Introduction to GAMS: The Transport Problem

1. Introduction

The ‘transport problem’ is a classic linear programming (LP) problem where the objective is to minimise the costs of transporting commodities from their place and/or places of manufacture to where they are to be sold. There are many such problems in the real world, e.g., the animal feed industry uses the same method for least cost ration formulation. Related problems using LP, and related, techniques include scheduling of bus/coach services, airline schedules and the allocation of teaching rooms and staff timetables. Such applications of LP techniques can become very complicated and very large, but they are all constrained optimisation problems.

Constrained optimisation problems are central to modern economics. Consumers seek to maximise utility given their preferences and subject to constraints imposed by incomes and prices. Producers seek to maximise profits subject to technology and constraints imposed by factor supplies and prices. There is a long history of LP being applied to economics problems.[1]

The advantage of an LP problem for this course is three-fold. First, all LP and CGE problems are optimisation problems and hence LP problems contain all the features in CGE problems; second, a simple LP problem is a simpler coding exercise; and third, the GAMS tutorial uses a LP problem to demonstrate all the key elements in a GAMS programme.

This course uses the GAMS transport tutorial, which is a standard introduction to GAMS. It is a simple linear programming problem where the objective is to determine the least cost means to transfer commodities from two production plants to three markets. The example covers all the major aspects of a standard GAMS programme – the declaration and assignment of sets, data (parameters, tables and scalars) and variables, the declaration and definition of equations, model statements and display commands. These exercises include various extensions to the standard GAMS tutorial, designed to help develop GAMS programming skills.

These notes are a brief introduction to the greater detail provided in the Tutorial chapter, by Rosenthal, which is included in the GAMS manual. Copies of the GAMS Tutorial and the

[1] The classic economics text is Dorfman, R., Samuelson, P.A. and Solow, R.M., (**1958**), *Linear programming and Economic Analysis*’, New York: Dover.

GAMS Manual are available as *.pdf files from the GAMS `Help` menu. (A copy of the GAMS tutorial is included in the user library.)

2. Basic Linear Programming Problems

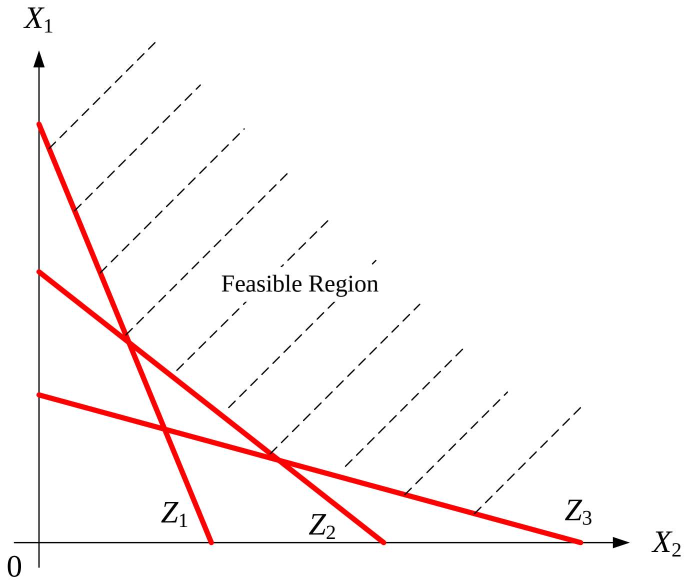
The basic principles behind LP problems can be illustrated by two simple figures.[1]

Consider, first, the simple diet problem. The objective is to minimise the total cost (*C*) of achieving a minimum defined level of consumption of three nutrients (Z_1,Z_2,Z_3) each of which is supplied by two different food commodity (X_1,X_2) in different ratios. This can be expressed as

$$\text{Min } C = p_1.X_1 + p_2.X_2$$
$$\text{sto}$$
$$a_{11}.X_1 + a_{12}.X_2 \geq Z_1$$
$$a_{21}.X_1 + a_{22}.X_2 \geq Z_2$$
$$a_{31}.X_1 + a_{32}.X_2 \geq Z_3$$

where a_{ij} is a coefficient that quantifies the amount of nutrient *i* provided by commodity *j*. This can be illustrated by Figure 1, where the sloping (linear) lines illustrate the nutrient constraints, i.e., the minimum quantities of the commodities needed to satisfy the nutrient constraints. Hence the feasible region, i.e., the area that satisfies the nutrient requirement is above and to the right of the boundary formed by the constraints.

