Systems Theory

Laboratory 6: *Linear optimisation.*

Purpose of the exercise:
Linear programming primal / dual problem formulation and solution using simplex method and MATLAB/Simulink environment.

1. Introduction

Linear optimisation or *linear programming* is used to find an extremum of a problem represented by linear relationships. Linear programming algorithms are efficient solutions capable of solving complex optimisation problems. These solutions are applicable for problems with linear objective function, subject to linear inequality and/or equality constraints.

Linear programming may be used to solve a problem when the objective is to maximize some linear function, and there is a linear system of inequalities (equalities) that defines the constraints. Its feasible region is a set defined as the intersection of finitely many half spaces, each of which is defined by a linear inequality. Its objective function is a real-valued linear function defined on such a polyhedron (polytope). A linear optimisation algorithm finds a point in this polyhedron (polytope), where the objective function has an extremum, if such a point exists.

In a classical linear programming (LP) problem we consider continuous values of decisive variables. If the decisive variables are required to be integers (or binary), then the problem is called an integer programming, IP (or binary integer programming, BIP) problem. If only some decisive variables are required to be integers, then it is a mixed integer programming (MIP) problem.

2. LP problem with inequality constraints

Determine a vector of variables: $\boldsymbol{x} = [x_1, \ldots, x_n]^T \in \boldsymbol{R}^n$, *maximising* an objective linear form:

$$max(c_1 x_1 + \cdots + c_n x_n) \qquad OR \quad max(\boldsymbol{c}^T \boldsymbol{x})$$

subject to (s.t.) *m* **inequality** constraints:

$$\begin{matrix} a_{11}x_1 + \cdots + a_{1n}x_n \le b_1 \\ \vdots \\ a_{m1}x_1 + \cdots + a_{mn}x_n \le b_m \end{matrix} \qquad OR \qquad \boldsymbol{Ax} \le \boldsymbol{b}$$

where:

$$x_1 \ge 0, \ldots, x_n \ge 0 \qquad\qquad OR \qquad \boldsymbol{x} \ge 0$$

defines a set of admissible elements $\boldsymbol{X}$, i.e. a *feasible region*, being a convex *polyhedron* (*polytope*) *solid*.

<u>Theorem 1</u> (existence)

If *the set of admissible elements of the LP problem is non-empty, and* if *the value of the problem is finite,* then *the problem has a solution.*

<u>Theorem 2</u> (globality and localisation)

The extremum of the LP problem objective function is always global*, and achieved in the* extremum point *of the convex polyhedron (polytope)* $\boldsymbol{X}$.

3. Augmented LP problem with equality constraints

Determine an augmented vector of variables: $\overline{\boldsymbol{x}} = [\boldsymbol{x} \quad \boldsymbol{x}^s]^T = [x_1, \dots, x_n, x_{n+1}, \dots x_{n+m}]^T \in \boldsymbol{R}^{n+m}$, *maximising* an objective:

$$max(c_1x_1 + \cdots + c_nx_n) \qquad OR \qquad max(\boldsymbol{c}^T\boldsymbol{x})$$

s.t. m **equality** constraints:

$$\begin{matrix} a_{11}x_1 + \cdots + a_{1n}x_n + x_{n+1} & & = b_1 \\ \vdots & & \\ a_{m1}x_1 + \cdots + a_{mn}x_n & + x_{n+m} & = b_m \end{matrix} \qquad OR \qquad \overline{\boldsymbol{A}}\overline{\boldsymbol{x}} = \boldsymbol{b}$$

where:

$$x_1 \geq 0, \dots, x_n \geq 0, x_{n+1} \geq 0, \dots, x_{n+m} \geq 0 \qquad OR \qquad \overline{\boldsymbol{x}} \geq 0$$

defines a *feasible region* being a convex *polyhedron* (*polytope*) *surface*, while: $\boldsymbol{x}^s = [x_{n+1}, \dots x_{n+m}]^T$ is a vector of additional (*slack*), non-negative variables, introduced to replace inequality constraints with equalities, and: $\overline{\boldsymbol{A}} = [\boldsymbol{A} \quad \boldsymbol{I}]$.

