PO-202 - Programação Linear

Atividade 5

Grupo:

Gabriel Telles Missailidis

Rafael Silva de Oliveira

Samir Nunes da Silva

Professor:

Luiz Leduino Salles Neto

23/11/2022

Instituto Tecnológico de Aeronáutica – ITA

1 Problema 1

a)

Modelaremos o problema com 6 variáveis de decisão. Seja x_{ij} a quantidade de toneladas do ingrediente j usado na receita do doce i. Temos $i \in \{1, 2\}$ e $j \in \{1, 2, 3\}$, representando, respectivamente, açúcar, amêndoas e chocolate. As restrições do problema envolvem a quantidade limitada de quilos de ingredientes em estoque, e também restrições envolvendo a composição de cada doce. Por exemplo, a mistura do doce 1 deve conter pelo menos 20% de amêndoas,ou seja,

$$x_{11} >= 0,2(x_{11} + x_{12} + x_{13})$$

$$-0,2x_{11} + 0,8x_{12} - 0,2x_{13} >= 0$$

$$-x_{11} + 4x_{12} - x_{13} >= 0$$

Desenvolvimentos semelhantes foram feitos para as restrições de composição da mistura do doce 2. Note que, para a função objetivo, \$0,50 por kg é equivalente a \$500 por ton. Assim, chega-se na formulação do problema:

$$\begin{aligned}
Maximizar \quad & z = 500(x_{11} + x_{12} + x_{13}) + 400(x_{21} + x_{22} + x_{23}) \\
sujeito\ a: \quad & x_{11} + x_{21} \leq 10 \\
& x_{12} + x_{22} \leq 2 \\
& x_{13} + x_{23} \leq 3 \\
& -x_{11} + 4x_{12} - x_{13} \geq 0 \\
& -x_{21} + 9x_{22} - x_{23} \geq 0 \\
& -x_{21} - x_{22} + 9x_{23} \geq 0 \\
& x_{ij} >= 0, \forall i, j
\end{aligned}$$

Problema 1

In [2]:
```python
prob1 = LpProblem('P1', LpMaximize)
```

In [3]:
```python
x11 = pulp.LpVariable('x11', lowBound = 0, cat = 'Continuous') # Massa de açúcar no doce 1
x12 = pulp.LpVariable('x12', lowBound = 0, cat = 'Continuous') # Massa de amêndoas no doce 1
x13 = pulp.LpVariable('x13', lowBound = 0, cat = 'Continuous') # Massa de chocolate no doce 1

x21 = pulp.LpVariable('x21', lowBound = 0, cat = 'Continuous') # Massa de açúcar no doce 2
x22 = pulp.LpVariable('x22', lowBound = 0, cat = 'Continuous') # Massa de amêndoas no doce 2
x23 = pulp.LpVariable('x23', lowBound = 0, cat = 'Continuous') # Massa de chocolate no doce 2
```

In [4]:
```python
z = 500*(x11+x12+x13) + 400*(x21+x22+x23)
prob1 += z
```

In [5]:
```python
rest1 = x11 + x21
rest2 = x12 + x22
rest3 = x13 + x23
rest4 = -x11 + 4*x12 - x13
rest5 = -x21 + 9*x22 - x23
rest6 = -x21 - x22 + 9*x23

prob1 += (rest1 <= 10)
prob1 += (rest2 <= 2)
prob1 += (rest3 <= 3)
prob1 += (rest4 >= 0)
prob1 += (rest5 >= 0)
prob1 += (rest6 >= 0)
```

In [6]:
```python
optimization_result = prob1.solve()
```

In [7]:
```python
assert optimization_result == LpStatusOptimal
```

In [8]:
```python
for var in [x11,x12,x13,x21,x22,x23]:
    print(f'{var.name}:{var.value()}')
z = 0.5*(x11.value()+x12.value()+x13.value()) + 0.4*(x21.value()+x22.value()+x23.value())
print('z:{}'.format(z))
```

```
x11:4.0
x12:1.0
x13:0.0
x21:6.0
x22:1.0
x23:3.0
z:6.5
```

Figura 1: Modelagem e obtenção da solução ótima do problema 2 através do PuLP Python.

