Bulletin of the *Transilvania* University of Braşov
Series V: Economic Sciences • Vol. 7 (56) No. 2 - 2014

BOOLEAN VARIABLES IN ECONOMIC MODELS SOLVED BY LINEAR PROGRAMMING

Dorin LIXĂNDROIU[1]

Abstract: *The article analyses the use of logical variables in economic models solved by linear programming. Focus is given to the presentation of the way logical constraints are obtained and of the definition rules based on predicate logic. Emphasis is also put on the possibility to use logical variables in constructing a linear objective function on intervals. Such functions are encountered when costs or unitary receipts are different on disjunct intervals of production volumes achieved or sold. Other uses of Boolean variables are connected to constraint systems with conditions and the case of a variable which takes values from a finite set of integers.*

Key Words: *linear programming, Boolean variable, economic model.*

1. Introduction

A wide class of optimization models in economics are solved by means of linear programming. The linear programming problem is comprised within the general mathematical programming models and is characterized by the fact that both the objective function and the constraints are expressed mathematically by linear functions [2].

The general form of the linear programming problem *(LPP)* in matrix notation is:

Max [Min] $f(X) = C^t \cdot X$

$$A \cdot X \leq B \qquad (1)$$
$$X \geq 0$$

where:

$A(m,n)$ – is the matrix of coefficients of the constraint system

$B(m,1)$ – is the column vector of free terms

$X(n,1)$ – is the column vector of the n variables

$C^t(1,n)$ – is the transposed column vector (whose components determine the unknown coefficients of the objective function).

In the general form of the *LPP*, it is considered that variables are real numbers.

There are many economic applications of great importance which lead to models which also impose other conditions on variables.

If the unknowns are Boolean variables, i.e. the final solutions for the linear programming problem is 0/1, a Boolean linear programming problem is obtained. There is also the possibility that only some of the variables are Boolean.

[1] Faculty of Economic Sciences and Business Administration, *Transilvania* University of Braşov.

These problems are encountered in models where the user wants, for example, to select projects out of a given set, to select certain courses of action out of a given set etc.

2. Logical constraints in linear programming models

A general model for choosing investment projects is considered. There are *n* investment projects whose ensuing benefits as a result of the implementation are known [4], [5].

We note:

$x_i = 1$, if the project *i* is accomplished

$x_i = 0$, if the project *i* is not accomplished

The question is to determine the investment projects to be performed so that the total benefit should be maximum. We further consider introducing constraints on some logical operations.

The following logical symbols will be used: $\bar{x}$ (negation, not), $x \vee y$ (disjunction, or), $x \wedge y$ (conjunction, and), $x \equiv y$ (equivalence), $x \Rightarrow y$ (implication).

The shift from logical formalism to algebraic formalism can be achieved by replacing terms of the form $(x=1)$ by the variable *x* and the conditions of the form $\overline{(x=1)}$ by $(1-x)$.

The logical operator *or* is replaced by the addition operator "+", and the logical operator *and* is replaced by the multiplication operator "." .

The logical truth condition requires that the algebraic expression obtained be greater than or equal to 1.

a) Incompatibility

Two projects are incompatible when they cannot be accomplished at the same time; only one or neither can be accomplished (the projects involve the use of the same limited resources, or are technical variants having the same purpose).

The truth table corresponding to the incompatibility situation is:

	$x_j = 0$	$x_j = 1$
$x_i = 0$	True	True
$x_i = 1$	True	False

Analysing the table, one can see that this leads to the constraint:

$$x_i + x_j \le 1 \qquad (2).$$

Constraint (2) may be deduced using algebraic methods. The incompatibility condition is expressed by means of the logical condition:

$\overline{(x_i = 1) \wedge (x_j = 1)}$ which must be true.

