1 Introduction

This software package uses a homogeneous and self-dual algorithm to solve linear programs:

$$
\text{(LP)} \qquad \begin{array}{ll} \text{Min or Max} & c^T x \\ \text{Subject to} & a_i x \ (\leq, =, \geq)\ b_i,\ i = 1, \ldots, m \\ & l_j \leq x_j \leq u_j,\ j = 1, \ldots, n \end{array}
$$

where

- $c \in R^n$: the objective function vector,
- $A \in R^{m \times n}$: the constraint matrix, and its ith row is denoted by a_i,
- $b \in R^m$: the right-hand-side vector,
- $l_j, u_j \in R$: the lower and upper bounds for jth decision variable x_j.

The program includes the following features:

1. It solves the linear programming problem without any regularity assumption concerning the existence of optimal, feasible, or interior feasible solutions.
2. If the LP problem has a solution, the program generates a sequence that approaches feasibility and optimality simultaneously; if the problem is infeasible or unbounded, the algorithm will produce an infeasibility certificate for at least one of the primal and dual problems.
3. If the problem is solvable, the program is able to generate an optimal basic solution, which is desired in many practical applications.

This package contains the following files:

1. Make files: used to compile source codes and link them into two executable codes, include files `Coplopt.mak`, `Coplrpt.mak`, and `Makefile`,
2. Source files for the executable code, say `opt`, to solve LP problems:

```
LPconst.h   LPdefine.h  LPmatrx.h
LPcross.h   LPfunct.h   LPmfunc.h   LPxfunc.h
Dmatrix.c   Xbasics.c   Xlusolv.c   copldens.c  coplopti.c
Dmemory.c   Xclean.c    Xprimal.c   coplhash.c  coplouta.c  coplshut.c
Dmmdpro.c   Xcross.c    Xsimplx.c   coplhomo.c  coplpara.c  coplstac.c
Dnumfac.c   Xdual.c     Zldpnt.c    coplmain.c
Dprique.c   Xlubasi.c   coplbasi.c  coplmemo.c  coplpres.c
Dvector.c   Xlufact.c   coplcrus.c  coplmpsf.c  coplpsub.c
```

Source files for the executable code, say `rpt`, to generate the readable solution file from several files, `coplfile.*`, resulted from executing `opt`:

```
LPreport.h
coplpost.c   coplrepo.c
```

3. COPLLP.spc, the default specification file.

4. Example data files:

```
afiro.mps  example.mps
```

2 Setup

Two executable codes named **opt** and **rpt** can be created by:

1. Download all the files into a directory of your choice,

2. Compile two sets of source codes and link them together to form two executable programs **opt** and **rpt**, respectively, or simply type command `make` to compile all source codes and to form two executable programs **opt** and **rpt**.

3 Usage

To solve a problem using **opt**, type any of the following command lines

```
opt [-f {NAME1}] [-o {NAME2}] [-s {NAME3}]
opt [-f:{NAME1}] [-o:{NAME2}] [-s:{NAME3}]
opt [-f{NAME1}]  [-o{NAME2}]  [-s{NAME3}]
```

The contents in '`[ ]`' are optional, '`/`' means 'or', and the contents in "{}" are required if the corresponding option appears in the command line. Here

-f: Specify the input data file name. If this option is omitted, the system will prompt the user for a file name.

-o: Specify the output file name. The output information will be saved in a file named "`coplfile.out`" if this option is omitted.

-s: Specify the SPC (specification) file name. The system will look COPLLP.spc as the default specification file name. The system will use the default settings if this option is omitted and file COPLLP.spc can not be found.

4 Parameter Specification: The SPC file

COPL_LP allows users to set, by SPC (specification) file, the algorithm parameters used by the system. The file has relatively free format, each line of the file defines only one parameter and consists of a key word and the value. Lines beginning with "*" are treated as comments. Here is the default parameter settings:

```
bound_set_name           : NULL
cache_memory_size        : 256
complementarity_tolerance: 1.0e-12
cross_over_switch        : off
dense_column_threshold   : 150
dual_feasible_tolerance  : 1.0e-8
infeasible_tolerance     : 1.0e-8
iteration_limit          : 500
lower_bound_default      : 0.0
nonzero_tolerance        : 1.0e-12
number_of_columns        : 100000
number_of_elements       : 500000
objective_set_name       : NULL
optimization_type        : min
pcg_starting_tolerance   : 1.0e-10
pivoting_tolerance       : 1.0e-14
presolve_level           : 5
presolve_loop            : 10
primal_dual_gap_tolerance: 1.0e-8
primal_feasible_tolerance: 1.0e-8
range_set_name           : NULL
right_hand_side_set_name : NULL
step_factor              : 0.99995
upper_bound_default      : 1.0e30
```

