FICO Xpress MATLAB Interface

Overview

The Xpress MATLAB interface is a tool that makes Xpress optimization algorithms available directly from within the MATLAB environment, enabling users to easily define mathematical programming models and solve them with Xpress from within the MATLAB environment.

The interface provides functions for solving linear, quadratic and quadratically constrained programming problems, and the mixed integer versions of these. All optimization functions are designed to take a model description as input and produce a solution as output.

Using the Xpress for MATLAB Toolbox

Please refer to the "Xpress Installation and Licensing User Guide" for instructions on Xpress installation. The MATLAB interface does not require a separate software license.

In order to make the Xpress functions available in MATLAB, the Xpress MATLAB path must be added to the MATLAB search path. This can be done either using the graphical "Set Path" dialog box or the command line.

Using the MATLAB graphical interface to set the search path

From the main MATLAB window, click on "File" → "Set Path…", then on the "Add Folder" button and select the "matlab" subfolder of your Xpress installation folder (typically 'c:\xpress\matlab').

You can also make this change permanent by clicking on the "Save" button.

Using the MATLAB command line to set the search path

The command to add the Xpress interface to MATLAB search path is:

```
>> addpath 'c:\xpressmp\matlab'
```

(assuming you installed Xpress on 'c:\xpressmp'), and this can be made permanent with the command

```
>> savepath
```

Verifying if Xpress works

You can verify that the Xpress MATLAB interface is working properly by executing the command

```
>> xprsver
```

inside MATLAB. In case everything is fine you should get something like:

```
FICO Xpress Optimizer 64-bit v21.00.02 (Hyper capacity)
(c) Copyright Fair Isaac Corporation 2010
```

Interface functions

The Xpress MATLAB interface is comprised of the following functions:

- 7 optimization functions (xprslp, xprsqp, xprsqcqp, xprsbip, xprsmip, xprsmiqp, xprsmiqcqp)
- 2 functions to set/get controls (xprsoptimset and xprsoptimget)
- 1 function to show the Xpress version (xprsver)

The next section documents each of these functions. Once the MATLAB search path has been configured, the same documentation will be also directly available in MATLAB, both from the drop down menu Help > Product Help, as a new Toolboxes section, and from the command line using the 'help' command(e.g. with 'help xprslp').

Problem matrices

Differently from MATLAB Optimization Toolbox minimization functions, that take two distinct matrices in input: one for inequality constraints and the other for equality constraints, Xpress interface functions take only one matrix for both types of constraints plus a vector that specifies the constraint type.

Therefore, if matrices A and Aeq (with rhs, respectively, b and beq) are used to solve a linear problem with the Optimization Toolbox's linprog function:

```
>> x = linprog(f, A, b, Aeq, beq, lb, ub);
```

the same problem can be solved with Xpress using the commands

```
>> rtype = [repmat('L',[1 size(A,1)]) repmat('E',[1 size(Aeq,1)])];
>> x = xprslp(f, [A; Aeq], [b; beq], rtype, lb, ub);
```

where the rtype vector indicates that rows from matrix A are of type 'L' (lower than) and rows from matrix Aeq are of type 'E' (equalities).

Setting and querying control and attributes

Optimization options can be specified with a mechanism similar to that used by the MATLAB Optimization Toolbox, that is via an options structure that specifies a list of Xpress controls and their

values. See function xprsoptimset and the 'Control Parameters' section of the Optimizer Reference manual for more details.

The xprsoptimset function also handle the conversion from the Optimization Toolbox options to the corresponding Xpress options for all cases where this makes sense.

Furthermore, after calling an Xpress optimization function, it is possible to retrieve the final value of any Xpress control or attribute. The list of control and attribute names to be returned must be specified in the "XPRSGET" field of the option argument, separated by blanks. For example

```
>> options= xprsoptimset('XPRSGET', 'LPOBJVAL LPSTATUS')
>> [x,fval,ef,output] = xprslp(f, A, b, [], lb, ub, options);
>> fval, output.LPOBJVAL
fval =
   -78
output =
    LPOBJVAL: -78
    LPSTATUS: 1
```

It is also possible to request that the output structure be filled with all Xpress control and attribute values by setting 'XPRSGET' to 'ALL'.