A resolução do problema de otimização linear foi feita através do PuLP em Python, conforme a Figura 1, da qual é possível extrair a solução ótima, com $x_{11} = 4$, $x_{12} = 1$, $x_{13} = 0$, $x_{21} = 6$, $x_{22} = 1$, $x_{23} = 3$, com lucro máximo $z_{max} = \$6500$. Interpretando as variáveis de decisão, percebe-se que a mistura do doce 1 foi de 80% açúcar, 20% amêndoas e 0% chocolate, enquanto a do doce 2 foi de 60% açúcar, 10% amêndoas e 30% chocolate. A solução ótima produz 5 toneladas de doce 1 e 10 toneladas de doce 2.

b)

Variando ligeiramente a quantidade de açúcar de 10 toneladas, foi obtida a Tabela 1. Note que para cada aumento (diminuição) de 0,1 ton de açúcar no estoque, o lucro total aumenta (diminui) em \$30. Adicionalmente, a cada 0,1 ton a mais no estoque, 0,08 ton deixavam de ir para a receita do doce 1 e 0,18 ton a mais iam para a receita do doce 2. Para respeitar as restrições de ingredientes, há semelhante movimento nos estoques de amêndoas, de modo que a cada 0,1 ton de açúcar adicionada, havia 0,02 ton de amêndoas que passavam da linha de produção do doce 1 e iam para o doce 2. Em todas as situações analisada, a distribuição do estoque de chocolate permaneceu a mesma, indo inteiramente para a receita do doce 2.

Açúcar	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	z
10,2	3,84	0,96	0	6,36	1,04	3	6560
10,1	3,92	0,98	0	6,18	1,02	3	6530
10	4,00	1,00	0	6,00	1,00	3	6500
9,9	4,08	1,02	0	5,82	0,98	3	6470
9,9	4,16	1,04	0	5,64	0,96	3	6440

Tabela 1: Análise de Sensibilidade para mudanças na quantidade estocada de açúcar afetando a solução ótima.

2 Problema 2

a)

Nutriente	1	2	3	4	5	NMD por nutriente
Vit. A	1	0	1	1	2	21
Vit. K	0	1	2	1	1	12
Custo (Centavos/Unid.)	20	20	31	11	12	

Figura 2: Cálculo do valor esperado do custo de 1 unidade de vitamina K no Júpiter Notebook.

Através da tabela de dieta disposta na Figura 2, modelou-se o problema de otimização linear através da biblioteca PuLP do Python, no Júpiter Notebook, conforme mostra a Figura 3. Nela, o conjunto de variáveis $\{x_1, x_2, x_3, x_4, x_5$ são números inteiros positivos que representam as quantidades de unidades de cada um dos alimentos, de 1 a 5. Por sua vez, as restrições foram obtidas da coluna "NMD por nutriente", que representa as necessidades mínimas diárias das vitaminas A e K. Por fim, pretende-se minimizar a função-objetivo z, que representa o custo total devido à compra dos alimentos de 1 a 5. O problema foi então modelado como segue:

$$
\begin{aligned}
&Minimizar && z = 20x_1 + 20x_2 + 31x_3 + 11x_4 + 12x_5 \\
&sujeito\ a: && x_1 + x_3 + x_4 + 2x_5 \geq 21 \\
& && x_2 + 2x_3 + x_4 + x_5 \geq 12 \\
& && x_1, x_2, x_3 \geq 0
\end{aligned}
$$