The parameters that can be specified in the SPC file are list below together with their default values in .

bound_set_name {`NULL`}. Let the system use the bound set with the specified name. The first bound name in the MPS file will be used if no bound name is specified or if `NULL` is specified in the SPC file.

cache_memory_size {`256`}. Define the size of cache size on the computer. The size is measured in Kilobytes. it is better to choose a smaller value if the cache size is unknown initially.

complementarity_tolerance {`1.0e-12`}. Define the complementarity tolerance.

cross_over_switch {`on`}. Request that cross-over procedure be performed `(on)` or not performed `(off)`.

dense_column_threshold {`150`}. Define the dense column threshold value. Each column is treated as a dense column if the number of nonzeros in this column is greater or equal to the specified threshold value.

dual_feasible_tolerance {`1.0e-8`}. Define the dual feasibility tolerance.

infeasible_tolerance {`1.0e-8`}. Define the infeasibility tolerance. This value is used to declare infeasibility of the problem.

iteration_limit {`500`}. An upper limit on the number of iterations.

lower_bound_default {`0`}. The default value for lower bound.

nonzero_tolerance {`1.0e-12`}. An element is treated as zero if the absolute value of the element is less this value.

number_of_columns {`100000`}. This value is used to set the maximum number of the variables or columns in an LP problem.

number_of_elements {`500000`}. This value is used to set the maximum number of nonzero constraint coefficients in an LP problem.

objective_set_name {`NULL`}. Set the objective row name used by the system. The last row name marked by letter "N" in section row in the MPS file will be used if no objective name is given or if `NULL` is specified in the SPC file.

optimization_type {`min`}. Define the LP problem as minimizing or maximizing if `min` or `max` is specified, respectively

pcg_starting_tolerance : {`1.0e-10`}. The system will start a pre-conditioned conjugate gradient process to improve accuracy of the solution of the K-K-T linear equation system, if the residual of the system is less than the specified value.

pivoting_tolerance {`1.0e-14`}. If a diagonal element of the symmetric matrix to be factorized is less than the setting value, the diagonal element is considered to be zero.

presolve_level {`5`}. Define the level of presolving. There are 6 possible levels (level 0 - 5) which can be set by users. Here
0: no presolving is performed.
1: dependent rows, null and fixed columns.
2: singleton rows, forcing rows, doubleton rows, dominated rows.
3: dominated columns.
4: duplicate rows.
5: duplicate columns.

presolve_loop {`10`}. After one loop reduction is done, further reduction is possible. Thus, presolving procedure may run more than one loop. The system will quit the presolving if no further reduction is possible or the number of presolving loops exceeds the setting loop value.

primal_dual_gap_tolerance {`1.0e-8`}. Define the relative primal dual gap tolerance.

primal_feasible_tolerance {`1.0e-8`}. Define the primal feasibility tolerance.

range_set_name {`NULL`}. Let the system use the range set with the specified name. The first range name in the MPS file will be used if no range set name is given or if `NULL` is specified.

right_hand_side_set_name {`NULL`}. Let the system use the right hand side with the specified name. The first right hand side name in the MPS file will be used if no right hand side name is given or if `NULL` is specified.

step_factor {`0.99995`}. Set the step size factor, which should be real number in $(0, 1)$.

upper_bound_default {`1.0e+30`}. The default value for upper bounds.

5 Data Input: The MPS file

An MPS file is required for each LP problem. It is used to name variables and constraints, and to input numerical data. The MPS file has a *fixed* format, i.e. each item of data must appear in specific columns.

The MPS file consists of lines, each line is called a "card". Several "header cards" divide the MPS file into different sections as follows:

```
NAME
ROWS
 .
COLUMNS
 .
RHS
 .
RANGES
 .
BOUNDS
 .
ENDATA
```

Each header card must begin in column 1.

