XPRESS

Contents

1 Introduction

This document describes the GAMS/XPRESS linear and mixed-integer programming solver. The GAMS/XPRESS solver is based on the XPRESS-MP Optimization Subroutine Library, and runs only in conjunction with the GAMS modeling system.

GAMS/XPRESS (also simply referred to as XPRESS) is a versatile, high - performance optimization system. The system integrates a powerful simplex-based LP solver, a MIP module with cut generation for integer programming problems and a barrier module implementing a state-of-the-art interior point algorithm for very large LP problems.

The GAMS/XPRESS solver is installed automatically with your GAMS system. Without a license, it will run in student or demonstration mode (i.e. it will solve small models only). If your GAMS license includes XPRESS, there is no size or algorithm restriction imposed by the license, nor is any separate licensing procedure required.

2 Usage

If you have installed the system and configured XPRESS as the default LP, RMIP[1] and MIP solver, all LP, RMIP and MIP models without a specific solver option will use XPRESS. If you installed another solver as the default, you can explicitly request a particular model to be solved by XPRESS by inserting the statement

    `option LP = xpress;` { or `MIP` or `RMIP` }

somewhere before the `solve` statement.

The following standard GAMS options can be used to control XPRESS-MP:

- `option reslim =x;` or `modelname.reslim = x;`

  Stop the algorithm after `x` seconds and report the best solution found so far. `Modelname` is the name of the model as specified in a previous `model` statement.

---

[1]RMIP means: Relaxed Mixed Integer Programming. You can solve a MIP model as an RMIP. This will ignore the integer restrictions and thus solves the problem as an LP.

- `option iterlim=n;` or `modelname.iterlim = n;`

  Stop the algorithm after n simplex iterations and report the best solution found so far. For MIP models, this places a cumulative limit on simplex iterations for the relaxed LP and the nodes of the B&B tree. `Modelname` is the name of the model as specified in a previous `model` statement.

- `option sysout=on;`

  Echo more detailed information about the solution process to the listing file.

- `option optcr=x;`

  In a MIP stop the search as soon as the relative gap is less than `x`.

- `option optca=x;`

  In a MIP stop the search as soon as the absolute gap is less than `x`.

- `option bratio=x;`

  Determines whether or not an advanced basis is passed on to the solver. `Bratio=1` will always ignore an existing basis (in this case XPRESS-MP will use a crash routine to find a better basis than an all-slack basis), while `bratio=0` will always accept an existing basis. Values between 0 and 1 use the number of non-basic variables found to determine if a basis is likely to be good enough to start from.

- `modelname.prioropt = 1;`

  Turns on usage of user-specified priorities. Priorities can be assigned to integer and binary variables using the syntax: `variablename.prior = x;`. Default priorities are 0.0. Variables with a priority v1 are branched upon earlier than variables with a priority v2 if v1<v2.

- `modelname.nodlim = n;`

  Specifies a node limit for the Branch-and-Bound search. When the number of nodes exceeds this number the search is stopped and the best integer solution found (if any) is reported back to GAMS. The default value of 0 indicates: no node limit.

In general this is enough knowledge to solve your models. In some cases you may want to use some of the XPRESS options to gain further performance improvements or for other reasons.

3   Options

Options can be specified in a file called *xpress.opt*. The syntax is rather simple: a line in the option file can be one of the following:

- An empty line or a line consisting only of blanks.

- A comment line, which is a line in which the first non-blank character is an asterisk '*'. The remainder of the line is ignored.

- An option, which consists of a keyword followed by a value.

An example of a valid option file is:

```
* sample XPRESS-MP options file
algorithm simplex
presolve    0
IterLim     50000
```

Keywords are not case sensitive. I.e. whether you specify `iterlim`, `ITERLIM`, or `Iterlim` the same option is set. To use an options file you specify a model suffix `modelname.optfile=1;` or use command line options `optfile=1`.

