IBM ILOG CPLEX Optimization Studio Getting Started with CPLEX for MATLAB

Version 12 Release 4

Further acknowledgements

IBM ILOG CPLEX states these additional registered trademarks and acknowledgements.

Additional registered trademarks

Python is a registered trademark of the Python Software Foundation.

MATLAB is a registered trademark of The MathWorks, Inc.

Acknowledgement of use: dtoa routine of the gdtoa package

IBM ILOG CPLEX acknowledges use of the `dtoa` routine of the `gdtoa` package, available at

`http://www.netlib.org/fp/`.

The author of this software is David M. Gay.

(end of acknowledgement of use of `dtoa` routine of the `gdtoa` package)

Contents

Chapter 1. Introduction

An overview of CPLEX® for MATLAB.

CPLEX for MATLAB is an extension to IBM® ILOG® CPLEX Optimizers that allows a user to define optimization problems and solve them within MATLAB. Thus a student or practitioner who is using MATLAB can easily solve optimization problems within that framework.

Chapter 2. Installation of CPLEX for MATLAB

Install and set up IBM ILOG CPLEX Optimization Studio before using the CPLEX connector for MATLAB.

If you have not yet set up CPLEX, first follow those procedures for installation before you begin using the CPLEX connector for MATLAB.

Instructions for installation are available at the IBM Support Portal under the topic *Choose your task: Installation*.

Chapter 3. Integration with MATLAB

The menu items and windows used to solve optimization models are described.

CPLEX for MATLAB should be integrated into your MATLAB environment in order for you to take full advantage of its features.

When you have installed CPLEX for MATLAB and set the paths as described in the `readme.html` file, a new item is added to the Toolboxes section of the MATLAB **Start Button**. You can use the items on this menu to find more help about using CPLEX.

In addition, the online manuals for CPLEX for MATLAB have been added to the MATLAB Product Help, available from the drop down menu **Help > Product Help**.

Within the MATLAB Command Window, inline help is available for the CPLEX classes and functions. For example typing 'help cplexlp' will display information about the function `cplexlp`.

Chapter 4. Using CPLEX for MATLAB

Presents an overview of how to solve an optimization problem.

IBM ILOG CPLEX Optimizers provides a tool for solving optimization, or mathematical programming, problems.

The most basic mathematical programming problem is commonly referred to as Linear Programming (LP) problem. The basic form of an LP problem is:

Maximize (or Minimize)	`f*x`
subject to	`Aeq*x = beq` `Aineq*x <= bineq`
with these bounds	`l <= x <= u`

where `Aeq` and `Aineq` are matrices, `f`, `beq`, `bineq`, `l` and `u` are vectors such that the upper bounds `u(i)` and lower bounds `l(i)` may be positive infinity, negative infinity, or any real number. Both sparse and dense format can be used in all places where matrices/vectors are used.

The elements of data you provide as input for this LP problem are:

Objective function coefficients	`f`
Constraint coefficients	`Aeq` `Aineq`
Righthand sides	`beq` `bineq`
Upper and lower bounds	`u` `l`

The optimal solution that CPLEX computes and returns is:

Variables	`x`

CPLEX for MATLAB can also solve several extensions to LP:

- Quadratic Programming (QP) problems, where the LP objective function is expanded to include quadratic terms.
- Quadratically Constrained Programming (QCP) problems that include quadratic terms among the constraints. In fact, CPLEX can solve Second Order Cone Programming (SOCP) problems.
- Mixed Integer Programming (MIP) problems, where any or all of the LP, QP, or QCP variables are further restricted to take integer values in the optimal solution and where MIP itself is extended to include constructs like Special Ordered Sets (SOS), semi-continuous variables, and indicator variables.

- Least Squares (LSQ) problems, where the objective is to minimize a norm. The problem can be constrained linearly or quadratically, and the variables may be restricted to take integer values in the solution.