Figure 1 The Diet Problem



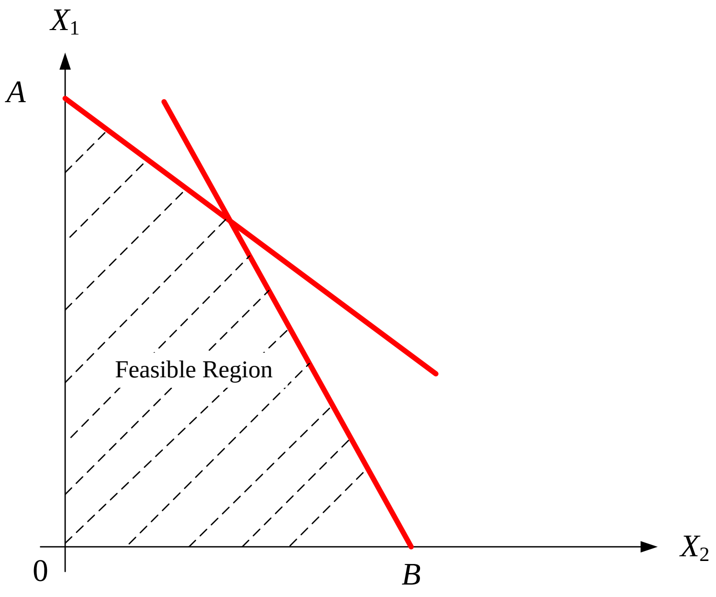
[1] These illustrations draw on Dorfman *et al.*, (1958), Fig 2-2 (p 24) and Fig 2-6 (p 35).

While it is tempting to think of this as akin to an indifference curve, Dorfman *et al*., (1958) demonstrate how this would be inappropriate. However, it could be considered akin to an isoquant that defines the minimum cost of ‘sustaining’ life given two inputs with different technologies that produce the necessities of life, i.e., a purely technical relationship.

The second problem relates to the theory of comparative advantage, where there are two countries that can both produce two commodities (X_1,X_2) with different technologies. The objective is to define the maximum amount of one commodity subject to prescribed amount of the other commodity. This can be illustrated by Figure 2.

In Figure 2, the intercept ***A*** identifies the maximum amount of X_1 that can be produced if both countries only produce X_1, while the intercept ***B*** identifies the maximum amount of X_2 that can be produced if both countries only produce X_2. As the minimum amount of X_2 is increased so the maximum amount of X_1 that can be produced declines, while similarly as the minimum amount of X_1 is increased so the maximum amount of X_2 that can be produced declines.

