TOMLAB – Unique Features for Optimization in MATLAB

Utah State University, Logan

October 29, 2004

Kenneth Holmström
Tomlab Optimization AB
Västerås Technology Park, Sweden
hkh@tomlab.biz

Professor in Optimization
Department of Mathematics and Physics
Mälardalen University, Sweden

Special thanks to Marcus Edvall and Anders Göran at Tomlab Optimization
Outline of the talk

- The TOMLAB Optimization Environment
  - Background and history
  - Design and usage
- Tests on customer supplied large-scale optimization examples
- Customer cases, embedded solutions, consultant work
- Business perspective
- Box-bounded global non-convex optimization
- Summary
Background

- **MATLAB** – a high-level language for mathematical calculations, distributed by MathWorks Inc.
- Matlab can be extended by Toolboxes that adds to the features in the software, e.g.: finance, statistics, control, and *optimization*
- Why develop the **TOMLAB Optimization Environment**?
  - A uniform approach to optimization didn’t exist in **MATLAB**
  - Good optimization solvers were missing
  - Large-scale optimization was non-existent in **MATLAB**
  - Other toolboxes needed robust and fast optimization
  - Technical advantages from the **MATLAB** languages
    - Fast algorithm development and modeling of applied optimization problems
    - Many built in functions (ODE, linear algebra, …)
    - GUI development fast
    - Interfaceable with C, Fortran, Java code
History of Tomlab

- President and founder: Professor Kenneth Holmström
- The company founded 1986, Development started 1989
- Two toolboxes NLPLIB och OPERA by 1995
- Integrated format for optimization 1996
- TOMLAB introduced, ISMP97 in Lausanne 1997
- TOMLAB v1.0 distributed for free until summer 1999
- TOMLAB v2.0 first commercial version fall 1999
- TOMLAB starting sales from web site March 2000
- TOMLAB v3.0 expanded with external solvers /SOL spring 2001
- Dash Optimization Ltd’s XpressMP added in TOMLAB fall 2001
- Tomlab Optimization Inc. founded in California December 2001
- ILOG S.A.’s CPLEX added to product list fall 2002
2003-2004...

- Around 100 algorithms for optimization.
- Partnerships with:
  - Arki Consulting & Development A/S (CONOPT),
  - Boeing Phantom Works (SOCS, BARNLP, SPRNLP),
  - Cranfield University (MAD, Automatic Differentiation),
  - Dash Optimization Ltd. (Xpress\textsuperscript{MP}),
  - ILOG S.A. (CPLEX),
  - Klaus Schittkowski (NLPJOB, DFNLP, NLPQL, MODFIT; PDEFIT)
  - Optimal Methods Inc. (OQNLP),
  - PENOPT GbR (PENNON, PENNLP, PENSDP, PENBMI),
  - Pinter Consulting (LGO)
  - SIEMENS (SENN),
  - Stanford Business Software Inc. (SNOPT, NPSOL, MINOS, LSSOL, SQOPT),
  - Sunset Software Technology (XA),
  - University of Dundee, Fletcher, Leyffer (MINLP\textsuperscript{BB}, Filter SQP, MIQP\textsuperscript{BB}, BQPD),
  - Ziena Optimization Inc. (KNITRO 4.0);

- New Products;
  - Toolbox for \textbf{Costly (CPU-intensive) Global Black-Box Optimization (CGO)}
  - Highly integrated support for AMPL problems

- New home page and InstallShield installations.
Optimization Problems

• General formulation
  – \[\min f(x)\]
  – S.t. \[x_L \leq x \leq x_U\]
  \[b_L \leq Ax \leq b_U\]
  \[c_L \leq c(x) \leq c_U\]
  a subset of \(x\) can be restricted to integer variables.

LP, MILP, QP, MIQP, MIQQ, NLP, MINLP, SDP, BMI, GOAL, L1, LLS, NLLS, GLB, GLC, MINIMAX, LCP, MCP, CGO.
Features in TOMLAB /BASE

- **General**
  - Entry level optimization solvers and interfaces.
  - Routines for interfacing.

