

In the name of God

Part 3. ILOG CPLEX

3.3. MPS File Format

Spring 2010

Instructor: Dr. Masoud Yaghini

Outline

- MPS Standard Format
- CPLEX MPS Format
- References



MPS Standard Format

Industry Standard

- **MPS format**

- was originally introduced by IBM to express linear and integer programs in a standard way.
- established on mainframe LP systems, has become a widely accepted standard for defining LP problems.
- is a column-oriented format: problems are specified by column (variable) rather than by row (constraint).

Records in MPS format

- Each line of the MPS file represents a single record.
- Records in an MPS data file consist of two types:
 - **Indicator records**
 - **Data records**

Indicator records

- **Indicator records**

- separate the individual sections of the MPS file.
- Each indicator record contains a single word that begins in the first column.

- **Indicator records:**

- **NAME**
- **ROWS**
- **COLUMNS**
- **RHS**
- **BOUNDS**
- **ENDATA**

Indicator records

- **NAME**

- specifies the problem name; unlike other indicator records, the name record contains data

- **ROWS**

- specifies name and sense for each constraint

- **COLUMNS**

- specifies the name assigned to each variable (column) and the nonzero constraint coefficients corresponding to that variable

Indicator records

- **RHS**

- specifies the names of right hand side vectors and values for each constraint (row)
- If no RHS section is present, CPLEX generates a warning message and set all right hand side values to 0.

- **BOUNDS**

- specifies the limits within which each variable (column) must remain
- If no BOUNDS section is present, all variables have their bounds set from 0 (zero) to $+\infty$ (positive infinity).
- The BOUNDS section is mandatory.

- **ENDATA**

- signals the end of the data; always the last entry in an MPS file

Data records

- Each data record comprises **six fields**, as in Fields of a data record in MPS file format.
- The fields must be separated by **white space** (that is, blank space, tab, etc.), and the **first field** must begin in column 2 or beyond.
- **Fields of a data record in MPS file format:**

Field	1	2	3	4	5	6
Columns	2-3	5-12	15-22	25-36	40-47	50-61
Length	2	8	8	12	8	12
Contents	Indicator	Name	Name	Value	Name	Value

Data records

- **Comments**

- If the first character in Field 3 or 5 is a dollar sign (\$), the remaining characters in the record are treated as a comment.
- Another method for inserting comments is to place an asterisk (*) in column 1.
- Everything on such a line is treated as a comment.

The ROWS section

- In the ROWS section, each row of the problem is specified with its name and sense, one row per record.
- **Field 1** contains a single letter designating the sense of each row. Acceptable values are:
 - **N** : indicates the objective function row.
 - **G** : a greater-than-or-equal-to row.
 - **L** : a less-than-or-equal-to row.
 - **E** : an equality row.
- **Field 2** contains a character identifier, maximum length of 8 characters, specifying the **name of the row**.
- **Fields 3-6** are not used in the ROWS section.

Example

Maximize $x_1 + 2x_2 + 3x_3$

subject to $-x_1 + x_2 + x_3 \leq 20$

$x_1 - 3x_2 + x_3 \leq 30$

with these bounds $0 \leq x_1 \leq 40$

$0 \leq x_2 \leq +\infty$

$0 \leq x_3 \leq +\infty$

Example: ROWS section

```
ROWS
```

```
  N  obj
```

```
  L  c1
```

```
  L  c2
```

The COLUMNS section

- In the COLUMNS section, all the columns of the constraint matrix are specified with **their name** and all of the **nonzero elements**, as follows:
 - **Field 1:** Blank
 - **Field 2:** Column identifier (name)
 - **Field 3:** Row identifier (name)
 - **Field 4:** Value of matrix coefficient specified by Fields 2 and 3
 - **Field 5:** Row identifier (optional)
 - **Field 6:** Value of matrix coefficient specified by Fields 2 and 5 (optional)

Example: COLUMNS section

COLUMNS

x1	obj	-1	c1	-1
x1	c2	1		
x2	obj	-2	c1	1
x2	c2	-3		
x3	obj	-3	c1	1
x3	c2	1		

The RHS section

- In the RHS section, the nonzero righthand-side values of the constraints are specified.
 - **Field 1:** Blank
 - **Field 2:** RHS identifier (name)
 - **Field 3:** Row identifier
 - **Field 4:** Value of RHS coefficient specified by Field 2 and 3
 - **Field 5:** Row identifier (optional)
 - **Field 6:** Value of RHS coefficient specified by Field 2 and 5 (optional)

- **Example:**

```
RHS
  rhs          c1                20   c2                30
```


The BOUNDS section

- In the BOUNDS section, bound values for variables may be specified.
 - **Field 1:** Type of bound.
 - **Field 2:** Bound identifier
 - **Field 3:** Column identifier to be bounded
 - **Field 4:** Value of the specified bound
 - **Fields 5 and 6** are not used in the BOUNDS section.

