**Atividade 3 - Otimização Linear**

**Entrega:** 09/09/2020

**PARTE 1**

**Exercício 01 (Lista 1) -** *Minimize o custo de uma ração composta por milho (M) e farelo de soja (FS), que custam respectivamente 0.26 reais e 0.32 reais o quilo. A ração deve ter no mínimo 0.34 kg de proteína e 2.64 kg de carboidratos. Cada quilo de milho contêm 0.07 kg de proteína e 0.82 kg de carboidratos, cada quilo de farelo de soja contêm 0,21 kg de proteína e 0,79 kg de carboidratos*

**Código implementado:**

| *import numpy as np*  *import pulp*  *#MIN = 0.26\*x1 + 0.32\*x2*  *#sj a:*  *#Reta r: 0.07\*x1 + 0.21\*x2 >= 0.34*  *#Reta s: 0.82\*x1 + 0.79\*x2 >= 2.64,*  *#sendo x1 = Milho e x2 = Farelo de Soja*  *# Definindo o problema como de maximização*  *prob = pulp.LpProblem('Exercício 1 da Lista 1', pulp.LpMinimize)*  *# Definindo as variáveis de decisão*  *x1 = pulp.LpVariable('Milho', lowBound=0, cat='Continuous')*  *x2 = pulp.LpVariable('Farelo de Soja', lowBound=0, cat='Continuous')*  *#Definindo a função a ser minimizada*  *MIN=0.26\*x1 + 0.32\*x2*  *#Adicionando a função-objetivo*  *prob += MIN*  *#Definindo a função da reta R*  *reta\_r = 0.07\*x1 + 0.21\*x2*  *#Adicionando a função da reta R nas restrições*  *prob += (reta\_r>=0.34)*  *#Definindo a função da reta S*  *reta\_s = 0.82\*x1 + 0.79\*x2*  *#Adicionando a função da reta S nas restrições*  *prob += (reta\_s>=2.64)*  *#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 var in (x1, x2):*  *print('Produção semanal ótima de {}: {:1.0f}'.format(var.name, var.value()))*  ***RESULTADO:***  *Exercício\_1\_da\_Lista\_1:*  *MINIMIZE*  *0.32\*Farelo\_de\_Soja + 0.26\*Milho + 0.0*  *SUBJECT TO*  *\_C1: 0.21 Farelo\_de\_Soja + 0.07 Milho >= 0.34*  *\_C2: 0.79 Farelo\_de\_Soja + 0.82 Milho >= 2.64*  *VARIABLES*  *Farelo\_de\_Soja Continuous*  *Milho Continuous*  *Produção semanal ótima de Milho: 2*  *Produção semanal ótima de Farelo\_de\_Soja: 1* |
| --- |

**Resposta:** O ponto que contém o valor ótimo é dado pelo x1 = 2 e x2 = 1, ou seja, 2 de Milho e 1 de Farelo de Soja.

**Exercício 02 (Lista 1) -** *Um investidor tem R$22000,00 para investir nos próximos 5 anos. No início de cada ano ele pode investir em depósitos de um ou dois anos. O banco paga 8% para o depósito de um ano e 17%(total) para depósito de dois anos. Além disso, há a possibilidade de investir em títulos a partir do segundo ano e que rendem após 3 anos 27% (total). Se o investidor reinveste seu dinheiro todo ano, formule o problema de modo a maximizar seu lucro total ao final de 5 anos.*