In the Xpress MATLAB interface, control and attribute names are always all uppercase and without the XPRS prefix.

Example

In this example we solve the sample problem from MATLAB's documentation page on the "linprog" function.

The problem at hand is:

minimize

$-5x1 - 4x2 - 6x3$

subject to

$x1 - x2 + x3 \leq 20$

$3x1 + 2x2 + 4x3 \leq 42$

$3x1 + 2x2 \leq 30$

$0 \leq x1, 0 \leq x2, 0 \leq x3$

First, enter the coefficients

```
>> f = [-5; -4; -6];
```

```
>> A = [1 -1  1
>>      3  2  4
>>      3  2  0];
>> b = [20; 42; 30];
>> lb = zeros(3,1);
```

Next, call the Xpress linear programming function.

```
>> [x,fval,exitflag,output,lambda] = xprslp(f,A,b,'L',lb);
```

Entering x, lambda.lin, and lambda.lower returns the following results:

```
x =
     0.0000
    15.0000
     3.0000
lambda.lin =
     0
     1.5000
     0.5000
lambda.lower =
     1.0000
     0
     0
```

Xpress MATLAB Interface functions

xprsbip

Solve binary integer programming problems with Xpress.

Syntax

x = xprsbip(f,A,b,rtype,x0,options)

[x,fval,exitflag,output] = xprsbip(...)

Description

Finds the minimum of a problem specified by

min f*x

st. A*x ≤/=/≥ b

x binary

A is an m x n matrix; f, b, rtype, lb, ub and x0 are vectors.

Input arguments rtype, x0 and options can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprsbip(f, A, b, rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprsbip(f, A, b, rtype)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

f	Linear objective function vector.
A	Matrix for linear constraints .
b	Vector for constraints rhs.
rtype	Character vector (string) giving the row types: L indicates a ≤ row; E indicates a = row; G indicates a ≥ row; N indicates a free row. If rtype = [], all rows are assumed to be of type 'L'. If rtype is a single character, all constraints are assigned the corresponding type.
x0	Optional initial known solution used to speed-up search.

options Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval Value of the objective function at the solution x.

exitflag Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1	Function converged to a solution x (MIPSTATUS=MIP_OPTIMAL).
0	Number of iterations exceeded iter limit (STOPSTATUS= STOP_ITERLIMIT).
-2	The problem is infeasible (MIPSTATUS=MIP_INFEAS) .
-4	Number of searched nodes exceeded limit (STOPSTATUS= STOP_NODELIMIT).
-5	Search time exceeded limit (STOPSTATUS= STOP_TIMELIMIT).
-8	Other stop reason, see MIPSTATUS and STOPSTATUS for details.

output Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See bintprog and the section 'Setting and querying control and attributes' for details.

See Also

xprsoptimset, bintprog

xprslp

Solve linear programming problems with Xpress.

Syntax

x = xprslp(f,A,b,rtype,lb,ub,options)

[x,fval,exitflag,output,lambda] = xprslp(...)

Description

Finds the minimum of a problem specified by

min f*x

st. A*x ≤/=/≥ b

lb ≤ x ≤ ub

A is an m x n matrix; f, b, rtype, lb, and ub are vectors.

Input arguments rtype, lb, ub and options can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprslp(f, A, b, rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprslp(f, A, b, rtype)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

f — Linear objective function vector.