The standard MATLAB vector and matrix format is used for the elements of data that you need to provide. For example, the CPLEX for MATLAB Toolbox function `cplexlp` solves the problem specified by

```
min      f*x
st.      Aineq*x <= bineq
         Aeq*x    = beq
         lb <= x <= ub
```

where `f`, `bineq`, `beq`, `lb`, and `ub` are MATLAB vectors, and `Aineq` and `Aeq` are MATLAB matrices.

The vector `x` returned by the function call

```
x = cplexlp(f,Aineq,beq,Aeq,beq,lb,ub)
```

contains the optimal solution to the specified linear programming problem.

Provided in CPLEX for MATLAB is both a toolbox of functions and a class API. The toolbox contains functions for solving optimization problems, where the input matrices are provided to the function and results returned. With the class API, objects can be created, and those objects carry a state.

The benefits of using the Cplex class API include the ability to:
- build up a model by manipulating a Cplex object.
- use computation methods such as `Cplex.solve()` and `Cplex.refineConflict()` that modify the object so results can be queried as needed.
- perform restarts after manipulation.
- attach an output parser, a GUI with stop buttons, and other controls.

Chapter 5. Overview of the CPLEX for MATLAB APIs

CPLEX for MATLAB provides two APIs for solving mathematical programming problems, the toolbox functions and the Cplex class.

CPLEX for MATLAB Toolbox

The toolbox provides functions for solving a variety of mathematical programming problems.

The CPLEX for MATLAB Toolbox provides functions for solving a variety of mathematical programming problems. The toolbox functions are designed to take a model description as input and produce a solution as output.

For example:

```
x = cplexlp(f,Aineq,bineq,Aeq,beq,lb,ub)
```

finds the minimum of a linear programming problem specified by

```
min     f*x
st.     Aineq*x <= bineq
        Aeq*x    = beq
        lb <= x <= ub
```

The toolbox provides the functions `cplexlp`, `cplexqp` and `cplexbilp` to solve linear programming problems (LP), quadratic programming problems (QP) and binary integer programming problems (BILP).

The toolbox provides functions that support the solution of the basic problem types handled by CPLEX are:

- `cplexlp` for linear programming problems (LP),
- `cplexqp` for quadratic programming problems (QP) and
- `cplexbilp` for binary integer programming problems (BILP).

Functions that support the solution of additional problem types handled by CPLEX are provided. These functions are:

- `cplexqcp` for quadratically constrained programming problems (QCP),
- `cplexmilp` for mixed integer linear programming problems (MIP),
- `cplexmiqp` for mixed integer quadratic programming problems (MIQP) and
- `cplexmiqcp` for mixed integer quadratically constrained mixed integer programming problems (MIQCP).

The solution of least square problems is supported through the functions:

- `cplexlsqlin` for linearly constrained least squares problems,
- `cplexlsqmilp` for linearly constrained mixed integer least squares problems,
- `cplexlsqbilp` for linearly constrained binary integer least squares problems,
- `cplexlsqmiqcp` for quadratically constrained mixed integer least squares problems,
- `cplexlsqqcp` for quadratically constrained programming problems,
- `cplexlsqnonneglin` for nonnegative least squares problems,

- `cplexlsqnonnegmilp` for nonnegative mixed integer least squares problems,
- `cplexlsqnonnegmiqcp` for nonnegative quadratically constrained mixed integer least squares problems and
- `cplexlsqnonnegqcp` for nonnegative quadratically constrained programming problems.

The advantage of the toolbox design is that you can reuse your code where you had used MATLAB Optimization Toolbox functions to solve linear programming, quadratic programming, binary integer programming, linearly constrained least squares, and nonnegative least squares problems.

Setting and querying parameters in the CPLEX for MATLAB Toolbox

Options, also called parameters, can be set to control the solution of problems. The toolbox provides two types of options input. One type corresponds to the MATLAB Optimization Toolbox options, and the other type is the CPLEX parameters. You can use either or both of these types of options. If you use both, the CPLEX parameters will override the MATLAB options.

The toolbox options are listed in the following table.

