

Chapter 11

Some Examples Using GAMS

11.1 Introduction

In this chapter we explain how some of the examples given in previous chapters can be solved using GAMS. To avoid repetition, we give a detailed explanation of the GAMS commands in the first example, and in the remaining examples we only emphasize the new interesting features.

For the sake of a better understanding, we write GAMS reserved words, data and equation names in capital letters, and the rest of code, in lowercase letters.

11.2 Linear Programming Examples

This section deals with the linear programming problems analyzed in previous chapters.

11.2.1 The Transportation Problem

The transportation problem was analyzed in detail in Section 1.2, page 4, and in Example 1.1, page 5. A particular example can be stated as:

Minimize

$$Z = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}. \quad (11.1)$$

subject to

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= u_i; \quad \forall i = 1 \dots m, \\ \sum_{i=1}^m x_{ij} &= v_j; \quad \forall j = 1 \dots n, \\ x_{ij} &\geq 0; \quad \forall i = 1 \dots m; \forall j = 1 \dots n, \end{aligned} \tag{11.2}$$

where $m = n = 3$ and

$$\mathbf{C} = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 1 & 2 \\ 3 & 2 & 1 \end{pmatrix}, \quad \mathbf{u} = \begin{pmatrix} 2 \\ 3 \\ 4 \end{pmatrix}, \quad \text{and } \mathbf{v} = \begin{pmatrix} 5 \\ 2 \\ 2 \end{pmatrix}.$$

From a GAMS-user point of view, the following considerations should be taken into account:

- This problem presents three kinds of equations. The first one is a single equation that represents the objective function, the second one is a group of three equations associated with the three origins, and the third one is a group of three equations associated with the three destinations. Therefore, two different sets for origins and destinations should be defined to represent, in a compact way, the last two groups of constraints. Notice the use of the symbol ‘*’ to list the members of a set.
- There are three different data arrays: c_{ij} , u_i , v_j . Data u_i and v_j should be declared using the parameter command, due to the fact that both of them are one-dimensional arrays. The c_{ij} array can be declared using either the parameter or the table command. In this case, the table command has been used.
- The optimization variable ‘ $\mathbf{z}$ ’ is declared to represent the value of the objective function to be minimized.
- The array x_{ij} is the array of the optimization variables. There are as many variables as the number of origins times the number of destinations.
- After declaring a variable, it is convenient to define its type. The x_{ij} variables are positive in sign, and ‘ $\mathbf{z}$ ’ is not restricted in sign (free by default).
- Once the necessary sets, data and optimization variables have been defined, the constraints are declared and defined. The objective function is declared as a single equation (it does not depend on any set). In addition, two groups of constraints are declared using two different names. The first group ‘SHIP(I)’ represents as many constraints as the number of origins, and therefore the corresponding set is indicated. The second group ‘RECEIVE(J)’ represents as many constraints as the number of destinations, and therefore the corresponding set is indicated.

- Once the constraints have been declared, they should be defined. Since the GAMS notation almost mimics mathematical notation, this is an easy task. Notice the use of the ‘SUM(J, x(I,J))’, that is the GAMS operator to represent $\sum_{j=1}^n x_{ij}$.
- After the definition of equations, we name the model using the term ‘transport’. This model includes the ‘COST’, ‘SHIP(I)’ and ‘RECEIVE(J)’ constraints. In this statement, equation indices are not allowed. Notice that the names enclosed within slashes can be replaced by the reserved word ‘all’.
- Once the model has been defined, it must be solved. This is a linear programming problem, therefore the reserved word ‘lp’ is used. We also indicate that the objective function should be minimized.

An input GAMS file to solve this problem is

```
$title THE TRANSPORTATION PROBLEM

** First, indices are declared and defined.
** Index I is used to refer to the three origins.
** Index J is used to refer to the three destinations.
** Note how the symbol '*' is used to list sets members in a compact way.

SETS
  I index of shipping origins      /I1*I3/
  J index of shipping destinations /J1*J3/;

** Vectors of data (U(I) and V(J)) are defined as parameters
** Data are assigned to vector elements

PARAMETERS
  U(I)  the amount of good to be shipped from origin I
        /I1 2
         I2 3
         I3 4/
  V(J)  the amount of good to be received in destination J
        /J1 5
         J2 2
         J3 2/;

** The C(I,J) data matrix is defined as a table
** Data are assigned to matrix elements

TABLE C(I,J) cost of sending a unit from origin I to destination J
      J1   J2   J3
I1    1    2    3
I2    2    1    2
I3    3    2    1;

** The optimization variables are declared.
** First, the objective function variable (z) is declared.
** Next, the remaining variables with controlling indices are declared.
```

Part of the output file is

---- VAR X		the amount of product to be shipped from origin I to destination J		
	LOWER	LEVEL	UPPER	MARGINAL
I1.J1	.	2.000	+INF	.
I1.J2	.	.	+INF	2.000
I1.J3	.	.	+INF	4.000
I2.J1	.	1.000	+INF	.
I2.J2	.	2.000	+INF	.
I2.J3	.	.	+INF	2.000
I3.J1	.	2.000	+INF	.
I3.J2	.	.	+INF	EPS
I3.J3	.	2.000	+INF	.

```

---- VAR Z          -INF    14.000    +INF      .
   Z                objective function variable

```

In the above output file, the optimal values of the variables are listed in the column identified by 'LEVEL'. Therefore, the solution of the transportation problem is:

$$Z = 14, \quad \mathbf{X} = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 2 & 0 \\ 2 & 0 & 2 \end{pmatrix}.$$

11.2.2 The Production-Scheduling Problem 1

The production-scheduling problem was analyzed in detail in Section 1.3, page 6, and in Example 1.2, page 8. A particular case can be stated as:

Maximize

$$Z = \sum_{t=1}^n (a_t y_t - b_t x_t - c_t s_t), \quad (11.3)$$

subject to

$$\begin{aligned} s_{t-1} + x_t - s_t &= y_t, \quad \forall t = 1 \dots n, \\ s_t, x_t, y_t &\geq 0. \end{aligned} \quad (11.4)$$

where $a_t = b_t = c_t = 1, \forall t = 1 \dots n; \quad n = 4; \quad s_0 = 2$ and $\mathbf{y} = (2, 3, 6, 1)^T$.

Some significant considerations of the GAMS code for this example are:

- The use of the 'fx' suffix to fix an element of a variable array to a single value.
- In the objective function equation, the '\$' command is used to sum specific (not all) elements of an array. This feature is also used to define a group of equations in terms of a subset.

The input GAMS file for this problem is

```

$title PRODUCTION_SCHEDULING PROBLEM

** The index T is declared. It has 5 elements which are defined
** using the symbol '*'.

SET
  T   The month index /0*4/;

** Once the set T has been defined, the data vector Y(T) is
** declared and all its elements are assigned except
** the first one (element '0'), which, by default, is set to 0.

PARAMETER
  Y(T) demand in month T
    /1 2
      2 3

```