Problema 2

```
In [9]: prob2 = LpProblem('P2', LpMinimize)
```

```
In [10]: x1 = pulp.LpVariable('x1', lowBound = 0, cat = 'Integer')
         x2 = pulp.LpVariable('x2', lowBound = 0, cat = 'Integer')
         x3 = pulp.LpVariable('x3', lowBound = 0, cat = 'Integer')
         x4 = pulp.LpVariable('x4', lowBound = 0, cat = 'Integer')
         x5 = pulp.LpVariable('x5', lowBound = 0, cat = 'Integer')
```

```
In [11]: z = 20*x1 + 20*x2 + 31*x3 + 11*x4 + 12*x5
         prob2 += z
```

```
In [12]: rest1 = 1*x1 + 1*x3 + 1*x4 + 2*x5
         rest2 = 1*x2 + 2*x3 + 1*x4 + 1*x5

         prob2 += (rest1 >= 21)
         prob2 += (rest1 >= 12)
```

```
In [13]: optimization_result = prob2.solve()
```

```
In [14]: assert optimization_result == LpStatusOptimal
```

```
In [15]: for var in [x1, x2, x3, x4, x5]:
             print(f'{var.name}:{var.value()}')

         x1:0.0
         x2:0.0
         x3:0.0
         x4:1.0
         x5:10.0
```

Figura 3: Modelagem e obtenção da solução ótima do problema 2 através do PuLP Python.

Da Figura 3, nota-se que a solução ótima foi: $x_1 = x_2 = x_3 = 0$, $x_4 = 1$ e $x_5 = 10$, de tal forma que o custo mínimo é $z_{min} = 131$ centavos.

$\therefore$ **Resposta:** $(x_1,x_2,x_3,x_4,\ x_5) = (0,\ 0,\ 0,\ 1,\ 10)$, ou seja, deve-se comprar 1 unidade do alimento 4 e 10 unidades do alimento 5 para cumprir as necessidades diárias das vitaminas A e K com custo mínimo.

b)

```
In [2]: Ek = (20*0 + 1*20 + 2*31 + 1*11 + 1*12)/(0 + 1 + 2 + 1 + 1)
        Ek

Out[2]: 21.0
```

Figura 4: Cálculo do valor esperado do custo de 1 unidade de vitamina K no Júpiter Notebook.

Considere a Figura 4, que indica o cálculo do valor esperado E_k do custo de 1 unidade de vitamina K. A partir dela, vemos que $E_k = 21$ centavos/unidade. Logo, concluímos que sim, o preço de p = 12 centavos/unidade proposto é competitivo com os alimentos disponíveis para satisfazer a necessidade dessa vitamina por dois motivos:

1º) Como $p < E_k$, o valor proposto está abaixo do valor esperado para uma unidade de vitamina K, dados os valores disponíveis no mercado.

2º) Dado que a otimização para satisfazer as necessidades mínimas diárias dos nutrientes foi de 1 unidade do alimento 4 (que vale 11 centavos/unidade) e de 10 unidades do alimento 5 (que vale 12 centavos/unidade), espera-se que as pessoas que buscam menor preço na satisfação das necessidades mínimas diárias comprem uma unidade de vitamina K por $(1 * 11 + 1 * 12 * 10)/11 \approx 11{,}91$ centavos/unidade, que é muito próximo do valor proposto pelo farmacêutico.

