## October 2010

## KNITRO 7.0 User Options for AMPL



KNITRO user options can be set from AMPL by typing the name of the option and a numeric value. When using AMPL's interactive mode, set all options in a single command; for example,

ampl: option knitro\_options "maxit=100 opttol=1.0e-5";

When running Knitro directly with an AMPL problem, set user options on the command line with the problem name; for example,

knitroampl testproblem.nl maxit=100 opttol=1.0e-5

A complete list of available KNITRO options can always be shown by typing: knitroampl -=

OPTION	DESCRIPTION	DEFAULT
alg	optimization algorithm used:	0
algorithm	0: let KNITRO choose the algorithm	
	1: Interior/Direct (barrier) algorithm	
	2: Interior/CG (barrier) algorithm	
	3: Active Set algorithm	
bar_directinterval	frequency for trying to force direct steps	10
bar_feasible	whether feasibility is given special emphasis:	0
	0: no special emphasis on feasibility	
	1: iterates must honor inequalities	
	2: emphasize first getting feasible before optimizing	
	3: implement both options 1 and 2 above	
bar_feasmodetol	tolerance for entering stay feasible mode	1.0e-4
bar_initmu	initial value for barrier parameter	1.0e-1
bar_initpt	initial point strategy for barrier algorithms	0
	0: let KNITRO choose the initial point strategy	
	1: shift the initial point to improve barrier performance	
	2: do not alter the initial point supplied by the user	
bar_maxbacktrack	maximum number of linesearch backtracks	3
bar_maxrefactor	maximum number of KKT refactorizations allowed	0
bar_murule	barrier parameter update rule:	0
	0: let KNITRO choose the barrier update rule	
	1: monotone decrease rule	
	2: adaptive rule based on complementarity gap	
	3: probing rule (Interior/Direct only)	
	4: safeguarded Mehrotra predictor-corrector type rule	
	5: Mehrotra predictor-corrector type rule	
	6: rule based on minimizing a quality function	

OPTION	DESCRIPTION	DEFAULT
bar_penaltycons	technique for penalizing constraints in the barrier algorithms:	0
• •	0: let KNITRO choose the strategy	
	1: do not apply penalty approach to any constraints	
	2: apply a penalty approach to all general constraints	
bar_penaltyrule	penalty parameter rule for step acceptance:	0
	0: let KNITRO choose the strategy	
	1: use single penalty parameter approach	
	2: use more tolerant, flexible strategy	
blasoption	specify the BLAS/LAPACK function library to use:	1
•	0: use KNITRO built-in functions	
	1: use Intel Math Kernel Library functions	
	2: use the dynamic library specified with "blasoptionlib"	
debug	enable debugging output:	0
	0: no extra debugging	
	1: print info to debug solution of the problem	
	2: print info to debug execution of the solver	
delta	initial trust region radius scaling	1.0e0
feastol	feasibility termination tolerance (relative)	1.0e-6
feastol_abs	feasibility termination tolerance (absolute)	0.0e-0
gradopt	gradient computation method:	1
	1: use exact gradients	
	2: compute forward finite-difference approximations	
	3: compute centered finite-difference approximations	
hessopt	Hessian (Hessian-vector) computation method:	1
-	1: use exact Hessian derivatives	
	2: use dense quasi-Newton BFGS Hessian approximation	
	3: use dense quasi-Newton SR1 Hessian approximation	
	4: compute Hessian-vector products by finite diffs	
	5: compute exact Hessian-vector products	
	6: use limited-memory BFGS Hessian approximation	
honorbnds	0: allow bounds to be violated during the optimization	2
	1: enforce bounds satisfaction of all iterates	
	2: enforce bounds satisfaction of initial point	
infeastol	tolerance for declaring infeasibility	1.0e-8
linsolver	linear system solver to use inside KNITRO:	0
	0: let KNITRO choose the linear system solver	
	1: (not currently used; same as 0)	
	2: use a hybrid approach; solver depends on system	
	3: use a dense QR method (small problems only)	
	4: use HSL MA27 sparse symmetric indefinite solver	
	5: use HSL MA57 sparse symmetric indefinite solver	
lmsize	number of limited-memory pairs stored in LBFGS approach	10
lpsolver	1: use internal LP solver in Active Set algorithm	1
ippoivoi	2: use ILOG-CPLEX LP solver in Active Set algorithm	
	_	
	(requires a valid CPLEX license)	

