

XPRESS

Contents

1	Introduction	1
2	Usage	1
3	Options	2
3.1	General Options	3
3.2	LP Options	4
3.3	MIP Options	5
3.4	Newton-Barrier Options	9
4	Helpful Hints	9

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, RMIP¹ 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.

¹RMIP 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
<code>advbasis</code>	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
<code>algorithm</code>	<code>simplex</code> : use simplex solver <code>barrier</code> : 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.	<code>simplex</code>
<code>basisout</code>	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.
<code>iterlim</code>	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
<code>mpsoutputfile</code>	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 <code>.MAT</code> even if an extension was specified. You can prefix the file name with a path.	Don't write an MPS file.
<code>rerun</code>	Applies only in cases where presolve is turned on and the model is diagnosed as infeasible or unbounded. If <code>rerun</code> is nonzero, we rerun the model using primal simplex with presolve turned off in hopes of getting better diagnostic information. If <code>rerun</code> is zero, no good diagnostic information exists, so we return no solution, only an indication of unboundedness/infeasibility.	1
<code>reslim</code>	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
<code>crash</code>	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
<code>extrapresolve</code>	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.	<code>automatic</code>
<code>lpiterlimit</code>	Sets the iteration limit for simplex algorithms. For MIP models, this is a per-node iteration limit for the B&B tree. Overrides the <code>iterlim</code> option.	<code>MAXINT</code>
<code>mpsnamelength</code>	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
<code>presolve</code>	<p>-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.</p> <p>0: Presolve not applied.</p> <p>1: Presolve applied.</p> <p>2: Presolve applied, but redundant bounds are not removed. This can sometimes increase the efficiency of the barrier algorithm.</p> <p>As XPRESS-MP does a basis preserving presolve, you don't have to turn off the presolver when using an advanced basis.</p>	1
<code>scaling</code>	<p>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.</p> <p>Bit 0 = 1: Row scaling.</p> <p>Bit 1 = 2: Column scaling.</p> <p>Bit 2 = 4: Row scaling again.</p> <p>Bit 3 = 8: Maximin.</p> <p>Bit 4 = 16: Curtis-Reid.</p> <p>Bit 5 = 32: Off implies scale by geometric mean, on implies scale by maximum element. Not applicable for maximin and Curtis-Reid scaling.</p>	35
<code>threads</code>	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 <code>threads</code> to 0 uses all available cores while setting <code>threads</code> to -1 leaves one core free for other tasks.	1
<code>trace</code>	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
<code>bigmmethod</code>	0: for phase I / phase II 1: if 'big M' method to be used	automatic
<code>bigm</code>	The infeasibility penalty used in the "Big M" method	automatic
<code>defaultalg</code>	1: automatic 2: dual simplex 3: primal simplex 4: Newton barrier	1
<code>etamol</code>	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 <code>etamol</code> are taken to be zero in this step.	1.0e-12
<code>feastol</code>	This is the zero tolerance on right hand side values, bounds and range values. If one of these is less than or equal to <code>feastol</code> in absolute value, it is treated as zero.	1.0e-6
Option	Description	Default
<code>invertfreq</code>	The frequency with which the basis will be inverted. A value of -1 implies automatic.	-1
<code>invertmin</code>	The minimum number of iterations between full inversions of the basis matrix.	3
<code>lplog</code>	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
<code>matrixtol</code>	The zero tolerance on matrix elements. If the value of a matrix element is less than or equal to <code>matrixtol</code> in absolute value, it is treated as zero.	1.0e-9
<code>optimalitytol</code>	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 <code>optimalitytol</code> .	1.0e-6
<code>penalty</code>	Minimum absolute penalty variable coefficient used in the "Big M" method.	automatic
<code>pivottol</code>	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 <code>pivottol</code> is treated as zero for this purpose.	1.0e-9
<code>pricingalg</code>	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
<code>relpivottol</code>	At each iteration a pivot element is chosen within a given column of the matrix. The relative pivot tolerance, <code>relpivottol</code> , 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 unbound-ness 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 `maxmipso1` 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
<code>loadmipso1</code>	If true, the initial point provided by GAMS will be passed to the optimizer <i>to be treated as an integer feasible point</i> . 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
<code>mipcleanup</code>	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
<code>miptrace</code>	A <code>miptrace</code> file with the specified name will be created. This file records the best integer and best bound values every <code>miptracenode</code> nodes and at <code>miptracetime</code> -second intervals.	none
<code>miptracenode</code>	Specifies the node interval between entries to the <code>miptrace</code> file.	100
<code>miptracetime</code>	Specifies the time interval, in seconds, between entries to the <code>miptrace</code> file.	5