Problem types: LP, MILP, QP, GLB, GLC, MINIMAX, L1, GOAL, LLS, NLLS.
Solver options: 30 including several interfaces.
Features in TOMLAB /CPLEX

• General formulation
  - \[ \min 0.5 \, x^T \, Q \, x + c \, x \]
  - S.t. \[ x_L \leq x \leq x_U \]
  \[ b_L \leq A \, x \leq b_U \]
  \[ x^T \, H \, x + a \, x \leq r_U \quad 1 \ldots n \]
  a subset of \( x \) can be restricted to integer variables.

Problem types: LP, MILP, QP, MIQP, MIQQ.
Solver options: Simplex, Dual simplex, Network simplex, Barrier, Sifting, Concurrent.
Fastest LP, MILP and MIQP solver. Only MIQQ solver.
Advantages with TOMLAB – 1

- No algebraic modeling language offer the unique problem formulations
  - Semi-definite programming problem, general objective, linear and bi-linear matrix inequalities (LMI, BMI), linear and nonlinear constraints
  - Global black-box optimization, both expensive and non-expensive
- If you know MATLAB (500,000+ users), you know TOMLAB:
  - Reduces training to a minimum
  - Gives novices direct access to the best optimization packages
  - Reduces development time of optimization applications to a minimum
- Multi platform support, TOMLAB is readily available for
  - Windows, Linux, HP, MAC, and SUN systems
- Easily embedded in other products using the MATLAB compiler.
  - Portability
Advantages with TOMLAB – 2

- Uniform and flexible approach to solving your problem
  - Gateway routines automatically convert the problem when using solvers designed for different purposes, e.g. NLP solvers efficiently solving sparse nonlinear least squares problems or a quadratic programming problem
  - Universal driver routine enables the user to switch solver by changing a keyword. Lists for the various problem types, so the user can easily iterate thought all solvers that handle the specific problem
  - Assign routines creating problem structure available for all problem types
  - Multiple solvers can easily be used as backup for one-another, making convergence ~100% safe and robust

- Performance
  - Many solvers implemented in Fortran and C for maximum speed
  - Critical parts of the MATLAB solvers coded in Fortran or C

- Compatibility
  - Automatic Differentiation with MAD and ADMAT
  - Fully call-compatible with MathWorks’ Optimization Toolbox

- User features
  - Dynamic GUI Design – create own database of problems
User feedback October 12, 2004

- The evaluation went very well; I was highly impressed with the simplicity with which I could replace Optimization Toolbox with Tomlab
- The sales documentation online probably does not do enough to make clear how easy it is to run the Tomlab solvers instead of Optimization Toolbox; I had expected a lot of recoding but there was none!
- I've also been pleased with the performance, especially using npsol. At least a 20x speedup over fmincon for my application
Solving an optimization problem - with TOMLAB

1. Define the linear parts and variable types of your problem in matrices and vectors
2. Define the nonlinear parts in the MATLAB m-file language
   - The following options are available for the gradients, Jacobian and constraints gradients
     • Automatic differentiation (2 MATLAB toolboxes)
     • Numerical differentiation (6 methods)
     • Code them yourself
3. Identify patterns in your problem, Hessian and Constraint Jacobian patterns, and other information that may be useful to the solver
4. Import your problem to a TOMLAB Prob structure array, by using one of the standard assign statements
5. Select solver, and run your problem by using driver routine tomRun
6. Debug and modify solver settings, try other solvers and/or sub-solvers

FIND THE BEST SOLVER FOR YOUR PROBLEM!
Large-Scale Optimization Benchmark Tests

- **Computer**
  - P4 2400 MHz, 1 GB RAM, 3 GB swap
  - Windows 2000, MATLAB 6.5, TOMLAB v4.3

- **Definitions**
  - **n**: number of variables
  - **m**: number of constraints
  - **nnz**: number of nonzeros
  - **d**: \( \text{nnz} / (\text{tot. # of elements, e.g. } n \times m) \)
**Large-Scale LP – Michael Board**