We have:

$$\overline{(x_i = 1) \wedge (x_j = 1)} \equiv \overline{(x_i = 1)} \vee \overline{(x_j = 1)} \Leftrightarrow$$
$$\Leftrightarrow (1 - x_i) + (1 - x_j) \ge 1 \Leftrightarrow x_i + x_j \le 1$$

(De Morgan's law has been applied $\overline{X \wedge Y} = \overline{X} \vee \overline{Y}$)

Generalization:

- out of *n* projects, one or none can be accomplished: $\sum_{i=1}^{n} x_i \le 1$
- out of *n* projects, at most *m* projects can be accomplished: $\sum_{i=1}^{n} x_i \le m$

b) Disjunction (or)

At least one of the projects *i* or *j* must be accomplished.

The truth table corresponding to this situation is:

	$x_j = 0$	$x_j = 1$
$x_i = 0$	False	True
$x_i = 1$	True	True

Analysing the table, one can see that this leads to the constraint:

$$x_i + x_j \geq 1 \tag{3}$$

Constraint (3) may be deduced using algebraic methods as follows:

- the disjunction condition *"at least one of the projects i or j must be accomplished"* is expressed by means of the logical condition:

$(x_i = 1) \vee (x_j = 1)$ which must be true

We have:

$(x_i = 1) \vee (x_j = 1) \Leftrightarrow x_i + x_j \geq 1$

Generalization:

- out of n projects, at least one must be accomplished: $\sum_{i=1}^{n} x_i \geq 1$
- out of n projects, at least m must be accomplished: $\sum_{i=1}^{n} x_i \geq m$

c) Alternative (xor)

It involves that, given two projects, either of them should be accomplished, but not both.

The truth table corresponding for this situation is:

	$x_j = 0$	$x_j = 1$
$x_i = 0$	False	True
$x_i = 1$	True	False

Analysing the table, one can that this leads to the constraint:

$$x_i + x_j = 1 \tag{4}$$

The alternative (xor) is expressed by means of the logical condition:

$$\left((x_i = 1) \wedge \overline{(x_j = 1)}\right) \vee \left(\overline{(x_i = 1)} \wedge (x_j = 1)\right)$$

which must be true.

We have:

$$\left((x_i = 1) \wedge \overline{(x_j = 1)}\right) \vee \left(\overline{(x_i = 1)} \wedge (x_j = 1)\right) \Leftrightarrow$$
$$\Leftrightarrow x_i \cdot (1 - x_j) + (1 - x_i) \cdot x_j \geq 1 \Leftrightarrow$$
$$x_i - x_i \cdot x_j + x_j - x_i \cdot x_j \geq 1 \Leftrightarrow$$
$$\Leftrightarrow x_i + x_j \geq 1 + 2 \cdot x_i \cdot x_j$$

Because the situations $x_i = x_j = 0$ and $x_i = x_j = 1$ are excluded, it results:

$x_i + x_j = 1$ and $x_i + x_j \geq 1 + 2 \cdot x_i \cdot x_j$

are equivalent.

In optimization models leading to the *LPP,* the constraint will be expressed under the linear form given by (4).

Generalization:

- out of n projects, only one must be accomplished: $\sum_{i=1}^{n} x_i = 1$
- out n projects, m projects must be accomplished: $\sum_{i=1}^{n} x_i = m$

d) Implication

If project i involves project j, project i cannot be accomplished without project j being accomplished.

But, if project i is not accomplished, project j can be accomplished or not.

The truth table corresponding for this situation is:

	$x_j = 0$	$x_j = 1$
$x_i = 0$	True	True
$x_i = 1$	False	True

Analysing the table, one can see that this leads to the constraint:

$$x_i \leq x_j \tag{5}$$

Constraint (5) may be deduced using algebraic methods.

The implication is expressed by means of the logical condition:
$(x_i = 1) \Rightarrow (x_j = 1)$ which must be true.

