Linear Programming

Mehrdad Majzoobi
Fall 2007
Farinaz Koushanfar

Linear Additive Delays Models
Linear Programming

- **Linear programming** (LP) problems involve the optimization of a linear objective function, subject to linear equality and inequality constraints.

\[
\begin{align*}
\text{Min or Max } & \quad f(X) \\
\text{subject to } & \quad AX (\leq, \geq, >, =, <) b \\
\text{where} & \quad \text{lower bound } \leq X \leq \text{upper bound}
\end{align*}
\]

Linear Programming

- An optimization problem can be categorized as linear and quadratic based on the objective function (F):
  - **Linear:** \( f(X) = c^T X \)
  - **Quadratic:** \( f(X) = 0.5 \ XQX^T + c^T X \)

- Also if the variables (\( X \)) have integer values the we would have an ILP or IQP problem.
MATLAB

- \( x = \text{linprog}(c, A, b, Aeq, beq, lb, ub, x0) \)
  - Min \( c^T x \) such that:
    - \( Ax \leq b \)
    - \( Aeqx = beq \)
    - \( Lb \leq x \leq ub \)
- \( x = \text{quadprog}(H, f, A, b, Aeq, beq, lb, ub, x0) \)
  - Min \( 0.5x^T Qx + c^T x \) such that:
    - \( Ax \leq b \)
    - \( Aeqx = beq \)
    - \( Lb \leq x \leq ub \)

CPLEX

- Interactive Optimizer
  - The problem must be fed line by line and variable by variable, not efficient for large problems
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 \[ 0 \leq x_1 \leq 40 \]
\[ 0 \leq x_2 \leq +\infty \]
\[ 0 \leq x_3 \leq +\infty \]

Steps:

CPLEX> enter
Enter name for problem: example1
Enter new problem ['end' on a separate line terminates]:

minimize
x1+2x2+2x3
st
-x1 + x2 + x3 <= 20
x1 - 3x2 + x3 <= 30
bounds
x1 <= 40
end
Displaying the problem:

```
CPLEX> display problem all
Minimize
   obj: x1 + 2 x2 + 2 x3
Subject To
   c1: - x1 + x2 + x3 <= 20
   c2: x1 - 3 x2 + x3 <= 30
Bounds
   0 <= x1 <= 40
   All other variables are >= 0.
```

Solve

```
CPLEX> optimize
```

Display the solution

```
CPLEX> display solution x1 x2 x3
```

CPLEX interface does not accept inputs in matrix and array format.

One needs to read in the problem through an input .mps file.
Linking CPLEX to MATLAB

- CPLEXMEX is a MATLAB Mex Interface for the CPLEX Callable Library
- It is a free and open source code developed by Nicolò Giorgetti, University of Oslo.

Licensing:
- "setenv ILOG_LICENSE_FILE /usr/site/cplex/share/etc/license"

Launch MATLAB (version 6)
Set the current directory to the path where the files are located
Type “makecplex”
Test to see if the integration is successful by typing: “testcplex”
Linking CPLEX to MATLAB

The calling syntax is:

\[
[xopt, opt, status, extra] = \text{cplexmex}(\text{maxmin}, Q, c, A, b, ctype, \ldots, \text{lb}, \text{ub}, \text{vartype}, x0, \text{params})
\]

- maxmin = 1 minimize, maxmin = -1 maximize.
- Q: A matrix containing the quadratic objective function coefficients.
- c: A column array containing the linear objective function coefficients.
- A: A matrix containing the constraints coefficients. A may be a sparse matrix.
- b: A column array containing the right-hand side value for each constraint.
- ctype: A column array containing the sense of each constraint in the constraint matrix.
  - ctype(i) = 'L' "\leq" Variable with upper bound
  - ctype(i) = 'E' "=" Fixed Variable
  - ctype(i) = 'G' "\geq" Variable with lower bound
  - ctype(i) = 'N' Nonbinding constraint
- lb (ub): An array of the lower (upper) bounds on each of the variables. If the i-th variable is lower (upper) bounded free put lb(i) or ub(i) = Inf. If all variables are lower (upper) bounded free put lb (or ub) = -Inf*ones(number_of_columns,1) or more simply lb=[] or ub=[];
- vartype: A column array containing the types of the variables.
  - vartype(i) = 'C' continuous variable
  - vartype(i) = 'I' Integer variable
  - vartype(i) = 'B' Boolean variable
- x0: A column vector containing the initial conditions. To leave free a value just put an Inf value in the corresponding value.
- xopt: The optimizer.
- opt: The optimum.
- status: Status of the optimization.
- extra: A cell array containing two fields: 1)lambda: Dual variables of the problem 2)rc: Reduced Costs.
Example

\[
\begin{align*}
\text{maximize} \quad & 5x_1 + 4x_2 + 3x_3 \\
\text{subject to} \quad & 2x_1 + 3x_2 + x_3 \leq 5 \\
& 4x_1 + 2x_2 + 2x_3 \leq 11 \\
& 3x_1 + 4x_2 + 2x_3 \leq 8 \\
& x_1, x_2, x_3 \geq 0
\end{align*}
\]

Example:

\begin{verbatim}
H=[]; %not a QP problem
maxmin = -1; %maximize
x_0 = []; %no initial solution
c=[5,4,3];
A=[2,3,1;4,1,2;3,4,2];
b=[5,11,8];
cctype = char('L'*ones(3,1));
vartype = char('C'*ones(3,1));
lb=zeros(3,1);
ub=[];
[xopt,fopt,status,extra]=cplexmex(maxmin,H,c,A,b,cctype,lb,ub,vartype,x_0);
\end{verbatim}
CPLEXMEX

>> xopt
xopt =
2.0000
0
1.0000

>> fopt
fopt =
13