Intervening card images (indicated by " · " above) have the following data format:

Columns	2-3	5-12	15-22	25-36	40-47	50-61
Contents	*Key*	*Name0*	*Name1*	*Value1*	*Name2*	*Value2*

In addition, "comment" cards are allowed, these have an asterisk " * " in column 1 and any characters begin at column 2 in the card.

5.1 The NAME card

```
NAME          EXP01         (for example)
```

This card contains the word `NAME` in columns 1-4, and the name of the problem beginning in column 15. The name may consist of 1 or more characters of any kind, or it may be blank. The `NAME` card is normally the first card in the MPS file, but it may be preceded or followed by comment cards.

5.2 The ROWS Section

This section is to name constraints. The general constraints are commonly referred to as rows. The ROWS section contains one card for each constraint or each row. *Key* defines what type the constraint is, and *Name0* gives the constraint an 8-character name. The various row-types are as follows:

Key	*Row-type*
`E`	$=$
`G`	$\geq$
`L`	$\leq$
`N`	Objective

Note that 1-character *Key* may be in either column 2 or column 3.

Row-type `E`, `G` and L are easily understood in terms of a linear function ax and a right-hand side b. They specify constraints of the form

$$ax = b, \quad ax \geq b \quad \text{and} \quad ax \leq b$$

respectively. Nonzero elements of the row-vector a will appear in appropriate parts of the COLUMNS section, and if b is nonzero then it will appear in the RHS section.

Row-type `N` stands for "Not binding", also known as "Free". It is used to define the *objective row*. The following is an example:

```
ROWS
 E  row1
 G  .......2
 L  r o w 3
 N  COST
 E   ABC  4
```

5.3 The COLUMNS Section

For each variable x_j (say), the COLUMNS section defines a name for x_j and list nonzero entries a_{ij} in the corresponding column of the constraint matrix. The nonzeros for the first column must be grouped together before those for the second column, and so on. If a column has several nonzeros, it does not matter what order they appear in (as long as all appear before the next column).

In COLUMNS section, *Key* is blank, *Name0* is the column name, and *Name1*, *Value1* give a row name and value for some coefficient in that column. If there is another row name and value for the same column, they may appear as *Name2*, *Value2* on the same card, or they may be on the next card.

If either *Name1* or *Name2* is blank, the corresponding value is ignored.

Here is an example

```
1   45      12  15     22  25          36    40    47  50          61 (fields)
+--++-------+  +-------+  +----------+    +-------+  +----------+
COLUMNS
    x0000001  row1               1.0    .......2      -3.0
    x0000001  COST              -1.0
    X      2  row1        -11.1111111   r o w 3       5.0E-02
    X      2  .......2          .32e2
    X      2  COST            -7
```

This example describes two columns (or variables), named "`x0000001`" and "`X      2`" respectively. If the row names are described in `ROWS` section in the order in the example given in the previous subsection, then there are three nonzeros in the first column, i.e column `x0000001`, they are $a_{11} = 1.0,\ a_{21} = -3.0,\ c_1 = -1.0$, and there are four nonzeros in the second row, they are $a_{12} = -11.1111111,\ a_{22} = 0.32e2,\ a_{32} = 5.0e-2,\ c_2 = -7$.

5.4 The RHS Section

This section specifies the nonzero elements of the RHS vector for the problem. These nonzeros may appear in any order. The format in this section is exactly the same as in the COLUMNS section, with *Name0* giving a name to the right-hand side. This section is optional, i.e., there may be no such a section in a MPS file, which means all the elements of the RHS vector for the problem are zeros.

The RHS section may contain more than one right-hand side. The *first* one will be used unless some other name is specified in the SPC file. If a right-hand side name is specified in the SPC file (i.e. option `right_hand_set_name` is set), the system will process the cards with the setting name and will ignore all other name card in the RHS section.

An example is given as follows:

```
1   45      12  15     22  25          36    40    47  50          61
+--++-------+  +-------+  +----------+    +-------+  +----------+
```

```
RHS
    rhs01      row1                  10.    .......2       30.
    rhs01      r o w 3                5.
    rhs01      row1                   4.    .......2        6.
    rhs01      r o w 3               10.
```

The RHS section may contain more than one right-hand side. The *first* one will be used unless some other name is specified in the SPC file.