3 Problema 3

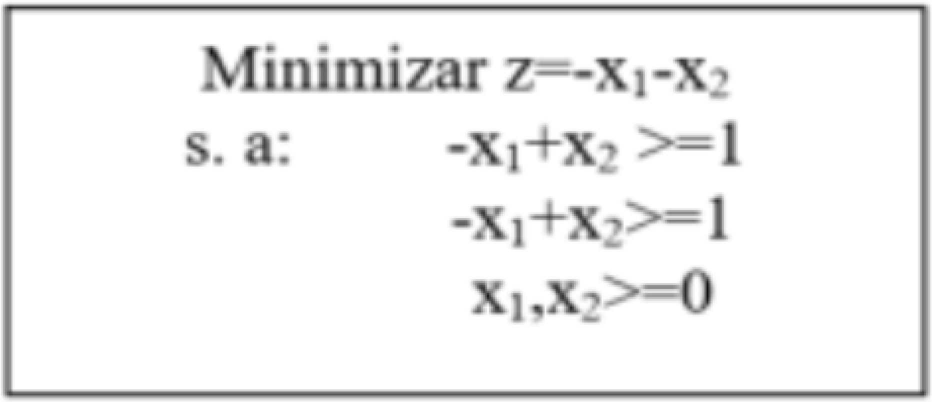
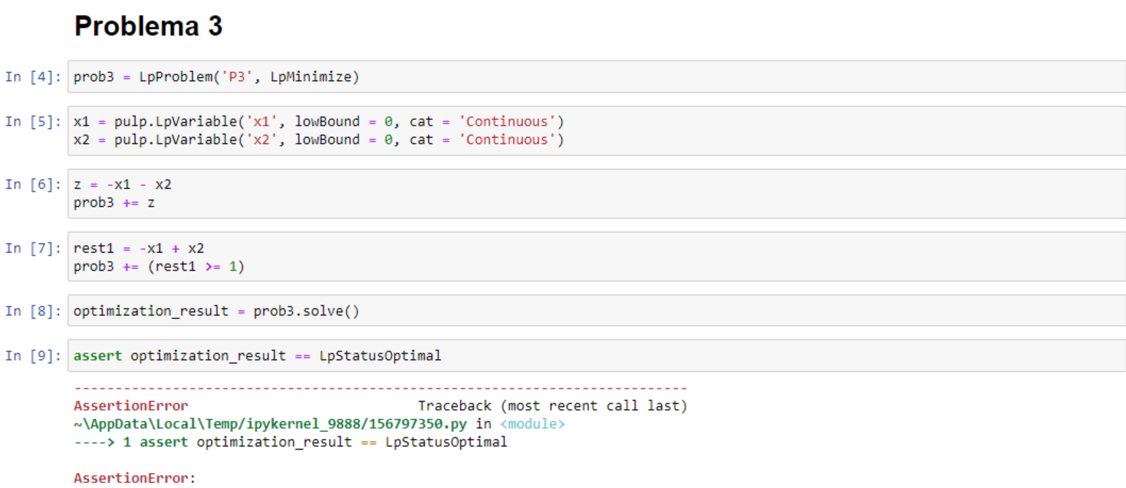


Figura 5: Problema original proposto.

Consideremos inicialmente o problema primal disposto na Figura 5, retirada diretamente do enunciado do problema 3. Nota-se que duas das restrições são idênticas, de tal forma que pode-se resumir o problema proposto conforme segue:

$$\begin{aligned} \textit{Minimizar} \quad & z = -x_1 - x_2 \\ \textit{sujeito a :} \quad & -x_1 + x_2 \geq 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$



Problema 3

```
In [4]: prob3 = LpProblem('P3', LpMinimize)

In [5]: x1 = pulp.LpVariable('x1', lowBound = 0, cat = 'Continuous')
        x2 = pulp.LpVariable('x2', lowBound = 0, cat = 'Continuous')

In [6]: z = -x1 - x2
        prob3 += z

In [7]: rest1 = -x1 + x2
        prob3 += (rest1 >= 1)

In [8]: optimization_result = prob3.solve()

In [9]: assert optimization_result == LpStatusOptimal

---------------------------------------------------------------------------
AssertionError                            Traceback (most recent call last)
~\AppData\Local\Temp/ipykernel_9888/156797350.py in <module>
----> 1 assert optimization_result == LpStatusOptimal

AssertionError:
```

Figura 6: Tentativa de resolução do primal no PuLP.

