Cali Coffee Company Production Optimisation

Michael Nef

July 5, 2021

Contents

4 Code 7

1 Client information

1.1 Available products

The Cali Coffee Company (CCC) are an international supplier of ground coffee mixes. In their production process they have 4 component beans:

1. Abundo
2. Colmado
3. Maximo
4. Saboro

CCC sells 3 blends:

1. Hotel (luxury) blend
2. Restaurant blend
3. Market (retail) blend

CCC has 4 bean suppliers:

1. Colombian Abundo
2. Peruvian Calmado
3. Brazilian Maximo
4. Chilean Saboro

1.2 Coffee Blend information

Each blend is comprised of a mix of different components. The relative proportion of these components is indicated for each blend as a percentage (%).

component	Hotel	Restaurant	Market	Cost/lbs($usd)	availability (lbs)
Abundo	20%	35%	10%	0.6	40000
Colmado	40%	15%	35%	0.8	25000
Maximo	15%	20%	40%	0.55	20000
Saboro	25%	30%	15%	0.7	45000
Wholesale price/lb ($usd)	2.25	2.5	2.4		

1.3 Additional Production Constraints

- The processing plant can handle no more than 100000 pounds per week.
- There are ongoing contracts for production: Minimum 10000, 25000, 30000 pounds of Hotel, Restaurant and Market blend must be produced to satisfy these contracts.

2 Results and Analysis

2.1 Optimal Allocation

Based on the current constraints and prices, the optimal production schedule is as follows:

Hotel	Restaurant	Market
20000	25000	30000

This production schedule generates $129350.0 (usd).

2.2 Binding Constraints:

Analysis has identified 3 limiting factors which are preventing rises in profits:

1. 20000lbs Maximo bean availability.
2. Existing contractual obligation to produce 25000lbs of Hotel blend.
3. Existing contractual obligation to produce 30000lbs of Market blend.

2.3 Sensitivity Analysis

2.3.1 Maximo bean availability

For every extra pound of Maximo beans made available we can expect to generate an additional $10 in profit. Similarly, if the availability of Maximo beans decreases, we expect to sustain losses of $10 per pound decrease.

2.3.2 Contractual obligation to supply Market blend

For every pound reduction in the market supply contract we can expect to generate an additional $2 in profit. Similarly, if our contractual obligations increase, we expect to sustain losses of $2 per pound increase.

2.3.3 Other

Changes to any other constraint will result in no difference in profits.

3 Linear Programming Formalism

3.1 Intro / Objective

Objective is to maximise profit by selecting production amounts for Hotel, Restaurant and Market blends: Profit = Revenue − Costs

3.2 Production schedule

Let our production schedule be denoted $\mathbf{x}$. $\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} \text{Hotel (lbs)} \\ \text{restaurant (lbs)} \\ \text{market (lbs)} \end{bmatrix}$

3.3 Coffee Blend Matrix

To get the vector of required beans for a given blend we define the following Coffee Blend Matrix A: $A = \begin{bmatrix} 0.2 & 0.35 & 0.1 \\ 0.4 & 0.15 & 0.35 \\ 0.15 & 0.2 & 0.4 \\ 0.25 & 0.3 & 0.15 \end{bmatrix}$

3.4 Bean Cost Vector

We also define a Bean Cost Vector $\mathbf{c}$, each element in this vector is the cost/lbs (\$usd) for a given component. $\mathbf{c} = \begin{bmatrix} 0.6 \\ 0.8 \\ 0.55 \\ 0.7 \end{bmatrix}$

3.5 Costs

Provided a production schedule, coffee blend matrix and bean cost vector, the total cost of this amount of production is given by: Costs $= \mathbf{c}^\top A\mathbf{x}$

3.6 Coffee Price Vector

We define a price vector, $\mathbf{p}$, each element is the price/lb CCC generates for selling a particular coffee blend: $\mathbf{p} = \begin{bmatrix} 2.25 \\ 2.5 \\ 2.4 \end{bmatrix}$

3.7 Revenue

Provided a production schedule and Coffee Price Vector, the total revenue of this amount of production is given by: Revenue = $\mathbf{p}^\top \mathbf{x}$

3.8 Objective

$\max_{\mathbf{x}}(\mathbf{p}^\top \mathbf{x} - \mathbf{c}^\top A\mathbf{x})$

3.9 Constraints

3.9.1 Ongoing coffee contracts

$$\mathbf{x} \succeq \begin{bmatrix} 10000 \\ 25000 \\ 30000 \end{bmatrix}$$

3.9.2 Supplier/bean availability

$$A\mathbf{x} \preceq \begin{bmatrix} 40000 \\ 25000 \\ 20000 \\ 45000 \end{bmatrix}$$

3.9.3 Processing Plant Capacity

$x_1 + x_2 + x_3 \leq 100000$

3.9.4 Non-negative Production

$\mathbf{x} \succeq \mathbf{0}$

4 Code

```
from scipy import optimize
import scipy
import numpy as np

A = np.matrix([[0.2, 0.35, 0.1],
               [0.4, 0.15, 0.35],
               [0.15, 0.2, 0.4],
               [0.25, 0.3, 0.15]])

c = np.array([0.6, 0.8, 0.55, 0.7])

p = np.array([2.25, 2.5, 2.4])

objective = -(p.T - c.T @ A)

A_ub = np.matrix([[0.2, 0.35, 0.1], # Abundo usage
                  [0.4, 0.15, 0.35], # Colmado usage
                  [0.15, 0.2, 0.4], # Maximo usage
                  [0.25, 0.3, 0.15], # Saboro usage
                  [1, 1 , 1], # Total production
                  [-1, 0, 0], # Hotel
                  [0, -1, 0], # Restaurant
                  [0, 0, -1]]) # Market

b_ub = np.array([40000, # Abundo availability
                 25000, # Colmado availability
                 20000, # Maximo availability
                 45000, # Saboro availability
                 100000, # Maximum production
                 -10000, # Min Hotel
                 -25000, # Min Restaurant
                 -30000]) # Min Market

primal_result = optimize.linprog(objective, A_ub, b_ub)
print('RESULTS OF PRIMAL LP')
print(f'Optimal assignment: {list(map(round, primal_result.x))}')
print(f'Maximum value: {round(primal_result.fun)}')
print(f'Slack values: {list(map(round, primal_result.slack))}')
```

```
print()

dual_objective = b_ub.T
dual_A_ub = -A_ub.T
dual_b_ub = objective.T

dual_result = optimize.linprog(dual_objective, dual_A_ub, dual_b_ub)
print('RESULTS OF DUAL LP')
print(f'Optimal assignment: {list(map(round, dual_result.x))}')
print(f'Maximum value: {round(dual_result.fun)}')
print(f'Slack values: {list(map(round, dual_result.slack))}')
```