The tables that follow contain the XPRESS options. They are organized by function (e.g. LP or MIP) and also by type: some options control the behavior of the GAMS/XPRESS link and will be new even to experienced XPRESS users, while other options exist merely to set control variables in the XPRESS library and may be familiar to XPRESS users.

3.1 General Options

The following general options control the behavior of the GAMS/XPRESS link.

Option	Description	Default
advbasis	0: don't use advanced basis provided by GAMS 1: use advanced basis provided by GAMS This option overrides the GAMS BRATIO option.	Determined by GAMS
algorithm	simplex: use simplex solver barrier: use barrier algorithm This option is used to select the barrier method to solve LP's. By default the barrier method will do a crossover to find a basic solution.	simplex
basisout	If specified an MPS basis file is written. In general this option is not used in a GAMS environment, as GAMS maintains basis information for you automatically.	Don't write a basis file.
iterlim	Sets the iteration limit for simplex algorithms. When this limit is reached the system will stop and report the best solution found so far. Overrides the GAMS ITERLIM option.	10000
mpsoutputfile	If specified XPRESS-MP will generate an MPS file corresponding to the GAMS model. The argument is the file name to be used. It can not have an extension: XPRESS-MP forces the extension to be .MAT even if an extension was specified. You can prefix the file name with a path.	Don't write an MPS file.
reform	If true, the link will try to reformulate the model by removing the objective variable and the equation it appears in and replacing these with an objective function. If false, or if reformulation is not possible, the objective variable is not removed and the problem passed to the optimizer will have a very simple objective: the objective variable.	1
rerun	Applies only in cases where presolve is turned on and the model is diagnosed as infeasible or unbounded. If rerun is nonzero, we rerun the model using primal simplex with presolve turned off in hopes of getting better diagnostic information. If rerun is zero, no good diagnostic information exists, so we return no solution, only an indication of unboundedness/infeasibility.	0
reslim	Sets the resource limit. When the solver has used more than this amount of CPU time (in seconds) the system will stop the search and report the best solution found so far. Overrides the GAMS RESLIM option.	1000.0

The following general options set XPRESS library control variables, and can be used to fine-tune XPRESS.

Option	Description	Default
crash	A crash procedure is used to quickly find a good basis. This option is only relevant when no advanced basis is available. 0: no crash 1: singletons only (one pass) 2: singletons only (multi-pass) 3: multiple passes through the matrix considering slacks 4: multiple passes (<= 10), but do slacks at the end >10: as 4 but perform n-10 passes 100: default crash behavior of XPRESS-MP version 6	0 when GAMS provides an advanced basis, and 2 otherwise

Option	Description	Default
extrapresolve	The initial number of extra elements to allow for in the presolve. The space required to store extra presolve elements is allocated dynamically, so it is not necessary to set this control. In some cases, the presolve may terminate early if this is not increased.	automatic
lpiterlimit	Sets the iteration limit for simplex algorithms. For MIP models, this is a per-node iteration limit for the B&B tree. Overrides the iterlim option.	MAXINT
mpsnamelength	Maximum length of MPS names in characters. Internally it is rounded up to the smallest multiple of 8. MPS names are right padded with blanks. Maximum value is 64.	0
presolve	-1: Presolve applied, but a problem will not be declared infeasible if primal infeasibilities are detected. The problem will be solved by the LP optimization algorithm, returning an infeasible solution, which can sometimes be helpful. 0: Presolve not applied. 1: Presolve applied. 2: Presolve applied, but redundant bounds are not removed. This can sometimes increase the efficiency of the barrier algorithm. As XPRESS-MP does a basis preserving presolve, you don't have to turn off the presolver when using an advanced basis.	1
scaling	Bitmap to determine how internal scaling is done. If set to 0, no scaling will take place. The default of 35 implies row and column scaling done by the maximum element method. Bit 0 = 1: Row scaling. Bit 1 = 2: Column scaling. Bit 2 = 4: Row scaling again. Bit 3 = 8: Maximin. Bit 4 = 16: Curtis-Reid. Bit 5 = 32: Off implies scale by geometric mean, on implies scale by maximum element. Not applicable for maximin and Curtis-Reid scaling.	35
threads	Controls the number of threads to use. Positive values will be compared to the number of available cores detected and reduced if greater than this amount. Non-positive values are interpreted as the number of cores to leave free so setting `threads` to 0 uses all available cores while setting `threads` to -1 leaves one core free for other tasks.	1
trace	Control of the infeasibility diagnosis during presolve - if nonzero, infeasibility will be explained.	0

3.2   LP Options

The following options set XPRESS library control variables, and can be used to fine-tune the XPRESS LP solver.