Figure 2 Efficient Frontier

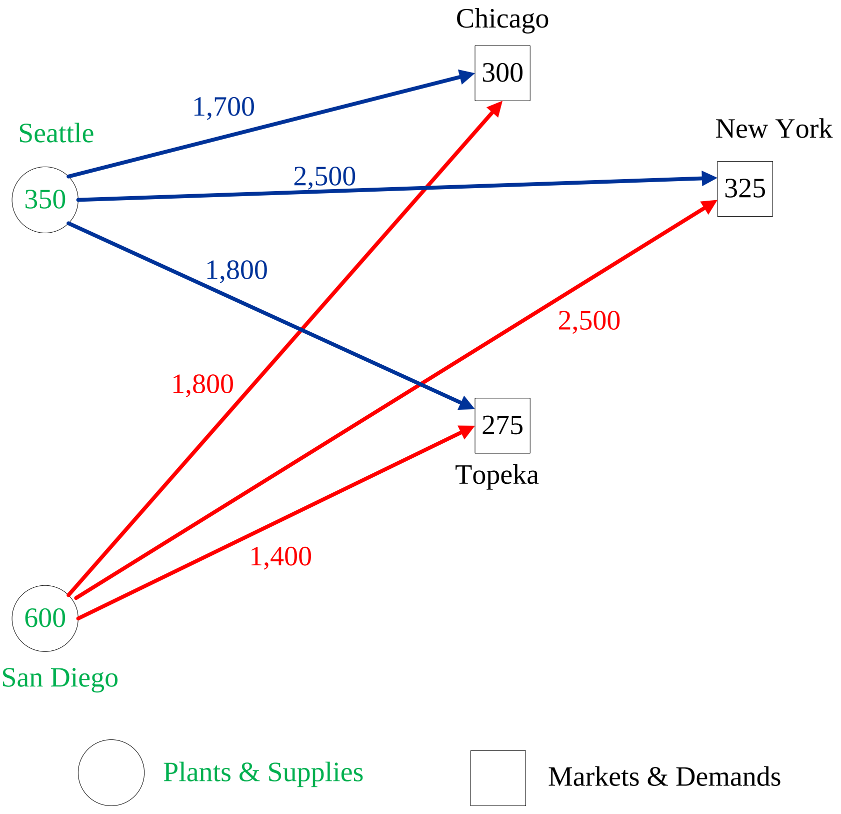


These two illustrations should be enough to demonstrate that the societal programming problem that underlies general equilibrium economics shares certain commonalities with LP problems; namely both are constrained optimisation problems. The biggest difference with CGE models is that the constraints are non-linear.

3. The GAMS Transport Problem

The GAMS transport problem determines the least cost means to transfer commodities from 2 production plants to three markets. In this simple example a food commodity is produced by two canning plants owned by a single firm and is then sold at three different markets. Each canning plant has a fixed capacity while each market has a given level of (expected) demand. This can be illustrated in a simple figure (Figure 3), where the plants and capacities are in GREEN and markets and demands are in BLACK. The distances from Seattle to the three markets are BLUE and distances from San Diego are in RED.

Figure 3 A Simple Transport Problem



The firm knows the distances between each canning plant and each market and the transport costs, per mile[1], associated with transporting each case of the commodity from the plants to the markets. The firm wants to know how much of the commodity should be supplied by each plant to each market to minimise transport costs

[1] An imperial mile is approximately 1.6 km.

This is a mathematical programming problem, so it is appropriate to express the problem algebraically. Since GAMS is a set based programming language we will make extensive use of sets. The benefits of set based programming languages will become evident later, although to start with they seem to make the problem more difficult.

Standard algebraic presentation

Indices/Sets

i = plants
j = markets

The set *i* identifies the plants while the set *j* identifies the markets.

Available Data

a_i = supply of commodity at plant i (in cases)
b_j = demand for commodity at market j (in cases)
d_{ij} = distances between plant i and market j (\$/mile//case)
f = freight cost (\$/case/1,000 miles)

The programme will need to know the capacity (*a*) of each plant *i* (in cases), the demand at each market j (in cases), the distances between each plant and market (*d)* and the freight costs (*f*). The combination of the freight cost and the distances

Decision Variables

X_{ij} = amount of commodity to ship from plant i to market j (cases)

The firm then needs to determine the amount of the commodity to be shipped from each plant to each market.

Constraints

Supply limit at plant *i*: $\sum_j X_{ij} \le a_i \qquad \forall i$

Demand at market *j*: $\sum_i X_{ij} \ge b_j \qquad \forall j$

$$X_{ij} \ge 0 \qquad \forall i, j$$

And then needs to formalise the constraints: the amount sent form each plant to the different markets cannot exceed the plant's capacity; the amount sent to each market must at least satisfy demand at each market; and that negative amounts cannot be sent from any plant.