A — Matrix for linear constraints.

b — Vector for constraints rhs.

rtype — Character vector (string) giving the row types:
L indicates a ≤ row;
E indicates a = row;
G indicates a ≥ row;
N indicates a free row.
If rtype = [], all rows are assumed to be of type 'L'.
If rtype is a single character, all constraints are assigned the corresponding type.

lb — Lower bounds. If lb = [] it means there are no lower bounds.
If lb is a scalar, x is uniformely bounded by that scalar.

ub — Upper bounds. If ub = [] it means there are no upper bounds.
If ub is a scalar, x is uniformely bounded by that scalar.

options — Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x — Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval — Value of the objective function at the solution x.

exitflag — Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1	Function converged to a solution x (LPSTATUS=OPTIMAL)
0	Number of iterations exceeded iter limit (LPSTATUS=UNFINISHED and STOPSTATUS=ITERLIMIT)
-2	No feasible point was found. (LPSTATUS=INFEAS) .
-3	Problem is unbounded (LPSTATUS=UNDOUNDED).
-8	Other stop reason, see LPSTATUS and STOPSTATUS for details.

output — Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See linprog and the section 'Setting and querying control and attributes' for details.

lambda — Structure containing the Lagrange multipliers at the solution x (separated by constraint type). The fields of the structure are:

lower	Lower bounds lb
upper	Upper bounds ub
lin	Linear constraints from matrix A

See Also

xprsoptimset, linprog

xprsmip

Solve mixed integer linear programming problems with Xpress.

Syntax

x = xprsmip(f,A,b,rtype,ctype, clim,sos,lb,ub,x0,options)

[x,fval,exitflag,output] = xprsmip(...)

Description

Finds the minimum of a problem specified by

min f*x

st. A*x ≤/=/≥ b

lb ≤ x ≤ ub

x in the domain specified by the ctype, clim and SOS arguments

A is an m x n matrix; f, b, rtype, ctype, clim, lb, ub and x0 are vectors; sos is a struct vector.

Input arguments rtype and following can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprsmip(f, A, b, rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprsmip(f, A, b, rtype)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

f Linear objective function vector.

A Matrix for linear constraints.

b Vector for constraints rhs.

rtype Character vector (string) giving the row types:
L indicates a ≤ row;
E indicates a = row;
G indicates a ≥ row;
N indicates a free row.
If rtype = [], all rows are assumed to be of type 'L'.
If rtype is a single character, all constraints are assigned the corresponding type.

ctype Character vector (string) giving the column types:

C or \0 continuous variables;
B binary variables;
I integer variables;
P partial integer variables;
S semi-continuous variables;
R semi-continuous integers.
If ctype = [], all columns are assumed to be of type 'C'.
If ctype is a single character, all columns are assigned the corresponding type.

clim — Vector containing the integer limits for the partial integer variables and lower bounds for semi-continuous and semi-continuous integer variables (column types 'P', 'S, 'R'). Values in the positions corresponding to all other columns are ignored.
clim is mandatory if there are any 'P', ' S' or 'R' columns.
If clim is a scalar, all columns are assigned to that same limit.

sos — Struct vector defining SOS sets. The number of SOS sets is given by the number of elements in the struct. The struct must contain the following fields:
sos(i).type: a character indicating the SOS type, either '1' or '2';
sos(i).ind: numeric vector with the indices of columns in the set (column indices start from 0);
sos(i).wt: numeric vector with the reference row weights corresponding to the columns in the sos(i).ind vector. It must have the same length as sos(i).ind.

lb — Lower bounds. If lb = [] it means there are no lower bounds.
If lb is a scalar, x is uniformely bounded by that scalar.

ub — Upper bounds. If ub = [] it means there are no upper bounds.
If ub is a scalar, x is uniformely bounded by that scalar.

x0 — Optional initial known solution used to speed-up search.

options — Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x — Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval — Value of the objective function at the solution x.

exitflag — Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1 Function converged to a solution x (MIPSTATUS=MIP_OPTIMAL).
0 Number of iterations exceeded iter limit (STOPSTATUS= STOP_ITERLIMIT).

-2 The problem is infeasible (MIPSTATUS=MIP_INFEAS) .
-4 Number of searched nodes exceeded limit (STOPSTATUS= STOP_NODELIMIT).
-5 Search time exceeded limit (STOPSTATUS= STOP_TIMELIMIT).
-8 Other stop reason, see MIPSTATUS and STOPSTATUS for details.

output Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See bintprog and the section ‘Setting and querying control and attributes' for details.