Option	Description	Default
bigmmethod	0: for phase I / phase II 1: if 'big M' method to be used	automatic
bigm	The infeasibility penalty used in the "Big M" method	automatic
defaultalg	1: automatic 2: dual simplex 3: primal simplex 4: Newton barrier	1

Option	Description	Default
etatol	Zero tolerance on eta elements. During each iteration, the basis inverse is premultiplied by an elementary matrix, which is the identity except for one column the eta vector. Elements of eta vectors whose absolute value is smaller than `etatol` are taken to be zero in this step.	1.0e-12
feastol	This is the zero tolerance on right hand side values, bounds and range values. If one of these is less than or equal to `feastol` in absolute value, it is treated as zero.	1.0e-6
invertfreq	The frequency with which the basis will be inverted. A value of -1 implies automatic.	-1
invertmin	The minimum number of iterations between full inversions of the basis matrix.	3
lplog	The frequency at which the simplex iteration log is printed. `n < 0:` detailed output every -n iterations `n = 0:` log displayed at the end of the solution process `n > 0:` summary output every n iterations	100
matrixtol	The zero tolerance on matrix elements. If the value of a matrix element is less than or equal to `matrixtol` in absolute value, it is treated as zero.	1.0e-9
optimalitytol	This is the zero tolerance for reduced costs. On each iteration, the simplex method searches for a variable to enter the basis which has a negative reduced cost. The candidates are only those variables which have reduced costs less than the negative value of `optimalitytol`.	1.0e-6
penalty	Minimum absolute penalty variable coefficient used in the "Big M" method.	automatic
pivottol	The zero tolerance for matrix elements. On each iteration, the simplex method seeks a nonzero matrix element to pivot on. Any element with absolute value less than `pivottol` is treated as zero for this purpose.	1.0e-9
pricingalg	This determines the pricing method to use on each iteration, selecting which variable enters the basis. In general Devex pricing requires more time on each iteration, but may reduce the total number of iterations, whereas partial pricing saves time on each iteration, although possibly results in more iterations. `-1:` if partial pricing is to be used `0:` if the pricing is to be decided automatically. `1:` if DEVEX pricing is to be used	0
relpivottol	At each iteration a pivot element is chosen within a given column of the matrix. The relative pivot tolerance, `relpivottol`, is the size of the element chosen relative to the largest possible pivot element in the same column.	1.0e-6

3.3 MIP Options

In some cases, the branch-and-bound MIP algorithm will stop with a proven optimal solution or when unboundedness or (integer) infeasibility is detected. In most cases, however, the global search is stopped through one of the generic GAMS options:

1. `iterlim` (on the cumulative pivot count), `reslim` (in seconds of CPU time),

2. `optca` & `optcr` (stopping criteria based on gap between best integer solution found and best possible) or

3. `nodlim` (on the total number of nodes allowed in the B&B tree).

It is also possible to set the `maxnode` and `maxmipsol` options to stop the global search: see the table of XPRESS control variables for MIP below.

The following options control the behavior of the GAMS/XPRESS link on MIP models.

