using JuMP

using GLPK

model = Model(GLPK.Optimizer)

# Karar değişkenleri

@variable(model, X[1:2, 1:5] >= 0, Int)

#PARAMETRELER

# Tenör Oranları

T = [0.08 0.48; 0.28 0.12; 0.11 0.05; 0.45 0.68; 0.1 0.09]

# Ton Başına Satış Fiyatları

D = [8000, 2600, 2050, 800000, 65500000]

# Ton Başına Maliyetler

M = [850 900; 200 350; 750 800; 1500 2000; 3000 4500]

#KISITLAR

# İşçilik Kısıtı

*@constraint(model, sum((10/6)\* X[i, j] for i in 1:2, j in 1:3) + (10/4) \* sum(X[i, j] for i in 1:2, j in 4:5) <= 2995200)*

# Kapasite Kısıtları

@constraint(model, sum(X[2, j] for j in 1:5) <= 2e7)

@constraint(model, sum(X[1, j] for j in 1:5) <= 12e6)

@constraint(model, sum(X[i, 4] for i in 1:2) <= 140000)

@constraint(model, sum(X[i, 5] for i in 1:2) <= 807992)

# Yasal Mevzuat Kısıtları

*@constraint(model, 0.00105 \* (X[1, 2] + X[2, 2]) <= 60)*

*@constraint(model, 0.0000105 \* (X[1, 1] + X[2, 1]) <=200)*

# Talep Kısıtları (Asgari Zenginleştirilmiş Cevher Miktarları)

@constraint(model, (X[1, 1]\*0.08 + X[2, 1]\*0.48) >= 160000)

@constraint(model, (X[1, 2]\*0.28 + X[2, 2]\*0.12) >= 20)

@constraint(model, (X[1, 3]\*0.11 + X[2, 3]\*0.05) >= 25)

@constraint(model, (X[1, 4]\*0.45 + X[2, 4]\*0.68) >= 2000)

@constraint(model, (X[1, 5]\*0.10 + X[2, 5]\*0.09) >= 1000)

# Amaç Fonksiyonu

@objective(model, Max, sum((X[i, j] \* (T[j, i] \* D[j]) - (X[i, j] \* M[j, i])) for i in 1:2, j in 1:5))

# Çözüm

optimize!(model)

println("Optimal Çözüm:")

println("EnbZ Değeri: ", objective\_value(model))

println("Karar Değişkenleri Değerleri:")

for i in 1:2

for j in 1:5

println("X$i$j = ", value(X[i, j]))

end

end