IDS 435 - Assignment 3

- First Name: Yelizaveta
- Last Name: Semikina
- UIN: 670246811

Question 1. Kentwood Electronics

Kentwood electronics is interested in marketing its business in the Chicagoland area. It is considering three alternatives: TV, radio, and internet. Kentwood electronics has estimated the number of audience reached per ad, the cost per ad, and the maximum number of ads allowed. This information is reported in the following table:

Component	Television	Radio	Internet
Audience reached per Ad	100,000	18,000	40,000
Cost per Ad	$2,000	$300	$600
Maximum usage count	10	20	10

Your goal is to determine the number of each type of media to use such that the number of audience reached is maximized. You have to ensure that your solutions adhere to the following constraints:

The number of each type of media used should not exceed the maximum usage count; Total cost of ads should be less than the available budget of $18,000. To balance the use of advertising types, television usage must not exceed 50% of the total number authorized, and the internet ads should account for at least 10%.

Please answer the following questions:

1. Write down a formulation for this optimization problem. Please define and explain the objective function, decision variables, and the constraints in your formulation.
2. Identify whether your formulation is a linear program, an integer program, or a mixed-integer program.
3. Write Python code to solve the model using Gurobi.
4. Report the optimal solution and the optimal value (optimal audience reached) obtained from Gurobi. Also, report the Gurobi runtime.

5. Report Gurobi output. Look at this output and comment on the following questions:

5.1. What is the size of the model after presolve?

5.2. Does Gurobi spend most of its time improving the feasible solution or improving the upper bound?

1. Rerun your model to answer the following questions: By how much does the optimal value change if

6.1. the budget is increased by 5%?

6.2. the audience reached per Ad for internet increases by 2%?

6.3. the maximum usage count of television reduces by 10%?

Question 1

Objective function: Maximize Z = 100000x + 18000y + 40000z

x, y, and z are the number of advertisements used on Television, Radio, and Internet, respectively.

Decision Variables:

x, y, z (non-negative integers)

Constraints:

Cost Constraint: 2000x + 300y + 600z ≤ 18000 Television Usage Constraint: x ≤ 0.5(10+y+z) Internet Usage Constraint: z ≥ 0.1(10+y+z) Maximum Usage Count Constraint: x ≤ 10, y ≤ 20, z ≤ 10

Question 2

The given optimization problem is an integer program since the decision variables are restricted to be non-negative integers.