Option	Description	Default
loadmipsol	If true, the initial point provided by GAMS will be passed to the optimizer *to be treated as an integer feasible point*. The optimizer uses the values for the discrete variables only: the values for the continuous variables are ignored and are calculated by fixing the integer variables and reoptimizing. In some cases, loading an initial MIP solution can improve performance. In addition, there will always be a feasible solution to return.	0
mipcleanup	If nonzero, clean up the integer solution obtained, i.e. round and fix the discrete variables and re-solve as an LP to get some marginal values for the discrete vars.	1
miptrace	A miptrace file with the specified name will be created. This file records the best integer and best bound values every `miptracenode` nodes and at `miptracetime`-second intervals.	none
miptracenode	Specifies the node interval between entries to the `miptrace` file.	100
miptracetime	Specifies the time interval, in seconds, between entries to the `miptrace` file.	5

The following options set XPRESS library control variables, and can be used to fine-tune the XPRESS MIP solver.

Option	Description	Default
backtrack	This determines how the next node in the tree search is selected for processing. 1: If `miptarget` is not set, choose the node with the best estimate. If `miptarget` is set (by the user or by the global search previously finding an integer solution), the choice is based on the Forrest-Hirst-Tomlin Criterion, which takes into account the best current integer solution and seeks a new node which represents a large potential improvement. 2: Always choose the node with the best estimated solution. 3: Always choose the node with the best bound on the solution.	3
breadthfirst	If `nodeselection = 4`, this determines the number of nodes to include in a breadth-first search.	10
covercuts	The number of rounds of lifted cover inequalities at the top node. A lifted cover inequality is an additional constraint that can be particularly effective at reducing the size of the feasible region without removing potential integral solutions. The process of generating these can be carried out a number of times, further reducing the feasible region, albeit incurring a time penalty. There is usually a good payoff from generating these at the top node, since these inequalities then apply to every subsequent node in the tree search.	20
cutdepth	Sets the maximum depth in the tree search at which cuts will be generated. Generating cuts can take a lot of time, and is often less important at deeper levels of the tree since tighter bounds on the variables have already reduced the feasible region. A value of 0 signifies that no cuts will be generated. N.B.: does not affect cutting on the root node.	-1: automatic
cutfreq	This specifies the frequency at which cuts are generated in the tree search. If the depth of the node modulo `cutfreq` is zero, then cuts will be generated.	-1: automatic
cutstrategy	This specifies the cut strategy. An aggressive cut strategy, generating a greater number of cuts, will result in fewer nodes to be explored, but with an associated time cost in generating the cuts. The fewer cuts generated, the less time taken, but the greater subsequent number of nodes to be explored. -1: Automatic selection of either the conservative or aggressive cut strategy. 0: No cuts. 1: Conservative cut strategy. 2: Aggressive cut strategy.	-1
degradefactor	Factor to multiply estimated degradations associated with an unexplored node in the tree. The estimated degradation is the amount by which the objective function is expected to worsen in an integer solution that may be obtained through exploring a given node.	1.0

Option	Description	Default
gomcuts	The number of rounds of Gomory cuts at the top node. These can always be generated if the current node does not yield an integral solution. However, Gomory cuts are not usually as effective as lifted cover inequalities in reducing the size of the feasible region.	2
maxmipsol	This specifies a limit on the number of integer solutions to be found (the total number, not necessarily the number of distinct solutions). 0 means no limit.	0
maxnode	The maximum number of nodes that will be explored. If the GAMS nodlim model suffix is set, that setting takes precedence.	1e8
mipabscutoff	If the user knows that they are interested only in values of the objective function which are better than some value, this can be assigned to mipabscutoff. This allows the Optimizer to ignore solving any nodes which may yield worse objective values, saving solution time.	automatic
mipabsstop	Absolute stopping criterion for the global search. The search will stop when the absolute gap (i.e. the difference between the best bound and the best incumbent) is within this tolerance. Similar to the GAMS optca option, but if XPRESS stops on this test it will claim a proven optimal solution.	0
mipaddcutoff	The amount to add to the objective function of the best integer solution found to give the new cutoff. Once an integer solution has been found whose objective function is equal to or better than mipabscutoff, improvements on this value may not be interesting unless they are better by at least a certain amount. If mipaddcutoff is nonzero, it will be added to mipabscutoff each time an integer solution is found which is better than this new value. This cuts off sections of the tree whose solutions would not represent substantial improvements in the objective function, saving processor time. Note that this should usually be set to a negative number for minimization problems, and positive for maximization problems. Notice further that the maximum of the absolute and relative cut is actually used.	+/-1.0e-5
miplog	Global print control 0: No printout in global. 1: Only print out summary statement at the end. 2: Print out detailed log at all solutions found. 3: Print out detailed eight-column log at each node. n < 0: Print out summary six-column log at each -n nodes, or when a new solution is found	-100
mippresolve	Bitmap determining type of integer processing to be performed. If set to 0, no processing will be performed. Bit 0: Reduced cost fixing will be performed at each node. This can simplify the node before it is solved, by deducing that certain variables' values can be fixed based on additional bounds imposed on other variables at this node. Bit 1: Primal reductions will be performed at each node. Uses constraints of the node to tighten the range of variables, often resulting in fixing their values. This greatly simplifies the problem and may even determine optimality or infeasibility of the node before the simplex method commences. Bit 2: Unused - previously controlled probing. Now controlled by the preprobing option. Bit 3: If node preprocessing is allowed to change bounds on continuous columns. Bit 4: Dual reductions will be performed at each node.	automatic