5.5 The RANGES Section

Ranges are used for constraints of the form

$$\alpha \le a^T x \le \beta,$$

where both α and β are finite. The range of the constraint is $r = \beta - \alpha$. Either α or β is specified in the RHS section (as b say), and r is defined in the RANGES section. The resulting α and β depend on the row-type of the constraint and the sign of r as follows:

Row-type	*Sign of r*	*Lower limit, α*	*Upper limit, β*
E	$+$	b	$b+\|r\|$
E	$-$	$b-\|r\|$	b
G	$+$ or $-$	b	$b+\|r\|$
L	$+$ or $-$	$b-\|r\|$	b

The format in this section is exactly the same as in the COLUMNS section, with *Name0* giving a name to the range set. Here is an example:

```
1  45      12  15     22  25         36   40     47  50         61
+--++------+  +------+  +----------+   +------+  +----------+
ROWS
 E  row1
 G  .......2
 L  r o w 3
 N  COST
 E   ABC  4
COLUMNS
 .
RHS
    rhs        row1                 10.     .......2        30.
    rhs        r o w 3               5.       ABC  4         6.
RANGES
    range1     row1                  1.     .......2         3.
    range1     r o w 3              -1.       ABC  4        -2.
```

The constraints listed in the example will have the following limits:

$$
\begin{array}{rcccl}
10 & \leq & \texttt{row1} & \leq & 11 \\
30 & \leq & \texttt{.......2} & \leq & 33 \\
4 & \leq & \texttt{r o w 3} & \leq & 5 \\
4 & \leq & \texttt{ ABC 4} & \leq & 6
\end{array}
$$

The RANGES section may contain more than one set of ranges. The *first* set will be used if no other name is specified in the SPC file.

5.6 The BOUNDS Section

The BOUNDS section defines bound-type and bound value for each variable whose lower bound value or upper bound value is not the default. The default bounds on all variables are $0 \leq x_j \leq +\infty$, however, if necessary, the default values 0 and $+\infty$ can be changed in the SPC file to $\alpha \leq x_j \leq \beta$ by the `lower_bound_default` and upper_bound_default options, respectively. So, a variable, say x_j, has the default bounds if it does not appear in the BOUNDS section. In general, different sets of bounds may be given, and the *first* one will be used if no other name is specified in the SPC file.

In BOUNDS section, *Key* gives the type of bound required, *Name0* is the name of bound set, and *Name1* and *Value1* are the column name and bound value. Note that *Name2* and *Value2* are ignored, i.e. one card just used for one bound.

Let x and ν be the column name and bound value specified, and let α and β are the default bound values. The various bound-types allowed are as follows:

Key	*Bound-type*	*Resulting bounds*
`LO`	Lower bound	$\nu \leq x \leq +\infty$
`UP`	Upper bound	$-\infty \leq x \leq \nu$
`FX`	Fixed variable	$\nu \leq x \leq \nu$
`FR`	Free variable	$-\infty \leq x \leq +\infty$
`MI`	Minus infinity	$-\infty \leq x \leq 0$
`PL`	Plus infinity	$0 \leq x \leq +\infty$

Here is an example:

```
1  45      12  15     22  25          36   40     47  50          61
+--++------+   +------+   +----------+     +------+   +----------+
BOUNDS
 UP  BOUND1    x01                  4.0
 UP  BOUND1    x02                  2.0
 LO  BOUND1    x01                 -2.0
 LO  BOUND1    x03                  1.0
 FX  BOUND1    x04                  0.5
 FR  BOUND1    x05
 MI  BOUND1    x06                 7.91
```

```
PL   BOUND1    x07            -1000.0
MI   BOUND1    x08
PL   BOUND1    x09
```

The effect of the example above is to give the following bounds:

$$
\begin{array}{lcccr}
-2 & \le & \texttt{x01} & \le & 4 \\
\alpha & \le & \texttt{x02} & \le & 2 \\
1 & \le & \texttt{x03} & \le & \beta \\
 & & \texttt{x04} & = & 0.5 \\
-\infty & \le & \texttt{x05} & \le & +\infty \\
-\infty & \le & \texttt{x06} & \le & 7.91 \\
-1000 & \le & \texttt{x07} & \le & +\infty \\
-\infty & \le & \texttt{x08} & \le & 0 \\
0 & \le & \texttt{x09} & \le & +\infty
\end{array}
$$

where α and β are specified lower and upper bounds in the spec file.