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

Option	Description	Default
<code>backtrack</code>	This determines how the next node in the tree search is selected for processing. 1: If <code>miptarget</code> is not set, choose the node with the best estimate. If <code>miptarget</code> 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
<code>breadthfirst</code>	If <code>nodeselection</code> = 4, this determines the number of nodes to include in a breadth-first search.	10
<code>covercuts</code>	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
<code>cpmaxcuts</code>	The initial maximum number of cuts that will be stored in the cut pool. During optimization, the cut pool is subsequently resized automatically.	100
<code>cpmaxelems</code>	The initial maximum number of nonzero coefficients which will be held in the cut pool. During optimization, the cut pool is subsequently resized automatically.	200
<code>cutdepth</code>	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.	0
<code>cutfreq</code>	This specifies the frequency at which cuts are generated in the tree search. If the depth of the node modulo <code>cutfreq</code> is zero, then cuts will be generated.	8

Option	Description	Default
<code>cutstrategy</code>	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
<code>degradefactor</code>	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
<code>gomcuts</code>	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
<code>maxmipsol</code>	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
<code>maxnode</code>	The maximum number of nodes that will be explored. If the GAMS nodlim model suffix is set, that setting takes precedence.	1e8
<code>mipabscutoff</code>	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 <code>mipabscutoff</code> . This allows the Optimizer to ignore solving any nodes which may yield worse objective values, saving solution time.	automatic
<code>mipaddcutoff</code>	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 <code>mipabscutoff</code> , improvements on this value may not be interesting unless they are better by at least a certain amount. If <code>mipaddcutoff</code> is nonzero, it will be added to <code>mipabscutoff</code> 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
<code>miplog</code>	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

Option	Description	Default
<code>mippresolve</code>	<p>Bitmap determining type of integer processing to be performed. If set to 0, no processing will be performed.</p> <p>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.</p> <p>Bit 1: Logical preprocessing will be performed at each node. This is performed on binary variables, often resulting in fixing their values based on the constraints. This greatly simplifies the problem and may even determine optimality or infeasibility of the node before the simplex method commences.</p> <p>Bit 2: Probing of binary variables is performed at the top node. This sets certain binary variables and then deduces effects on other binary variables occurring in the same constraints.</p>	automatic
<code>miprelcutoff</code>	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 <code>mipabscutoff</code> . 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
<code>miptarget</code>	The target objective value for integer solutions. This is automatically set after solving the LP unless set by the user.	1e40
<code>miptol</code>	This is the tolerance within which a decision variable's value is considered to be integral.	5.0e-6
<code>nodeselection</code>	<p>This determines which nodes will be considered for solution once the current node has been solved.</p> <p>1: <i>Local first:</i> Choose among the two descendant nodes, if none among all active nodes.</p> <p>2: <i>Best first:</i> All nodes are always considered.</p> <p>3: <i>Depth first:</i> Choose deepest node, but explore two descendants first.</p> <p>4: <i>Best first, then local first:</i> All nodes are considered for the first <code>breadthfirst</code> nodes, after which the usual default behavior is resumed.</p>	automatic
<code>pseudocost</code>	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
<code>treecovercuts</code>	The number of rounds of lifted cover inequalities generated at nodes other than the top node in the tree. Compare with the description for <code>covercuts</code> .	2
<code>treegomcuts</code>	The number of rounds of Gomory cuts generated at nodes other than the first node in the tree. Compare with the description for <code>gomcuts</code> .	0
<code>varselection</code>	<p>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 <code>backtrack</code> has been set to 1.</p> <p>1: Sum the minimum of the 'up' and 'down' pseudo costs.</p> <p>2: Sum all of the 'up' and 'down' pseudo costs.</p> <p>3: Sum the maximum, plus twice the minimum of the 'up' and 'down' pseudo costs.</p> <p>4: Sum the maximum of the 'up' and 'down' pseudo costs.</p> <p>5: Sum the 'down' pseudo costs.</p> <p>6: Sum the 'up' pseudo costs.</p>	1

3.4 Newton-Barrier Options

The barrier method is invoked 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
<code>bariterlimit</code>	Maximum number of barrier iterations	200
<code>crossover</code>	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

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.