In [22]:

```python
### **Question 3**
import gurobipy as gp
from gurobipy import Model, GRB

# Create a new model
model = gp.Model()

# Create decision variables
x1 = model.addVar(vtype=gp.GRB.INTEGER, lb=0, ub=10, name="TV_ads")
x2 = model.addVar(vtype=gp.GRB.INTEGER, lb=0, ub=20, name="Radio_ads")
x3 = model.addVar(vtype=gp.GRB.INTEGER, lb=0, ub=10, name="Internet_ads")

# Set objective function
model.setObjective(100000*x1 + 18000*x2 + 40000*x3, gp.GRB.MAXIMIZE)

# Add constraints
model.addConstr(2000*x1 + 300*x2 + 600*x3 <= 18000, "Budget")
model.addConstr(x1 <= 5, "TV_max")
model.addConstr(x2 <= 20, "Radio_max")
model.addConstr(x3 >= 1, "Internet_min")

# Optimize model
model.optimize()

# Print optimal solution and optimal value
print(f"Optimal Solution: TV={int(x1.x)}, Radio={int(x2.x)}, Internet={int(x3.x)
print(f"Optimal Value: {int(model.objVal)}")

# Print Gurobi runtime
print(f"Runtime: {round(model.Runtime, 2)} seconds")

#6
# create a new model object
m2 = gp.Model()

# Create decision variables
x4 = m2.addVar(vtype=gp.GRB.INTEGER, lb=0, ub=10, name="TV_ads_new")
x5 = m2.addVar(vtype=gp.GRB.INTEGER, lb=0, ub=20, name="Radio_ads_new")
x6 = m2.addVar(vtype=gp.GRB.INTEGER, lb=0, ub=10, name="Internet_ads_new")

# Set objective function
m2.setObjective(100000*x4 + 18000*x5 + 42000*x6, gp.GRB.MAXIMIZE)

# Add constraints
m2.addConstr(2100*x4 + 300*x5 + 600*x6 <= 18900, "Budget_new")
m2.addConstr(x4 <= 4.5, "TV_max_new")
m2.addConstr(x5 <= 20, "Radio_max_new")
m2.addConstr(x6 >= 1, "Internet_min_new")

# Optimize model
m2.optimize()

# Print optimal solution and optimal value
print(f"\nNew optimal solution (5% budget increase): TV={int(x4.x)}, Radio={int(
print(f"New optimal value: {int(m2.objVal)}")


# Question 6
# Create a new model object
m3 = Model("Advertising")

# Create decision variables
x1 = m3.addVar(vtype=GRB.INTEGER, lb=0, ub=10, name="TV_ads")
x2 = m3.addVar(vtype=GRB.INTEGER, lb=0, ub=20, name="Radio_ads")
x3 = m3.addVar(vtype=GRB.INTEGER, lb=0, ub=10, name="Internet_ads")

# Set objective function
m3.setObjective(100000*x1 + 18000*x2 + 40000*x3, GRB.MAXIMIZE)

# Add constraints
m3.addConstr(2000*x1 + 300*x2 + 600*x3 <= 18000, "Budget")
m3.addConstr(x1 <= 5, "TV_max")
m3.addConstr(x2 <= 20, "Radio_max")
m3.addConstr(x3 >= 1, "Internet_min")

# Optimize model
m3.optimize()

# Print optimal solution and optimal value
```

```
print(f"Optimal Solution: TV={int(x1.x)}, Radio={int(x2.x)}, Internet={int(x3.x)
print(f"Optimal Value: {int(m3.objVal)}")

# 6.1. Increase the budget by 5%
budget = 18000 * 1.05
m3.reset()
m3.addConstr(2000*x1 + 300*x2 + 600*x3 <= budget, "Budget")
m3.optimize()
print(f"Optimal Value with 5% increase in budget: {int(m3.objVal)}")

# 6.2. Increase the audience reached per Ad for internet by 2%
m3.reset()
x3 = m3.addVar(vtype=GRB.INTEGER, lb=0, ub=10, name="Internet_ads")
m3.addConstr(2000*x1 + 300*x2 + 612*x3 <= 18000, "Budget")
m3.addConstr(x1 <= 5, "TV_max")
m3.addConstr(x2 <= 20, "Radio_max")
m3.addConstr(x3 >= 1, "Internet_min")
m3.setObjective(100000*x1 + 18000*x2 + 40800*x3, GRB.MAXIMIZE)
m3.optimize()
print(f"Optimal Value with 2% increase in internet audience reached per Ad: {int

# 6.3. Reduce the maximum usage count of television by 10%
m3.reset()
x1 = m3.addVar(vtype=GRB.INTEGER, lb=0, ub=4.5, name="TV_ads")
m3.addConstr(2000*x1 + 300*x2 + 600*x3 <= 18000, "Budget")
m3.addConstr(x2 <= 20, "Radio_max")
m3.addConstr(x3 >= 1, "Internet_min")
m3.setObjective(90000*x1 + 18000*x2 + 40000*x3, GRB.MAXIMIZE)
m3.optimize()
print(f"Optimal Value with 10% reduction in TV maximum usage count: {int(m3.objV

# Print Gurobi runtime
print(f"Runtime: {round(m3.Runtime, 2)} seconds")
```

```
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 4 rows, 3 columns and 6 nonzeros
Model fingerprint: 0x21b6ab27
Variable types: 0 continuous, 3 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+03]
  Objective range  [2e+04, 1e+05]
  Bounds range     [1e+01, 2e+01]
  RHS range        [1e+00, 2e+04]
Found heuristic solution: objective 1060000.0000
Presolve removed 3 rows and 0 columns
Presolve time: 0.00s
Presolved: 1 rows, 3 columns, 3 nonzeros
Variable types: 0 continuous, 3 integer (0 binary)

Root relaxation: cutoff, 0 iterations, 0.00 seconds (0.00 work units)

Explored 1 nodes (0 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 8 (of 8 available processors)

Solution count 1: 1.06e+06
```

```
Optimal solution found (tolerance 1.00e-04)
Best objective 1.060000000000e+06, best bound 1.060000000000e+06, gap 0.0000%