See Also

xprsoptimset, bintprog

xprsmiqcqp

Solve mixed integer quadratically constrained quadratic programming problems with Xpress.

Syntax

x = xprsmiqcqp(H,f,A,Q,b,rtype,ctype, clim,sos,lb,ub,x0,options)

[x,fval,exitflag,output] = xprsmiqcqp(...)

Description

Finds the minimum of a problem specified by

min $0.5*x^T*H*x+f*x$

st. $A*x + x^T *Q_i*x \leq/=/\geq b$

$lb \leq x \leq ub$

x in the domain specified by the ctype, clim and SOS arguments

H is an n x n matrix; A is an m x n matrix; f, b, rtype, ctype, clim,lb, ub and x0 are vectors; sos is a struct vector.

Input arguments rtype and following can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprsmiqcqp(H, f, A,Q,b,rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprsmiqcqp(H,f, A, b, rtype)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

H	Matrix for quadratic objective terms .
f	Linear objective function vector.
A	Matrix for the linear part of the constraints.
Q	cell array of length m with the n x n matrices for the quadratic terms of the constraints. If there is only one constraint (m=1), Q can be a simple double matrix instead of a cell array. For linear constraint, the corresponding Q{i} matrix can be set to [].
b	Vector for constraints rhs.
rtype	Character vector (string) giving the row types: L indicates a ≤ row;

E indicates a = row;
G indicates a ≥ row;
N indicates a free row.
If rtype = [], all rows are assumed to be of type 'L'.
If rtype is a single character, all constraints are assigned the corresponding type.

ctype — Character vector (string) giving the column types:
C or \0 continuous variables;
B binary variables;
I integer variables;
P partial integer variables;
S semi-continuous variables;
R semi-continuous integers.
If ctype = [], all columns are assumed to be of type 'C'.
If ctype is a single character, all columns are assigned the corresponding type.

clim — Vector containing the integer limits for the partial integer variables and lower bounds for semi-continuous and semi-continuous integer variables (column types 'P', 'S, 'R'). Values in the positions corresponding to all other columns are ignored.
clim is mandatory if there are any 'P', ' S' or 'R' columns.
If clim is a scalar, all columns are assigned to that same limit.

sos — Struct vector defining SOS sets. The number of SOS sets is given by the number of elements in the struct. The struct must contain the following fields:
sos(i).type: a character indicating the SOS type, either '1' or '2';
sos(i).ind: numeric vector with the indices of columns in the set (column indices start from 0);
sos(i).wt: numeric vector with the reference row weights corresponding to the columns in the sos(i).ind vector. It must have the same length as sos(i).ind.

lb — Lower bounds. If lb = [] it means there are no lower bounds.
If lb is a scalar, x is uniformely bounded by that scalar.

ub — Upper bounds. If ub = [] it means there are no upper bounds.
If ub is a scalar, x is uniformely bounded by that scalar.

x0 — Optional initial known solution used to speed-up search.

options — Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x — Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval — Value of the objective function at the solution x.

exitflag — Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1	Function converged to a solution x (MIPSTATUS=MIP_OPTIMAL).
0	Number of iterations exceeded iter limit (STOPSTATUS= STOP_ITERLIMIT).
-2	The problem is infeasible (MIPSTATUS=MIP_INFEAS) .
-4	Number of searched nodes exceeded limit (STOPSTATUS= STOP_NODELIMIT).
-5	Search time exceeded limit (STOPSTATUS= STOP_TIMELIMIT).
-8	Other stop reason, see MIPSTATUS and STOPSTATUS for details.

output — Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See quadprog and the section 'Setting and querying control and attributes' for details.

See Also

xprsoptimset, bintprog, quadprog

xprsmiqp

Solve mixed integer quadratic programming problems with Xpress.

Syntax

x = xprsmiqp(H,f,A,b,rtype,ctype, clim,sos,lb,ub,x0,options)

[x,fval,exitflag,output] = xprsmiqp(...)