Transport problem converted to regular LP

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>m (all eq.)</th>
<th>nnz(A)</th>
<th>d(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>674,190</td>
<td>1,671</td>
<td>1,348,380</td>
<td>0.0012</td>
</tr>
<tr>
<td>2</td>
<td>1,483,527</td>
<td>2,488</td>
<td>2,967,054</td>
<td>0.000805</td>
</tr>
<tr>
<td>3</td>
<td>4,841,370</td>
<td>4,407</td>
<td>9,682,740</td>
<td>0.000405</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Board-1</th>
<th></th>
<th></th>
<th>Board-2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Iterations</td>
<td>Time</td>
<td></td>
<td>Iterations</td>
<td>Time</td>
<td></td>
</tr>
<tr>
<td>Xpress\textsuperscript{MP} (Barrier)</td>
<td>22</td>
<td>58 s</td>
<td></td>
<td>34</td>
<td>179 s</td>
<td></td>
</tr>
<tr>
<td>CPLEX (Barrier)</td>
<td>29</td>
<td>44 s</td>
<td></td>
<td>31</td>
<td>196 s</td>
<td></td>
</tr>
<tr>
<td>CPLEX (Network)</td>
<td>43,675</td>
<td>5 s</td>
<td></td>
<td>108,356</td>
<td>14 s</td>
<td></td>
</tr>
<tr>
<td>MINOS</td>
<td>142,959</td>
<td>576 s</td>
<td>n/a</td>
<td>&gt;45 m</td>
<td></td>
<td></td>
</tr>
<tr>
<td>linprog</td>
<td>42</td>
<td>301 s</td>
<td>51</td>
<td>16 m\textsuperscript{(1)}</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Board-3:** Out of memory, all solvers. (MATLAB limit of 1.4GB)

(1) – Not optimal solution
## Large-Scale MILP – Nikos Laoutaris

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>m</th>
<th>nnz(A)</th>
<th>d(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nikos-1</td>
<td>23,000</td>
<td>7,001</td>
<td>26,000</td>
<td>1.6147e-004</td>
</tr>
<tr>
<td>Nikos-2</td>
<td>230,000</td>
<td>70,001</td>
<td>260,000</td>
<td>1.6149e-005</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>Nikos-1</th>
<th>Nikos-2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XpressMP</td>
<td>&lt;1.5 s</td>
<td>&lt;140 s</td>
</tr>
<tr>
<td>CPLEX</td>
<td>&lt;1.5 s</td>
<td>&lt;140 s</td>
</tr>
<tr>
<td>mipSolve (1)</td>
<td>16,16 s</td>
<td>n/a</td>
</tr>
</tbody>
</table>

Opt Tlbx – no suitable solver available

---

(1) MINOS 5.5 used as LP sub solver
Large-Scale QP – Kostas Skouras
Portfolio optimization, Credit Suisse First Boston
dense F (QP term), sparse A (constraints)

<table>
<thead>
<tr>
<th>n</th>
<th>m (all eq.)</th>
<th>nnz(F)</th>
<th>d(F)</th>
<th>nnz(A)</th>
<th>d(A)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>524</td>
<td>273</td>
<td>524²</td>
<td>1.0</td>
<td>1572</td>
</tr>
<tr>
<td>2.</td>
<td>4,046</td>
<td>2,119</td>
<td>4046²</td>
<td>1.0</td>
<td>12138</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Solver</th>
<th>Iterations</th>
<th>Time</th>
<th>Iterations</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Xpress&lt;sup&gt;MP&lt;/sup&gt; (Barrier)</td>
<td>8</td>
<td>1.68 s</td>
<td>11</td>
<td>203 s</td>
</tr>
<tr>
<td>CPLEX (Barrier)</td>
<td>9</td>
<td>1.22 s</td>
<td>11</td>
<td>305 s</td>
</tr>
<tr>
<td>SNOPT</td>
<td>20</td>
<td>0.24 s</td>
<td>20</td>
<td>6.17 s</td>
</tr>
<tr>
<td>SNOPT - SQOPT</td>
<td>184</td>
<td>1.25 s</td>
<td>Aborted &gt; 40 min</td>
<td></td>
</tr>
<tr>
<td>MINLP - BQPDs</td>
<td>70</td>
<td>1.34 s</td>
<td>Aborted &gt; 25 min</td>
<td></td>
</tr>
<tr>
<td>Opt Tlbx quadprog</td>
<td>529</td>
<td>223 s</td>
<td>n/a – out of memory</td>
<td></td>
</tr>
</tbody>
</table>
Large-Scale NLP – ABB
Model Predictive Control (MPC) for paper mill production

1246 variables, 495 linear constraints (99 eq.), 300 nonlinear equalities
Density of linear constraint matrix: 0.0029; 1782 nonzeros
Density of nonlinear constraint gradient: 0.017; 6200 nonzeros
feastol=opttol=1E-4, MATLAB 7.0