Equivalent, partitioned matrix form of the augmented LP problem is to maximise $f = \boldsymbol{c}^T\boldsymbol{x}$ s.t.:

$$\begin{bmatrix} 1 & -\boldsymbol{c}^T & 0 \\ 0 & \boldsymbol{A} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} f \\ \boldsymbol{x} \\ \boldsymbol{x}^s \end{bmatrix} = \begin{bmatrix} 0 \\ \boldsymbol{b} \end{bmatrix}$$

where: $\boldsymbol{x} \geq 0, \boldsymbol{x}^s \geq 0$.

4. Dual linear programming (DLP) problem

A problem dual to the original (*primal*) LP problem with inequality constraints is formulated as follows (DLP):

Determine a vector of *dual* variables: $\boldsymbol{y} = [y_1, \dots, y_m]^T \in \boldsymbol{R}^m$, *minimising* an objective linear form:

$$min(b_1y_1 + \cdots + b_my_m) \qquad OR \qquad min(\boldsymbol{b}^T\boldsymbol{y})$$

s.t. n **inequality** constraints:

$$\begin{matrix} a_{11}y_1 + \cdots + a_{1m}y_m \geq c_1 \\ \vdots \\ a_{n1}y_1 + \cdots + a_{nm}y_m \geq c_n \end{matrix} \qquad OR \qquad \boldsymbol{A}^T\boldsymbol{y} \geq \boldsymbol{c}$$

where:

$$y_1 \geq 0, \dots, y_m \geq 0 \qquad OR \qquad \boldsymbol{y} \geq 0$$

<u>Remarks 1 ÷ 4</u>

- The *constraints* $b_1, \dots, b_m$ of the primal problem are the *objective function* coefficients in the dual problem.
- The *constraints* $c_1, \dots, c_n$ of the dual problem are the *objective function* coefficients in the primal problem.
- Each *constraint* in the primal problem corresponds to a *variable* in the dual problem, and *vice versa*.
- The problem *dual* to DLP is again the *primal* LP problem, and it makes sense to speak of *a pair of dual* (complementary) *linear programming problems*.

Theorem 3 (duality)

The following alternative holds for a pair of dual linear programming problems: EITHER *the values of both problems are* finite *and* equal AND *both problems have solutions,* OR *the set of admissible elements of one of the problems is empty* AND *in the other problem* EITHER *the set of admissible elements is empty* OR *the value of the problem is* infinite.

In the former *case, vectors* $\boldsymbol{x}^* \in \boldsymbol{R}^n$ *and* $\boldsymbol{y}^* \in \boldsymbol{R}^m$ *are solutions of the* primal *and* dual *problems, respectively,* IF AND ONLY IF *they are* admissible *in these problems,* AND *satisfy one of the following two equivalent relations:*

$$\boldsymbol{c}^T\boldsymbol{x}^* = \boldsymbol{b}^T\boldsymbol{y}^* \quad \text{(D.1)}$$

$$\boldsymbol{y}^{*T}(\boldsymbol{A}\boldsymbol{x}^* - \boldsymbol{b}) = \boldsymbol{x}^{*T}(\boldsymbol{A}^T\boldsymbol{y}^* - \boldsymbol{c}) = 0 \quad \text{(D.2)}$$

5. Simplex method

(a) Basic feasible solution

Consider the augmented LP problem. A **basis** $\boldsymbol{B}$ is a *square*, *non-singular* matrix consisting of *m* columns of $\overline{\boldsymbol{A}}$. A column vector of variables $\boldsymbol{x}^{\boldsymbol{B}}$, associated with the columns in $\boldsymbol{B}$ arranged in the same order, is the **basic vector** corresponding to it. Let $\boldsymbol{D}$ be the matrix consisting of *n* columns of $\overline{\boldsymbol{A}}$ not in $\boldsymbol{B}$, and let $\boldsymbol{x}^{\boldsymbol{D}}$ be the vector of variables associated with these columns. Then:

- The columns in $\boldsymbol{B}$ are *called* the **basic**, in $\boldsymbol{D}$ – **nonbasic columns**.
- The variables in $\boldsymbol{x}^{\boldsymbol{B}}$ are *called* the **basic**, in $\boldsymbol{x}^{\boldsymbol{D}}$ – **nonbasic variables**.

