**Atividade 4 - Otimização Linear**

**Entrega:** 23/09/2020

**PARTE 1**

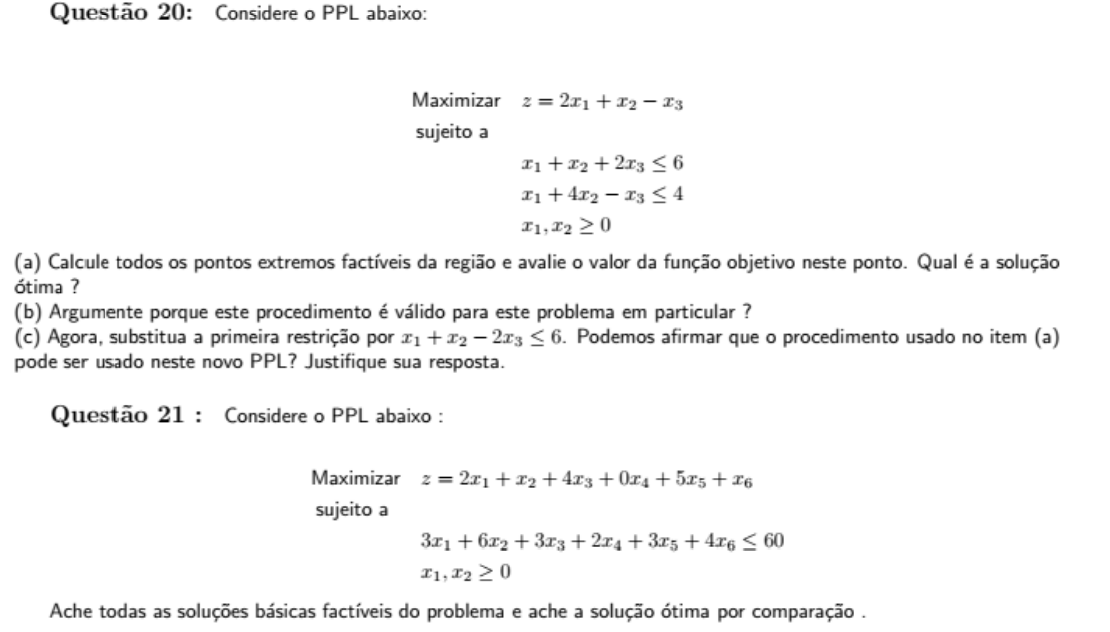
**Exercício (Parte 1) -** *Este problema consiste em atender a demanda de diferentes contratos a partir de várias filiais com o objetivo de minimizar os custos de entrega dos produtos. Os contratos exigem que o fornecimento seja bastante diversificado, indicando um número mínimo de filiais diferentes para seu atendimento, de forma a garantir o atendimento da demanda no caso de uma das filiais sofrerem uma queda de oferta. As capacidades de entrega de cada filial são conhecidas e para garantir a lucratividade, cada filial estabelece sua política de quantidade mínima para entrega. O objetivo é indicar o esquema de distribuição que, atendendo todas as restrições do problema, minimize o custo total de atendimento.*

