# Week 2: Modellering og Løsning af Optimeringsproblemer

Thomas

February 3, 2026

## LP-2

A company produces two products, a normal version $N$ and a luxury version $L$. We wish to maximize the profit from the production given some constraints. The profit on $N$ is 70 and on $L$ it is 100. Thus we can define the objective function as

$$Profit = 70N + 100L.$$

The product needs construction. Each department, wood, plastic and assemply, has 30, 10 and 20 people available respectively. Each person can work 30 hours per week. The assemply time for each product given a material category can be represented in matrix form as:

$$A = \begin{bmatrix} 2.25 & 2.5 \\ 1 & 0.5 \\ 1 & 2 \end{bmatrix}$$

where the rows represent wood, plastic and assemply respectively and the columns represent product $N$ and $L$. Furthermore, we can a constraint from the marketing department that the number of luxury products must be between 1/3 and 2/3 of the total production. We can augment the assemply matrix $A$ to get a total constraint matrix $C$ as:

$$C = \begin{bmatrix} 2.25 & 2.5 \\ 1 & 0.5 \\ 1 & 2 \\ \frac{1}{3} & -\frac{2}{3} \\ -\frac{1}{3} & \frac{2}{3} \end{bmatrix}$$

Thus we have the following constraints:

$$\begin{aligned} \mathcal{C}: \quad & 2.25N + 2.5L \leq 30 \cdot 30 \quad \text{(wood)} \\ & 1N + 0.5L \leq 10 \cdot 30 \quad \text{(plastic)} \\ & 1N + 2L \leq 20 \cdot 30 \quad \text{(assemply)} \\ & L \geq \frac{1}{3}(N + L) \iff \frac{N}{3} - \frac{2L}{3} \leq 0 \quad \text{(marketing lower bound)} \\ & L \leq \frac{2}{3}(N + L) \iff -\frac{N}{3} + \frac{2L}{3} \leq 0 \quad \text{(marketing upper bound)} \\ & N, L \geq 0 \quad \text{(non-negativity)} \end{aligned}$$

Thus we can state the model as:

$$\begin{aligned} \text{Maximize} \quad & Profit = 70N + 100L \\ \text{subject to} \quad & \mathcal{C} \end{aligned}$$

This model is now ready to be solved using an pythons PuLP library.

### Python Code for LP-2

```python
import pulp as PLP
from CustomFunctions import construct_constraints

# Maximization problem
model = PLP.LpProblem(name="LP2", sense=PLP.LpMaximize)

variable_names = ["N", "L"]
n_variables = len(variable_names)
objective_coefficients = [70, 100]
lower_bounds = [0, 0]

# Define decision variables using dictionaries
x = {
    name: PLP.LpVariable(name=name, lowBound=lb)
    for name, lb in zip(variable_names, lower_bounds)
}
# Define objective function coefficients
model += PLP.lpSum(objective_coefficients[i] * x[variable_names[i]] for i in
                   range(n_variables)), "Objective"

# Coefficient matrix for constraints
coef_matrix = [[2.25, 2.5],
               [1.0, 0.5],
                [1., 2.],
               [1/3, -2/3],
               [-1/3, 2/3]]
# all are less than or equal to constraints
constraint_types = ["<="] * len(coef_matrix)
# Right-hand side values for constraints
rhs = [30*30, 10*30, 20*30, 0, 0]
# Construct constraints
construct_constraints(model,
                      x,
                      variable_names,
                      coef_matrix,
                      rhs,
                      constraint_types)

from CustomFunctions import print_solution
print_solution(model)
```

## 12.1 Food manufacture 1