Rearranging the variables, (augmented) LP problem can be written in partitioned form:

$$\overline{\boldsymbol{A}}\overline{\boldsymbol{x}} = \boldsymbol{b} \quad \xrightarrow{\text{yields}} \quad \boldsymbol{B}\boldsymbol{x}^{\boldsymbol{B}} + \boldsymbol{D}\boldsymbol{x}^{\boldsymbol{D}} = \boldsymbol{b}, \quad \boldsymbol{x}^{\boldsymbol{B}} \geq 0, \boldsymbol{x}^{\boldsymbol{D}} \geq 0$$

The **basic solution**, corresponding to the *basis* $\boldsymbol{B}$ is obtained by setting $\boldsymbol{x}^{\boldsymbol{D}} = 0$, and then solving the remaining system for the values of the basic variables: $\boldsymbol{x}^{\boldsymbol{B}} = \boldsymbol{B}^{-1}\boldsymbol{b}$. This solution is *feasible*, if $\boldsymbol{B}^{-1}\boldsymbol{b} \geq 0$, and in this case $\boldsymbol{B}$ is said to be a **feasible basis**, and the solution: $\boldsymbol{x}^{\boldsymbol{B}} = \boldsymbol{B}^{-1}\boldsymbol{b}$, $\boldsymbol{x}^{\boldsymbol{D}}$=0 is called the **basic feasible solution** (BFS) corresponding to it.

(b) Pivot step

Given a basis $\boldsymbol{B}$, the canonical *tableau* with *respect to it* is obtained by multiplying the system of equality constraints on the left by $\boldsymbol{B}^{-1}$:

$$[\boldsymbol{B} \quad \boldsymbol{D}]\begin{bmatrix}\boldsymbol{x}^{\boldsymbol{B}}\\ \boldsymbol{x}^{\boldsymbol{D}}\end{bmatrix} = \boldsymbol{b}$$

$$[\boldsymbol{I} \quad \boldsymbol{B}^{-1}\boldsymbol{D}]\begin{bmatrix}\boldsymbol{x}^{\boldsymbol{B}}\\ \boldsymbol{x}^{\boldsymbol{D}}\end{bmatrix} = \boldsymbol{B}^{-1}\boldsymbol{b} = \overline{\boldsymbol{b}}$$

When the basic and nonbasic columns are rearranged in proper order, the canonical *tableau* becomes:

basic variables	$\boldsymbol{x}^{\boldsymbol{B}}$ $\boldsymbol{x}^{\boldsymbol{D}}$	
$\boldsymbol{x}^{\boldsymbol{B}}$	$\boldsymbol{I}$ $\boldsymbol{B}^{-1}\boldsymbol{D}$	$\boldsymbol{B}^{-1}\boldsymbol{b} = \overline{\boldsymbol{b}}$

The main step in the *simplex* algorithm for linear programs is the (Gauss Jordan elimination) *pivot* step. In each stage of the algorithm, the *basis* is changed by *bringing* into the *basic vector* exactly one *nonbasic* variable known as the *entering* variable. Its updated column vector is the pivot column for this basis change. The *dropping* variable has to be determined accordingly to guarantee that the new basis obtained after the pivot step will also be a *feasible* basis (more details in [6]).

(c) Implementation procedure

Simplex method may be effectively used to obtain an extreme point of a convex n-dimensional polyhedron (polytope) being a feasible region. For greater number of decisive variables (n), computer algorithms are developed to execute the pivotal transformations of the problem *tableau*, searching the feasible polyhedron (polytope) vertices for extremum.

In MATLAB/Simulink environment, *linprog* is intended for standard LP problems. For MIP and IP problems, *intlinprog* may be used. Regarding all: MIP, IP, BIP, and LP problems, also *solve* command may be used to solve a problem created with *optimproblem* and optimisation variable(s) defined with *optimvar* (see LP problem example: search for *optimproblem* in MATLAB Documentation).