<table>
<thead>
<tr>
<th>Solver</th>
<th>Iter</th>
<th>Time</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>SNOPT</td>
<td>57</td>
<td>4.75 s</td>
<td></td>
</tr>
<tr>
<td>KNITRO 4.0</td>
<td>169</td>
<td>33.4 s</td>
<td>alg=1 Int/Direct, exact grad, sparse Q-N</td>
</tr>
<tr>
<td>KNITRO 4.0</td>
<td>108</td>
<td>14.8 s</td>
<td>alg=2 Interior / CG</td>
</tr>
<tr>
<td>KNITRO 4.0</td>
<td>292</td>
<td>62.2 s</td>
<td>alg=3 Active set (feas 1E-12)</td>
</tr>
<tr>
<td>FMINCON</td>
<td>n/a</td>
<td>&gt;2.5 h</td>
<td>Manually terminated</td>
</tr>
<tr>
<td>SNOPT</td>
<td>156</td>
<td>12.6 s</td>
<td>feastol=opttol=1E-6</td>
</tr>
</tbody>
</table>

ABB now embeds Tomlab /SNOPT in their product
Taiwan Semiconductor Manufacturing Company (TSMC)

- Design improvement projects. Test problem: box constrained nonlinear problem with 63,000 variables.
- Function quite costly. Failed with Optimization Toolbox - fmincon.
- Tried TOMLAB /SNOPT after recoding.
- TOMLAB /KNITRO 3.x solved the problem after optimizing all code in about 2 minutes.
- A full design project was initiated based on the promising results, and a problem with about 480,000 decision variables was solved in about 1 hour.
- Solution implemented with Cadence, Mentor Graphics, and SpringSoft EDA tools. Solution verified from Silicon Data.
Embedded solutions with TOMLAB

- Philips embeds Tomlab /Xpress in their production planning system for Lumileds’ facilities in Malaysia and the Netherlands.

- Gaz de France embeds Tomlab /SNOPT for production planning. Solution based on MCC and distributed for non-MATLAB use.

- US Naval Research Lab: Has developed a package for dynamic optimization and control, DIDO. Tomlab is a required part. Now Tomlab /DIDO is one toolbox option in MATLAB, maintained and developed by Dr. Michael Ross.
Halliburton Energy Services

- Service provider to the big oil companies.

- Tomlab /NPSOL has been compiled and embedded in Halliburton’s new commercial software NMRStudio for real-time analysis of exploration data.

- Optimization is performed in real-time while measurements from the drilling equipment are coming in. Also available for offline analysis.

- More robust and faster mathematical analysis is the deliverable, with estimated savings of $200,000 per drilled hole. The system will cut the measurement time in half.

- More advanced parameter estimation of very noisy NMR-data (Nuclear Magnetic Resonance) is implemented.

- Tomlab’s consulting team in the US is continuously improving the models, and optimization methods. A full time position has been introduced at Halliburton to further implement the systems.
Optimization for the pulp and paper industry

- Tomlab has been selected as the optimization platform for the European Union project DOTS (dynamic optimization of pulp and paper mills) with partners in Finland, France, Germany and Sweden.

- Several problems for the pulp and paper industry:
  
  Paper quality  
  Energy consumption  
  White-water System Management (closing up the mill)  
  Stabilize regulatory controls
Unilever multiple fat-blend production planning

- Mixed-integer nonlinear problem
- Solved with standard tools, but too slow (OQNLP, MINLPBB, LGO, glcCluster, glcFast)
- The Tomlab team has developed new algorithms that solve the problems order of magnitude faster
NASA use of TOMLAB

- Claremont University in cooperation with JPL/NASA uses TOMLAB

- Optimizing dataflow from the Mars landers through the orbiters and back to Earth.

- TOMLAB /SNOPT is 10-20 times as fast as MathWorks’ Optimization Toolbox, and finds better solutions (nonsmooth problem).

- Will hopefully be embedded in the mission planning system,
TOMLAB in the market place

- The TOMLAB software sales are growing, factor 2 in 2003. Roughly factor 2 in 2004.
  - Rapidly being established as the standard for optimization in MATLAB
- Recognized for the modeling features.