Objective Function

Minimise $\sum_i \sum_j c_{ij} X_{ij}$

<u>Example Data</u>

The required data can be presented as a matrix, as illustrated in Table 1. The distances between plants and markets are in thousands of miles, e.g., Seattle to New York is 2,500 miles. The maximum supplies from each plant are the cases, e.g., 600 from San Diego, while the minimum demands at each market are also in cases, e.g., 300 at Chicago.

Table 1 Transport Problem Data

		Markets		
Plants	**New York**	**Chicago**	**Topeka**	**Supplies**
		(Distances ‘000 m)		
Seattle	2.5	1.7	1.8	350
San Diego	2.5	1.8	1.4	600
Demands	325	300	275	

The only other data element required is the freight cost, *f*.

4. Structure of a GAMS Model

All GAMS programmes have a common structure, illustrated in Table 2. This is the ordering used in the transport tutorial and is very similar to the order used in the CGE models in this course. The order used other models may differ, but a basic principle cannot be violated: nothing can be used in a model before it has been declared, i.e., the code must declare the existence of a set/parameter/variable/equation before it can be assigned a value.

Table 2 GAMS Programme Structure

SETS	Declaration Assignment of Members
Data (PARAMETERS, TABLES, SCALARS)	Declaration Assignment of Values
VARIABLES	Declaration Assignment of Type (optional) Assignment of bounds/initial values
EQUATIONS	Declaration Definition
MODEL and SOLVE statements (optional) DISPLAY statements	

A copy of the code for the GAMS tutorials transport problem is given below, with the text that gives the references removed. This version differs marginally from that in the GAMS model library in that the conventions used in this course have been adopted. Namely, all sets and parameters are in lower case and all variables are in upper case; other than these changes the codes are identical. Three features justify highlighting

1. The font used – COURIER NEW – is a fixed pitch font, i.e., every character occupies the same amount of horizontal space.
2. In tables the values must be aligned below the column labels and in line with the row labels, as in Table *d*(*i*,*j*).
3. The use of white space and alignments to make it easy to read the code.

It is important to spend some time NOW going through the GAMS code and relating each line of code to each part of the standard algebraic presentation. The similarities should soon be apparent, as should be how algebraic expressions are presented the code to accommodate the need to work as a simple text file.

5. The Transport Problem in GAMS Code

```
$TITLE  A TRANSPORTATION PROBLEM (TRNSPORT,SEQ=1)
$OFFUPPER
* This problem finds a least cost shipping schedule that meets
* requirements at markets and supplies at factories


  SETS
       i   canning plants   / SEATTLE, SAN-DIEGO /
       j   markets          / NEW-YORK, CHICAGO, TOPEKA / ;

  PARAMETERS

       a(i)  capacity of plant i in cases
         /    SEATTLE     350
              SAN-DIEGO   600  /

       b(j)  demand at market j in cases
         /    NEW-YORK    325
              CHICAGO     300
              TOPEKA      275  / ;

  TABLE d(i,j)  distance in thousands of miles
                    NEW-YORK       CHICAGO      TOPEKA
      SEATTLE          2.5           1.7          1.8
      SAN-DIEGO        2.5           1.8          1.4   ;

  SCALAR f  freight in dollars per case per thousand miles  /90/ ;

  PARAMETER c(i,j)  transport cost in thousands of dollars per case ;

            c(i,j) = f * d(i,j) / 1000 ;

  VARIABLES
       X(i,j)  shipment quantities in cases
       Z       total transportation costs in thousands of dollars ;

  POSITIVE VARIABLE X ;

  EQUATIONS
       COST        define objective function
       SUPPLY(i)   observe supply limit at plant i
       DEMAND(j)   satisfy demand at market j ;

  COST..        Z  =E=  SUM((i,j), c(i,j)*X(i,j)) ;

  SUPPLY(i)..   SUM(j, X(i,j))  =L=  a(i) ;

  DEMAND(j)..   SUM(i, X(i,j))  =G=  b(j) ;

  MODEL TRANSPORT /ALL/ ;

  SOLVE TRANSPORT USING LP MINIMIZING Z ;

  DISPLAY X.L, X.M ;
```