**Código implementado:**

| *import numpy as np*  *import pulp*  *#MAX = x51 + x42 + x33*  *#sj a:*  *#Ano 1:x11 + x12<=22000*  *#Ano 2:x21 +x22 + x23 <= 1.08\*x11*  *#Ano 3:x31 + x32 + x33 <= 1.08\*x21 + 1.17\*x12*  *#Ano 4:x41 + x42 <= 1.08x\*31 + 1.17\*x22*  *#Ano 5:x51 <= 1.08\*x41 + 1.17\*x32 + 1.27\*x23*  *# Definindo o problema como de maximização*  *prob = pulp.LpProblem('Exercício 2 da Lista 1', pulp.LpMaximize)*  *# Definindo as variáveis de decisão*  *x11 = pulp.LpVariable('Quantidade aplicada na aplicação 1 no ano 1', lowBound=0, cat='Continuous')*  *x21 = pulp.LpVariable('Quantidade aplicada na aplicação 1 no ano 2', lowBound=0, cat='Continuous')*  *x31 = pulp.LpVariable('Quantidade aplicada na aplicação 1 no ano 3', lowBound=0, cat='Continuous')*  *x41 = pulp.LpVariable('Quantidade aplicada na aplicação 1 no ano 4', lowBound=0, cat='Continuous')*  *x51 = pulp.LpVariable('Quantidade aplicada na aplicação 1 no ano 5', lowBound=0, cat='Continuous')*  *x12 = pulp.LpVariable('Quantidade aplicada na aplicação 2 no ano 1', lowBound=0, cat='Continuous')*  *x22 = pulp.LpVariable('Quantidade aplicada na aplicação 2 no ano 2', lowBound=0, cat='Continuous')*  *x32 = pulp.LpVariable('Quantidade aplicada na aplicação 2 no ano 3', lowBound=0, cat='Continuous')*  *x42 = pulp.LpVariable('Quantidade aplicada na aplicação 2 no ano 4', lowBound=0, cat='Continuous')*  *x23 = pulp.LpVariable('Quantidade aplicada na aplicação 3 no ano 2', lowBound=0, cat='Continuous')*  *x33 = pulp.LpVariable('Quantidade aplicada na aplicação 3 no ano 3', lowBound=0, cat='Continuous')*  *#Definindo a função a ser maximizada*  *MAX = x51 + x42 + x33*  *#Adicionando a função-objetivo*  *prob += MAX*  *#Definindo a função do Ano 1*  *ano1 = x11 + x12*  *#Adicionando a função do Ano 1 nas restrições*  *prob += (ano1<=22000)*  *#Definindo a função do Ano 2*  *ano2 = x21 +x22 + x23*  *#Adicionando a função do Ano 2 nas restrições*  *prob += (ano2<=1.08\*x11)*  *#Definindo a função do Ano 3*  *ano3 = x31 + x32 + x33*  *#Adicionando a função do Ano 3 nas restrições*  *prob += (ano3<=1.08\*x21 + 1.17\*x12)*  *#Definindo a função do Ano 4*  *ano4 = x41 + x42*  *#Adicionando a função do Ano 4 nas restrições*  *prob += (ano4<=1.08\*x31 + 1.17\*x22)*  *#Definindo a função do Ano 5*  *ano5 = x51*  *#Adicionando a função do Ano 5 nas restrições*  *prob += (ano5<=1.08\*x41 + 1.17\*x32 + 1.27\*x23)*  *#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 var in (x11, x21, x31, x41, x51, x12, x22, x32, x42, x23, x33):*  *print('Obtenção de lucro ótimo na {}: {:1.0f}'.format(var.name, var.value()))*  ***RESULTADO:***  *Exercício\_2\_da\_Lista\_1:*  *MAXIMIZE*  *1\*Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_5 + 1\*Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_4 + 1\*Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_3 + 0*  *SUBJECT TO*  *\_C1: Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_1*  *+ Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_1 <= 22000*  *\_C2: - 1.08 Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_1*  *+ Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_2*  *+ Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_2*  *+ Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_2 <= 0*  *\_C3: - 1.08 Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_2*  *+ Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_3*  *- 1.17 Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_1*  *+ Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_3*  *+ Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_3 <= 0*  *\_C4: - 1.08 Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_3*  *+ Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_4*  *- 1.17 Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_2*  *+ Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_4 <= 0*  *\_C5: - 1.08 Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_4*  *+ Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_5*  *- 1.17 Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_3*  *- 1.27 Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_2 <= 0*  *VARIABLES*  *Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_1 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_2 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_3 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_4 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_5 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_1 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_2 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_3 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_4 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_2 Continuous*  *Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_3 Continuous*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_1: 22000*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_2: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_3: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_4: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_1\_no\_ano\_5: 30175*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_1: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_2: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_3: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_2\_no\_ano\_4: 0*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_2: 23760*  *Obtenção de lucro ótimo na Quantidade\_aplicada\_na\_aplicação\_3\_no\_ano\_3: 0* |
| --- |

**Resposta:** Para uma produção ótima no sistema de equações, deve-se investir na aplicação 1 no primeiro ano, logo após, deve-se investir na aplicação 3 no ano 2 e após isso, deve-se aplicar na aplicação 1 no último e quinto ano. Assim, resultará em produção ótima de dinheiro numa sequência de aplicações