Option	Description	Default
miprelcutoff	Percentage of the LP solution value to be added to the value of the objective function when an integer solution is found, to give the new value of `mipabscutoff`. The effect is to cut off the search in parts of the tree whose best possible objective function would not be substantially better than the current solution.	1.0e-4
miprelstop	Relative stopping criterion for the global search. The search will stop when the relative gap is within this tolerance, i.e when: —bestFound - bestBound— ¡= miprelstop * bestFound. Similar to the GAMS `optcr` option, but if XPRESS stops on this test it will claim a proven optimal solution.	0
miptarget	The target objective value for integer solutions. This is automatically set after solving the LP unless set by the user.	1e40
miptol	This is the tolerance within which a decision variable's value is considered to be integral.	5.0e-6
nodeselection	This determines which nodes will be considered for solution once the current node has been solved. 1: *Local first*: Choose among the two descendant nodes, if none among all active nodes. 2: *Best first*: All nodes are always considered. 3: *Depth first*: Choose deepest node, but explore two descendents first. 4: *Best first, then local first*: All nodes are considered for the first `breadthfirst` nodes, after which the usual default behavior is resumed.	automatic
objGoodEnough	If set, stop the search once a solution if found with objective value as good or better than this value.	none
preprobing	This determines the amount of probing to perform on binary variables during presolve. This is done by fixing a binary to each of its values in turn and analyzing the implications. -1: Choose automatically. 0: Disabled. 1: Light probing - only a few implications will be examined. 2: Full probing - all implications for all binaries will be examined. 3: Full probing and repeat as long as the problem is significantly reduced.	automatic
pseudocost	The default pseudo cost used in estimation of the degradation associated with an unexplored node in the tree search. A pseudo cost is associated with each integer decision variable and is an estimate of the amount by which the objective function will be worse if that variable is forced to an integral value.	0.01
sleepOn-ThreadWait	Threads during a MIP solve can now busy-wait instead of going to sleep when waiting for work. This is to overcome a performance issue with modern speed-stepping CPUs, which might step down to a lower clock frequency when the load is less than 100%.	no
treecovercuts	The number of rounds of lifted cover inequalities generated at nodes other than the top node in the tree. Compare with the description for `covercuts`.	2
treegomcuts	The number of rounds of Gomory cuts generated at nodes other than the first node in the tree. Compare with the description for `gomcuts`.	0
varselection	This determines how to combine the pseudo costs associated with the integer variables to obtain an overall estimated degradation in the objective function that may be expected in any integer solution from exploring a given node. It is relevant only if `backtrack` has been set to 1. 1: Sum the minimum of the 'up' and 'down' pseudo costs. 2: Sum all of the 'up' and 'down' pseudo costs. 3: Sum the maximum, plus twice the minimum of the 'up' and 'down' pseudo costs. 4: Sum the maximum of the 'up' and 'down' pseudo costs. 5: Sum the 'down' pseudo costs. 6: Sum the 'up' pseudo costs.	1

3.4   MIP Solution Pool Options

Typically, XPRESS finds a number of integer feasible points during its global search, but only the final solution is available. The MIP solution pool capability makes it possible to store multiple integer feasible points (aka solutions) for later processing. The MIP solution pool operates in one of two modes: by default (`solnpoolPop = 1`) the global search is not altered, but with (`solnpoolPop = 2`) a selected set (potentially all) of the integer feasible solutions are enumerated.