Resolvendo-se tal problema de minimização no PuLP, de acordo com a Figura 6, nota-se que a solução é ilimitada, ou seja, pode-se fazer x_2 tender ao infinito enquanto se mantém $x_1 = 0$, por exemplo. Dessa forma, o problema primal é de fato inviável. Utilizando-se o método Simplex, pode-se achar a tabela Simplex ótima, que é descrita pelo sistema a seguir, no qual x_3 é variável de excesso.

$$\begin{aligned} Maximizar \quad & z = -1 + 2x_1 + x_2 \\ sujeito\ a: \quad & -x_1 + x_2 - x_3 = 1 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Assim, o problema dual pode ser escrito como:

$$\begin{aligned} Minimizar \quad & w = y_1 - 1 \\ sujeito\ a: \quad & -y_1 \geq 2 \\ & y_1 \geq 1 \\ & -y_1 \geq 0 \end{aligned}$$

Simplificando-se o dual, vem:

$$\begin{aligned} Minimizar \quad & w = y_1 - 1 \\ sujeito\ a: \quad & y_1 \leq -2 \\ & y_1 \geq 1 \end{aligned}$$

Nota-se, pelas restrições, que o dual é de fato inviável, tal como o primal, pois não há y_1 que satisfaz o sistema, conforme queríamos demonstrar.

4 Problema 4

O problema foi modelado com duas variáveis de decisão: x_1 se refere à quantidade de cargas frágeis na seção pressurizada (tarifa R$3000 por tonelada) e x_2, à quantidade de cargas ordinárias na seção comum (tarifa R$1000 por tonelada). O objetivo é maximizar a receita, identificado como função z.

Entre as restrições, a primeira seção pode levar no máximo 6 toneladas, e a segunda, até 12. A carga total não pode super 16 toneladas, e a primeira seção deve ser menos de 2 toneladas mais pesada que metade da seção 2. Além disso, os valores das cargas devem ser, claro, positivos.

Portanto, o problema deve ser modelado como segue:

$$\begin{aligned} Maximizar \quad & z = 3000x_1 + 1000x_2 \\ sujeito\ a: \quad & x_1 \leq 6 \\ & x_2 \leq 12 \\ & x_1 + x_2 \leq 16 \\ & 2x_1 - x_2 \leq 4 \\ & x_i \geq 0, \forall i \end{aligned}$$

Portanto, devemos resolver pelo método gráfico. Impostas as condições, a região na qual a solução do problema se localiza é a entre os pontos pretos:

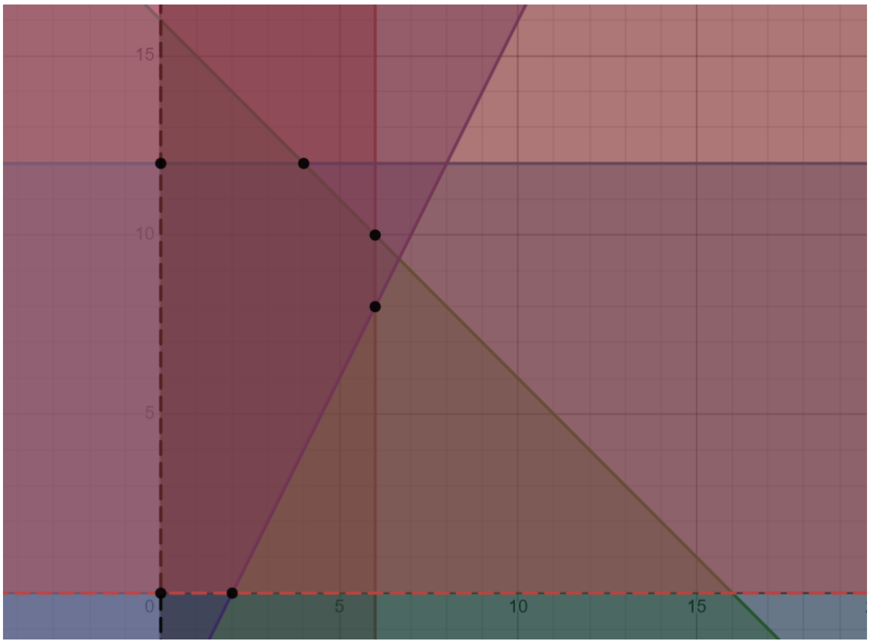


Figura 7: Modelagem da solução ótima do problema 4 através do Método Gráfico.