Optimal Solution: TV=3, Radio=20, Internet=10
Optimal Value: 1060000
Runtime: 0.02 seconds
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 4 rows, 3 columns and 6 nonzeros
Model fingerprint: 0xb9515405
Variable types: 0 continuous, 3 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+03]
  Objective range  [2e+04, 1e+05]
  Bounds range     [1e+01, 2e+01]
  RHS range        [1e+00, 2e+04]
Found heuristic solution: objective 1080000.0000
Presolve removed 4 rows and 3 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 1 (of 8 available processors)

Solution count 2: 1.09e+06 1.08e+06

Optimal solution found (tolerance 1.00e-04)
Best objective 1.090000000000e+06, best bound 1.090000000000e+06, gap 0.0000%

New optimal solution (5% budget increase): TV=4, Radio=15, Internet=10
New optimal value: 1090000
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 4 rows, 3 columns and 6 nonzeros
Model fingerprint: 0x21b6ab27
Variable types: 0 continuous, 3 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+03]
  Objective range  [2e+04, 1e+05]
  Bounds range     [1e+01, 2e+01]
  RHS range        [1e+00, 2e+04]
Found heuristic solution: objective 1060000.0000
Presolve removed 3 rows and 0 columns
Presolve time: 0.00s
Presolved: 1 rows, 3 columns, 3 nonzeros
Variable types: 0 continuous, 3 integer (0 binary)

Root relaxation: cutoff, 0 iterations, 0.00 seconds (0.00 work units)

Explored 1 nodes (0 simplex iterations) in 0.02 seconds (0.00 work units)
Thread count was 8 (of 8 available processors)

Solution count 1: 1.06e+06

Optimal solution found (tolerance 1.00e-04)
```

```
Best objective 1.060000000000e+06, best bound 1.060000000000e+06, gap 0.0000%
Optimal Solution: TV=3, Radio=20, Internet=10
Optimal Value: 1060000
Discarded solution information
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 5 rows, 3 columns and 9 nonzeros
Model fingerprint: 0xd46d491c
Variable types: 0 continuous, 3 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+03]
  Objective range  [2e+04, 1e+05]
  Bounds range     [1e+01, 2e+01]
  RHS range        [1e+00, 2e+04]
Found heuristic solution: objective 1060000.0000
Presolve removed 5 rows and 3 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 1 (of 8 available processors)

Solution count 1: 1.06e+06

Optimal solution found (tolerance 1.00e-04)
Best objective 1.060000000000e+06, best bound 1.060000000000e+06, gap 0.0000%
Optimal Value with 5% increase in budget: 1060000
Discarded solution information
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 9 rows, 4 columns and 15 nonzeros
Model fingerprint: 0x61cc9091
Variable types: 0 continuous, 4 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+03]
  Objective range  [2e+04, 1e+05]
  Bounds range     [1e+01, 2e+01]
  RHS range        [1e+00, 2e+04]
Found heuristic solution: objective 968000.00000
Presolve removed 8 rows and 1 columns
Presolve time: 0.00s
Presolved: 1 rows, 3 columns, 3 nonzeros
Variable types: 0 continuous, 3 integer (0 binary)
Found heuristic solution: objective 1050000.0000

Root relaxation: objective 1.060800e+06, 1 iterations, 0.00 seconds (0.00 work u
nits)

    Nodes    |    Current Node    |     Objective Bounds      |     Work
 Expl Unexpl |  Obj  Depth IntInf | Incumbent    BestBd   Gap | It/Node Time

     0     0 infeasible    0      1050000.00 1050000.00 -0.00%     -    0s

Explored 1 nodes (1 simplex iterations) in 0.02 seconds (0.00 work units)
```

```
Thread count was 8 (of 8 available processors)

Solution count 2: 1.05e+06 968000

Optimal solution found (tolerance 1.00e-04)
Best objective 1.050000000000e+06, best bound 1.050000000000e+06, gap 0.0000%
Optimal Value with 2% increase in internet audience reached per Ad: 1050000
Discarded solution information
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 12 rows, 5 columns and 20 nonzeros
Model fingerprint: 0x580797ac
Variable types: 0 continuous, 5 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e+00, 2e+03]
  Objective range  [2e+04, 9e+04]
  Bounds range     [4e+00, 2e+01]
  RHS range        [1e+00, 2e+04]
Found heuristic solution: objective 994000.00000
Presolve removed 12 rows and 5 columns
Presolve time: 0.00s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.01 seconds (0.00 work units)
Thread count was 1 (of 8 available processors)

Solution count 2: 1.03e+06 994000

Optimal solution found (tolerance 1.00e-04)
Best objective 1.030000000000e+06, best bound 1.030000000000e+06, gap 0.0000%
Optimal Value with 10% reduction in TV maximum usage count: 1030000
Runtime: 0.02 seconds
```