The BOUNDS section

- Acceptable values for type of bound (Field 1):
 - **LO** Lower bound
 - **UP** Upper bound
 - **FX** Fixed value (upper and lower bound the same)
 - **FR** Free variable (lower bound $-\infty$ and upper bound $+\infty$)
 - **MI** Minus infinity (lower bound = $-\infty$)
 - **PL** Plus infinity (upper bound = $+\infty$)
- In our example, the BOUNDS section looks like this:

```
BOUNDS
UP BOUND      x1      40
```

The BOUNDS section

- **The BOUNDS section:**

- If no bounds are specified, ILOG CPLEX assumes a lower bound of 0 (zero) and an upper bound of $+\infty$.
- If only a single bound is specified, the unspecified bound remains at 0 or $+\infty$.
- If an upper bound of less than 0 is specified and no other bound is specified, the lower bound is automatically set to $-\infty$.

Example of MPS file format

```
NAME                example2.mps
ROWS
  N  obj
  L  c1
  L  c2
COLUMNS
  x1      obj      -1      c1      -1
  x1      c2        1
  x2      obj      -2      c1        1
  x2      c2      -3
  x3      obj      -3      c1        1
  x3      c2        1
RHS
  rhs      c1      20      c2      30
BOUNDS
  UP BOUND  x1      40
ENDATA
```

CPLEX MPS Format

CPLEX MPS Format

- MPS format included restrictions, such as:
 - requiring input fields to occupy fixed columnar positions and
 - limiting all names to a length of 8 characters or fewer.
- In CPLEX version 3.0 and subsequent releases, these restrictions were relaxed.
- The current ILOG CPLEX MPS format is actually an extended version of the historical MPS format.
- To maintain compatibility with earlier versions as well as MPS files from other systems, ILOG CPLEX provides an MPS file conversion utility which translates older files into the newer CPLEX MPS format.

Objective sense and name

- CPLEX extends the MPS standard by allowing two additional sections:
 - **OBJSEN**
 - **OBJNAME**
- They must be specified after the **NAME** section.
- **OBJSEN**
 - sets the objective function sense, and
- **OBJNAME**
 - selects an objective function from among the free rows within the file.
 - The values for **OBJSENSE** can be **MAX** or **MIN** .

Objective sense and name

- If neither of these sections appears in the MPS file,
 - CPLEX assumes that the problem is a **minimization** and
 - the objective function is the first free row encountered in the ROWS section.
- Here is an example of these optional sections:

```
NAME          example.mps
OBJSENSE
  MAX
OBJNAME
  rowname
```


Integer variables in MPS files

- CPLEX accepts two commonly used ways of extending the MPS file format to include integer variables:
 - In the COLUMNS section or
 - In the BOUNDS section
- In the first way, integer variables are identified within the COLUMNS section of the MPS file by **marker lines**.
- A marker line is placed at the beginning and end of a range of integer variables.
- Multiple sets of marker lines are allowed.

Integer variables in MPS files

- Integer marker lines have a field format consisting of Fields 2 through 4.
 - **Field 2:** Marker name
 - **Field 3:** '**MARKER**' (including the single quotation marks)
 - **Field 4:** Keyword '**INTORG**' and '**INTEND**' to mark beginning and end respectively (including the single quotation marks)
 - **Fields 5 and 6** are ignored.
- If no bounds are specified for the variables within markers, bounds of 0 (zero) and 1 (one) are assumed.

Integer variables in MPS files

- Example, column x4 is an integer variable:

```

NAME
ROWS
  N  obj
  L  c1
  L  c2
  E  c3
COLUMNS
  x1      obj      -1      c1      -1
  x1      c2        1
  x2      obj      -2      c1        1
  x2      c2       -3      c3        1
  x3      obj      -3      c1        1
  x3      c2        1
  MARK0000 'MARKER'          'INTORG'
  x4      obj      -1      c1      10
  x4      c3      -3.5
  MARK0001 'MARKER'          'INTEND'
RHS
  rhs      c1      20      c2      30
BOUNDS
  UP BOUND  x1      40
  LO BOUND  x4        2
  UP BOUND  x4        3
ENDATA

```

Integer variables in MPS files

- In the second way of treating integer variables, integer variables are declared in the BOUNDS section with special bound types in Field 1.
- The acceptable special bound types are:
 - **BV** Binary variable
 - ◆ **Field 4** must be 1.0 or blank
 - **LI** Integer lower bound
 - ◆ **Field 4** is the lower bound value and must be an integer
 - **UI** Integer upper bound
 - ◆ **Field 4** is the upper bound value and must be an integer
- To specify general integers with no upper bounds, use **LI** with the value 0.0.

MPS CPLEX Format

- **Example:** column x4 is an integer variable declared in the BOUNDS section

```
NAME
ROWS
  N  obj
  L  c1
  L  c2
  E  c3
COLUMNS
  x1      obj      -1      c1      -1
  x1      c2         1
  x2      obj     -2      c1         1
  x2      c2     -3      c3         1
  x3      obj     -3      c1         1
  x3      c2         1
  x4      obj     -1      c1        10
  x4      c3     -3.5
RHS
  rhs      c1        20      c2        30
BOUNDS
  UP BOUND  x1        40
  LI BOUND  x4         2
  UI BOUND  x4         3
ENDATA
```

References

References

- **ILOG CPLEX, ILOG CPLEX File Formats, ILOG CPLEX, 2008.**



The End