# Homework Assignment (Problem Set) 3:

Jerry Zhao

**1.** An engineer at Fertilizer Company has synthesized a sensational new fertilizer made of just two interchangeable basic raw materials. The company wants to take advantage of this opportunity and produce as much as possible of the new fertilizer. The company currently has $50,000 to buy raw materials at a unit price of $9000 and $6000 per unit, respectively. When amounts $x_1$ and $x_2$ of the basic raw materials are combined, a quantity $q$ of fertilizer results given by: $q = 4x_1 + 2x_2 - 0.5x_1^2 - 0.25x_2^2$

Part A: Formulate as a constrained nonlinear program. Clearly indicate the variables, objective function, and constraints.

Part B: Solve the Program (provide exact values for all variables and the optimal objective function).

**Solution:**

**Decision Variables:**

- x1≥0: Amount of the first raw material used
- x2≥0: Amount of the second raw material used

Objective Function:

The company aims to maximize the quantity of fertilizer produced:

Max q = 4x_1 + 2x_2 - 0.5x_1^2 - 0.25x_2^2

- 4x_1 + 2x_2 represents the linear contribution of raw materials to production.
- -0.5x_1^2 - 0.25x_2^2 accounts for diminishing returns as more raw material is used.

**Constraints:**

1. Budget Constraint: 9000x_1 + 6000x_2 <= 50000

The total cost of raw materials cannot exceed the available budget of $50,000.

2. Non-Negativity Constraints: x_1≥0, x_2≥0

The company cannot use negative amounts of raw materials.

```python
from scipy.optimize import minimize


def objective(vars):
    x1, x2 = vars
    return -(4*x1 + 2*x2 - 0.5*x1**2 - 0.25*x2**2)


def constraint(vars):
    x1, x2 = vars
    return 50000 - (9000*x1 + 6000*x2)  # Budget constraint


bounds = [(0, None), (0, None)]

# Initial guess
x0 = [1, 1]


constraints = {'type': 'ineq', 'fun': constraint}

# Solve the optimization problem
solution = minimize(objective, x0, bounds=bounds, constraints=constraints, method='SLSQP')

# Extract the results
optimal_x1, optimal_x2 = solution.x
optimal_q = -solution.fun  # Convert back to maximization

round(optimal_x1,2), round(optimal_x2,2), round(optimal_q,2)
```

```
(3.41, 3.22, 11.67)
```

**2.** The area of a triangle with sides of length $a$, $b$, and $c$ is $\sqrt{s(s-a)(s-b)(s-c)}$, where $s$ is half the perimeter of the triangle. We have 90 feet of fence and want to fence a triangular-shaped area.

Part A: Formulate the problem as a constrained nonlinear program that will enable us to maximize the area of the fenced area, with constraints. Clearly indicate the variables, objective function, and constraints.

Hint: *The length of a side of a triangle must be less than or equal to the sum of the lengths of the other two sides.*

Part B: Solve the Program (provide exact values for all variables and the optimal objective function).

**Solution:**

**Variables:**

- Let a,b,c be the three sides of the triangle.
- Let s be the semi-perimeter: s=(a+b+c)/2

**Objective Function:**

Maximize A=sqrt((s−a)(s−b)(s−c))

**Constraints:**

1. Perimeter Constraint: a+b+c=90
2. Triangle Inequality Constraints: a+b>c, a+c>b, b+c>a
3. Non-Negativity Constraints: a,b,c>0

```
import scipy.optimize as opt
import numpy as np

def heron_area(x):
    a, b, c = x
    s = (a + b + c) / 2
    area = np.sqrt(s * (s - a) * (s - b) * (s - c))
    return -area

# Constraints
def perimeter_constraint(x):
    return 90 - sum(x)

def triangle_inequality_1(x):
    return x[0] + x[1] - x[2]

def triangle_inequality_2(x):
    return x[0] + x[2] - x[1]

def triangle_inequality_3(x):
    return x[1] + x[2] - x[0]

# Define bounds (positive values)
bounds = [(1, 90), (1, 90), (1, 90)]  # Triangle sides must be less than 90 feet total and more than 0

# Initial guess
x0 = [30, 30, 30]  # Start with an equilateral triangle

# Constraints dictionary
constraints = [
    {'type': 'eq', 'fun': perimeter_constraint},
    {'type': 'ineq', 'fun': triangle_inequality_1},
    {'type': 'ineq', 'fun': triangle_inequality_2},
    {'type': 'ineq', 'fun': triangle_inequality_3},
]

# Solve the optimization problem
result = opt.minimize(heron_area, x0, bounds=bounds, constraints=constraints)

# Extract results
optimal_sides = result.x
max_area = -result.fun  # Convert back to positive area

optimal_sides, max_area
```

```
(array([30., 30., 30.]), 389.7114317029974)
```

Optimal triangle is an equilateral triangle

**3.** The Tiny Toy Company makes three types of new toys: the tiny tank, the tiny truck, and the tiny turtle. Plastic used in one unit of each is 1.5, 2.0 and 1.0 pounds, respectively. Rubber for one unit of each toy is 0.5, 0.5, and 1.0 pounds, respectively. Also, each tank uses 0.3 pounds of metal and the truck uses 0.6 pounds of metal during production. The average weekly availability for plastic is 16,000 pounds, 9,000 pounds of metal, and 5,000 pounds of rubber. It takes two hours of labor to make one tank, two hours for one truck, and one hour for a turtle. The company allows no more than 40 hours a week for production (priority #1). Finally, the cost of manufacturing one tank is $7, 1 truck is $5 and 1 turtle is $4; a target budget of $164,000 is initially used as a guideline for the company to follow.

a) Minimize over-utilization of the weekly available supply of materials used in making the toys and place twice as much emphasis on the plastic (priority #2)

b) Minimize the under and over-utilization of the budget. Maximize available labor hour usage (priority #3).

