USER'S GUIDE FOR TOMLAB /LGO[1]

János D. Pintér[2], Kenneth Holmström[3], Anders O. Göran[4] and Marcus M. Edvall[5]

November 6, 2006

[1]More information available at the TOMLAB home page: http://tomopt.com. E-mail: **tomlab@tomopt.com**.

[2]Pintér Consulting Services, Inc., 129 Glenforest Drive, Halifax, NS, Canada B3M 1J2, jdpinter@hfx.eastlink.ca.

[3]Professor in Optimization, Mälardalen University, Department of Mathematics and Physics, P.O. Box 883, SE-721 23 Västerås, Sweden, kenneth.holmstrom@mdh.se.

[4]Tomlab Optimization AB, Västerås Technology Park, Trefasgatan 4, SE-721 30 Västerås, Sweden, anders@tomopt.com.

[5]Tomlab Optimization Inc., 855 Beech St #121, San Diego, CA, USA, medvall@tomopt.com.

Contents

1 Introduction

1.1 Overview

Welcome to the TOMLAB /LGO User's Guide. TOMLAB /LGO includes the LGO solver from Pintér Consulting Services and an interface to MATLAB, by MathWorks'.

The *Lipschitz-Continuous Global Optimizer* (LGO) solver suite serves for the analysis and global solution of general nonlinear programming (NLP) models. The LGO solver system has been developed and gradually extended for more than a decade and it now incorporates a suite of robust and efficient global and local nonlinear solvers. It can also handle smaller LP models. LGO is documented elsewhere in detail: see for example, Pintér (1996, 2001, 2003). [1, 2, 3]

TOMLAB /LGO integrates the following global scope algorithms:

- Branch-and-bound (adaptive partition and sampling) based global search (BB)
- Adaptive global random search (GARS)
- Adaptive multistart global random search (MS)

LGO also includes the following local solver strategies:

- Constrained local search, based on a generalized reduced gradient approach (GRG).

The overall solution approach followed by TOMLAB /LGO is based on the seamless combination of the global and local search strategies. This allows for a broad range of operations. In particular, a solver suite approach supports the flexible usage of the component solvers: one can execute fully automatic (global and/or local search based) optimization, and can design customized interactive runs.

TOMLAB /LGO does not rely on any sub-solvers, and it does not require any in-depth structural information about the model. It is particularly suited to solve even 'black box' (closed, confidential), or other complex models, in which the available analytical information may be limited. TOMLAB /LGO needs only computable function values (without a need for higher order analytical information). TOMLAB /LGO can even solve models having constraints involving continuous, but non-differentiable functions. Thus, within TOMLAB, LGO is well suited to solve nonlinear - global and convex - optimization models.

TOMLAB /LGO can also be used in conjunction with other TOMLAB solvers. Local solvers (when available) can be used to verify the solution found by LGO, and to provide additional local information.

1.2 Contents of this manual

- Section 1 provides a basic overview of the TOMLAB /LGO solver package.
- Section 2 provides an overview of the Matlab interface to LGO.
- Section 3 describes how to set LGO solver options from Matlab.
- Section 4 provides information regarding TOMLAB /LGO test examples.
- Section 5 gives detailed information about the interface routine *lgoTL*.

1.3 More information

Please visit the following links for more information and see the illustrative references at the end of this manual.

- http://tomopt.com/tomlab/products/lgo/

1.4 Prerequisites

In this concise manual we assume that the user is familiar with global optimization and nonlinear programming, setting up problems in TOMLAB (in particular global constrained nonlinear (**glc**) problems) and with the Matlab language in general.

2 Using the Matlab Interface

The LGO solver package is accessed via the *tomRun* driver routine, which calls the *lgoTL* interface routine. The solver itself is located in the MEX file *lgo*.

Table 1: The interface routines.

Function	Description	Section	Page
lgoTL	The interface routine called by the TOMLAB driver routine *tomRun*. This routine then calls the MEX file *lgo*	5.1	6

3 Setting LGO Options

All control parameters can be set directly from Matlab.

