The Buyer-Seller Problem

Takemyprocs

September 8, 2021

Abstract

At first, we deal with a simplified case of the buyer-seller problem which we refer as The Seller Problem. We provide a simple model along with an illustrative application. Then, we depart from this basic framework to deal with more realistic scenarios where the decision maker jointly determines buying and selling choices. Finally, we discuss some issues on the buying and selling process that are not related to the choice problem.

1 The Seller Problem: a basic framework

Suppose a seller is endowed with a fixed amount of m inputs, namely: $s_1, s_2, ..., s_m$. With them the seller can craft n products, namely: $x_1, x_2, ..., x_n$. Not all inputs may be required for crafting a particular product, so we define A as the technical matrix that captures the input requirements for each output:

$$A = \begin{array}{c c} & \begin{array}{cccc} x_1 & x_2 & \dots & x_n \end{array} \\ \begin{array}{c} s_1 \\ s_2 \\ \vdots \\ s_m \end{array} & \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \end{array}$$

For instance, in order to craft 1 unit of product x_1 we need to use a_{11} units of input s_1, a_{21} units of input s_2, and so on. We assume technical coefficients a_{ij} are fixed. Recall that since not all inputs are required for crafting a particular product some of the elements in this matrix may be 0. Finally, we assume that only outputs can be sold freely at fixed prices: $p_1, p_2, ..., p_n$.

Given those conditions, we can write the seller problem as follows:

$$\begin{aligned} \max_{x_1, x_2, ..., x_n} \quad & p_1 x_1 + p_2 x_2 + ... + p_n x_n \\ \text{s.t.} \quad & a_{11} x_1 + a_{12} x_2 + ... + a_{1n} x_n \leq s_1 \\ & a_{21} x_1 + a_{22} x_2 + ... + a_{2n} x_n \leq s_2 \\ & \vdots \\ & a_{m1} x_1 + a_{m2} x_2 + ... + a_{mn} x_n \leq s_m \\ & x_1, x_2, ..., x_n \geq 0 \end{aligned} \tag{1}$$

Intuitively, the seller problem is to choose production quantities $x_1, x_2, ..., x_n$ in order to maximize earnings subject to input availability constraints. This resembles the linear programming canonical form. Therefore, we can use almost any mathematical software (e.g. Matlab, R or Python) to solve this problem. In order to apply this model we need information on technical coefficients, input endowments and prices. We provide an example in the next section.

2 An example

Suppose a seller is endowed with 98 units of input 1, 147 units of input 2, 182 units of input 3, 157 units of input 4 and 114 units of input 5. By using those inputs the seller can craft up to 6 different products, say $x_1, ..., x_6$. The input requirements for crafting each product are summarized in the technical matrix shown below:

$$A = \begin{array}{c} \\ s_1 \\ s_2 \\ s_3 \\ s_4 \\ s_5 \end{array} \begin{array}{c} \begin{array}{cccccc} x_1 & x_2 & x_3 & x_4 & x_5 & x_6 \end{array} \\ \begin{pmatrix} 5 & 0 & 0 & 0 & 0 & 4 \\ 0 & 5 & 0 & 0 & 0 & 4 \\ 0 & 0 & 5 & 0 & 3 & 4 \\ 0 & 0 & 0 & 5 & 3 & 4 \\ 0 & 0 & 0 & 0 & 1 & 3 \end{pmatrix} \end{array}$$

Finally, market prices for products 1 to 6 are 100.2, 80, 110.96, 120.11, 115.90 and 314 respectively. According to the basic framework described in the previous section we formulate the seller problem for this particular application as follows:

$$\begin{aligned} \max_{x_1, x_2, ..., x_n} \quad & 100.2x_1 + 80x_2 + 110.96x_3 + 120.11x_4 + 115.90x_5 + 314x_6 \\ \text{s.t.} \quad & 5x_1 + 4x_6 \leq 98 \\ & 5x_2 + 4x_6 \leq 147 \\ & 5x_3 + 3x_5 + 4x_6 \leq 182 \\ & 5x_4 + 3x_5 + 4x_6 \leq 157 \\ & x_5 + 3x_6 \leq 114 \\ & x_1, x_2, x_3, x_4, x_5, x_6 \geq 0 \end{aligned} \tag{2}$$

We use the package `lpSolve` (linear programming) in the R environment to solve this problem. As shown below, our code[1] provides us with optimal quantities for every product.

```
     product quantity
[1,] "x1"    "17"
[2,] "x2"    "27"
[3,] "x3"    "34"
[4,] "x4"    "29"
[5,] "x5"    "0"
[6,] "x6"    "3"
```

[1]See the seller.R script in GitHub.

Notice that quantity for product x_5 is null, this is an example of *corner solution* which is very common in the linear programming framework. This solution implies that, at faced prices, endowments and current technical requirements, is not profitable to craft product x_5 at all. It is worth noting that although we use a linear programming approach what we use indeed is an *integer programming* approach without loss of generality. The difference lies in the solution's domain, while linear programming admits solutions in the real domain, integer programming offers solutions within the integer domain which is a more realistic approach (Who would sugest you to produce π units of any product?).