Formulate the above decision problem as a single linear goal program. Clearly identify your achievement vector (i.e., hierarchy of priority levels for the goals). Do not solve.

**Solution:**

**Decision Variables**

- x1 = Number of tiny tanks produced per week
- x2 = Number of tiny trucks produced per week
- x3 = Number of tiny turtles produced per week

Over-utilization (d+) and under-utilization (d−) of resources:

- dp+, dp− = Over/under-utilization of plastic supply
- dr+, dr− = Over/under-utilization of rubber supply
- dm+, dm− = Over/under-utilization of metal supply
- dl+, dl− = Over/under-utilization of labor hours
- db+, db− = Over/under-utilization of budget

**Constraints**

Each toy requires a specific amount of plastic, rubber, and metal:

**Plastic Constraint:** 1.5x1 + 2.0x2 + x3 + dp+ − dp− =16,000

**Rubber Constraint:** 0.5x1 + 0.5x2 + x3 + dr+ − dr− =5,000

**Metal Constraint:** 0.3x1 + 0.6x2 + dm+ − dm− = 9,000

**Labor Constraint (Priority #1)** 2x1 + 2x2 + x3 + dl+ − dl− =40

**Budget Constraint** 7x1 + 5x2 + 4x3 + db+ − db− =164,000

**Objective Function:**

1. Priority #1: Ensure labor hours do not exceed the weekly limit (minimize $dl+$).

2. Priority #2: Minimize over-utilization of materials, with plastic over-utilization weighted twice as much as other materials. Minimize $(2dp+ + dr+ + dm+)$

3. Priority #3: Minimize both under-utilization and over-utilization of the budget, and maximize labor hour usage by minimizing Minimize $(db+ + db-)$ and Minimize $dl-$

4. XYZ Company is planning an advertising campaign for its new product. The media considered are television and radio. Rated exposures per thousand dollars of advertising expenditure are 10,000 for TV and 7,500 for radio. Management has agreed that the campaign cannot be judged successful if total exposures are under 750,000. The campaign would be viewed as superbly successful if 1 million exposures occurred. In addition, the company has realized that the two most important audiences for its product are persons 18 to 21 years of age and persons 25 to 30 years of age. The following table estimates the number of individuals in the two age groups expected to be exposed to advertisements per $ 1,000 of expenditures:

Exposures per $1000

| Age | Television | Radio |
|---|---|---|
| 18-21 | 2,500 | 3,000 |
| 25-30 | 3,000 | 1,500 |

Management has rank ordered five goals it wishes to achieve, arranged from highest to lowest priorities.

a) Achieve total exposures of at least 750,000 persons.
b) Avoid expenditures of more than $100,000.
c) Avoid expenditures of more than $70,000 for television advertisements.
d) Achieve at least 1 million total exposures.
e) Reach at least 250,000 persons in each of the two age groups, 18-21 and 25-30 years. In addition, management realizes and wishes to account for the fact that the purchasing power of the 25-30 age group is twice that of the 18-21 age group.

Formulate the above decision problem as a single linear goal program. Clearly identify your achievement vector (i.e., hierarchy of priority levels for the goals). Do not solve.

**Solution:**

Decision Variables

- xT = Thousands of dollars spent on television advertising
- xR = Thousands of dollars spent on radio advertising

deviation variables for over-utilization (d+) and under-utilization (d−) of each goal:

- de+, de−= Over/under-exposure for total audience
- db+, db− = Over/under-expenditure (total budget)
- dbT+, dbT− = Over/under-expenditure for television budget
- ds+, ds− = Over/under-exposure for superb success
- d18+, d18− = Over/under-exposure for age group 18-21
- d25+, d25− = Over/under-exposure for age group 25-30

**Constraints**

1. Total Exposure Constraints: 10,000xT + 7,500xR + de+ − de− = 750,000

For **superb success**: 10,000xT + 7,500xR + ds+ − ds−=1,000,000

*2. Budget Constraints*

Total expenditure should not exceed \$100,000: $xT + xR + db+ - db- = 100$

Television advertising should not exceed \$70,000: $xT + dbT+ - dbT- = 70$

*3. Age Group Exposure Constraints*

For **18-21 years**:

$2{,}500xT + 3{,}000xR + d18+ - d18- = 250{,}000$

For **25-30 years**:

$3{,}000xT + 1{,}500xR + d25+ - d25- = 250{,}000$

Since the **purchasing power** of the **25-30 age group** is twice that of **18-21**, we account for this by **minimizing the underachievement** for this group with twice the emphasis.