The MIP enumeration proceeds until all MIP solutions are enumerated or cut off, or until a user-defined limit is reached. Whenever a new solution is generated by the enumerator, it is presented to the solution pool manager. If there is room in the pool, the new solution is added. If the pool is full, a *cull round* is performed to select a number of solutions to be thrown out - these solutions can be those stored in the pool and/or the new solution. Solutions can be selected for culling based on their MIP objective value and/or the overall diversity of the solutions in the pool. If neither is chosen, a default choice is made to throw out one solution based on objective values. Whenever a solution is thrown out based on its MIP objective, the enumeration space is pruned based on the cutoff defined by this objective value.

By default, the capacity of the pool is set very large, as is the number of cull rounds to perform, so selecting only `solnpoolPop = 2` will result in full enumeration. However, many different strategies can be executed by setting the solution pool options. For example, to choose the *N*-best solutions, simply set the solution pool capacity to *N*. When the pool is full, new solutions will force a cull round, and the default is to reject one solution based on its objective and update the cutoff accordingly. To generate all solutions with an objective as good as *X*, leave the pool capacity set at a high level but set the cutoff to *X* using the `mipabscutoff` option. To return the *N*-first solutions, set the solution pool capacity to *N* and `solnpoolCullRounds = 0`: as soon as the pool is full the enumeration will stop on the cull round limit.

A number of other strategies for controlling the solution pool behavior are possible by combining the different options provided. Several working examples are provided in the GAMS Test Library in models `xpress03.gms`, `xpress04.gms`, and `xpress05.gms`.

Option	Description	Default
solnpool	If set, the integer feasible solutions generated during the global search will be saved to a solution pool. A GDX file whose name is given by this option will be created and will contain an index to separate GDX files containing the individual solutions in the solution pool.	none
solnpool-Capacity	Maximum number of solutions to store in the solution pool. This is only relevant when using the MIP solution enumerator.	999999999
solnpool-CullRounds	Limits the rounds of culls performed due to a full solution pool.	999999999
solnpool-CullDiversity	When performing a round of culls due to a full solution pool, this control sets the maximum number to cull based on the diversity of the solutions in the pool.	-1
solnpool-CullObj	When performing a round of culls due to a full solution pool, this control sets the maximum number to cull based on the MIP objective function.	-1
solnpool-DupPolicy	Determines whether to check for duplicate solutions when adding to the MIP solution pool, and what method is used to check for duplicates. 0: keep all solutions found 1: compare all vars, exact matches discarded 2: compare rounded discrete and exact continuous vars 3: compare rounded discrete vars only	3
solnpoolPop	Determines what method is used to populate the solution pool. 1: the default global search routine 2: the MIP solution enumerator By default the MIP solution pool merely stores the incumbent solutions that are found during the global search, without changing the behavior of the search itself. In constrast, the MIP solution enumerator makes it possible to enumerate all or many of the feasible solutions for the MIP, instead of searching for the best solution.	1

Option	Description	Default
solnpoolPrefix	Prefix for the names of the GDX files containing the solutions from the solution pool.	soln
solnpool-Verbosity	Controls the amount of output generated for solution pool activity. -1: no output 0: output only messages coming from the XPRESS libraries 1: add some messages logging the effect of solution pool options 2: debugging mode	0

3.5   Newton-Barrier Options

The barrier method is invoked by default for quadratic problems, and can be selected for linear models by using one of the options

```
algorithm      barrier
defaultalg     4
```

The barrier method is likely to use more memory than the simplex method. No warm start is done, so if an advanced basis exists, you may not wish to use the barrier solver.

