# Boolean optimization algorithm

So, this are the data,

- There are multiple sellers (S).
- Each of the seller has (J) number of products in his store.
- The sellers have different prices on their products i.e. seller (i) sells product (j) in ($p_{ij}$).
- Each seller takes fixed/constant commission for buying at his store (C), the commission is fixed regardless of how many products are bought from his store.

If a client buys number of products (A) only from $S_i$, the function will be

$$y_i = \overline{c_i} + \sum_J a_{ij} p_{ij}$$

$$\overline{c_i} = \begin{cases} 0, & \sum_J a_{ij} p_{ij} = 0 \\ c_i, & \sum_J a_{ij} p_{ij} > 0 \end{cases}$$

The actual situation is, the client buys from multiple sellers and the equation looks like that

$$\boldsymbol{Y = \sum_i y_i = \sum_I \overline{c_i} + \sum_I \sum_J a_{ij} p_{ij}}$$

$$Min(Y) = ?$$

There are 2 more arguments:

1. In the elaborated algorithm the seller can give discounts on the commission if the buyer is buying over a certain amount of money.

$$\overline{c_i} = \begin{cases} 0, & \sum_J a_{ij}p_{ij} = 0 \\ c_{i1}, & 0 \leq \sum_J a_{ij}p_{ij} < K \\ c_{i2}, & K \leq \sum_J a_{ij}p_{ij} < L \\ c_{i3}, & L \leq \sum_J a_{ij}p_{ij} < M \\ & \dots \end{cases}$$

2. The algorithm takes in to calculations only specific sellers (the system already knows how to find this sellers).

The module needs to find the minimum of Y, **Min (Y)=?**