5.7 The ENDATA Card

This card contains word "`ENDATA`". Each MPS file must end with it.

5.8 An example

Suppose we have an LP problem in the form:

$$
\begin{array}{ll}
\text{minimize} & -7x_1 - 12x_2 \\
\text{subject to} & 9x_1 + 4x_2 \le 360 \\
 & 4x_1 + 5x_2 \le 200 \\
 & 3x_1 + 10x_2 \le 300 \\
 & 1 \le x_1 \le 100 \\
 & 1 \le x_2 \le 100.
\end{array}
$$

Here, the data set of the problem is $L(A, b, c, l, u)$, where

$$
A = \begin{bmatrix} 9 & 4 \\ 4 & 5 \\ 3 & 10 \end{bmatrix}, \quad b = \begin{bmatrix} 360 \\ 200 \\ 300 \end{bmatrix}, \quad c = \begin{bmatrix} -7 \\ -12 \end{bmatrix}, \quad l = \begin{bmatrix} 1 \\ 1 \end{bmatrix}, \quad u = \begin{bmatrix} 100 \\ 100 \end{bmatrix}.
$$

To create the corresponding MPS data file for the problem, you can do in this way. Firstly, form the following augmented matrix by combining A with b, c, l and u.

$$
\left[\begin{array}{cccc|c}
 & x_1 & x_2 & & rhs \\
a_1 & 9 & 4 & & 360 \\
a_2 & 4 & 5 & & 200 \\
a_3 & 3 & 10 & & 300 \\
c & -7 & -12 & & \\
l & 1 & 1 & & \\
u & 100 & 100 & &
\end{array}\right]
$$

Give names for all rows, except for bound rows l and u. Here we use `r1`, `r2`, and `cost` to name rows 1, 2, and 3, respectively, and their row-types are `L`, `L`, `L`, and `N`, respectively. And then give names for all columns. We denote the four columns by `x01`, `x02`, `x03`, and `rhs`, respectively.

Now, we can write the MPS file for the problem as follows.

```
1   5      12  15     22  25          36   40     47  50          61
+---+------+   +------+   +-----------+    +------+   +-----------+
NAME           example
ROWS
 L  r1
 L  r2
 L  r3
 N  cost
COLUMNS
    x01        r1                 9.0    r2                    4.0
    x01        r3                 3.0    cost                 -7.0
    x02        r1                 4.0    r2                    5.0
    x02        r3                10.0    cost                -12.0
RHS
    rhs        r1               360.0    r2                  200.0
    rhs        r3               300.0
BOUNDS
 LO            x01                1.0
 UP            x01              100.0
 LO            x02                1.0
 UP            x02              100.0
ENDATA
```

6 Data Output

6.1 Run-Time Output

When a problem is solved by **opt**, the system will print some information on screen. For example, after type the command line

opt -fexample

you will see on screen algorithm parameter specification, the MPS file statistics, the presolving process, the iterative progress, the cross-over phase, the solution statistics, and the cpu time summary. They are similar to what is contained in an output file described below.

