GLPKMEX Parameters list

Nicolò Giorgetti

ver. 0.5.8
(based on GLPK 4.1)

This document describes all control parameters currently implemented in the GLPKMEX, a Matlab MEX interface for the GLPK library. Symbolic names of control parameters and corresponding codes of GLPK are given on the left. Types, default values, and descriptions are given on the right.

1 Integer parameters

`msglev`	type: integer, default: 3
LPX_K_MSGLEV	Level of messages output by solver routines: 0 — no output 1 — error messages only 2 — normal output 3 — full output (includes informational messages)
`scale`	type: integer, default: 3
LPX_K_SCALE	Scaling option: 0 — no scaling 1 — equilibration scaling 2 — geometric mean scaling, then equilibration scaling
`dual`	type: integer, default: 0
LPX_K_DUAL	Dual simplex option: 0 — do not use the dual simplex 1 — if initial basic solution is dual feasible, use the dual simplex
`price`	type: integer, default: 1
LPX_K_PRICE	Pricing option (for both primal and dual simplex): 0 — textbook pricing 1 — steepest edge pricing
`round`	type: integer, default: 0
LPX_K_ROUND	Solution rounding option: 0 — report all primal and dual values "as is" 1 — replace tiny primal and dual values by exact zero
`itlim`	type: integer, default: -1
LPX_K_ITLIM	Simplex iterations limit. If this value is positive, it is decreased by one each time when one simplex iteration has been performed, and reaching zero value signals the solver to stop the search. Negative value means no iterations limit.
`itcnt`	type: integer, initial: 0
LPX_K_ITCNT	Simplex iterations count. This count is increased by one each time when one simplex iteration has been performed.
`outfrq`	type: integer, default: 200
LPX_K_OUTFRQ	Output frequency, in iterations. This parameter specifies how frequently the solver sends information about the solution to the standard output.
`branch`	type: integer, default: 2
LPX_K_BRANCH	Branching heuristic option (for MIP only): 0 — branch on the first variable 1 — branch on the last variable 2 — branch using a heuristic by Driebeck and Tomlin

`btrack` type: integer, default: 2

`LPX_K_BTRACK` Backtracking heuristic option (for MIP only):
0 — depth first search
1 — breadth first search
2 — backtrack using the best projection heuristic

`presol` type: int, default: 0

`LPX_K_PRESOL` If this flag is set, the routine `lpx_simplex` solves the problem using the built-in LP presolver. Otherwise the LP presolver is not used.

2 Real parameters

`relax` type: real, default: 0.07

`LPX_K_RELAX` Relaxation parameter used in the ratio test. If it is zero, the textbook ratio test is used. If it is non-zero (should be positive), Harris' two-pass ratio test is used. In the latter case on the first pass of the ratio test basic variables (in the case of primal simplex) or reduced costs of non-basic variables (in the case of dual simplex) are allowed to slightly violate their bounds, but not more than (`RELAX` · `TOLBND`) or (`RELAX` · `TOLDJ`) (thus, `RELAX` is a percentage of `TOLBND` or `TOLDJ`).

`tolbnd` type: real, default: 10^{-7}

`LPX_K_TOLBND` Relative tolerance used to check if the current basic solution is primal feasible. (Do not change this parameter without detailed understanding its purpose.)

`toldj` type: real, default: 10^{-7}

`LPX_K_TOLDJ` Absolute tolerance used to check if the current basic solution is dual feasible. (Do not change this parameter without detailed understanding its purpose.)

`tolpiv` type: real, default: 10^{-9}

`LPX_K_TOLPIV` Relative tolerance used to choose eligible pivotal elements of the simplex table. (Do not change this parameter without detailed understanding its purpose.)

`objll` type: real, default: `-DBL_MAX`

`LPX_K_OBJLL` Lower limit of the objective function. If on the phase II the objective function reaches this limit and continues decreasing, the solver stops the search. (Used in the dual simplex only.)

`objul` type: real, default: `+DBL_MAX`

`LPX_K_OBJUL` Upper limit of the objective function. If on the phase II the objective function reaches this limit and continues increasing, the solver stops the search. (Used in the dual simplex only.)

`tmlim` type: real, default: -1.0

`LPX_K_TMLIM` Searching time limit, in seconds. If this value is positive, it is decreased each time when one simplex iteration has been performed by the amount of time spent for the iteration, and reaching zero value signals the solver to stop the search. Negative value means no time limit.

`outdly` type: real, default: 0.0

`LPX_K_OUTDLY` Output delay, in seconds. This parameter specifies how long the solver should delay sending information about the solution to the standard output. Non-positive value means no delay.

`tolint` type: real, default: 10^{-5}

`LPX_K_TOLINT` Relative tolerance used to check if the current basic solution is integer feasible. (Do not change this parameter without detailed understanding its purpose.)

`tolobj` type: real, default: 10^{-7}

`LPX_K_TOLOBJ` Relative tolerance used to check if the value of the objective function is not better than in the best known integer feasible solution. (Do not change this parameter without detailed understanding its purpose.)

3 Matlab Example

```
% Problem data
s=-1;
c=[10,6,4]';
a=[1,1,1;...
   10,4,5;...
   2,2,6];
b=[100,600,300]';
ctype=['U','U','U']';
lb=[0,0,0]';
ub=[];
vartype=['C','C','C']';

% Setting parameters
param.msglev=1;   % error messages only
param.itlim=100;  % Simplex iterations limit = 100

[xmin,fmin,status,lambda,extra]=glpkmex(s,c,a,b,ctype,lb,ub,vartype,param)
```