6. Tasks

Consider stiff platform support optimisation LP (IP) problem, as follows.

Maximise load capacity of a stiff platform *support*, assuming that:

- the platform is supported by conceptual columns,
- a total number of columns L is limited, as well as their total mass M, and total material price P,
- two types of material may be used, each characterised with specific price per unit (P_1 and P_2, respectively, e.g. in thousands of PLN), mass per unit (M_1 and M_2, in tons), and load capacity per unit (S_1 and S_2, in MN).

The parameters of the problem:

Material Grade A

Parameter	Value
P_1	2.5
M_1	0.8
S_1	6.0

Material Grade B

Parameter	Value
P_2	2.0
M_2	0.6
S_2	5.0

Total Means

Parameter	MAX Value
P	110
M	36
L	50

The regarded problem is an example of a linear (integer) optimisation task, whereas the detailed distribution of load and columns layout analysis is out of the scope here, as well as possible consolidations of multiple conceptual columns to obtain the required spot load capacity (e.g. *5* columns made of *GradeA* and *6* made of *GradeB*, or even: *2.5* of *GradeA* and *3.3* of *GradeB*).

Remark 5

If we search for an optimum in the whole set of *Reals+*, and the found x^* is from *Integers+* subset of *Reals+*, this is a solution of *IP* problem. *IP* problems are generally NP-hard (non-deterministic polynomial-time hardness).

For the task described above:

1. Formulate a primal LP / IP problem with the inequality constraints.
2. Using *linprog* / *intlinprog MATLAB* function, solve the primal problem and determine optimal vector of decisive variables: $\boldsymbol{x}^* = [{x_1}^*, {x_2}^*]^T \in \boldsymbol{R}^2$ ($x_1 \geq 0$ and $x_2 \geq 0$ are number of columns made of material *Grade A* and *Grade B*, respectively), as well as corresponding *maximum* value of the *objective function*: $f = S_1{x_1}^* + S_2{x_2}^*$.
3. Formulate an augmented primal problem with the equality constraints by introduction of slack variables.
4. Using *linprog* / *intlinprog MATLAB* function, solve the augmented primal problem and determine optimal vector of variables, as well as corresponding maximum objective function value.
5. Formulate a dual linear programming problem (DLP) with the inequality constraints: *minimise* total *means* (price, mass, and number of columns) necessary to fulfil (equalise or exceed) the demanded dual problem *constraints* i.e. load capacity.
6. Using *linprog* / *intlinprog MATLAB* function, solve the DLP problem and determine optimal vector of variables: $\boldsymbol{y}^* = [{y_1}^*, {y_2}^*, {y_3}^*]^T \in \boldsymbol{R}^3$, as well as corresponding *minimum* objective function $f_d = P{y_1}^* + M{y_2}^* + L{y_3}^*$ value.
7. Verify applicability of *Theorem 3*. Substitute the obtained $\boldsymbol{x}^*$ and $\boldsymbol{y}^*$ into equations (D.1) and (D.2).
8. SUPPLEMENTARY: Solve IP versions of the regarded problems defined with *optimproblem*, using *solve* command (maximum of 1 additional point).

References:

[1] MATLAB/Simulink documentation: http://www.mathworks.com
[2] G.B. Dantzig „Linear Programming and Extensions", The RAND Co., Princeton University Press, 1963.
[3] W.A. Poe and S. Mokhatab „Modeling, Control, and Optimization of Natural Gas Processing Plants", Elsevier, 2017.
[4] A.D. Ioffe and V.M. Tihomirov „Theory of extremal problems. Studies in mathematics and its applications", Amsterdam–New York–Oxford: North-Holland Publishing Company, 1979.
[5] W. Pogorzelski „Teoria systemów i metody optymalizacji", Oficyna Wydawnicza PW, Warszawa, 1996.
[6] K.G. Murty „Linear Complementarity, Linear And Nonlinear Programming", 1997.
[7] D.S. Naidu „Optimal Control Systems", CRC Press, 2003.