**Exercício 02 (Lista 2) -** *A pequena empresa de jardinagem Jardinaço planeja plantar flores nos parques da cidade de Macondo. Para isso pretende usar tulipas, rosas e girassóis em três tipos de layouts. O tipo 1 utiliza 30 tulipas, 20 rosas e 4 girassóis. O tipo 2 utiliza 10 tulipas, 40 rosas e 3 girassóis. O tipo 3 utiliza 20 tulipas, 50 rosas e 2 girassóis. A Jardinaço lucra 50,00 reais com cada layout do tipo 1, 30,00 reais com cada layout do tipo 2 e 60,00 reais com cada layout do tipo 3. A empresa possui 1000 tulipas, 800 rosas e 100 girassóis em estoque. Como a empresa pode maximizar seus lucros?*

**Código implementado:**

| *import numpy as np*  *import pulp*  *#MAX = 50\*x1 + 30\*x2 + 60\*x3*  *#sj a:*  *#Tulipas: 30\*x1 + 10\*x2 + 20\*x3 <= 1000*  *#Rosas: 20\*x1 + 40\*x2 + 50\*x3 <= 800*  *#Girassóis 3: 4\*x1 + 3\*x2 + 2\*x3 <= 100,*  *#sendo x1 = Layout do tipo 1, x2 = Layout do tipo 2 e x3 = Layout do tipo 3*  *# Definindo o problema como de maximização*  *prob = pulp.LpProblem('Exercício 2 da Lista 2', pulp.LpMaximize)*  *# Definindo as variáveis de decisão*  *x1 = pulp.LpVariable('Layout do Tipo 1', lowBound=0, cat='Continuous')*  *x2 = pulp.LpVariable('Layout do Tipo 2', lowBound=0, cat='Continuous')*  *x3 = pulp.LpVariable('Layout do Tipo 3', lowBound=0, cat='Continuous')*  *#Definindo a função a ser maximizada*  *MAX = 50\*x1 + 30\*x2 + 60\*x3*  *#Adicionando a função-objetivo*  *prob += MAX*  *#Definindo a função das tulipas*  *tulipas = 30\*x1 + 10\*x2 + 20\*x3*  *#Adicionando a função das tulipas nas restrições*  *prob += (tulipas<=100)*  *#Definindo a função das rosas*  *rosas = 20\*x1 + 40\*x2 + 50\*x3*  *#Adicionando a função das rosas nas restrições*  *prob += (rosas<=800)*  *#Definindo a função dos girassóis*  *girassois = 4\*x1 + 3\*x2 + 2\*x3*  *#Adicionando a função dos girassóis nas restrições*  *prob += (girassois<=100)*  *#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 var in (x1, x2, x3):*  *print('Produção ótima do {}: {:1.0f}'.format(var.name, var.value()))*  ***RESULTADO:***  *Exercício\_1\_da\_Lista\_1:*  *MAXIMIZE*  *50\*Layout\_do\_Tipo\_1 + 30\*Layout\_do\_Tipo\_2 + 60\*Layout\_do\_Tipo\_3 + 0*  *SUBJECT TO*  *\_C1: 30 Layout\_do\_Tipo\_1 + 10 Layout\_do\_Tipo\_2 + 20 Layout\_do\_Tipo\_3 <= 100*  *\_C2: 20 Layout\_do\_Tipo\_1 + 40 Layout\_do\_Tipo\_2 + 50 Layout\_do\_Tipo\_3 <= 800*  *\_C3: 4 Layout\_do\_Tipo\_1 + 3 Layout\_do\_Tipo\_2 + 2 Layout\_do\_Tipo\_3 <= 100*  *VARIABLES*  *Layout\_do\_Tipo\_1 Continuous*  *Layout\_do\_Tipo\_2 Continuous*  *Layout\_do\_Tipo\_3 Continuous*  *Produção ótima do Layout\_do\_Tipo\_1: 0*  *Produção ótima do Layout\_do\_Tipo\_2: 0*  *Produção ótima do Layout\_do\_Tipo\_3: 5* |
| --- |

**Resposta:** A obtenção da maximização de lucros se dá pela venda única e exclusiva do Layout do Tipo 3 de acordo com as restrições.