We have:
$$(x_i = 1) \Rightarrow (x_j = 1) \equiv \overline{(x_i = 1)} \vee (x_j = 1) \Leftrightarrow$$
$$\Leftrightarrow (1 - x_i) + x_j \geq 1 \Leftrightarrow x_i \leq x_j$$

(We have applied the definition of implication $X \Rightarrow Y \equiv \overline{X} \vee Y$)

The implication may appear in more complex situations. For analysis we build the logic situations table for *3* projects:

Case	x_i	x_h	x_k
A	0	0	0
B	0	0	1
C	0	1	0
D	0	1	1
E	1	0	0
F	1	0	1
G	1	1	0
H	1	1	1

The accomplishment of project i involves the accomplishment of at least one of the projects h and k.

The situation is defined by the compliance with case F, G or H and can be expressed by the constraint:

$$x_i \leq x_h + x_k, \tag{6}$$

which verifies the 3 cases: *(F):* $1 \leq 0 + 1$, *(G):* $1 \leq 1 + 0$, *(H):* $1 \leq 1 + 1$.

The constraint (6) may be deduced using algebraic methods:
$$(x_i = 1) \Rightarrow (x_h = 1) \vee (x_k = 1) \equiv$$
$$\equiv \overline{(x_i = 1)} \vee ((x_h = 1) \vee (x_k = 1)) \Leftrightarrow$$
$$\Leftrightarrow (1 - x_i) + x_h + x_k \geq 1 \Leftrightarrow x_i \leq x_h + x_k$$

The accomplishment of project i involves the accomplishment of projects h and k.

The situation is defined by the compliance with case *H* and can be expressed by the constraint:

$$2 \cdot x_i \leq x_h + x_k, \tag{7}$$

which verifies *(H):* $2 \cdot 1 \leq 1 + 1$, i.e. $2 \leq 2$.

Using algebraic methods, we obtain:
$$(x_i = 1) \Rightarrow (x_h = 1) \wedge (x_k = 1) \equiv$$
$$\equiv \overline{(x_i = 1)} \vee ((x_h = 1) \wedge (x_k = 1)) \Leftrightarrow$$
$$\Leftrightarrow (1 - x_i) + x_h \cdot x_k \geq 1 \Leftrightarrow x_i \leq x_h \cdot x_k$$

The constraint $x_i \leq x_h \cdot x_k$ is equivalent to the constraint $2 \cdot x_i \leq x_h + x_k$.

Obviously, in an optimization model that results in the *LPP,* the constraint will be expressed in the linear form given by (7).

The accomplishment of projects h or k involves the accomplishment of project i.

The situation is defined by the compliance with case F, G or H and can be expressed by the constraint:

$$x_h + x_k \leq 2 \cdot x_i, \tag{8}$$

which verifies the 3 cases:
(F): $0 + 1 \leq 2 \cdot 1$, *(G):* $1 + 0 \leq 2 \cdot 1$,
(H): $1 + 1 \leq 2 \cdot 1$.

Using algebraic formalism, we obtain:

$$(x_h = 1) \vee (x_k = 1) \Rightarrow (x_i = 1) \equiv$$
$$\equiv \overline{(x_h = 1) \vee (x_k = 1)} \vee (x_i = 1)$$
$$\equiv \left(\overline{(x_h = 1)} \wedge \overline{(x_k = 1)}\right) \vee (x_i = 1) \Leftrightarrow$$
$$\Leftrightarrow (1 - x_h) \cdot (1 - x_k) + x_i \geq 1$$
$$\Leftrightarrow x_i \geq x_h + x_k - x_h \cdot x_k$$

Obviously, the two constraints are equivalent, but in the optimization model that leads to the *LPP,* it is required to express the constraint in the linear form given by (8).

3. Rules for defining logical constraints

Using Boolean variables in modelling business processes by means of linear programming allows us to define a large number of logical constraints.

We consider *n* sentences $P_1, P_2, ..., P_n$ and let the binary variables be $x_1, x_2, ..., x_n$, defined as follows:

$x_i = 1$, if P_i is true

$x_i = 0$, if P_i is false.