**Products:** Base module + 20 options
Marketing

- Internet site [http://tomlab.biz](http://tomlab.biz). Full website optimization program to drive in maximum number of users.
- Newsletters to attract users back for more information, testing
- Follow-ups with each registered customer
- Newsgroup coverage. Active participation in Matlab and mathematical newsgroups
- Information messages to mailing lists
- Talks at scientific conferences
- Network of satisfied users and scientists
- Rapid and advanced support in the initial sales process
The TOMLAB Optimization Environment
- For fast and robust large-scale optimization in MATLAB®

What is TOMLAB?
The TOMLAB Optimization Environment is a powerful optimization platform for solving applied optimization problems in Matlab. TOMLAB provides a wide range of features, tools and services for your solution process. Read more about Tomlab >>

The leading optimal control tool is now available: TOMLAB /DIDO.

Purchase TOMLAB online >>

Are you looking for a solver to embed in your system?

If you register you can test TOMLAB for free for 21 days.

What can Tomlab do for you? See our customer examples >>

MOST POPULAR PRODUCTS

TOMLAB /SOL v4.4
TOMLAB /SOL v4.4 efficiently integrates the well-known solvers developed by the Stanford Systems Optimization Laboratory (SOL) with MATLAB and TOMLAB. The toolbox includes the solvers MINOS, LPOPT, QPOPT, NPSOL, NLSSOL, LSSOL, NPSOPT, SQOPT.
Read more >>
Buy now >>

TOMLAB /CPLEx v9.0
Solver package CPLEx 9.0, including Matlab interface. State-of-the-art mixed-integer linear and quadratic programming with quadratic constraints (MIQP, MIQCP, MIQCO), and large-scale simplex and barrier methods for LP and QP. On request we can also provide TOMLAB /CPLEx MEX for CPLEx 9.x and more recent versions.
Georgia Tech Aerospace Initial Test Problem - SNOPT

- Problem with $n = 752$, $m = 755$ was evaluated. Initial problem was solved in about 15 min.
- Turns out the major feasibility tolerance was set to $1e-9$ using numerical derivatives. Problem now solved in about 44.25 s.
- Adding ConsPattern: 20.6 s.
- Modifying objective function: 18.8 s.
- Modifying constraint calculation: 15.4 s.
- Removing redundant constraints, re-writing to a QP problem: 0.6 s.
  - 255 constraints were fixing the decision variables.
- Switching to SQOPT: 0.45 s.

EXAMPLE OF ADVANCED USER SUPPORT DURING THE SALES PROCESS
Box-bounded global optimization

\[ \min_x f(x) \]

s/t \[ -\infty < x_L \leq x \leq x_U < \infty \]

\[ x_L, x, x_U \in \mathbb{R}_d, \quad f(x) \in \mathbb{R} \]

No derivative information available.

\[ f(x) \text{ is either non-costly or costly.} \]

Dimension \( d \) is not too large, say 1-20.
Solution of non-costly subproblems

- DIRECT algorithm (Jones et al. 1993)
  efficient strategy, global convergence proof
- No tuning parameters
- Dividing space into rectangles
- Estimating all possible Lipschitz constants
- Handles nonsmooth, nonlinear, nonconvex, multimodal functions
- Fast, robust and efficient implementation of the DIRECT algorithm in Tomlab’s glbFast
DIRECT algorithm – initial phase
TOMLAB GUI – global optimization
Constrained box-bounded global optimization

\[
\begin{align*}
\text{min } & \quad f(x) \\
\text{s/t } & \quad -\infty < x_L \leq x \leq x_U < \infty \\
& \quad b_L \leq A x \leq b_U \\
& \quad c_L \leq c(x) \leq c_U \\
\end{align*}
\]

\(x_L, x, x_U \in \mathbb{R}^d, \quad f(x) \in \mathbb{R}\)
\(b_L, b_U \in \mathbb{R}^m, \quad c(x), c_L, c_U \in \mathbb{R}^p,\)

No derivative information available.
\(c_i(x)\) either costly or non-costly.
Subset of \(x\) could be integer valued.
Non-costly constrained optimization

- **glcFast** - Fast and efficient implementation of the extended DIRECT algorithm (Jones 2001) that handles constraints and some \( x \) integer valued.
- **glcSolve** – Matlab version of the extended DIRECT algorithm
glcCluster