6.2 Output file

The system also save data summary and computation information in a file named by user using command line option "`-o`" or in "`coplfile.out`" named by default. Here is a sample.

```
SPC FILE
--------
     1     bound_set_name            : NULL
     2     cache_memory_size         : 256
     3     complementarity_tolerance: 1.0e-12
     4     cross_over_switch         : off
     5     dense_column_threshold    : 150
     6     dual_feasible_tolerance   : 1.0e-8
     7     infeasible_tolerance      : 1.0e-8
     8     iteration_limit           : 500
     9     lower_bound_default       : 0.0
    10     nonzero_tolerance         : 1.0e-12
    11     number_of_columns         : 100000
    12     number_of_elements        : 500000
    13     objective_set_name        : NULL
    14     optimization_type         : min
    15     pcg_starting_tolerance    : 1.0e-10
    16     pivoting_tolerance        : 1.0e-14
    17     presolve_level            : 5
    18     presolve_loop             : 10
    19     primal_dual_gap_tolerance: 1.0e-8
    20     primal_feasible_tolerance: 1.0e-8
    21     range_set_name            : NULL
    22     right_hand_side_set_name  : NULL
    23     step_factor               : 0.99995
    24     upper_bound_default       : 1.0e30
    25


MPS file
--------
     1     NAME            example
     2     ROWS
     7     COLUMNS
    12     RHS
    15     BOUNDS

xxxxxx     Total number of errors in MPS file      0

Problem Size
------------
 ROWS
   total ....................................................... 3
   equalities .................................................. 0
```

```
  inequalities ................................................... 3
  ranged ......................................................... 0
 COLUMNS
  total .......................................................... 2
  lower bounded .................................................. 2
  upper bounded .................................................. 2
  fixed .......................................................... 0
  free ........................................................... 0
  total .......................................................... 4
  lower .......................................................... 2
  upper .......................................................... 2
  fixed .......................................................... 0
  free ........................................................... 0
  minus infinity ................................................. 0
  plus infinity .................................................. 0
 DENSITY
  original .................................... 100.000% : 6/(3*2)
  after crushing ............................... 60.000% : 9/(3*5)
  final model .................................. 20.000% : 3/(3*5)

 ITER        POBJ              DOBJ           GAP       PINF     DINF
 0     -3.800000000E+01  -1.900000000E+01  9.5E-01  2.1E+02  3.7E+00
 1     -1.500173611E+02  -2.692435908E+01  4.4E+00  2.3E+00  1.1E+00
 2     -3.372990603E+02  -2.377936414E+02  4.2E-01  4.1E-02  1.0E+00
 3     -4.046935675E+02  -4.135642174E+02  2.1E-02  1.4E-02  1.3E-01
 4     -4.278624000E+02  -4.286566821E+02  1.8E-03  9.8E-04  2.6E-03
 5     -4.279999792E+02  -4.280000368E+02  1.3E-07  5.6E-08  2.0E-07
 6     -4.280000000E+02  -4.280000000E+02  6.8E-12  2.8E-12  1.0E-11

Computational Results
---------------------
 Interior-Point Optimization Results
  number of iterations ........................................... 6
  primal objective ............................... -4.280000000E+02
  dual objective ................................. -4.280000000E+02
  primal dual gap ................................. 6.845855662E-12
  primal dual barrier ............................. 0.000000000E+00
  primal infeasibility ............................ 2.849728882E-12
  dual infeasibility .............................. 9.980796971E-12

Time distribution
-----------------
 time for initial data input ............................. 0.70 sec
 time for presolve process ............................... 0.02 sec
```

```
 time for symbolic computation ............................ 0.00 sec
 time for numerical computation ........................... 0.03 sec
 time for cross-over computation .......................... 0.00 sec
 time for the whole process ............................... 0.76 sec

------------------------------  EOF  ------------------------------
```

6.3 Solution file

The program `rpt` can be used to view the optimal solutions for the optimization problem solved by `opt`. `opt` always generated solution stack files named `coplfile.sol`, `coplfile.bin`, and coplfile.stk. When you type `rpt` immediately after `opt`, one report file named `name.rpt` will be generated (if your input file is named `name.mps`). The file basically has two sections. The first is ROWS section, each line of which contains the constraint name and type, its lower bound, its active value (evaluated at the optimal solution), its upper bound, and its dual variable (multiplier) value. The second row is COLUMNS section, each line of which contains the variable name and type, its lower bound, its value at the optimal solution, its upper bound, and its dual variable (multiplier) value.

For the example in the preceding subsection, the corresponding report file, `example.rpt`, looks like

```
INPUT FILE NAME  :    example
NUMBER OF ROWS   :          3
NUMBER OF COLUMNS:          2
SOLUTION STATUS  :    optimal

ROWS
----
 order  name   type lower         active        upper         dual value
 1      r1     LE  -infinity     +2.760000e+02 +3.600000e+02 -2.500773e-12
 2      r2     LE  -infinity     +2.000000e+02 +2.000000e+02 -1.360000e+00
 3      r3     LE  -infinity     +3.000000e+02 +3.000000e+02 -5.200000e-01

COLUMNS
-------
 order  name   type lower         active        upper         comp value
 1      x01    BO  +1.000000e+00 +2.000000e+01 +1.000000e+02 +8.331469e-11
 2      x02    BO  +1.000000e+00 +2.400000e+01 +1.000000e+02 +1.990408e-12

----------------------------------- EOF -----------------------------------
```