Solving Linear and Integer Programs in MATLAB

Saeed Bastani

Department of Electrical and Information Technology, Lund University

Notations:

- The transposition operation is denoted by a superscript T (apostrophe in MATAB)

$$[1,2,3]^T = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}, \quad \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}^T = [1,2,3], \quad \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}^T = \begin{bmatrix} 1 & 4 \\ 2 & 5 \\ 3 & 6 \end{bmatrix}$$

- Row vector: $[1,2,3]$, Column vector: $\begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$

Definition 1: Let $\boldsymbol{c}$ be a column vector of length n, $\boldsymbol{b}$ a column vector of length m, and let A be a $m \times n$ matrix. A linear program associated with $\boldsymbol{c}$, A, and $\boldsymbol{b}$ is the minimum problem

$$\min \boldsymbol{c}^T \boldsymbol{x} \qquad (1)$$

Or the maximum problem

$$\max \boldsymbol{c}^T \boldsymbol{x} \qquad (2)$$

Subject to the constraint

$$A\boldsymbol{x} \le \boldsymbol{b} \qquad (3)$$

Note that $\boldsymbol{x}$ is a column vector of length n.

The general version of a linear program may involve equality constraints, in addition to inequality constraints expressed by (3).

Definition 2: Let $\boldsymbol{c}$ be a column vector of length n, $\boldsymbol{b}$ a column vector of length m, $\boldsymbol{b}_{eq}$ a column vector of length k, and let A and A_{eq} be $m \times n$ and $k \times n$ matrices, respectively. A linear program associated with $\boldsymbol{c}$, A, $\boldsymbol{b}$, A_{eq}, $\boldsymbol{b}_{eq}$ is the minimum problem (1) or the maximum problem (2), subject to the inequality constraint (3) and the equality constraint

$$A_{eq}\boldsymbol{x} = \boldsymbol{b}_{\boldsymbol{eq}} \qquad (4)$$

Example: $max\ z = 70\ x_1 + 50x_2$

Subject to

$$4\ x_1 + 3x_2 \le 240$$

$$2\ x_1 + \ x_2 \le 100$$

$$x_1 + 2x_2 = 60$$

$$3x_1 + 5x_2 = 90$$

$$x_1 \ge 0 \quad x_2 \ge 0$$

In this example, the objective function and constraints are characterized by the following vectors and matrices:

$\boldsymbol{c} = \begin{bmatrix} 70 \\ 50 \end{bmatrix}, A = \begin{bmatrix} 4 & 3 \\ 2 & 1 \end{bmatrix},\ \boldsymbol{b} = \begin{bmatrix} 240 \\ 100 \end{bmatrix}, A_{eq} = \begin{bmatrix} 1 & 2 \\ 3 & 5 \end{bmatrix},\ \boldsymbol{b_{eq}} = \begin{bmatrix} 60 \\ 90 \end{bmatrix}$

In MATLAB, the default operation mode of solvers is minimization, so you need to transform your problem to a minimization LP if your problem was originally a maximization problem. This is performed by changing the sign of $\boldsymbol{c}$ to $-\boldsymbol{c}$. Hence, in the previous example, we must change to $-\boldsymbol{c} = \begin{bmatrix} -70 \\ -50 \end{bmatrix}$. In that case, after running your optimization problem, you should change the sign of the output value of the objective function in order to match your original objective function.

Solving with MATLAB:

1. **Linear Programs**

For linear programs you should use *linprog* command. Before using this command, all constraints must be in the form of ≤ and =.

The complete format of *linprog* command is:

[x, fval, exitflag, output, lambda]= linprog(c, A, b, Aeq, beq, lb, ub, x0, options);

x: the optimal solution to your problem

fval: the optimal value of the objective function

exitflag: describes the exit condition

output: contains information about the optimization process

lambda: contains the Lagrange multipliers at the solution x (useful for sensitivity analysis)

c: the coefficient of objective function (in vector format)

A: the coefficients of the inequality constraints (in matrix format)

b: the right hand side (RHS) of inequality constraints (in vector format)

Aeq: the coefficients of the equality constraints (in matrix format)

beq: the right hand side (RHS) of equality constraints (in vector format)

lb, **ub**: the lower bound and upperbound of decision variables (in vector formats)

x0: the initial values of the decision variables. If you leave this blank ([]), the default value of zero is assumed.

options: contains parameters that describe how *linprog* should run

Your problem, however, may not involve all parameters in the *linprog* command, so you should use blank vectors [] instead of non-used parameters.

- If the problem only has inequality constraints, use this command:
  - [x, fval, exitflag, output, lambda] = linprog(c, A, b, [], [], lb, [], [], options);
- If the problem has inequality as well as equality constraints, use this format instead:
  - [x, fval, exitflag, output, lambda] = linprog(c, A, b, Aeq, beq, lb, [], [], options);

Before running *linprog,* you can specify how it should run, by manipulating the *options* parameter. For example, if you want to force *linprog* to use dual simplex algorithm, then do this way:
options = optimoptions('linprog', 'Algorithm', 'dual-simplex', 'Display', 'iter');

If you want to force *linprog* to use interior point algorithm instead of dual simplex, then do like this:
options = optimoptions('linprog', 'Algorithm', 'interior-point', 'Display', 'iter');

The 'Display' parameter tells *linprog* whether to enable or disable the display of the iterations of the algorithm on your screen. You can disable screen display by using '*off*' instead of '*iter*'.

Find out more information and examples of *linprog* from this link:
http://se.mathworks.com/help/optim/ug/linprog.html?requestedDomain=www.mathworks.com

2. Integer Programs

To solve integer programming problem, you should use *intlinprog* instead of *linprog*:

[x, fval, exitflag, output] = intlinprog(c, intcon, A, b, Aeq, beq, lb, ub, options);

The parameter *intcon* specifies those variables that admit integer values only. For example, intcon = [1,2,7] means x(1), x(2), and x(7) take only integer values.

The *options* parameter should look like this:

```
options = optimoptions('intlinprog', 'Display', 'off');
```

If you are interested to trace the run iterations on the screen, then *option* should be:

```
options = optimoptions('intlinprog', 'Display', 'iter');
```

Find out more information and examples of *intlinprog* from this link:

http://se.mathworks.com/help/optim/ug/intlinprog.html