Estudo de Caso 02
Problema de Transporte

Alícia Marzola Chaves
José Gabriel Vieira de Souza

1 Problema

O objetivo deste trabalho é abordar o problema do transporte, que se caracteriza pela necessidade de determinar a quantidade de produtos a serem transportados de um ponto de origem para um ponto de destino. Cada ponto apresenta uma oferta e uma demanda mínima, e o desafio consiste em criar um modelo que atenda a essas restrições e minimize os custos envolvidos.

O modelo proposto para resolver este problema em Python foi adaptado a partir de um modelo desenvolvido para o problema do caixeiro viajante, conforme apresentado em trabalhos anteriores. Nele, foram definidas as instâncias e seus respectivos valores de custo, oferta e demanda, além de serem incluídas outras duas instâncias de maior complexidade do que as especificadas originalmente no trabalho. Dessa forma, ao longo deste relatório, será realizada uma análise e discussão dos resultados obtidos para cada instância.

Modelo

O código em Python foi utilizado para todas as três instâncias escolhidas:
Com as bibliotecas necessárias importadas, os valores de custo, oferta e demanda foram definidos a fim do modelo ser declarado.

```
[ ] from math import sqrt
    import numpy as np

    c = 3
    m = 4

    C = range(c)
    M = range(m)
    A = [(i,j) for (i,j) in product(C,M)]
    Custo = [[10,2,20,11],
         [12,7,9,20],
         [4,14,16,18]]

    oferta = [15, 25, 10]
    demanda = [5,15,15,15]
```

```
[ ] # declaracao do modelo
    model = Model('Centros de Distribuicao', solver_name=CBC)

    x = {(i,j): model.add_var(lb=0) for (i,j) in A}
    x
```

Após isso, a função objetivo para minimização do custo foi chamada e as funções de restrição também foram modeladas.

```
[ ] # funcao objetivo: minimizacao do custo
    model.objective = minimize(xsum(Custo[i][j] * x[i,j] for (i,j) in A))
    print(model.objective)
```

```
[ ] for i in C:
        model += xsum(x[i,j] for j in M) == oferta[i]
```

```
[ ] for j in M:
        model += xsum(x[i,j] for i in C) == demanda[j]
```

Por fim, o modelo foi otimizado e o custo mínimo total foi calculado.

```
[ ] status = model.optimize()

    print("Custo Total: {:12.2f}.".format(model.objective_value))
```

2 Instâncias utilizadas

1. Instância dada na especificação: **três** centros de distribuição e **quatro** mercados
Custo = [[10,2,20,11],
[12,7,9,20],
[4,14,16,18]]
Oferta = [15, 25, 10]
Demanda = [5,15,15,15]

2. **Cinco** centros de distribuição e **seis** mercados.
Custo = [
[10, 2, 20, 11, 15, 18],
[12, 7, 9, 20, 14, 16],
[4, 14, 16, 18, 17, 10],
[8, 11, 13, 15, 19, 22],
[6, 9, 21, 23, 12, 13]]
Oferta = [20, 15, 25, 30, 10]
Demanda = [25, 20, 15, 15, 10, 15]

3. **Seis** centros de distribuição e **sete** mercados
Custo = [
[10, 2, 20, 11, 15, 18, 25],
[12, 7, 9, 20, 14, 16, 21],
[4, 14, 16, 18, 17, 10, 19],
[8, 11, 13, 15, 19, 22, 20],
[6, 9, 21, 23, 12, 13, 17],

[5, 8, 17, 15, 23, 14, 16]]
Oferta = [30, 25, 20, 35, 15, 25]
Demanda = [30, 20, 15, 25, 25, 20, 15]

3 Discussão dos resultados

Ao analisar os resultados das diferentes instâncias do problema de transporte, observamos variações significativas no custo total conforme o aumento do número de pontos envolvidos. Na instância 1, composta por 12 pontos, o custo total alcançou $435.00, enquanto na instância 2, com 30 pontos, o custo aumentou para $830.00. Essa tendência de aumento de custo se mantém na instância 3, onde temos 42 pontos, resultando em um custo total de $1480.00. Essa relação direta entre o número de pontos e o custo total sugere que, à medida que mais pontos são incluídos, as complexidades do transporte e os custos associados aumentam significativamente. Essa análise ressalta a importância de um planejamento eficiente no transporte de mercadorias, visando otimizar os recursos e minimizar os custos operacionais.