Description

Finds the minimum of a problem specified by

min $0.5* x^T *H*x+f*x$

st. $A*x \leq/=/\geq b$

$lb \leq x \leq ub$

x in the domain specified by the ctype, clim and SOS arguments

H is an n x n matrix; A is an m x n matrix; f, b, rtype, ctype, clim,lb, ub and x0 are vectors; sos is a struct vector.

Input arguments rtype and following can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprsmiqp(H, f, A, b, rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprsmiqp(H,f, A, b, rtype)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

H	Matrix for quadratic objective terms.
f	Linear objective function vector.
A	Matrix for linear constraints.
b	Vector for constraints rhs.
rtype	Character vector (string) giving the row types: L indicates a ≤ row; E indicates a = row; G indicates a ≥ row;

N indicates a free row.
If rtype = [], all rows are assumed to be of type 'L'.
If rtype is a single character, all constraints are assigned the corresponding type.

ctype — Character vector (string) giving the column types:
C or \0 continuous variables;
B binary variables;
I integer variables;
P partial integer variables;
S semi-continuous variables;
R semi-continuous integers.
If ctype = [], all columns are assumed to be of type 'C'.
If ctype is a single character, all columns are assigned the corresponding type.

clim — Vector containing the integer limits for the partial integer variables and lower bounds for semi-continuous and semi-continuous integer variables (column types 'P', 'S, 'R'). Values in the positions corresponding to all other columns are ignored.
clim is mandatory if there are any 'P', ' S' or 'R' columns.
If clim is a scalar, all columns are assigned to that same limit.

sos — Struct vector defining SOS sets. The number of SOS sets is given by the number of elements in the struct. The struct must contain the following fields:
sos(i).type: a character indicating the SOS type, either '1' or '2';
sos(i).ind: numeric vector with the indices of columns in the set (column indices start from 0);
sos(i).wt: numeric vector with the reference row weights corresponding to the columns in the sos(i).ind vector. It must have the same length as sos(i).ind.

lb — Lower bounds. If lb = [] it means there are no lower bounds.
If lb is a scalar, x is uniformely bounded by that scalar.

ub — Upper bounds. If ub = [] it means there are no upper bounds.
If ub is a scalar, x is uniformely bounded by that scalar.

x0 — Optional initial known solution used to speed-up search.

options — Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x — Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval — Value of the objective function at the solution x.

exitflag — Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1	Function converged to a solution x (MIPSTATUS=MIP_OPTIMAL).
0	Number of iterations exceeded iter limit (STOPSTATUS= STOP_ITERLIMIT).
-2	The problem is infeasible (MIPSTATUS=MIP_INFEAS) .
-4	Number of searched nodes exceeded limit (STOPSTATUS= STOP_NODELIMIT).
-5	Search time exceeded limit (STOPSTATUS= STOP_TIMELIMIT).
-8	Other stop reason, see MIPSTATUS and STOPSTATUS for details.

output — Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See quadprog and the section 'Setting and querying control and attributes' for details.

See Also

xprsoptimset, bintprog, quadprog

xprsoptimget

Retrieve Xpress optimization options values.

Syntax

val = xprsoptimget(options,'param')
val = xprsoptimget(options,'param',default)

Description

val = xprsoptimget(options,'param') returns the value of the specified parameter in the optimization options structure options. The parameter name is case sensitive and must be a valid Xpress control parameter name.

val = xprsoptimget(options,'param',default) returns default if the specified parameter is not defined in the optimization options structure options.

Examples

This statement returns the value of the FEASTOL optimization control parameter in the structure called my_options.

```
val = xprsoptimget(my_options,'FEASTOL')
```

This statement returns the value of the FEASTOL optimization control parameter in the structure called my_options (as in the previous example) except that if the FEASTOL parameter is not defined, it returns the value 1e-6.

```
optnew = xprsoptimget(my_options,'FEASTOL',1e-6);
```

See Also

xprsoptimset

xprsoptimset

Create or edit Xpress optimization options structures.