3 Some extentions

Insofar we have described a basic framework of the seller problem. Under this model we assumed that inputs cannot be sold in the market as we do with products. Also we have not considered the possibility of purchasing inputs in the market and then crafting products with them. In this section we make some departures from the basic framework to land into a more realistic trading scenario.

3.1 What if we sell inputs as well?

We extend the previous model by including the possibility to sell inputs directly through the market and not only use them for crafting purposes. This is easily done by attaching an identity matrix to our technical matrix, which is equivalent to add m extra products that uses 1 unit of each input. Put in simple, to "craft" and sell 1 unit of input s_i we require 1 unit of input i and 0 units of the remain inputs. The new technical matrix looks as follows:

$$A = \begin{array}{c} \\ s_1 \\ s_2 \\ \vdots \\ s_m \end{array} \begin{array}{c} \begin{array}{cccccccc} x_1 & x_2 & ... & x_n & \bar{s}_1 & \bar{s}_2 & ... & \bar{s}_m \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} & 1 & 0 & ... & 0 \\ a_{21} & a_{22} & ... & a_{2n} & 0 & 1 & ... & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & ... & a_{mn} & 0 & 0 & ... & 1 \end{pmatrix} \end{array}$$

At this point we must distinguish between the quantity of input i that we choose to sell in the market, which we denote as $\bar{s}_i$, and the available stock of input i, which we still refer to as s_i. Finally, assume that along with product prices $p_1, p_2, ..., p_n$ we are facing the following prices for inputs: $r_1, r_2, ..., r_m$. Given those conditions, we reformulate our problem as follows:

$$\begin{aligned} \max_{x_1,x_2,...,x_n,\bar{s}_1,\bar{s}_2,...,\bar{s}_m} \quad & p_1x_1 + p_2x_2 + ... + p_nx_n + r_1\bar{s}_1 + r_2\bar{s}_2 + ... + r_m\bar{s}_m \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n + \bar{s}_1 \leq s_1 \\ & a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n + \bar{s}_2 \leq s_2 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n + \bar{s}_m \leq s_m \\ & x_1, x_2, ..., x_n, \bar{s}_1, \bar{s}_2, ..., \bar{s}_m \geq 0 \end{aligned} \tag{3}$$

Notice that the objective function (or revenue function if you prefer) is now composed by earnings coming from both product and input sales. Also, inputs are now allocated in either product crafting or direct sale. This last condition is captured along the m declared constraints [2].

3.2 What if we buy inputs?

Now suppose we came back to the situation where we cannot sell inputs but we are able to buy them in order to craft and sell products. Of course, we cannot buy inputs indefinitively as we may have a budget constraint so we introduce capital K as a new parameter in the model. For the sake of simplicity, we understand capital as available cash to buy items (like inputs in our model). Technology stays the same so we adopt the technical matrix described in section 1:

$$A = \begin{array}{c} \\ \bar{s}_1 \\ \bar{s}_2 \\ \vdots \\ \bar{s}_m \end{array} \begin{array}{c} \begin{array}{cccc} x_1 & x_2 & ... & x_n \end{array} \\ \begin{pmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & ... & a_{mn} \end{pmatrix} \end{array}$$

However, it is worth noting that instead of having exogenous input quantities s_i we simultaneously determine product and input quantities. Therefore, we use the $\hat{s}_i$ notation to indicate that input availability is endogenous and determined within the model. Finally, we assume that along with product prices $p_1, p_2, ..., p_n$ we are facing the following prices for inputs: $r_1, r_2, ..., r_m$. Given those conditions, we reformulate our problem as follows:

$$\begin{aligned} \max_{x_1,x_2,...,x_n,\bar{s}_1,\bar{s}_2,...,\bar{s}_m} \quad & p_1x_1 + p_2x_2 + ... + p_nx_n - r_1\bar{s}_1 - r_2\bar{s}_2 - ... - r_m\bar{s}_m \\ \text{s.t.} \quad & a_{11}x_1 + a_{12}x_2 + ... + a_{1n}x_n - \bar{s}_1 \leq 0 \\ & a_{21}x_1 + a_{22}x_2 + ... + a_{2n}x_n - \bar{s}_2 \leq 0 \\ & \vdots \\ & a_{m1}x_1 + a_{m2}x_2 + ... + a_{mn}x_n - \bar{s}_m \leq 0 \\ & r_1\bar{s}_1 + r_2\bar{s}_2 + ... + r_m\bar{s}_m \leq K \\ & x_1, x_2, ..., x_n, \bar{s}_1, \bar{s}_2, ..., \bar{s}_m \geq 0 \end{aligned} \tag{4}$$

In this case the objective function is composed by overall earnings due to product sales and the overall spending due to input purchases. Regarding constraints, we see that input availability depends on the input purchase decision while this in turn is limited by the capital budget [3].

[2]Take a look at our script sell_inputs.R in GitHub for an application of this model

[3]See the script seller_buyer.R in GitHub for an application of this model