitive to the choice of initial set of sample values. The best choice seems to be to evaluate the non-costly constraints for enough points to obtain a feasible initial set of points to evalute the costly objective for.

Further Work

- Develop methods to deal with costly constraints.
- Develop strategies to deal with failure in the sub problem solvers,
- Improve the local-global target value cycle strategy, especially for flat functions. Add distance constraints in the global-local cycle strategy.
- Parallel computation of every cycle. Update RBF interpolation with a cluster of points.

E-mail: hkh@tomlab.biz

TOMLAB home page:

http://tomlab.biz

References

[1] Mattias Björkman and K. Holmström. Global Optimization of Costly Nonconvex Functions Using Radial Basis Functions. *Optimization and Engineering*, **1**4, 373–397, 2000.

[2] Powell, M. J. D.: 1999, 'Recent Research at Cambridge on Radial Basis Functions'. In: M. D. Buhmann, M. Felten, D. Mache, and M. W. Müller (eds.): *New Developments in Approximation Theory*. Basel: Birkhäuser, pp. 215–232.

[3] Hans-Martin Gutmann. On the semi-norm of radial basis function interpolants. *Journal of Approximation Theory*, 2001.

[4] Hans-Martin Gutmann. A radial basis function method for global optimization. *Journal of Global Optimization*, 19:201–227, 2001.

[5] Rommel G. Regis and Christine A. Shoemaker. Efficient Constrained Greedy Response Surface Methods for Global Optimization. *Journal of Global Optimization*, accepted

[6] Holmström, K.: 1999b, 'The TOMLAB Optimization Environment in Matlab'. *Advanced Modeling and Optimization* **1**(1), 47–69.

[7] Dixon, L. C. W. and G. P. Szegö: 1978, 'The Global Optimisation Problem: An Introduction'. In: L. Dixon and G. Szego (eds.): *Toward Global Optimization*. New York, pp. 1–15.

[8] Jones, D. R., M. Schonlau, and W. J. Welch: 1998, 'Efficient Global Optimization of Expensive Black-Box Functions'. *Journal of Global Optimization* **13**, 455–492.

[9] Huyer, W. and A. Neumaier: 1999, 'Global Optimization by Multilevel Coordinate Search'. *Journal of Global Optimization* **14**, 331–355.

[10] D. R. Jones, C. D. Perttunen, and B. E. Stuckman. Lipschitzian optimization without the Lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, October 1993.

[11] Donald R. Jones. DIRECT. *Encyclopedia of Optimization*, Kluwer Academic Publishing, 2001.

[12] Donald R. Jones. A taxonomy of global optimization methods based on response surfaces. *Journal of Global Optimization*, 16, 2001.