Syntax

```
options = xprsoptimset('param1',value1,'param2',value2,...)
options = xprsoptimset
options = xprsoptimset(oldopts,'param1',value1,...)
options = xprsoptimset(oldopts,newopts)
```

Description

The function xprsoptimset creates an options structure that you can pass as an input argument to the Xpress optimization functions. You can use the options structure to change the default parameters for these functions.

options = xprsoptimset('param1',value1,'param2',value2,...) creates an optimization options structure called options, in which the specified parameters (param) have specified values. The parameter names are case sensitive and must be valid Xpress control parameter names.

xprsoptimset with no input returns a complete list of parameters with their default values.

options = xprsoptimset(oldopts,'param1',value1,...) creates a copy of oldopts, modifying or adding the specified parameters with the specified values.

options = xprsoptimset(oldopts,newopts) combines an existing options structure oldopts with a new options structure newopts. Any parameters in newopts with nonempty values overwrite the corresponding old parameters in oldopts.

In the last two cases, oldopts can be a MATLAB Toolbox option structure, in which case the following parameters are converted to the corresponding Xpress controls (others are ignored):

Display -> OUTPUTLOG, MIPLOG, LPLOG
MaxIter -> LPITERLIMIT
TolRLPFun ->OPTIMALITYTOL
MaxTime -> MAXTIME
MaxNode -> MAXNODE
NodeDisplayInterval -> MIPLOG
NodeSearchStrategy -> NODESELECTION
TolXInteger -> MIPTOL

Only options that are set to a non-empty value are taken into consideration.

Examples

This statement creates an optimization options structure called options in which the FEASTOL parameter is set to 1e-8 and the MAXMIPSOL parameter is set to 10.

```
options = xprsoptimset('FEASTOL',1e-8,'MAXMIPSOL',10)
```

This statement makes a copy of the options structure called options, changing the value of the PRESOLVE parameter and storing new values in optnew.

```
optnew = xprsoptimset(options,'PRESOLVE',0);
```

This statement creates an Xpress optimization options structure with control values corresponding to the 'final' value of the MATLAB Toolbox option Display.

```
options = xprsoptimset(optimset('Display', 'final'));
```

This statement returns an optimization options structure that contains all the parameter names and default values..

```
defaults = xprsoptimset
```

See Also

xprsoptimget

xprsqcqp

Solve quadratically constrained quadratic programming problems with Xpress.

Syntax

x = xprsqcqp(H,f,A,Q,b,rtype,lb,ub,options)

[x,fval,exitflag,output,lambda] = xprsqcqp(...)

Description

Finds the minimum of a problem specified by

min $0.5* x^T *H*x+f*x$

st. $A*x + x^T *Q_i*x \leq/=/\geq b$

$lb \leq x \leq ub$

H is an n x n matrix; A is an m x n matrix; Q is a cell array of n x n matrices; f, b, rtype, lb, and ub are vectors.

Input arguments rtype, lb, ub and options can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprsqcqp(H,f,A,Q, b,rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprsqcqp(…)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

H	Matrix for quadratic objective terms.
f	Linear objective function vector.
A	Matrix for the linear part of the constraints.
Q	cell array of length m with the n x n matrices for the quadratic terms of the constraints. If there is only one constraint (m=1), Q can be a simple double matrix instead of a cell array. For linear constraint, the corresponding Q{i} matrix can be set to [].
b	Vector for constraints rhs.
rtype	Character vector (string) giving the row types: L indicates a ≤ row;

E indicates a = row;
G indicates a ≥ row;
N indicates a free row.
If rtype = [], all rows are assumed to be of type 'L'.
If rtype is a single character, all constraints are assigned the corresponding type.

lb — Lower bounds. If lb = [] it means there are no lower bounds.
If lb is a scalar, x is uniformely bounded by that scalar.

ub — Upper bounds. If ub = [] it means there are no upper bounds.
If ub is a scalar, x is uniformely bounded by that scalar.

options — Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x — Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval — Value of the objective function at the solution x.