Para descobrir o frete máximo, devemos maximizar a função $z = 3000x_1 + 1000x_2$ de forma que ela passe pelo ponto mais distante da origem. Isso resulta no seguinte gráfico:

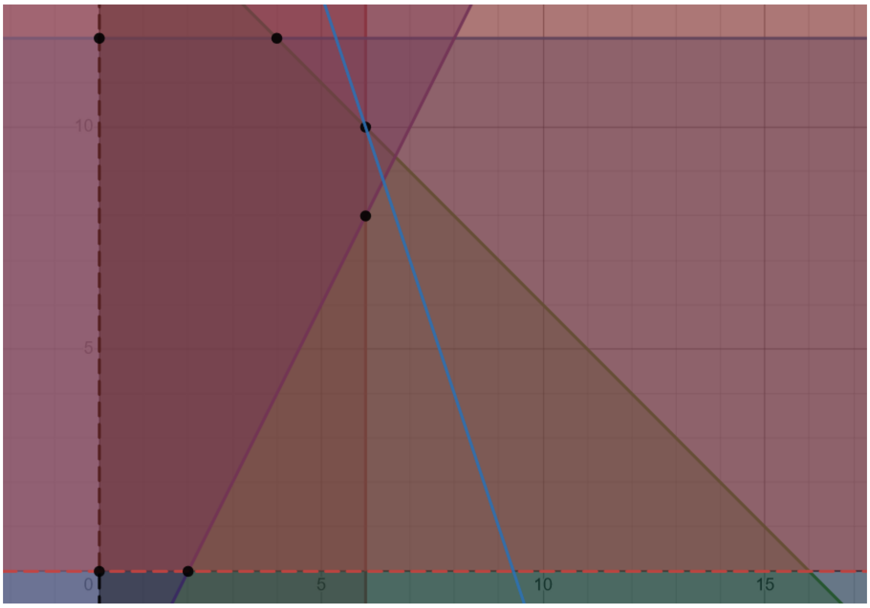


Figura 8: Obtenção da solução ótima do problema 4 através do Método Gráfico.

Portanto, a função que leva ao máximo frete é $3000x + 1000y = 28000$, ou seja, frete de R$28.000,00, que é obtido carregando 6 toneladas de cargas frágeis mais 10 de cargas ordinárias.

Com respeito aos preços-sombra, é possível encontrar 4, cada um associado a uma das quatro restrições descritas (a quinta e última é associada com a modelagem das variáveis, e não com as limitações do problema em si), ao aumentar uma unidade em cada restrição, isto é, equivalente ao ganho adicional com uma unidade a mais de esforço. Dessa forma, obtem-se como preços-sombra:

- Primeira restrição: novo máximo de R\$29.333,33; custo-sombra de $c_1 =$R\$1.333,33.
- Segunda restrição: novo máximo de R\$28.000,00; custo-sombra de $c_2 =$R\$0,00.
- Terceira restrição: novo máximo de R\$29.000,00; custo-sombra de $c_3 =$R\$1.000,00.
- Quarta restrição: novo máximo de R\$28.000,00; custo-sombra de $c_4 =$R\$0,00.

É possível ver que um acréscimo de esforço a mais nas segunda e quarta restrição não acarreta em aumento de receita. De fato, a solução ótima, que é de $x_1 = 6$ e $x_2 = 10$, já atende a $x_2 \leq 13$ e $x_1 - x_2/2 \leq 3$, ou seja, não é necessário modificar tais restrições para mais receita.

Agora, um acréscimo nas primeira e terceira restrições modificaria a solução ótima, sugerindo que há um custo escondido nessas variáveis.