Week 1: Modellering og Løsning af Optimeringsproblemer

Thomas

January 29, 2026

LP-1

We are given the respective sales prices of three products A, B and C:

- Product A: 8 DKK
- Product B: 16 DKK
- Product C: 12 DKK

We have the restrictions on material usage:

$$R_1 \leq 150, \quad R_2 \leq 170, \quad R_3 \leq 120, \quad R_4 \leq 140 \tag{1}$$

Naturally, we also have the non-negativity constraints:

$$R_1, R_2, R_3, R_4 \geq 0 \tag{2}$$

The sales values function can be expressed as:

$$\text{Sales Value} = 8A + 16B + 12C \tag{3}$$

Let P be the price vector for each of the materials R_i:

$$p = \begin{bmatrix} 0.3 \\ 0.7 \\ 0.4 \\ 0.8 \end{bmatrix} \tag{4}$$

The material usage constraints can be expressed in matrix form as:

$$M = \begin{bmatrix} 0 & 3 & 2 \\ 2 & 5 & 4 \\ 3 & 1 & 2 \\ 2 & 4 & 3 \end{bmatrix} \tag{5}$$

Where the rows correspond to the materials R_1, R_2, R_3, R_4 and the columns correspond to products A, B and C respectively.

Let x be the vector of products:

$$x = \begin{bmatrix} A \\ B \\ C \end{bmatrix} \tag{6}$$

Then we can express spending on materials as:

$$\text{Spending} = p^T M x \tag{7}$$

and thus profit can be expressed as:

$$\text{Profit} = \text{Sales Value} - \text{Spending} = 8A + 12B + 16C - p^T M x \tag{8}$$

We can now formulate the optimization problem as as a three decision variable linear programming problem:

$$\text{Maximize } z = 8A + 12B + 16C - p^T M x \tag{9}$$

Subject to:

$$\begin{aligned} 0A + 3B + 2C &\leq 150 \\ 2A + 5B + 4C &\leq 170 \\ 3A + 1B + 2C &\leq 120 \\ 2A + 4B + 3C &\leq 140 \\ A, B, C &\geq 0 \end{aligned} \tag{10}$$

We get an optimal solution of:

$$A = 10, \quad B = 30, \quad C = 0 \tag{11}$$

With a maximum profit of:

$$\text{Profit} = 278 \text{ DKK} \tag{12}$$

Note we can also reformulate this problem to a 7 decision variable linear programming problem by price variables for each product. Then we condition on the prices being lower than what we can aquire the materials for. This has the advantage the out solution will also provide the expenses for each material.

Python Code

```
# noinspection PyUnresolvedReferences
import pulp as PLP
"""
3-decision variable version.
"""
A = PLP.LpVariable(name="A",lowBound=0)
B = PLP.LpVariable(name="B",lowBound=0)
C = PLP.LpVariable(name="C",lowBound=0)

model = PLP.LpProblem(name="LP1", sense=PLP.LpMaximize)

# Define objective function
sales_price = 8*A + 16*B + 12*C

cost1 = 0.3*(3*B + 2*C)
cost2 = 0.7*(2*A + 5*B + 4*C)
cost3 = 0.4*(3*A + 1*B + 2*C)
cost4 = 0.8*(2*A + 4*B + 3*C)

total_cost = cost1 + cost2 + cost3 + cost4
profit = sales_price - total_cost
model += profit, "Objective"

```

```python
# Add constraints
model += 3*B + 2*C <= 150, "R1"
model += 2*A + 5*B + 4*C <= 170, "R2"
model += 3*A + 1*B + 2*C <= 120, "R3"
model += 2*A + 4*B + 3*C <= 140, "R4"


def print_solution(Model):
    # Loesning af modellen vha. PuLP's valg af Solver
    Model.solve()
    # Print af loesningens status
    print("Status:", PLP.LpStatus[Model.status])
    # Print af hver variabel med navn og loesningsvaerdi
    for v in Model.variables():
        print(v.name, "=", v.varValue)
    # Print af den optimale objektfunktionsvaerdi
    print("Obj. = ", PLP.value(Model.objective))
print_solution(model)

"""
7 - decision variable version. Add costs1 to costs4 for resources 1 to 4.
"""
model = PLP.LpProblem(name="LP1", sense=PLP.LpMaximize)

A = PLP.LpVariable(name="A",lowBound=0)
B = PLP.LpVariable(name="B",lowBound=0)
C = PLP.LpVariable(name="C",lowBound=0)
costs1 = PLP.LpVariable(name="costs1", lowBound=0)
costs2 = PLP.LpVariable(name="costs2", lowBound=0)
costs3 = PLP.LpVariable(name="costs3", lowBound=0)
costs4 = PLP.LpVariable(name="costs4", lowBound=0)

model += 8*A + 16*B + 12*C - (costs1 + costs2 + costs3 + costs4), "Objective"
# Add explicit cost calculations
model += costs1 == 0.3*(3*B + 2*C), "Cost1"
model += costs2 == 0.7*(2*A + 5*B + 4*C) , "Cost2"
model += costs3 == 0.4*(3*A + 1*B + 2*C) , "Cost3"
model += costs4 == 0.8*(2*A + 4*B + 3*C) , "Cost4"
# Add cost constraints instead of resource constraints
model += costs1 <= 150*0.3, "R1"
model += costs2 <= 170*0.7, "R2"
model += costs3 <= 120*0.4, "R3"
model += costs4 <= 140*0.8, "R4"

print_solution(model)
```