exitflag — Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1	Function converged to a solution x (LPSTATUS=OPTIMAL)
0	Number of iterations exceeded iter limit (LPSTATUS=UNFINISHED and STOPSTATUS=ITERLIMIT)
-2	No feasible point was found. (LPSTATUS=INFEAS) .
-3	Problem is unbounded (LPSTATUS=UNDOUNDED).
-8	Other stop reason, see LPSTATUS and STOPSTATUS for details.

output — Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See quadprog and the section 'Setting and querying control and attributes' for details.

lambda — Structure containing the Lagrange multipliers at the solution x (separated by constraint type). The fields of the structure are:

lower	Lower bounds lb
upper	Upper bounds ub
lin	Linear constraints from matrix A

See Also

xprsoptimset, quadprog

xprsqp

Solve quadratic programming problems with Xpress.

Syntax

x = xprsqp(H,f,A,b,rtype,lb,ub,options)

[x,fval,exitflag,output,lambda] = xprsqp(...)

Description

Finds the minimum of a problem specified by

min $0.5* x^T *H*x+f*x$

st. $A*x \leq/=/\geq b$

$lb \leq x \leq ub$

H is an n x n matrix; A is an m x n matrix; f, b, rtype, lb, and ub are vectors.

Input arguments rtype, lb, ub and options can be omitted, with the condition that, if one is omitted, also all the following ones must be omitted (as in x= xprsqp(H,f, A, b, rtype)). Omitting an input argument has the same effect as passing an empty array [].

All output arguments can be omitted too, again with the condition that, if one is omitted, also all the following ones must be omitted (as in [x, fval]= xprsqp(H,f, A, b, rtype)).

Note If the specified input bounds for a problem are inconsistent, the output x and fval are set to [].

Input Arguments

H	Matrix for quadratic objective terms.
f	Linear objective function vector.
A	Matrix for linear constraints.
b	Vector for constraints rhs.
rtype	Character vector (string) giving the row types: L indicates a ≤ row; E indicates a = row; G indicates a ≥ row; N indicates a free row. If rtype = [], all rows are assumed to be of type 'L'. If rtype is a single character, all constraints are assigned the corresponding type.
lb	Lower bounds. If lb = [] it means there are no lower bounds.

If lb is a scalar, x is uniformely bounded by that scalar.

ub — Upper bounds. If ub = [] it means there are no upper bounds.
If ub is a scalar, x is uniformely bounded by that scalar.

options — Options structure created with optimset or xprsoptimset functions. See xprsoptimset for more details.

Output Arguments

x — Solution found by the optimization function. If exitflag > 0, then x is a solution; otherwise, x is the value of the optimization routine when it terminated prematurely.

fval — Value of the objective function at the solution x.

exitflag — Integer identifying the reason the optimization algorithm terminated. The following lists the values of exitflag and the corresponding reasons the algorithm terminated.

1 Function converged to a solution x (LPSTATUS=OPTIMAL)
0 Number of iterations exceeded iter limit (LPSTATUS=UNFINISHED and STOPSTATUS=ITERLIMIT)
-2 No feasible point was found. (LPSTATUS=INFEAS) .
-3 Problem is unbounded (LPSTATUS=UNDOUNDED).
-8 Other stop reason, see LPSTATUS and STOPSTATUS for details.

output — Structure containing information about the optimization and, eventually, values of Xpress controls and attributes. See quadprog and the section 'Setting and querying control and attributes' for details.

lambda — Structure containing the Lagrange multipliers at the solution x (separated by constraint type). The fields of the structure are:

lower Lower bounds lb
upper Upper bounds ub
lin Linear constraints from matrix A

See Also

xprsoptimset, quadprog

xprsver

Display version number for Xpress .

Syntax

xprsver

Description

xprsver prints the version and release number for the Xpress software currently running.

Examples

Display the version:

xprsver

MATLAB display:

```
FICO Xpress Optimizer 64-bit v21.00.02 (Hyper capacity)
(c) Copyright Fair Isaac Corporation 2010
```

See Also

xprsoptimset, linprog