3.1 Setting options using the Prob.LGO structure

The parameters can be set as subfields in the *Prob.LGO* structure. The following example shows how to set a limit on the maximum number of merit function evaluations before termination of global search phase.

```
Prob = glcAssign(...)     % Setup problem, see help glcAssign for more information

Prob.LGO.options.G_maxfct = 10000; % Setting the maximum number of global search
                                   % phase model function evaluations.
```

In the cases where a solver specific parameter has a corresponding TOMLAB general parameter, the latter is used only if the user has not given the solver specific parameter.

A complete description of the available LGO parameters can be found in Section 5.1.

4 TOMLAB /LGO Test Examples

There are several LGO test examples included in the TOMLAB /LGO distribution. The examples are located in the *testprob* folder in TOMLAB. *lgo1_prob* contains one dimensional test problems while *lgo2_prob* includes two- and higher-dimensional.

To test the solution of these problem sets by LGO, the following type of code can be used:

```
Prob = probInit('lgo1_prob', 1);

Result = tomRun('lgo', Prob, 1);
```

It is also possible to run all the problems located in *glb_prob* and *glc_prob* if no integer variables are defined.

5 TOMLAB /LGO Solver Reference

A detailed description of the TOMLAB /LGO solver interface is given below. Also see the M-file help for *lgoTL.m*.

5.1 lgoTL

Purpose

Solves global and local (convex and mildly non-convex) constrained nonlinear programming problems.

LGO solves problems of the form

$$
\begin{array}{cccccc}
\min\limits_{x} & f(x) & & & & \\
 & & & & & \\
s/t & x_L & \leq & x & \leq & x_U \\
 & b_L & \leq & Ax & \leq & b_U \\
 & c_L & \leq & c(x) & \leq & c_U
\end{array}
\tag{1}
$$

where $x, x_L, x_U \in \mathbb{R}^n$, $A \in \mathbb{R}^{m_1 \times n}$, $b_L, b_U \in \mathbb{R}^{m_1}$ and $c(x), c_L, c_U \in \mathbb{R}^{m_2}$.

Calling Syntax

Prob = glcAssign(...);
Result = tomRun('lgo',Prob,...);

Description of Inputs

Prob Problem description structure. The following fields are used:

A	Linear constraints coefficient matrix.
x_L, x_U	Bounds on variables.
b_L, b_U	Bounds on linear constraints.
c_L, c_U	Bounds on nonlinear constraints. For equality constraints (or fixed variables), set e.g. b_L(k) == b_U(k).
PriLevOpt	Print level in MEX interface.
optParam	Structure with optimization parameters.
MaxFunc	Maximum number of function evaluations. (Prob.LGO.options.g_maxfct)
LGO.PrintFile	Name of LGO Print file. Amount and type of printing determined by PriLevOpt.
LGO.SummFile	Name of LGO summary file.
LGO.options	Structure with special fields for LGO optimization parameters. See Table 4.

Description of Outputs

Result	Structure with result from optimization. The following fields are set:	
	f_k	Function value at optimum.
	x_k	Solution vector.
	x_0	Initial solution vector.
	c_k	Nonlinear constraint residuals.
	xState	State of variables. Free == 0; On lower == 1; On upper == 2; Fixed == 3;
	bState	State of linear constraints. Free == 0; Lower == 1; Upper == 2; Equality == 3;
	cState	State of nonlinear constraints. Free == 0; Lower == 1; Upper == 2; Equality == 3;
	Ax	Values of linear constraints.
	ExitFlag	Exit status from LGO (TOMLAB standard).
	ExitText	Exit text from LGO.
	Inform	LGO information parameter. 1 = Normal completion. 2 = Iteration interrupt. 3 = Time limit exceeded. 4 = Terminated by solver. 7 = Size limitation. Other = Optimal solution found.
	FuncEv	Number of function evaluations.
	ConstrEv	Number of constraint evaluations. In the context of LGO, ConstrEv = FuncEv.
	QP.B	Basis vector in TOMLAB QP standard.
	Solver	Name of the solver (LGO).
	SolverAlgorithm	Description of the solver.
	LGO	Subfield with LGO specific results.
	sstat	Solver status.
	mstat	Model status.
	runtime	Time spent, measured by LGO solver.