Options corresponding to the MATLAB Optimization Toolbox

`Diagnostics`	'on' | {'off'}
`Display`	'off' | 'iter' | 'final' | 'notify'
`MaxIter`	refer to `cplex.Param.simplex.limits.iterations`
`Simplex`	refer to `cplex.Param.lpmethod`
	refer to `cplex.Param.qpmethod`
	refer to `cplex.Param.mip.strategy.startalgorithm`
`BranchStrategy`	refer to `cplex.Param.mip.strategy.variableselect`
`MaxNodes`	refer to `cplex.Param.mip.limits.nodes`
`MaxTime`	refer to `cplex.Param.timelimit`
`NodeDisplayInterval`	refer to `cplex.Param.mip.interval`
`NodeSearchStrategy`	refer to `cplex.Param.mip.strategy.nodeselect`
`TolFun`	refer to `cplex.Param.simplex.tolerances.optimality`
`TolXInteger`	refer to `cplex.Param.mip.tolerances.integrality`
`TolRLPFun`	refer to `cplex.Param.simplex.tolerances.optimality`

.

These options can be set using the toolbox function `optimset`. For example, the following code turns on the optimizer output and sets a node limit of 400,

```
options = cplexoptimset('Diagnostics', 'on', 'MaxNodes', 400);
```

Alternatively, these options can be set directly on the fields of the structure. For example, the following code has the same result as the previous one.

```
options = cplexoptimset;
options.Diagnostics = 'on';
options.MaxNodes = 400;
```

The current and default values of the options can be queried with the function `optimget`.

If you need to use CPLEX parameters that do not correspond to the options in the MATLAB Optimization Toolbox, you can create a structure which contains all of the CPLEX parameters. For example, to set a node limit of 400 and instruct CPLEX to use traditional branch and cut style search:

```
opt = cplexoptimset('cplex');
opt.mip.limits.nodes=400;
opt.mip.strategy.search=1;
```

Tip:

If you are already familiar with the names of parameters in the Interactive Optimizer, then you quickly recognize names of parameters in CPLEX for MATLAB. For example, the command “`set mip limits nodes 1`” in the Interactive Optimizer corresponds to “`opt.mip.limits.nodes = 1;`” in the CPLEX for MATLAB Toolbox where `opt` was created with the line “`opt = optimset('cplex');`”

To assist in setting parameters, auto-completion of parameter names is available in the MATLAB environment.

Cplex Class API

Describes the Cplex class

While the CPLEX for MATLAB Toolbox functions provide the ability to solve a multitude of mathematical programming problems, the toolbox design does not support restart. To enable users to use decomposition algorithms, the Cplex Class API is also provided in CPLEX for MATLAB.

The Cplex class stores the model and provides methods for the solution, analysis, manipulation and reading/writing of the model file. All of the data associated with the problem is stored in the properties of a Cplex object. These class properties are standard MATLAB data structures and can be manipulated directly within MATLAB. However, modifying the problem using methods provided in the Cplex class enforces consistency, such as ensuring that vectors are of the proper length.

The documentation of the Cplex class provides an introduction of properties and methods of the Cplex class in more detail.

The properties of the Cplex class include:

`Cplex.Model`	stores the data of the model
`Cplex.Solution`	stores the solution of the model
`Cplex.Param`	stores the parameters (options) of the model
`Cplex.Start`	stores the start of the LP model
`Cplex.MipStart`	stores the start of the MIP model
`Cplex.InfoCallback`	pointer to an informational callback
`Cplex.Conflict`	stores the conflict information of a conflicted model
`Cplex.Order`	stores the priority order information

`Cplex.DisplayFunc`	pointer to a function which provides control of display of output

The following informative methods are provided:

`Cplex.getVersion`	returns the CPLEX version
`Cplex.getProbType`	returns the problem type of the model

The following methods are provided for reading from and writing to files:

```
Cplex.readModel
Cplex.writeModel
Cplex.readBasis
Cplex.writeBasis
Cplex.readMipStart
Cplex.writeMipStart
Cplex.readParam
Cplex.writeParam
Cplex.writeConflict
```