Based on predicate logic, we can state the following rules [1]:

Rule 1. *At most one of the propositions $P_1, P_2, ..., P_n$ can be true*, leads to the condition:

$$\sum_{i=1}^{n} x_i \leq 1.$$

Generalizations are immediate *($k \leq n$):*

- *At most k of the propositions $P_1, P_2, ..., P_n$ can be true,* it results: $$\sum_{i=1}^{n} x_i \leq k$$
- *At least k of the propositions $P_1, P_2, ..., P_n$ are true,* it results: $\sum_{i=1}^{n} x_i \geq k$.

Rule 2. *One and only one of the propositions $P_1, P_2, ..., P_n$ is true,* leads to the condition: $\sum_{i=1}^{n} x_i = 1$.

The generalization is immediate *($k \leq n$):*

- *Exactly k of the propositions $P_1, P_2, ..., P_n$ are true,* it results: $\sum_{i=1}^{n} x_i = k$.

Rule 3. *If proposition P_1 is true, then proposition P_2 is true,* leads to the condition: $x_1 \leq x_2$, which forces x_2 to take the value 1, if x_1 is equal to *1*. We can see that proposition P_2 can be true, and proposition P_1 false.

The generalization of this reasoning will be immediate:

P_2 true $\Rightarrow$ P_3 true, i.e. $x_2 \leq x_3$.

Rule 4. *Proposition P_1 is true if and only if P_2 is true,* leads to the condition: $x_1 = x_2$. The generalization for *n* propositions which should be all simultaneously true or simultaneously false leads to *n-1* constraints: $x_1 = x_2$, $x_2 = x_3, ..., x_{n-1} = x_n$.

Rule 5. *If proposition P_1 is true or proposition P_2 is true, then proposition P_3 is true,* leads to the condition: $x_1 + x_2 \leq 2 \cdot x_3$. The generalization is immediate: *if at least one of the propositions $P_1, P_2, ..., P_n$ is true, then the proposition P_{n+1} is true*, therefore:

$$\sum_{i=1}^{n} x_i \leq n \cdot x_{n+1}.$$

Rule 6. *If proposition P_1 is true and proposition P_2 is true, then proposition P_3 is true,* leads to the following conditions:

$$x_1 + x_2 \leq 1 + x_3, \quad x_3 \leq x_1, \quad x_3 \leq x_2.$$

The generalization is immediate: *if all propositions $P_1, P_2, ..., P_n$ are true, then proposition P_{n+1} is true*, therefore *n+1* constraints result:

$$\sum_{i=1}^{n} x_i \leq n - 1 + x_{n+1},$$

$$x_{n+1} \leq x_i, \quad i = 1,2,...,n$$

4. Other applications of Boolean variables in linear programming

a) Variables with values in a set of integers

The concrete type of certain modelled economic phenomena require that variables subject to constraints should only take values from certain discrete sets (most of the times integers). For example, if variable x can take only values from the set of integers $\{k_1, k_2, ..., k_s\}$ we can write:

$$x = \alpha_1 \cdot k_1 + \alpha_2 \cdot k_2 + ... + \alpha_s \cdot k_s \quad \text{with}$$
$$\sum_{i=1}^{s} \alpha_i = 1 \quad \text{and}$$
$$\alpha_i \in \{0, 1\}, \quad i = 1,2,...,s \tag{9}$$

A mixed linear programme results with Boolean variables $\{\alpha_1, \alpha_2, ..., \alpha_s\}$.

Obviously, in order for this replacement to be possible, variable x must be upper bounded, so that the number of Boolean variables considered should be relatively small. Generally, such a superior increase is generally natural in economic models.

b) Constraint systems with conditions

There are situations in which the construction of a model leads to programmes in which constraints or groups of constraints occur which are mutually exclusive or which are not necessarily totally complied with. Let us analyse the following cases, in which the objective function is linear and in the constraint system:

$$\begin{array}{l} g_1(X) \geq 0 \\ g_2(X) \geq 0 \\ ...... \\ g_m(X) \geq 0 \end{array} \tag{10}$$

we formulate the following hypotheses:

I1. One and only one of the inequalities (10) is true.