The following options set XPRESS library control variables, and can be used to fine-tune the XPRESS barrier solver.

Option	Description	Default
barcrash	Determines the type of crash used for the crossover. A good crash basis will reduce the number of iterations required to crossover to an optimal basis. The possible values increase proportionally to their time consumption. 0: Turn off all crash procedures. 1 - 6: Available strategies with 1 being conservative and 6 being aggressive.	4
bardualstop	Absolute zero tolerance for dual infeasibilities. The dual constraint residuals must be be smaller than this value for the current point to be considered dual feasible.	automatic
bargapstop	Absolute zero tolerance for duality gap. The gap between the primal and dual solutions must be smaller than this value for the current point to be considered optimal.	automatic
barindeflimit	Maximum number of consecutive indefinite barrier iterations to perform for a QP before reporting that the problem is indefinite.	15
bariterlimit	Maximum number of barrier iterations	200
barorder	Controls the Cholesky factorization in the barrier method. 0: Choose automatically. 1: Minimum degree method. This selects diagonal elements with the smallest number of nonzeros in their rows or columns. 2: Minimum local fill method. This considers the adjacency graph of nonzeros in the matrix and seeks to eliminate nodes that minimize the creation of new edges. 3: Nested dissection method. This considers the adjacency graph and recursively seeks to separate it into non-adjacent pieces.	automatic
baroutput	Controls the level of solution output generated by the barrier method. 0: No output. 1: Output at each iteration.	1
barprimalstop	Absolute zero tolerance for primal infeasibilities. The constraint residuals must be be smaller than this value for the current point to be considered primal feasible.	automatic

Option	Description	Default
barstart	Controls the computation of the starting point for the barrier method. 0: Choose automatically. 1: Uses simple heuristics to compute the starting point based on the magnitudes of the matrix entries. 2: Uses the pseudoinverse of the constraint matrix to determine primal and dual initial solutions. Less sensitive to scaling and numerically more robust, but in several case less efficient than 1.	automatic
barstepstop	Minimal step size for barrier iterations. If the step size is smaller, the current solution will be returned.	1e-10
crossover	Determines whether the barrier method will cross over to the simplex method when at optimal solution has been found, in order to provide an end basis. 0: No crossover. 1: Crossover to a basic solution.	1
eigenvaluetol	Zero tolerance for negative eigenvalues of quadratic matrices: used in the convexity checker. A matrix with eigenvalue(s) smaller than the negative of this tolerance is not considered to be positive semi-definite.	1e-6
ifCheck-Convexity	Determines if the convexity of the problem is checked before optimization (for quadratic models). This check takes some time, so can be disabled for models known to be convex. 0: Turn off convexity checking. 1: Turn on convexity checking.	1

4 Helpful Hints

The comments below should help both novice and experienced GAMS users to better understand and make use of GAMS/XPRESS.

- **Infeasible and unbounded models** The fact that a model is infeasible/unbounded can be detected at two stages: during the presolve and during the simplex or barrier algorithm. In the first case we cannot recover a solution, nor is any information regarding the infeasible/unbounded constraint or variable provided (at least in a way that can be returned to GAMS). In such a situation, the GAMS link will automatically rerun the model using primal simplex with presolve turned off (this can be avoided by setting the rerun option to 0). It is possible (but very unlikely) that the simplex method will solve a model to optimality while the presolve claims the model is infeasible/unbounded (due to feasibility tolerances in the simplex and barrier algorithms).

- The barrier method does not make use of iterlim. Use bariterlim in an options file instead. The number of barrier iterations is echoed to the log and listing file. If the barrier iteration limit is reached during the barrier algorithm, XPRESS continues with a simplex algorithm, which will obey the iterlim setting.

- Semi-integer variables are not implemented in the link, nor are they supported by XPRESS; if present, they trigger an error message.

- SOS1 and SOS2 variables are required by XPRESS to have lower bounds of 0 and nonnegative upper bounds.