The following table shows all the options that the user can set for the solver. The TOMLAB /LGO options are divided into two main categories:

- General options
- Limits and tolerances

Description of Prob.LGO.options

Table 4: User options for the TOMLAB /LGO solver. The following fields are used:

Option	Description	Default
General options		
opmode	Specifies the algorithm to be used.	3
0	Local search from the given nominal solution without a preceding local search (LS)	
1	Global branch-and-bound search and local search (BB+LS)	
2	Global adaptive random search and local search (GARS+LS)	
3	Global multistart random search and local search (MS+LS)	
	Note that an option of 0 will work for many slightly nonconvex, as well as convex models. See note below this table.	
tlimit	Time limit in seconds.	600
Limits and Tolerances		
g_maxfct	Maximum number of merit function evaluations before termination of global search phase (BB, GARS, or MS). The value of `-1` uses `500(nvars+ncons)`, where `nvars` is the number of variables and `ncons` the number of constraints. The difficulty of global optimization models varies greatly: for difficult models, `g_maxfct` can be increased to as needed.	-1
max_nosuc	Maximum number of merit function evaluations in global search phase (BB, GARS, or MS) where no improvement is made. Global search terminates upon reaching this limit. The value of `-1` uses `100(nvars+ncons)`, where `nvars` is the number of variables and `ncons` the number of constraints. For difficult models, this value can also be increased as deemed necessary.	-1

Table 4: User options for the TOMLAB /LGO solver. The following fields are used:, continued

Option	**Description**	**Default**
penmult	Constraint penalty multiplier. Global search phase merit function is defined as objective + the violated constraints weighted by `penmult`.	100
acc_tr	Global search termination criterion parameter (acceptability threshold). The global search phase (BB, GARS, or MS) ends, if an overall merit function value found in the global search phase is less than `acc_tr`. If a good estimate is known, then its usage may results in a considerably faster search.	-1.0E+10
fct_trg	Target merit function value in local search phase.	-1.0E+10
fi_tol	Local search merit function improvement tolerance.	1.0E-06
con_tol	Maximal admissible constraint violation in local search.	1.0E-06
kt_tol	Kuhn-Tucker local optimality condition tolerance.	1.0E-06
irngs	Random number seed. An arbitrary integer value can be used instead of the default.	0

Note that the local search operational mode (`Opmode 0`) is the fastest, and that it will work for convex, as well as for many mildly non-convex models. If the model has a highly non-convex (multiextremal) structure, then at least one of the global search modes should be used. In difficult or complex models, it may be a good idea to apply all three global search modes, to verify the global solution, or perhaps to find alternative good solutions. Usually, `Opmode 3` is the safest (and slowest), since it applies several local searches; Opmodes 1 and 2 launch only a single local search from the best point found in the global search phase.
Note that if model-specific information is known (more sensible target objective/merit function value, tolerances, tighter variable bounds), then such information should always be used, since it may help to solve the model far more efficiently than by using 'blind' defaults.

References

[1] J.D. Pintér. *Global Optimization in Action (Continuous and Lipschitz Optimization: Algorithms, Implementations and Applications).* Kluwer Academic Publishers, Dordrecht / Boston / London., See http://www.wkap.nl/prod/b/0-7923-3757-3, 1996.

[2] J.D. Pintér. *Computational Global Optimization in Nonlinear Systems - An Interactive Tutorial.* Lionheart Publishing, Inc. Atlanta, GA., See http://www.lionhrtpub.com/books/globaloptimization.html, 2001.

[3] Pintér Consulting Services, Inc., Halifax, NS. See www.dal.ca/ jdpinter for further professional details. *LGO - A Model Development and Solver System for Continuous Global Optimization. Users Guide.*, 2003.