Python Project on Optimization : Transportation Network for distributing products

Problem Description A company has two warehouses from which it distributes products to five carefully chosen distribution centers. The company would like to have an interactive computer program which they can run week by week to tell them which warehouse should supply which distribution center so as to minimize the costs of the whole operation. For example, suppose that at the start of a given week the company has 2050 cases at warehouse A, and 8010 cases at warehouse B, and that the distribution centers or customer points require 1000, 1800, 4000, 500, and 1350 cases respectively. Which warehouse should supply which customer point?

from pulp import \*

Warehouses = ["A", "B"]

supply = {"A": 2050,

"B" : 8010}

CustomerPoint = ["1", "2", "3", "4", "5"]

demand = {"1":1000,

"2":1800,

"3" :4000,

"4" :500,

"5":1350,}

costs =[

[2,4,5,2,1] ,[3,1,3,2,3]

]

costs = makeDict([Warehouses,CustomerPoint],costs,0)

prob = LpProblem("Product Distribution Problem",LpMinimize)

Routes = [(w,b) for w in Warehouses for b in CustomerPoint]

vars = LpVariable.dicts("Route",(Warehouses,CustomerPoint),0,None,LpInteger)

prob += lpSum([vars[w][b]\*costs[w][b] for (w,b) in Routes]), "Sum\_of\_Transporting\_Costs"

for w in Warehouses:

prob += lpSum([vars[w][b] for b in CustomerPoint])<=supply[w], "Sum\_of\_Products\_out\_of\_Warehouse\_%s"%w

customer\_demand\_constraint = {}

for b in CustomerPoint:

constraint = lpSum([vars[w][b] for w in Warehouses])>=demand[b]

prob += constraint, "Sum\_of\_Products\_into\_customer\_%s"%b

customer\_demand\_constraint[b] = constraint

prob.writeLP("ProductDistributionProblem.lp")

for demand in range(500, 601, 10):

customer\_demand\_constraint['1'].constant = - demand

prob.solve()

print("Status:", LpStatus[prob.status])

for v in prob.variables():

print(v.name, "=", v.varValue)

print("Total Cost of Transportation = ", value(prob.objective))