**Código implementado:**

| *import numpy as np*  *import pulp*  *#Pela primeira tabela, teremos a demanda e o número mínimo de fornecedores:*  *demanda = [100, 65, 100, 70, 120, 60, 75, 100, 95, 85]*  *min\_fornecedores = [3, 2, 3, 2, 3, 2, 2, 3, 3, 3]*  *#Note que são aplicadas para 10 locais (Washington, Oregon, Califórnia, Idaho, Nevada, Montana, Wyoming, Arizona, Utah, Colorado)*  *#Pela segunda tabela, teremos a entrega mínima de cada equação é dada por:*  *ent\_min = [25, 30, 30, 35, 25]*  *#Note que são aplicadas para 5 cidades (Seattle, San Francisco, Las Vegas, Phoenix, Denver)*  *#Custos de cada uma das entregas*  *custo = [ #Destino*  *# 0 1 2 3 4 5 6 7 8 9*  *[10,15,10,15,20,20,20,40,10,30], #0*  *[30,15,10,20,10,20,20,30,20,30], #1*  *[20,10, 5,15,10,15,15,10, 5, 5], #2 Origem*  *[40,25,15,20,10,30,30,10,15,10], #3*  *[30,30,25,10, 5,35,35,15, 5,10] #4*  *]*  *#Nomes das origens*  *Origem = ["Seattle", "San Francisco", "Las Vegas", "Phoenix", "Denver"]*  *#Nomes dos destinos*  *Destino = ["Washington", "Oregon", "California", "Idaho", "Nevada", "Montana", "Wyoming", "Arizona", "Utah", "Colorado"]*  *# Definindo o problema como de minimização*  *prob = pulp.LpProblem('Exercício da Lista 4', pulp.LpMinimize)*  *x = pulp.LpVariable.dicts('Investimento ',[(i,j)for i in range(0,5) for j in range(0,10)] ,lowBound=0, cat='Continuous')*  *y = pulp.LpVariable.dicts('y',[(i,j)for i in range(0,5) for j in range(0,10)] ,lowBound=0, cat='Binary')*  *#Função à ser minimizada*  *prob += pulp.lpSum([custo[i][j] \* x[i,j] for i in range(0,5) for j in range(0,10)])*  *#Restrição à demanda dos destinos*  *for j in range(0,10):*  *prob += pulp.lpSum([x[i,j] for i in range(0,5)]) == demanda[j]*  *#Restrição ao numero mínimo de cada destino*  *for j in range(0,10):*  *prob += pulp.lpSum([y[i,j] for i in range(0,5)]) >= min\_fornecedores[j]*  *#Restrição à capacidade de cada origem*  *for i in range(0,5):*  *prob += pulp.lpSum([x[i,j] for j in range(0,10)]) <= 175*  *#Restrição à número mínimo de cada origem*  *for i in range(0,5):*  *for j in range(0,10):*  *prob += x[i,j] >= ent\_min[i] \* y[i,j]*  *#Restrição no relacionamento entre y e x*  *for j in range(0,10):*  *for i in range (0,5):*  *prob += x[i,j] <= 175 \* y[i,j]*  *#escrevendo o problema de otimização linear*  *print(prob)*  *# Resolvendo o problema*  *optimization\_result = prob.solve()*  *# Verificando se a solução ótima foi encontrada*  *assert optimization\_result == pulp.LpStatusOptimal*  *#mostrando o resultado*  *for i in range(0,5):*  *for j in range(0,10):*  *if y[i,j].varValue == 1:*  *print ("Investimento\_( ", Origem[i], " , ", Destino[j], " ) = ", x[i,j].varValue, " unidades ")*  ***RESULTADO:***  *Investimento\_( Seattle , Washington ) = 45.0 unidades*  *Investimento\_( Seattle , Idaho ) = 25.0 unidades*  *Investimento\_( Seattle , Montana ) = 25.0 unidades*  *Investimento\_( Seattle , Wyoming ) = 25.0 unidades*  *Investimento\_( Seattle , Utah ) = 25.0 unidades*  *Investimento\_( Seattle , Colorado ) = 25.0 unidades*  *Investimento\_( San Francisco , Oregon ) = 30.0 unidades*  *Investimento\_( San Francisco , California ) = 30.0 unidades*  *Investimento\_( San Francisco , Nevada ) = 30.0 unidades*  *Investimento\_( San Francisco , Montana ) = 35.0 unidades*  *Investimento\_( San Francisco , Wyoming ) = 50.0 unidades*  *Investimento\_( Las Vegas , Washington ) = 30.0 unidades*  *Investimento\_( Las Vegas , Oregon ) = 35.0 unidades*  *Investimento\_( Las Vegas , California ) = 35.0 unidades*  *Investimento\_( Las Vegas , Arizona ) = 30.0 unidades*  *Investimento\_( Las Vegas , Utah ) = 45.0 unidades*  *Investimento\_( Phoenix , California ) = 35.0 unidades*  *Investimento\_( Phoenix , Nevada ) = 60.0 unidades*  *Investimento\_( Phoenix , Arizona ) = 45.0 unidades*  *Investimento\_( Phoenix , Colorado ) = 35.0 unidades*  *Investimento\_( Denver , Washington ) = 25.0 unidades*  *Investimento\_( Denver , Idaho ) = 45.0 unidades*  *Investimento\_( Denver , Nevada ) = 30.0 unidades*  *Investimento\_( Denver , Arizona ) = 25.0 unidades*  *Investimento\_( Denver , Utah ) = 25.0 unidades*  *Investimento\_( Denver , Colorado ) = 25.0 unidades* |
| --- |

**Resposta: (**Ocultei o print do problema de otimização linear propositalmente, devido à quantidade excessiva de linhas e de informações sobre as 100 variáveis, contando x e y). Com ajuda do PulP do Python, conseguimos encontrar os valores ótimos para cada investimento e o quanto deve se dispor para cada um.