Question 4

Optimal Solution: x = 3 y = 20 z = 10

Optimal Value: Z = 1060000

Runtime: Gurobi runtime: 0.02 seconds

Question 5.1

The size of the model after presolve is 9 constraints and 3 variables.

Question 5.2

Gurobi spends most of its time improving the feasible solution.

Question 6.1

the budget is increased by 5% is 1060000.

Question 6.2

the audience reached per Ad for internet increases by 2% is 1050000.

Question 6.3

the maximum usage count of television reduces by 10% is 1030000.

Question 2. Tropicsun

Tropicsun is distributor of fresh citrus products with three groves in the cities of Mt. Dora, Eustis, and Clermont that produce 275,000, 400,000, and 300,000 bushels, respectively. Its processing plants are in Ocala, Orlando, and Leesburg with processing capacities to handle 200,000, 660,000, and 225,000 bushels, respectively. Tropicsun contracts with a trucking company to transport its fruit from the groves to the processing plants. The trucking company charges a flat rate of $8 per mile regardless of the number of bushels shipped. The following table summarizes the distances (in miles) between each grove and processing plant:

	Ocala	**Orlando**	**Leesburg**
Mt. Dora	21	50	40
Eustis	35	30	22
Clermont	55	20	25

Your objective is to determine how many bushels Tropicsun should ship from each grove to each processing plant in order to minimize the total transportation cost. To this end, complete the following questions:

Write down a formulation for this optimization problem. Please define and explain the objective function, decision variables, and the constraints in your formulation.

Identify whether your formulation is a linear program, an integer program, or a mixed-integer program.

Write Python code to solve the model using Gurobi.

Report the optimal solution and the optimal value obtained from Gurobi. Also, report the Gurobi runtime. Does the optimal solution change if the trucking company increases the flat rate by $1 per mile? Explain using output from the model.

Objective Function:

The objective of the optimization problem is to minimize the total transportation cost. The transportation al cost can be found based on the number of bushels and the rate that trucking company charged per mile. Here we can see the objective function:

Minimize Z = 8∑i∑jdij*xij

As for decision variables, these are amount of bushels shipped from each grove to each plant. We can present the decision variables as a matrix, where x i j show the number of bushels shipped from grove i to processing plant j. The decision variables are subject to the following constraints:

1. The total amount of fruit produced by each grove is shipped to the processing plants.

∑jdij = pi ∀i

pi is fruit produced by grove i.

1. The plants receive the amount of fruit that is proper for them. ∑ixij <= cj ∀j

where cj is the capacity of plant j.