**Objective Function:**

1. **Priority #1:** Achieve total exposures of at least 750,000 (minimize de−).
2. **Priority #2:** Avoid exceeding the total budget of \$100,000 (minimize db+).
3. **Priority #3:** Avoid exceeding \$70,000 in TV advertisements (minimize dbT+).
4. **Priority #4:** Achieve **superb success** with 1,000,000 exposures (minimize ds−).
5. **Priority #5:** Reach at least 250,000 persons in each age group, emphasizing the **25-30 age group** twice as much: Minimize (d18− + 2*d25−)

5. A large food chain owns a number of pharmacies that operate in a variety of settings. Some are situated in small towns and are open for only 8 hours a day, 5 days per week. Others are located in shopping malls and are open for longer hours. The analysts on the corporate staff would like to develop a model to show how a store's revenues depend on the number of hours that it is open. They have collected the following information from a sample of stores.

| **Hours of Operation** | **Average Revenue ($)** |
|---|---|
| 40 | 5958 |
| 44 | 6662 |
| 48 | 6004 |
| 48 | 6011 |
| 60 | 7250 |
| 70 | 8632 |
| 72 | 6964 |
| 90 | 11097 |
| 100 | 9107 |
| 168 | 11498 |

a) Use a linear function (e.g., $y = ax + b$; where a and b are parameters to optimize) to represent the relationship between revenue and operating hours and find the values of the parameters using the nonlinear solver that provide the **best fit** to the given data. What revenue does your model predict for 120 hours?

b) Suggest a two-parameter nonlinear model (e.g., $y = ax^b$; where a and b are parameters to optimize) for the same relationship and find the parameters using the Nonlinear Solver that provide the **best fit**. What revenue does your model predict for 120 hours? Which if the models in (a) and (b) do you prefer and why?

**Solution (Next Page):**

```python
import numpy as np
import pandas as pd
import scipy.optimize as opt

# Given data
hours = np.array([40, 44, 48, 48, 60, 70, 72, 90, 100, 168])
revenue = np.array([5958, 6662, 6004, 6011, 7250, 8632, 6964, 11097, 9107, 11498])

# Linear model: y = ax + b
def linear_model(x, a, b):
    return a * x + b

# Nonlinear model: y = a * x^b
def nonlinear_model(x, a, b):
    return a * (x ** b)

# Fit the linear model
params_linear, _ = opt.curve_fit(linear_model, hours, revenue)
a_linear, b_linear = params_linear

# Predict revenue for 120 hours using linear model
predicted_revenue_linear = linear_model(120, a_linear, b_linear)

# Fit the nonlinear model
params_nonlinear, _ = opt.curve_fit(nonlinear_model, hours, revenue)
a_nonlinear, b_nonlinear = params_nonlinear

# Predict revenue for 120 hours using nonlinear model
predicted_revenue_nonlinear = nonlinear_model(120, a_nonlinear, b_nonlinear)

# Creating a DataFrame for analysis
df_results = pd.DataFrame({
    "Hours of Operation": hours,
    "Actual Revenue ($)": revenue,
    "Linear Model Revenue ($)": linear_model(hours, a_linear, b_linear),
    "Nonlinear Model Revenue ($)": nonlinear_model(hours, a_nonlinear, b_nonlinear)
})


display(df_results)

# Return results
print(f"Best fitted parameter Linear, slope {round(a_linear,2)}, intercept {round(b_linear,2)}")
print("linear model predicted 120 hours: ", round(predicted_revenue_linear,2))
print("-------------------------------------------------------------------")
print(f"Best fitted parameter NonLinear, a {round(a_nonlinear,2)}, b {round(b_nonlinear,2)}")
print("Nonlinear model predicted 120 hours: ", round(predicted_revenue_nonlinear,2))
```

| | Hours of Operation | Actual Revenue ($) | Linear Model Revenue ($) | Nonlinear Model Revenue ($) |
|---|---|---|---|---|
| **0** | 40 | 5958 | 6317.903345 | 6043.860090 |
| **1** | 44 | 6662 | 6506.185305 | 6327.858934 |
| **2** | 48 | 6004 | 6694.467264 | 6598.766510 |
| **3** | 48 | 6011 | 6694.467264 | 6598.766510 |
| **4** | 60 | 7250 | 7259.313142 | 7347.719692 |
| **5** | 70 | 8632 | 7730.018041 | 7914.191567 |
| **6** | 72 | 6964 | 7824.159020 | 8022.337697 |
| **7** | 90 | 11097 | 8671.427838 | 8932.864738 |
| **8** | 100 | 9107 | 9142.132736 | 9398.013048 |
| **9** | 168 | 11498 | 12342.926045 | 12066.650688 |

```
Best fitted parameter Linear, slope 47.07, intercept 4435.08
linear model predicted 120 hours:  10083.54
-------------------------------------------------------------------
Best fitted parameter NonLinear, a 1022.03, b 0.48
Nonlinear model predicted 120 hours:  10260.88
```