**PARTE 2**

**

**Questão 20 (Parte 2) -**

*Maximizar: z = 2x1 + x2 - x3*

*Sujeito à:*

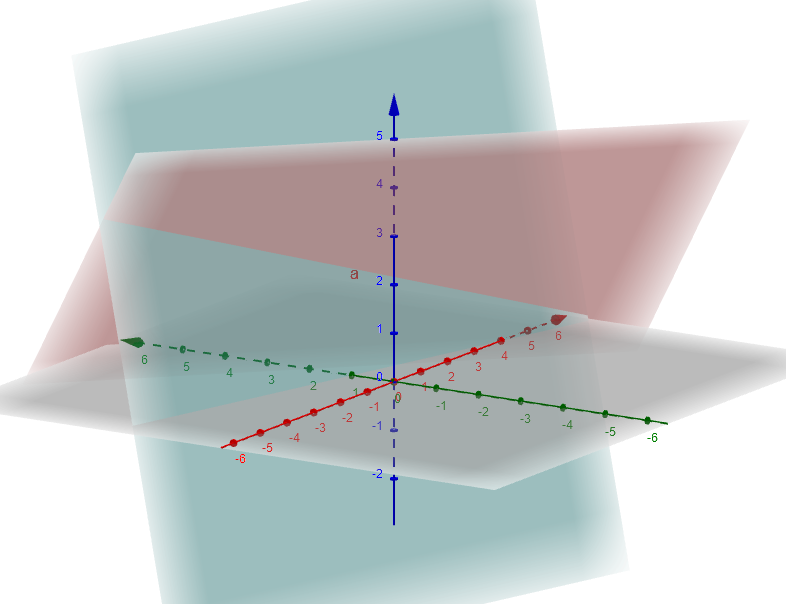
*x1 + x2 + 2x3 ≤ 6*

*x1 + 4x2 - x3 ≤ 4*

*x1 + x2 ≥ 0*

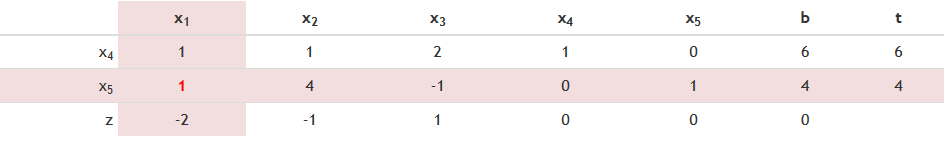
1. *Calcule todos os pontos extremos factíveis da região e avalie o valor da função objetivo neste ponto. Qual é a solução ótima?*

***Resposta:*** *Visualizando o gráfico das retas de restrição (Note que o problema está em R3), temos:*

**

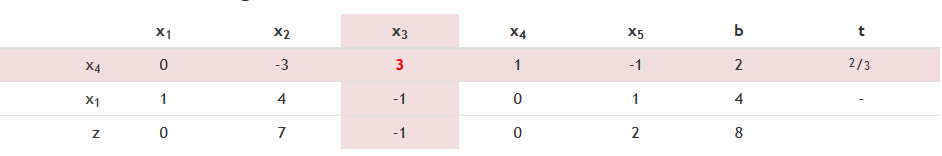
*Figura - Gráfico das Restrições.*

*Utilizando o método Simplex, teremos:*

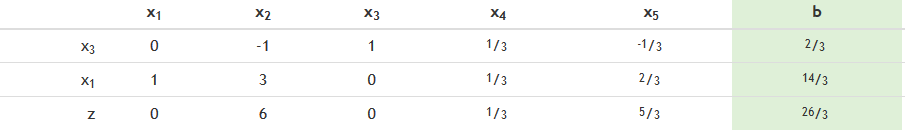
**

*Figura - Tabela Inicial do SIMPLEX.*

*Note que existe a variável x4 e x5 que são as variáveis de folga no algoritmo SIMPLEX.*

**

*Figura - Tabela após a obtenção do primeiro pivô.*

**

*Figura - Tabela de obtenção dos valores de z em função de x1 e x3 .*

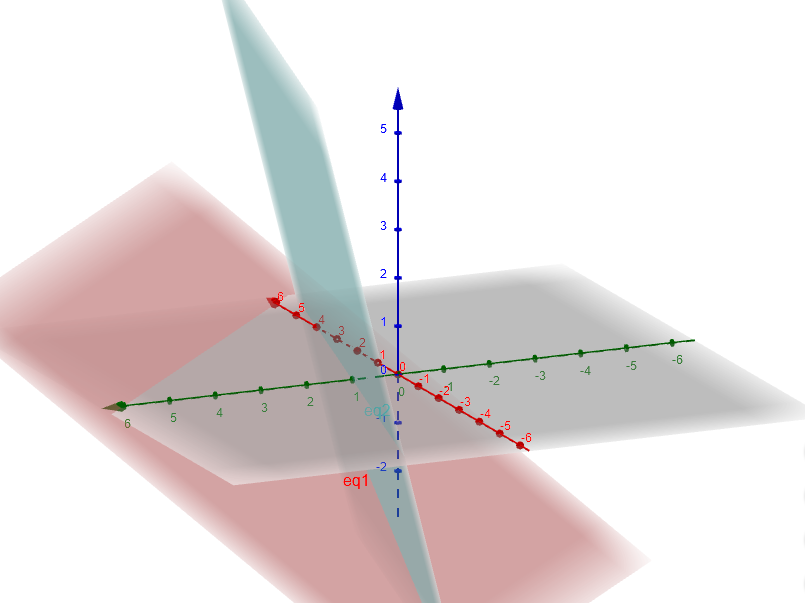
*Após rodar o algoritmo SIMPLEX, teremos como resultado os valores de x1 = ⅔ e x3 = 14/3, sendo que o valor solução de Z, sem contar os valores de folga em x4, x5 e até mesmo x2. Ou seja, o valor ótimo de Z(⅔, 0, 14/3) = 26/3*