The following methods are provided to solve and analyze the model, solution and mipstart:

```
Cplex.solve
Cplex.populate
Cplex.feasOpt
Cplex.refineConflict
Cplex.refineMipStartConflict
Cplex.terminate
```

The following methods are provided to solve, set and query parameters:

```
Cplex.tuneParam
Cplex.setDefault
Cplex.getChgParam
```

Although a model can be modified by manipulating the MATLAB data structures directly, the following functions are provided to make modifications easier:

```
Cplex.addCols
Cplex.addRows
Cplex.delCols
Cplex.delRows
Cplex.addSOSs
Cplex.addQCs
Cplex.addIndicators
```

Setting and querying parameters in the CPLEX Class API

To set parameters using the CPLEX Class API, you set the current values of the fields in the `Param` structure property of the Cplex class. For example, to set a node limit of 400 and instruct CPLEX to use traditional branch and cut style search:

```
cpx.Param.mip.limits.nodes.Cur=400;
cpx.Param.mip.strategy.search.Cur=1;
```

Tip:

If you are already familiar with the names of parameters in the Interactive Optimizer, then you quickly recognize names of parameters in CPLEX for MATLAB. For example, the command "`set mip limits nodes 1`" in the Interactive Optimizer corresponds to "`cpx.Param.mip.limits.nodes.Cur = 1;`" in the CPLEX Class API where `cpx` is an instance of the Cplex class.

To assist in setting parameters, auto-completion of parameter names is available in the MATLAB environment.

Chapter 6. Programming tips

As you model and solve mathematical programming problems, you may find it useful to refer to the documentation to find answers to your questions.

As you model and solve mathematical programming problems with CPLEX for MATLAB, you may find it useful to refer to the documentation to find answers to your questions.

Some common questions are answered in this documentation as well as the *CPLEX User's Manual* available in your CPLEX distribution.

- **How do I use the MATLAB sparse matrix format with the CPLEX for MATLAB functions and classes?**

  Either sparse or dense format can be used in the connector in all places where double matrices and vectors are accepted as arguments.
- **How do I understand and deal with my infeasible (or unbounded) model?**

  The section *Infeasibility and unboundedness* of the *CPLEX User's Manual* documents tools to help you analyze the source of the infeasibility in a model: the preprocessing reduction parameter for distinguishing infeasibility from unboundedness, the conflict refiner for detecting minimal sets of mutually contradictory bounds and constraints, and FeasOpt for repairing infeasibilities.
- **How do I obtain a pool of multiple solutions?**

  The section *Discrete optimization > Solution pool: generating and keeping multiple solutions* of the *CPLEX User's Manual* introduces the solution pool for storing multiple solutions to a mixed integer programming problem (MIP) and explains techniques for generating and managing those solutions.
- **How do I get this difficult model to solve?**

  The section *Programming considerations > Tuning tool* of the *User's Manual* documents the tuning tool, a utility to aid you in improving the performance of your optimization applications, analyzes a model or a group of models and suggests a suite of parameter settings for you to use that provide better performance than the default parameter settings for your model or group of models.
- **How do I invoke parallel threads to solve my model faster?**

  The section *Advanced programming techniques > Parallel optimizers* of the *CPLEX User's Manual* documents the CPLEX parallel optimizers.
- **How do I set algorithm control parameters?**

  The *CPLEX Parameters Reference Manual* lists all of the parameters and explains their settings. To learn about setting parameters when using the toolbox functions, see the function `cplexoptimset`. For information about setting parameters while using the Cplex class API, see `Cplex.Param`.
- **How do I check the consistency of my data?**

  There is a consistency check which can be controlled by the datacheck parameter. The default value of `datacheck` is off, which can improve the performance in the case of valid model data. If the `datacheck` is on, then a detailed error message will be displayed in the case of invalid data. For information about setting parameters, see `Cplex.Param` and `cplexoptimset`.

Printed in USA