Advanced Automation and Control

Laboratory 1

Prof. Davide M. Raimondo with the collaboration
of Dr. Marcello Torchio and Dr. Andrea Pozzi

Linear Programming

Problem description

The ACME company has a project to build at least 900 smart washing machines. The production process of such devices can be conducted in three different ways : (*i*) manually, (*ii*) semi-automatically, and (*iii*) automatically. Each of the available approaches involves the allocation of different amounts of human resources. In particular, the manual production demands 1 minute of qualified work, 40 minutes of non-qualified work and 3 minutes of assemblage. If the semi-automatic solution would be chosen, 4 minutes of qualified work, 30 minutes of non-qualified work, and 2 minutes for the assemblage would be required. Finally, 8, 20 and 4 minutes respectively would be required for the automatic method. ACME has a pool of 4500 minutes of qualified work, 36000 minutes of non-qualified work and 2700 minutes of assembly. The production costs of a washing machine are 70 euros if produced manually, 80 euros if produced semi-automatically, and 85 euros if produced automatically. Each smart washing machine is sold at 130 euros. From a commercial point of view, the ACME company is interested in one of the following objectives:

1. Find the optimal number of washing machines to be produced in order to minimize the costs.
2. Find the optimal number of washing machines to be produced in order to maximize the profit.

According to the chosen strategy, a corresponding optimization problem can be derived in order to support the decision-making process at ACME.

Problem formulation

The optimization variables of the problem are x_1, x_2 and x_3 which are used to represent the number of washing machines produced using the manual, semi-automatic, and automatic methods respectively.

If the company is interested in minimizing the production costs, then the following optimization problem needs to be solved

$$\min_{x_1,x_2,x_3} 70x_1 + 80x_2 + 85x_3$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &\geq 900 \\ x_1 + 4x_2 + 8x_3 &\leq 4500 \\ 40x_1 + 30x_2 + 20x_3 &\leq 36000 \\ 3x_1 + 2x_2 + 4x_3 &\leq 2700 \\ x_1, x_2, x_3 &\geq 0 \end{aligned} \; .$$

On the other hand, if ACME is interested in maximizing profit, the optimization problem has to be defined as

$$\max_{x_1,x_2,x_3} 60x_1 + 50x_2 + 45x_3$$

subject to

$$\begin{aligned} x_1 + x_2 + x_3 &\geq 900 \\ x_1 + 4x_2 + 8x_3 &\leq 4500 \\ 40x_1 + 30x_2 + 20x_3 &\leq 36000 \\ 3x_1 + 2x_2 + 4x_3 &\leq 2700 \\ x_1, x_2, x_3 &\geq 0 \end{aligned} .$$

The solution of the problems above has been addressed in the scripts:

- **LP_example_linprog.m** (it relies on the MATLAB function *linprog*)
- **LP_example_yalmip.m** (it makes use of *Yalmip*, a free MATLAB Toolbox for rapid prototyping of optimization problems)

Both scripts make use of the *MPT Toolbox* which, among the many features, includes Yalmip and allows the plotting of polyhedra. To execute the scripts use the commands:

- `[x,fval,exitflag]=LP_example_linprog(900,1,'max')`
- `[x,fval,exitflag]=LP_example_yalmip(900,1,'max')`

Both functions accept three arguments which represent the minimum number of washing machines to be produced, an index that if set to *1* allows the plotting of isocost lines and a string which allows to select whether we are interested in maximizing profit (*max*) or minimizing costs only (*min*).

The ouput arguments x, $fval$ and $exitflag$ provide respectively the optimal value of the optimization variables, the optimal cost and a flag indicating whether the optimization was completed successfully or any problem occurred (problem not feasible, unbounded, etc.).

Both scripts make use of the function Polyhedron (use *help Polyhedron* to see how it works). Please refer to the help also for the details of the function *linprog* and the symbolic variables *sdpvar* used by Yalmip. The scripts come with detailed comments (use the editor to see inside the files) which should help the reader in understanding the routines. The scripts return also the time required by linprog and yalmip to solve the LP programs. While Yalmip is quicker to program, it is usually slower in solving the optimization (conversion overhead).

Using the *linprog* script and then the *yalmip* one:

1. Solve the minimization problem by requiring a minimum of 900, 901, 902, 903, and 1200 machines. Are the obtained outcomes meaningful? Why?
2. Modify the optimization problem by removing the positivity constraints over the variable x_2. Solve the minimization problem with a minimum number of washing machines equal to 500. Do the results make sense? Why?
3. Modify the optimization problem by removing the constraints $x_2 \geq 0$ and $x_1 + x_2 + x_3 \geq 900$. Solve again the minimization problem with a random number as the minimum number of washing machines. What kind of result is obtained? Why?
4. According to the modified optimization problem of the above point, solve now a maximization problem. Why is there a difference with respect to the previous case?
5. Recover manually at least one the vertices of the feasible set of the original optimization problem. The MPT toolbox allows to obtain the vertices of a polyhedron P by using the command *P.v*

6. Coming back to the original problem, the number of smart washing machines produced manually, automatically and semi-automatically must be an integer. For this reason, solve the optimization problem with the *glpk* function, specifying the variable type integer (i.e. 'I'). Write *help glpk* further details

7. The same procedure can be done with *yalmip* by specifying *intvar* instead of *sdpvar*