Solution. We assume constraints are bounded and let L_i be the lower bound of the function $g_i(X), \quad i,1,2,...,m.$

The constraint system (10) is equivalent with:

$$\begin{array}{l} g_i(X) - (1-\alpha_i) \cdot L_i \geq 0, \quad i = 1,2,...,m \\ \sum_{i=1}^{m} \alpha_i = 1, \quad \alpha_i \in \{0, 1\}, \quad i = 1,2,...,m \end{array} \tag{11}$$

I2. Out of the m inequalities (10), k must be verified.

Solution. If L_i has the same significance as before, then the constraint system (10) is equivalent with:

$$\begin{array}{l} g_i(X) - \alpha_i \cdot L_i \geq 0, i = 1,2,...,m \\ \sum_{i=1}^{m} \alpha_i = m - k, \alpha_i \in \{0, 1\}, i = 1,2,...,m \end{array} \tag{12}$$

I3. Out of the m inequalities (10), at least k must be satisfied.

Solution. The constraint system (10) becomes:

$$\begin{array}{l} g_i(X) - \alpha_i \cdot L_i \geq 0, \quad i = 1,2,...,m \\ \sum_{i=1}^{m} \alpha_i \leq m - k, \ \alpha_i \in \{0, 1\}, i = 1,2,...,m \end{array} \tag{13}$$

In each of the cases considered, the initial problem becomes a mixed linear programming problem by adding Boolean variables $\alpha_1, \alpha_2, ..., \alpha_m$.

c) The linear objective function on intervals

In economic models leading to linear programming, the objective function to be optimized (maximization or minimization) is linear. For example, in the models for optimizing production, for a production volume x and a unitary variable cost c, the objective function takes the form: $z = c \cdot x$.

This approach does not square with reality when fixed costs vary on intervals or direct variable costs (or unitary receipts) are different on disjunct intervals according to the size of the production achieved or sold.

In order to write the objective functions in these complex economic situations, in what follows we focus on the *linear cost functions on intervals* [1], [3].

The form of the objective function presented above cannot be accepted when the production decision involves a fixed cost as well. We formulate the hypothesis: if production is null $(x=0)$, then the production cost of the product considered is *0* and equal to $c \cdot x + K$, if $x > 0$. For the resolution, the binary variable y is introduced, defined as follows:

$y = 0$, if $x = 0$

$y = 1$, if $x > 0$.

This allows the introduction of the fixed cost in the objective function, which takes the form:

$$\min z, \text{ with } z = c \cdot x + K \cdot y + ... \quad (14)$$

The connection between x and y is accomplished by means of the additional constraint:

$x \le M \cdot y$,

where M is a constraint greater than or equal to the maximum value that x can take.

Generalization

For each linear objective function (involving costs or receipts) on intervals, variables x_k can be introduced, which take values in the interval $(M_{k-1}, M_k]$ with $M_0 = 0$.

On each interval, the variable cost is constant, and values x_k will all be null except for the interval which includes value *x*. The generalization allows addressing uniform and progressive discounts in provisioning activities. Figure 1 shows an example of objective function of the total cost.

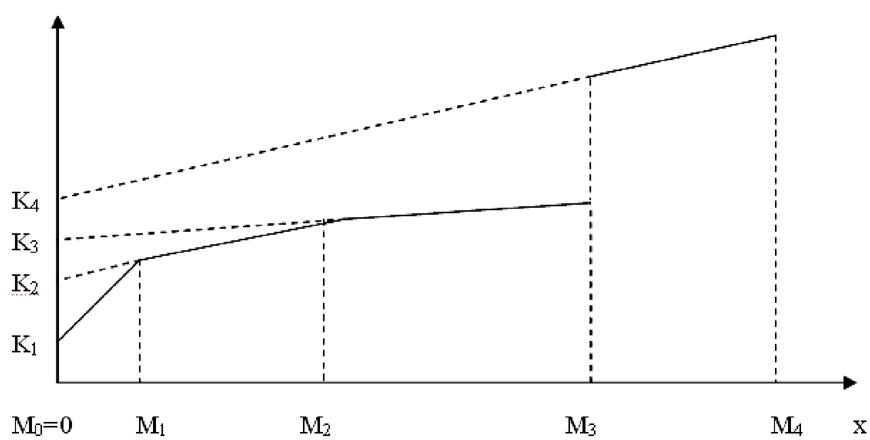

Fig. 1. *Total cost function* [1]

In the construction of the model represented in Figure 1, variable *x* will be replaced by $x_1 + x_2 + x_3 + x_4$.