- Hybrid-clustering algorithm for constrained box-bounded global mixed-integer non-convex optimization.
- Uses all sampled points in the constrained DIRECT algorithm (glcFast) as input to a clustering algorithm.
- The best point in each cluster created is used as an initial point in a local search.
- Local search by any NLP solver, e.g. NPSOL, with fixed integer values.
- The best point found from the local search, if any better, is then used in a second DIRECT search with glcFast. If new improvement, a local search is done.
Tomlab /CGO – Costly box-bounded global optimization

- $f(x)$ takes considerable CPU time, e.g. 30 minutes
- $f(x)$ is often a complex computer program, or the result of an advanced simulation, e.g. Monte-Carlo simulation, CFD or trading strategy evaluation.
- $f(x)$ often noisy, no derivatives known or usable.
Basic algorithm for costly global optimization

- Find initial set of $n \geq d+1$ sample points $x$. Experimental design problem.
- Compute costly $f(x)$ for initial set of points.
- Iteration until target $f(x)$ achieved or no time left:
  1. Use the $n$ sampled points to build a response surface model as an approximation of $f(x)$ surface.
  2. Optimize a cheap function of the approximating surface to obtain a new trial point to compute the costly $f(x)$ for.
  3. Compute and validate new $(x, f(x))$, increase $n$. 
RBF algorithm

- Interpolation of all sampled point by radial basis function interpolation.
- Cycle of target values on surface gives trade off between local and global search.
- Global convergence proof (Gutmann 2001).
- 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.
RBF algorithm - improvements

- Efficient numerical implementation of interpolation updates.
- Efficient trial point computations of interpolation coefficients and values, i.e. \( f(x) \) for non-costly global optimization subproblem.
- Each RBF iteration takes only order of seconds.
- Scaling of \( x \) to unit hypercube.
- Reduce influence of large \( f(x) \).
- Removal of too close points, keeping the best.
- Warm start of RBF (and/or EGO).
Tomlab /CGO – more improvements

- Extended RBF and EGO algorithm to handle non-costly constraints and mixed-integer problems (presented at Informs Annual Conference, Denver, October 2004)
TOMLAB /CGO – 2 dim test case

• Testing with Coolit, a CFD (Computational Fluid Dynamics) software from DAAT Inc.
• Function evaluations between 15 minutes and 4 hours.
• One continuous and one integer variable.
• 13 points used for initialization.
• glcFast global subsolver for local and global problem.
• 45 function evaluations establish a good optimum.
TOMLAB /CGO – 2 dim test case

Case 1 - All Data

Temperature vs. Function evaluations
Case 1 - Final 15

Temperature vs. Function evaluations
TOMLAB /CGO – 3 dim test case

- Testing with Coolit, a CFD (Computational Fluid Dynamics) software from DAAT Inc.
- Function evaluations between 15 mins and 1 hour.
- Two continuous and one integer variable.
- 4 corner points and mid point used for initialization (Gutmann).
- TOMLAB /OQNLP subsolver for local and global problem.
- Still good progress after 115 function evaluations.
TOMLAB /CGO – 3 dim test case

CASE2

DEGREES C

FUNC EVALS
## Global Optimization - Solvers

<table>
<thead>
<tr>
<th>Solver</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>glbSolve</td>
<td>DIRECT algorithm, box-bounded, both Matlab and fast MEX Fortran version.</td>
</tr>
<tr>
<td>glbFast</td>
<td>DIRECT by Jones (2001) for nonlinear and integer constraints, box-bounded, Matlab and fast MEX Fortran version.</td>
</tr>
<tr>
<td>glcSolve</td>
<td>DIRECT, combined with clustering techniques and local search from best cluster points found.</td>
</tr>
<tr>
<td>glcFast</td>
<td>Efficiency Global Optimization (EGO) for costly functions (Jones et. al 1998)</td>
</tr>
<tr>
<td>glcCluster</td>
<td>Algorithm for costly (expensive) functions using radial basis interpolation (RBF) and response model techniques.</td>
</tr>
</tbody>
</table>
Summary

• Tomlab is a powerful environment for all sorts of optimization in MATLAB
• Tomlab can be embedded in products. Ongoing work to make Tomlab more independent on MATLAB.
• The goal is that Tomlab should provide all state-of-the-art software in optimization for all types of optimization problems.