**Exercício 03 (Lista 2) -** *A cidade de Macondo produz 500 toneladas de lixo por dia, enquanto Dogvile produz 400 toneladas de lixo por dia. O lixo deve ser incinerado em dois incineradores, 1 e 2, e cada incinerador pode processar até 500 toneladas de lixo por dia. O custo por incineração do lixo é de 40 reais/ton no incinerador 1 e 30 reais/ton no 2. A incineração reduz cada tonelada de lixo à 0,2 toneladas de resíduos que devem ser armazenadas em dois aterros. Cada aterro pode receber até 200 toneladas de resíduos por dia. Para transportar uma tonelada de material, lixo ou resíduo, há um custo de 3,00 reais por quilômetro. As distâncias são mostradas na tabela 2. Como minimizar o custo total com a coleta e destino do lixo nas duas cidades?*

**Código implementado:**

| *import numpy as np*  *import pulp*  *#MIN = 40\*t1+30\*t2+40\*t3+30\*t4+40\*t5+30\*t6+40\*t7+30\*t8*  *#MIN = (35\*3\*0,2)\*t1+(10\*3\*0,2)\*t2+(38\*3\*0,2)\*t3+(13\*3\*0,2)\*t4+(45\*3\*0,2)\*t5+(51\*3\*0,2)\*t6+(48\*3\*0,2)\*t7+(48\*3\*0,2)\*t8*  *#sj a: t1+t2+t3+t4=500*  *# t5+t6+t7+t8=400*  *# t1+t3+t5+t7<=500*  *# t2+t4+t6+t8<=500*  *# t1+t2+t5+t6<=200*  *# t3+t4+t7+t8<=200*  *# t1,t2,t3,t4,t5,t6,t7,t8 >=0*  *# Definindo o problema como de maximização*  *prob = pulp.LpProblem('Exercício 3 da Lista 2', pulp.LpMinimize)*  *# Definindo as variáveis de decisão*  *t1 = pulp.LpVariable('Cidade 1 no incinerador 1 no aterro 1', lowBound=0, cat='Continuous')*  *t2 = pulp.LpVariable('Cidade 1 no incinerador 2 no aterro 1', lowBound=0, cat='Continuous')*  *t3 = pulp.LpVariable('Cidade 1 no incinerador 1 no aterro 2', lowBound=0, cat='Continuous')*  *t4 = pulp.LpVariable('Cidade 1 no incinerador 2 no aterro 2', lowBound=0, cat='Continuous')*  *t5 = pulp.LpVariable('Cidade 2 no incinerador 1 no aterro 1', lowBound=0, cat='Continuous')*  *t6 = pulp.LpVariable('Cidade 2 no incinerador 2 no aterro 1', lowBound=0, cat='Continuous')*  *t7 = pulp.LpVariable('Cidade 2 no incinerador 1 no aterro 2', lowBound=0, cat='Continuous')*  *t8 = pulp.LpVariable('Cidade 2 no incinerador 2 no aterro 2', lowBound=0, cat='Continuous')*  *#Definindo a função a ser minimizada*  *MIN = 61\*t1 + 36\*t2 + 62.8\*t3 + 37.8\*t4 + 67\*t5 + 60.6\*t6 + 68.8\*t7 + 58.8\*t8*  *#Adicionando a função-objetivo*  *prob += MAX*  *#Definindo a função da Soma 1*  *soma1 = t1+t2+t3+t4*  *#Adicionando a função da Soma 1 nas restrições*  *prob += (soma1==500)*  *#Definindo a função da Soma 2*  *soma2 = t5+t6+t7+t8*  *#Adicionando a função da Soma 2 nas restrições*  *prob += (soma2==400)*  *#Definindo a função da Restrição 1*  *rest1 = t1+t3+t5+t7*  *#Adicionando a função da Restrição 1 nas restrições*  *prob += (rest1<=500)*  *#Definindo a função da Restrição 2*  *rest2 = t2+t4+t6+t8*  *#Adicionando a função da Restrição 1 nas restrições*  *prob += (rest2<=500)*  *#Definindo a função da Restrição 2*  *rest3 = t1+t2+t5+t6*  *#Adicionando a função da Restrição 1 nas restrições*  *prob += (rest3<=200)*  *#Definindo a função da Restrição 2*  *rest4 = t3+t4+t7+t8*  *#Adicionando a função da Restrição 1 nas restrições*  *prob += (rest4<=200)*  *#escrevendo o problema de otimização linear*  *print(prob)*  *# Resolvendo o problema*  *optimization\_result = prob.solve()*  *#mostrando o resultado*  *for var in (t1, t2, t3, t4, t5, t6, t7, t8):*  *print('Pontos mínimos dos locais dados {}: {:1.0f}'.format(var.name, var.value()))*  ***RESULTADO:***  *SUBJECT TO*  *\_C1: Cidade\_1\_no\_incinerador\_1\_no\_aterro\_1*  *+ Cidade\_1\_no\_incinerador\_1\_no\_aterro\_2*  *+ Cidade\_1\_no\_incinerador\_2\_no\_aterro\_1*  *+ Cidade\_1\_no\_incinerador\_2\_no\_aterro\_2 = 500*  *\_C2: Cidade\_2\_no\_incinerador\_1\_no\_aterro\_1*  *+ Cidade\_2\_no\_incinerador\_1\_no\_aterro\_2*  *+ Cidade\_2\_no\_incinerador\_2\_no\_aterro\_1*  *+ Cidade\_2\_no\_incinerador\_2\_no\_aterro\_2 = 400*  *\_C3: Cidade\_1\_no\_incinerador\_1\_no\_aterro\_1*  *+ Cidade\_1\_no\_incinerador\_1\_no\_aterro\_2*  *+ Cidade\_2\_no\_incinerador\_1\_no\_aterro\_1*  *+ Cidade\_2\_no\_incinerador\_1\_no\_aterro\_2 <= 500*  *\_C4: Cidade\_1\_no\_incinerador\_2\_no\_aterro\_1*  *+ Cidade\_1\_no\_incinerador\_2\_no\_aterro\_2*  *+ Cidade\_2\_no\_incinerador\_2\_no\_aterro\_1*  *+ Cidade\_2\_no\_incinerador\_2\_no\_aterro\_2 <= 500*  *\_C5: Cidade\_1\_no\_incinerador\_1\_no\_aterro\_1*  *+ Cidade\_1\_no\_incinerador\_2\_no\_aterro\_1*  *+ Cidade\_2\_no\_incinerador\_1\_no\_aterro\_1*  *+ Cidade\_2\_no\_incinerador\_2\_no\_aterro\_1 <= 200*  *\_C6: Cidade\_1\_no\_incinerador\_1\_no\_aterro\_2*  *+ Cidade\_1\_no\_incinerador\_2\_no\_aterro\_2*  *+ Cidade\_2\_no\_incinerador\_1\_no\_aterro\_2*  *+ Cidade\_2\_no\_incinerador\_2\_no\_aterro\_2 <= 200*  *VARIABLES*  *Cidade\_1\_no\_incinerador\_1\_no\_aterro\_1 Continuous*  *Cidade\_1\_no\_incinerador\_1\_no\_aterro\_2 Continuous*  *Cidade\_1\_no\_incinerador\_2\_no\_aterro\_1 Continuous*  *Cidade\_1\_no\_incinerador\_2\_no\_aterro\_2 Continuous*  *Cidade\_2\_no\_incinerador\_1\_no\_aterro\_1 Continuous*  *Cidade\_2\_no\_incinerador\_1\_no\_aterro\_2 Continuous*  *Cidade\_2\_no\_incinerador\_2\_no\_aterro\_1 Continuous*  *Cidade\_2\_no\_incinerador\_2\_no\_aterro\_2 Continuous*  *Pontos mínimos dos locais dados Cidade\_1\_no\_incinerador\_1\_no\_aterro\_1: 0*  *Pontos mínimos dos locais dados Cidade\_1\_no\_incinerador\_2\_no\_aterro\_1: 200*  *Pontos mínimos dos locais dados Cidade\_1\_no\_incinerador\_1\_no\_aterro\_2: 0*  *Pontos mínimos dos locais dados Cidade\_1\_no\_incinerador\_2\_no\_aterro\_2: 0*  *Pontos mínimos dos locais dados Cidade\_2\_no\_incinerador\_1\_no\_aterro\_1: 0*  *Pontos mínimos dos locais dados Cidade\_2\_no\_incinerador\_2\_no\_aterro\_1: 0*  *Pontos mínimos dos locais dados Cidade\_2\_no\_incinerador\_1\_no\_aterro\_2: 200*  *Pontos mínimos dos locais dados Cidade\_2\_no\_incinerador\_2\_no\_aterro\_2: 0* |
| --- |

**Resposta:** Como não há uma quantidade suficiente de equações para a análise do processo de otimização e não há um resultado de acordo com as restrições, portanto não é o valor ótimo.