The objective function for the relative part of this production becomes:

$\min z$, with

$$z = (c_1 \cdot x_1 + K_1 \cdot y_1) + (c_2 \cdot x_2 + K_2 \cdot y_2) + \\ + (c_3 \cdot x_3 + K_3 \cdot y_3) + (c_4 \cdot x_4 + K_4 \cdot y_4) + ...$$

with the constraints:

$$\begin{aligned} &0 \le x_1 \le M_1 \cdot y_1 \\ &M_1 \cdot y_2 \le x_2 \le M_2 \cdot y_2 \\ &M_2 \cdot y_3 \le x_3 \le M_3 \cdot y_3 \\ &M_3 \cdot y_4 \le x_4 \le M_4 \cdot y_4 \\ &y_1 + y_2 + y_3 + y_4 = 1 \end{aligned} \tag{15}$$

Remark. If x varies discontinuously, at a small variation ε of the cost function, we have to decide if $x = M_i$ belongs to the interval i or $i+1$. Supposing the upper bounds of the intervals are excluded, i.e.:

$$x_1 \in [0, M_1), \quad x_2 \in [M_1, M_2), \ldots$$

we will have:

$$\begin{aligned} &0 \le x_1 \le (M_1 - \varepsilon) \cdot y_1 \\ &M_1 \cdot y_2 \le x_2 \le (M_2 - \varepsilon) \cdot y_2 \\ &M_2 \cdot y_3 \le x_3 \le (M_3 - \varepsilon) \cdot y_3 \\ &M_3 \cdot y_4 \le x_4 \le (M_4 - \varepsilon) \cdot y_4 \\ &y_1 + y_2 + y_3 + y_4 = 1 \end{aligned} \tag{16}$$

The last constraint does not allow x to belong to the interval $(M_i - \varepsilon, M_i)$.

Model (15) can be generalized for an N number of intervals:

$$\min z, \quad \text{with } z = \sum_{i=1}^{N} (c_i \cdot x_i + K_i \cdot y_i)$$

and the constraints:

$$\begin{aligned} &M_{i-1} \cdot y_i \le x_i \le M_i \cdot y_i, \; i = 1,2,\ldots,N \\ &y_1 + y_2 + \ldots + y_N = 1 \end{aligned} \tag{17}$$

It can be noticed that model (17) comprises $(2 \cdot N + 1)$ restrictions.

5. Conclusions

The use of logical variables in economic models solved by means of linear programming allows the construction of optimization models for productive processes which are closer to reality.

Thus, there can be a decrease in the inflexibility of linearity constraints frequently encountered in linear programming and of the simplifying hypotheses which are made regarding the definition of the objective function and of the model constraints.

References

1. Giard, V.: *Processus productifs et Programmation linéaire.* Paris. Ed. Economica, 1998.
2. Hillier, F., Lieberman, G.: *Introduction to Operations Research.* McGraw Hill, International Editions, 2005.
3. Lixăndroiu D.: *Solving Economic Models by Means of Linear Programming with Objective Function Linear on Intervals.* In: Review of Management and Economical Engineering, 2007, Vol.6, No.5, p.354-358.
4. Lixăndroiu, D.: *On Logical Restrictions in Economic Models Solved through Means of Linear Programming.* In: Proceedings of the 6 Biennial International Economic Symposium SIMPEC 2006, Braşov, vol.1, p. 175-181.
5. Vallin, Ph., Vanderpooten, D. : *Aide à la décision.* Paris. Édition Ellipses, 2002.