OPTION	DESCRIPTION	DEFAULT
maxcgit	maximum allowable conjugate gradient (CG) iterations: 0: let KNITRO set the number based on the problem size $n$ : maximum of $n>0$ CG iterations per minor iteration	0
maxcrossit	maximum number of allowable crossover iterations	0
maxit	maximum number of iterations before terminating 0: let KNITRO set the number based on the problem $n\colon$ maximum limit of $n>0$ iterations	0
maxtime_cpu	maximum CPU time in seconds before terminating	1.0e8
maxtime_real	maximum real time in seconds before terminating	1.0e8
mip_branchrule	MIP branching rule:  0: let KNITRO choose the branching rule  1: most-fractional branching  2: pseudo-cost branching  3: strong branching	0
mip_debug	MIP debugging level  0: no MIP debugging output  1: print MIP debugging information	0
mip_gub_branch	Branch on GUBs  0: do not branch on GUB constraints  1: allow branching on GUB constraints	0
mip_heuristic	heuristic search approach  0: let KNITRO decide whether to apply a heuristic  1: do not apply any heuristic  2: use feasibility pump heuristic  3: use MPEC heuristic	0
mip_heuristic_maxit	heuristic search iteration limit	100
$\mathtt{mip}_{-}\mathtt{implications}$	Add logical implications  0: do not add constraints from logical implications  1: add constraints from logical implications	1
mip_integer_tol	threshold for deciding integrality	1.0e-8
mip_integral_gap_abs	absolute integrality gap stop tolerance	1.0e-6
mip_integral_gap_rel	relative integrality gap stop tolerance	1.0e-6
mip_knapsack	add knapsack cuts  0: do not add knapsack cuts  1: add knapsack inequality cuts only  2: add knapsack inequality and equality cuts	1
mip_lpalg	LP subproblem algorithm  0: let KNITRO decide the LP algorithm  1: Interior/Direct (barrier) algorithm  2: Interior/CG (barrier) algorithm  3: Active Set (simplex) algorithm	0
mip_maxnodes	maximum nodes explored	100000
mip_maxsolves	maximum subproblem solves	200000
mip_maxtime_cpu	maximum CPU time in seconds for MIP	1.0e8
mip_maxtime_real	maximum real time in seconds for MIP	1.0e8

OPTION	DESCRIPTION	DEFAULT
mip_method	MIP method	0
-	0: let KNITRO choose the method	
	1: branch and bound method	
	2: hybrid method for convex nonlinear models	
mip_outinterval	MIP node output interval	10
mip_outlevel	MIP output level	1
mip_outsub	enable MIP subproblem debug output	0
mip_pseudoinit	method to initialize pseudo-costs	0
	0: let KNITRO choose the method	
	1: use average value	
	2: use strong branching	
mip_rootalg	root node relaxation algorithm	0
	0: let KNITRO decide the root algorithm	
	1: Interior/Direct (barrier) algorithm	
	2: Interior/CG (barrier) algorithm	
	3: Active Set algorithm	
mip_rounding	MIP rounding rule	0
1 0	0: let KNITRO choose the rounding rule	
	1: do not attempt rounding	
	2: use fast heuristic	
	3: apply rounding solve selectively	
	4: apply rounding solve always	
mip_selectrule	MIP node selection rule	0
•	0: let KNITRO choose the node select rule	
	1: use depth first search	
	2: use best bound node selection	
	3: use a combination of depth first and best bound	
mip_strong_candlim	strong branching candidate limit	10
mip_strong_level	strong branching level limit	10
mip_strong_maxit	strong branching subproblem iteration limit	1000
mip_terminate	termination condition for MIP	0
•	0: terminate at optimal solution	
	1: terminate at first integer feasible solution	
ms_enable	0: multi-start not enabled	0
	1: multi-start enabled	
ms_maxbndrange	maximum range to vary unbounded $x$ when generating start points	1.0e3
ms_maxsolves	maximum number of start points to try during multi-start	0
	0: let KNITRO set the number based on problem size	
	n: try exactly $n > 0$ start points	
ms_maxtime_cpu	maximum CPU time for multi-start, in seconds	1.0e8
ms_maxtime_real	maximum real time for multi-start, in seconds	1.0e8
ms_num_to_save	number feasible points to save in "knitro_mspoints.log"	0
		1.0e-6
		1.0e20
ms_oorminaoo		
	<u>-</u>	
ms_savetol ms_startptrange ms_terminate	tolerance for feasible points to be considered distinct maximum range to vary all x when generating start points  termination condition for multi-start  0: terminate after ms_maxsolves  1: terminate at first local optimum (if before ms_maxsolves)  2: terminate at first feasible solution (if before ms_maxsolves)	

OPTION	DESCRIPTION	DEFAULT
newpoint	0: no action	0
	1: save the latest new point to file "knitro_newpoint.log"	
	2: append all new points to file "knitro_newpoint.log"	
objrange	maximum allowable objective function magnitude	1.0e20
opttol	optimality termination tolerance (relative)	
opttol_abs	optimality termination tolerance (absolute)	0.0e-0
outappend	append output to existing files:	0
	0: do not append	
	1: do append	
outdir	directory where output files are created	
outlev	printing output level:	2
	0: no printing	
	1: just print summary information	
	2: print basic information every 10 iterations	
	3: print basic information at each iteration	
	4: print all information at each iteration	
	5: also print final (primal) variables	
	6: also print final Lagrange multipliers (sensitivies)	
outmode	0: direct KNITRO output to standard out (e.g., screen)	0
	1: direct KNITRO output to the file "knitro.log"	
	2: print to both the screen and file "knitro.log"	
pivot	initial pivot threshold for matrix factorizations	1.0e-8
$presolve\_dbg$	0: no debugging information	0
	2: print the KNITRO problem with AMPL model names	
scale	0: do not scale the problem	1
	1: perform automatic scaling of functions	
soc	0: do not allow second order correction steps	1
	1: selectively try second order correction steps	
	2: always try second order correction steps	
xtol	stepsize termination tolerance	1.0e-15