

## 2. Comquat Computers [15 pts]

Comquat owns four production plants at which personal computers are produced. Comquat can sell up to 20,000 computers per year at a price of \$3,500 per computer. For each plant the production capacity, cost per computer, and fixed cost of operating the plant for a year are given below. Determine how Comquat can maximize its yearly profit from computer production.

Plant	Production capacity	Plant fixed cost (\$ Million)	Cost per computer (\$)
1	10,000	9	1,000
2	8,000	5	1,700
3	9,000	3	2,300
4	6,000	1	2,900

```
In [1]: # data
prod_caps = [10000, 8000, 9000, 6000]
plant_fixed_cost = [9, 5, 3, 1]
cost_per_computer = [1000, 1700, 2300, 2900]

max_computers_per_year = 20000
price_per_computer = 3500
```

```
In [24]: using JuMP, Cbc, Gurobi, Mosek, GLPK

m = Model(solver = MosekSolver())

@variable(m, x[1:4] >= 0, Int) # quantities from each plant
@variable(m, z[1:4], Bin) # whether plant is active
@constraint(m, x .<= 10000*z) # if x > 0 then z = 1

@constraint(m, x[1:4] .<= prod_caps[1:4]) # cap plant capacities according to data

@constraint(m, sum(x[1:4]) <= 20000) # sale constraint

@expression(m, max_revenue, sum(x[1:4]) * 3500) # price per computer constraint

@expression(m, total_prod_cost, (cost_per_computer[1]*x[1] + cost_per_computer[2]*x[2] + cost_per_computer[3]*x[3] + cost_per_computer[4]*x[4]))
@expression(m, total_plant_cost, (plant_fixed_cost[1]*z[1] + plant_fixed_cost[2]*z[2] + plant_fixed_cost[3]*z[3] + plant_fixed_cost[4]*z[4]))

@objective(m, Max, max_revenue - (total_plant_cost + total_prod_cost))

solve(m)
```

Out[24]: :Optimal

```
In [25]: for i = 1:4
println("Plant ", i, " produced ", getvalue(x[i]), " computers")
end

Plant 1 produced 10000.0 computers
Plant 2 produced 8000.0 computers
Plant 3 produced 0.0 computers
Plant 4 produced 2000.0 computers
```

```
In [26]: println("With this configuration, Comquat Computers will make a profit of \$2.56e7")
```