Question 2 Code

Mark Ma

Sep 21, 2024

[1]:

```
import gurobipy as gp
from gurobipy import GRB
```

[2]:

```
# Data: Supplier information , cost per unit (c_jk), and maximum supply␣
↪capacity (s_jk)
suppliers = ['Supplier1', 'Supplier2', 'Supplier3']
resources = ['Toilet␣Paper', 'Liquid␣Soap', 'Detergent', 'Cloths',
'Toothpaste', 'Toothbrushes', 'Sanitary␣Pads', 'Shampoo']

# Cost and maximum supply capacity for each supplier and resource
cost_per_unit = {
('Supplier1', 'Toilet␣Paper'): (0.80, 150),
('Supplier1', 'Liquid␣Soap'): (6.40, 25),
('Supplier1', 'Detergent'): (6.80, 20),
('Supplier1', 'Cloths'): (10.00, 10),
('Supplier1', 'Toothpaste'): (2.60, 50),
('Supplier1', 'Toothbrushes'): (0.80, 50),
('Supplier1', 'Sanitary␣Pads'): (0.20, 150),
('Supplier1', 'Shampoo'): (2.30, 20),
('Supplier2', 'Toilet␣Paper'): (0.95, 100),
('Supplier2', 'Liquid␣Soap'): (3.98, 15),
('Supplier2', 'Detergent'): (4.60, 10),
('Supplier2', 'Cloths'): (11.00, 10),
('Supplier2', 'Toothpaste'): (3.00, 60),
('Supplier2', 'Toothbrushes'): (0.85, 60),
('Supplier2', 'Sanitary␣Pads'): (0.18, 100),
('Supplier2', 'Shampoo'): (1.20, 20),
('Supplier3', 'Toilet␣Paper'): (0.84, 70),
('Supplier3', 'Liquid␣Soap'): (5.50, 30),
('Supplier3', 'Detergent'): (7.50, 15),
('Supplier3', 'Cloths'): (10.50, 15),
('Supplier3', 'Toothpaste'): (2.80, 30),
('Supplier3', 'Toothbrushes'): (0.82, 30),
('Supplier3', 'Sanitary␣Pads'): (0.15, 100),
('Supplier3', 'Shampoo'): (3.00, 30)
}
```

```
# Minimum required quantities as a dataset for a shelter housing 20 people for␣
 ↪one month
minimum_quantities = {
'Toilet␣Paper': 200,
'Liquid␣Soap': 40,
'Detergent': 30,
'Cloths': 20,
'Toothpaste': 100,
'Toothbrushes': 100,
'Sanitary␣Pads': 300,
'Shampoo': 40
}
```

```
[3]: budget = 2000
```

```
[4]: model = gp.Model("Resource_Procurement")
```

```
Set parameter Username
Academic license - for non-commercial use only - expires 2025-09-09
```

```
[7]: x = model.addVars(cost_per_unit.keys(), name="x", vtype=GRB.INTEGER, lb=0)
     x
```

```
[7]: {('Supplier1', 'Toilet␣Paper'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Liquid␣Soap'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Detergent'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Cloths'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Toothpaste'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Toothbrushes'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Sanitary␣Pads'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier1', 'Shampoo'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Toilet␣Paper'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Liquid␣Soap'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Detergent'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Cloths'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Toothpaste'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Toothbrushes'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Sanitary␣Pads'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier2', 'Shampoo'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Toilet␣Paper'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Liquid␣Soap'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Detergent'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Cloths'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Toothpaste'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Toothbrushes'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Sanitary␣Pads'): <gurobi.Var *Awaiting Model Update*>,
      ('Supplier3', 'Shampoo'): <gurobi.Var *Awaiting Model Update*>}
```

[8]:
```
model.setObjective(gp.quicksum(x[supplier, resource] * cost_per_unit[(supplier,␣
 ↪resource)][0] for supplier, resource in cost_per_unit.keys()), GRB.MINIMIZE)
```

[9]:
```
for resource in resources:
    model.addConstr(gp.quicksum(x[supplier, resource] for supplier in␣
 ↪suppliers) >= minimum_quantities[resource], name=f"Demand_{resource}")
```

[10]:
```
for (supplier, resource), (_, max_supply) in cost_per_unit.items():
    model.addConstr(x[supplier, resource] <= max_supply,␣
 ↪name=f"Supply_{supplier}_{resource}")
```

[11]:
```
model.addConstr(gp.quicksum(x[supplier, resource] * cost_per_unit[(supplier,␣
 ↪resource)][0] for supplier, resource in cost_per_unit.keys()) <= budget,␣
 ↪name="Budget")
```

[11]:
```
<gurobi.Constr *Awaiting Model Update*>
```

[12]:
```
model.optimize()
```

```
Gurobi Optimizer version 11.0.3 build v11.0.3rc0 (win64 - Windows 10.0
(19045.2))

CPU model: Intel(R) Core(TM) i7-10875H CPU @ 2.30GHz, instruction set
[SSE2|AVX|AVX2]
Thread count: 8 physical cores, 16 logical processors, using up to 16 threads

Optimize a model with 33 rows, 72 columns and 72 nonzeros
Model fingerprint: 0xdeb5e035
Variable types: 0 continuous, 72 integer (0 binary)
Coefficient statistics:
  Matrix range     [1e-01, 1e+01]
  Objective range  [1e-01, 1e+01]
  Bounds range     [0e+00, 0e+00]
  RHS range        [1e+01, 2e+03]
Found heuristic solution: objective 1332.1000000
Presolve removed 33 rows and 72 columns
Presolve time: 0.01s
Presolve: All rows and columns removed

Explored 0 nodes (0 simplex iterations) in 0.02 seconds (0.00 work units)
Thread count was 1 (of 16 available processors)

Solution count 2: 1224.8 1332.1

Optimal solution found (tolerance 1.00e-04)
Best objective 1.224800000000e+03, best bound 1.224800000000e+03, gap 0.0000%
```

```
[13]: if model.status == GRB.OPTIMAL:
          print("Optimal solution found:")
          for supplier, resource in cost_per_unit.keys():
              if x[supplier, resource].x > 0:  # Only print positive purchase␣
       ↪quantities
                  print(f"Buy {x[supplier, resource].x:.2f} units of {resource} from␣
       ↪{supplier}")
          print(f"Total cost: {model.objVal:.2f}")
      else:
          print("No optimal solution found")
```

```
Optimal solution found:
Buy 150.00 units of Toilet␣Paper from Supplier1
Buy 20.00 units of Detergent from Supplier1
Buy 10.00 units of Cloths from Supplier1
Buy 50.00 units of Toothpaste from Supplier1
Buy 50.00 units of Toothbrushes from Supplier1
Buy 100.00 units of Sanitary␣Pads from Supplier1
Buy 20.00 units of Shampoo from Supplier1
Buy 15.00 units of Liquid␣Soap from Supplier2
Buy 10.00 units of Detergent from Supplier2
Buy 20.00 units of Toothpaste from Supplier2
Buy 20.00 units of Toothbrushes from Supplier2
Buy 100.00 units of Sanitary␣Pads from Supplier2
Buy 20.00 units of Shampoo from Supplier2
Buy 50.00 units of Toilet␣Paper from Supplier3
Buy 25.00 units of Liquid␣Soap from Supplier3
Buy 10.00 units of Cloths from Supplier3
Buy 30.00 units of Toothpaste from Supplier3
Buy 30.00 units of Toothbrushes from Supplier3
Buy 100.00 units of Sanitary␣Pads from Supplier3
Total cost: 1224.80
```