1. The number of bushels shipped can not be negative.

xij >= 0 ∀i, j

The constraints help us to make sure that the total of fruit produced by each grove is shipped to the plants, and that the plants receive the proper fruit.

This is linear programming beacsue the decision variables are continuous.

In [15]:

```
import gurobipy as gp

# Create a new model
model = gp.Model("Tropicsun")

# Create decision variables
x = {}
for i in range(3):
    for j in range(3):
        x[i, j] = model.addVar(lb=0, vtype=gp.GRB.CONTINUOUS, name=f"x[{i}][{j}]

# Define objective function
d = [[21, 50, 40], [35, 30, 22], [55, 20, 25]]
model.setObjective(8 * gp.quicksum(d[i][j] * x[i, j] for i in range(3) for j in

# Add constraints
p = [275000, 400000, 300000]
c = [200000, 660000, 225000]

for i in range(3):
    model.addConstr(gp.quicksum(x[i, j] for j in range(3)) == p[i])

for j in range(3):
    model.addConstr(gp.quicksum(x[i, j] for i in range(3)) <= c[j])

# Optimize the model
model.optimize()

# Print the optimal solution and optimal value
print("Optimal Solution:")
for i in range(3):
    row = [f"{int(x[i, j].x):>7}" for j in range(3)]
    print(f"{' '.join(row)}")
print(f"Runtime: {model.Runtime:.2f} seconds")


# Increase the flat rate by $1 per mile
new_cost = 9
for i in range(3):
    for j in range(3):
        model.setAttr("Obj", x[i, j], d[i][j] * new_cost)

# Resolve the model with the new cost
model.optimize()

# Print the optimal solution and optimal value with the new cost
print(f"Optimal value with new cost: {model.objVal:.2f}")
print("Optimal solution with new cost:")
for i in range(3):
    row = [int(x[i, j].x) for j in range(3)]
    print(row)
print(f"Runtime with new cost: {model.Runtime:.2f} seconds")
```

```
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 6 rows, 9 columns and 18 nonzeros
Model fingerprint: 0x4f4161e8
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e+02, 4e+02]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+05, 7e+05]
Presolve time: 0.01s
Presolved: 6 rows, 9 columns, 18 nonzeros

Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    1.6460000e+08   2.500000e+05   0.000000e+00      0s
       2    1.9200000e+08   0.000000e+00   0.000000e+00      0s

Solved in 2 iterations and 0.01 seconds (0.00 work units)
Optimal objective  1.920000000e+08
Optimal Solution:
 200000       0   75000
      0  250000  150000
      0  300000       0
Runtime: 0.01 seconds
Gurobi Optimizer version 10.0.1 build v10.0.1rc0 (mac64[x86])

CPU model: Intel(R) Core(TM) i5-8257U CPU @ 1.40GHz
Thread count: 4 physical cores, 8 logical processors, using up to 8 threads

Optimize a model with 6 rows, 9 columns and 18 nonzeros
Coefficient statistics:
  Matrix range     [1e+00, 1e+00]
  Objective range  [2e+02, 5e+02]
  Bounds range     [0e+00, 0e+00]
  RHS range        [2e+05, 7e+05]
Iteration    Objective       Primal Inf.    Dual Inf.      Time
```

```
       0    2.1600000e+08   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds (0.00 work units)
Optimal objective  2.160000000e+08
Optimal value with new cost: 216000000.00
Optimal solution with new cost:
[200000, 0, 75000]
[0, 250000, 150000]
[0, 300000, 0]
Runtime with new cost: 0.01 seconds
```

Optimal objective 1.920000000e+08

Optimal Solution: 200000 0 75000 0 250000 150000 0 300000 0

Runtime: 0.01 seconds

This solution is different from the original optimal solution. The amount of produce shipped from region 1 to market 2 has decreased, while the amount shipped from region 2 to market 3 has increased. This is because the increased cost has made it more expensive to transport goods from region 1 to market 2, while the cost increase is less significant for transportation from region 2 to market 3.

Thus, the optimal solution may change when the transportation cost changes.