1. *Argumente porque este procedimento é válido para este problema em particular ?*

***Resposta:*** *Contemplando o gráfico das restrições, temos que existe uma certa área delimitada, ou seja, existe uma reta como intersecção dos planos e o valor ótimo leva o valor do vértice (Pelo Teorema Fundamental da Programação Linear) contido nas intersecções com o plano y = 0 (Visualizando o método SIMPLEX aplicado).*

1. *Agora, substitua a primeira restrição por x1 + x2 - 2x3 ≤ 6. Podemos afirmar que o procedimento usado no item (a) pode ser usado neste novo PPL? Justifique sua resposta.*

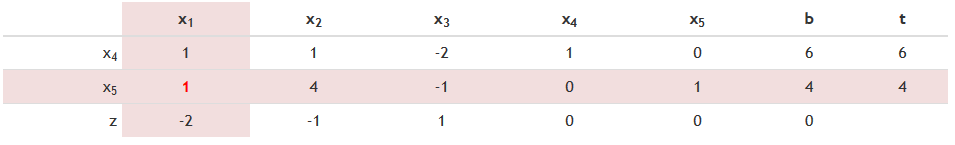
***Resposta:*** *Visualizando o gráfico das retas de restrição, temos:*

**

*Figura - Gráfico da Nova Restrição Dada na Letra c.*

*A reta de interseção entre os planos é dada por uma reta que não passa por x, y, z > 0. Ou seja, adquire intersecções apenas para valores negativos.*

*Aplicando SIMPLEX, teremos:*

**

*Figura - Tabela Inicial do SIMPLEX na letra c.*

*Fazendo a primeira operação de pivô, teremos em x3 uma tabela toda de -1, e quando temos uma variável nula ou negativa em todas as linhas, temos um problema de solução ilimitada. Um problema de solução ilimitada consiste em: À medida que os valores das variáveis aumentam, o valor Z também aumenta sem violar qualquer restrição.*

**Questão 20 (Parte 2) -**

*Maximizar: z = 2x1 + x2 + 4x3 + 0x4 + 5x5 + x6*

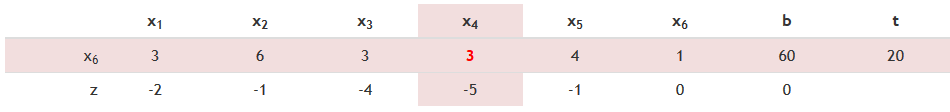
*Sujeito à:*

*3x1 + 6x2 + 3x3 + 2x4 + 3x5 + 4x6 ≤ 60*

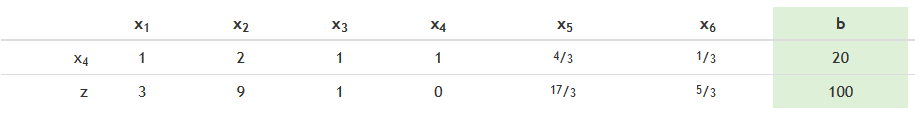
*x1 + x2 ≥ 0*

*Ache todas as soluções básicas factíveis do problema e ache a solução ótima por comparação.*

***Resposta:*** *Visualizando que o valor de x5 é o maior dos maximizadores, utilizaremos ele no processo do SIMPLEX:*

**

*Figura - Tabela Inicial do SIMPLEX*

**

*Figura - Tabela de obtenção dos valores de z em função de x4*

*Foram analisadas apenas 5 variáveis no SIMPLEX, visto que uma variável da função Z é 0x4, portanto ela não entra na função objetivo. E o valor ótimo por comparação é quando x5 (x4 no algoritmo) adquire valor de 20 e o Z se torna 100. Portanto, esse é o valor ótimo usando comparação e o algoritmo do SIMPLEX.*