

Pesquisa Operacional - ELE082

Prof. Eduardo Gontijo Carrano

Nome: Antônio Pedro de Amorim 1018010443

Questão 1 0%

A Viação Aérea Brasileira está estudando a compra de três tipos de aviões: Boeing 717 para as pontes aéreas de curta distância, Boeing 737-500 para vôos domésticos e internacionais de média distância e MD-11 para vôos internacionais de longa distância. Em um estudo preliminar, considerou-se que a capacidade máxima dos aviões a serem comprados será sempre preenchida para efeito de planejamento. Os dados de planejamento constam na Tabela 1.

Tabela 1: Dados operacionais dos aviões

Tipo de aviões	Custo ($\times 10^6$ US\$)	Receita ($\times 10^6$ US\$)	Pilotos aptos
Boeing 717	5,1	330	30
Boeing 737-500	3,6	300	20
MD-11	6,8	420	10

A verba disponível para as compras é de US\$ 220 milhões. Os pilotos de MD-11 podem pilotar todos os aviões da empresa, mas os demais pilotos só podem ser escalados às aeronaves a que foram habilitados. Cada aeronave necessita de dois pilotos para operar. As oficinas de manutenção têm capacidade equivalente à 40 Boeings 717. Em termos de esforço de manutenção, um Boeing 737-500 equivale a 3/4 de um Boeing 717 e um MD-11 equivale a 5/3 de um Boeing 717.

Escreva um modelo de programação linear inteira que maximiza o lucro obtido com os aviões adquiridos.

Resolva o problema resultante computacionalmente (Excel, OpenOffice, Matlab, Gurobi ou CPLEX).

Modelo:

$$L(x) = \text{Receita} - \text{Custo} \Rightarrow L(x) = 324.9x_1 + 296.4x_2 + 413.2x_3$$

Restrições:

$$\left\{ \begin{array}{l} x_1 \geq 15 \\ x_2 \geq 10 \\ x_3 \geq 0 \\ 5.1x_1 + 3.6x_2 + 6.8x_3 \leq 220 \\ x_1 + \frac{3}{4}x_2 + \frac{5}{3}x_3 \leq 40 \\ x_1 \leq 20 \\ x_2 \leq 15 \\ x_3 \leq 5 \end{array} \right.$$

Solução no Omega, cplex-python

$$B717 = 20$$

$$\rightarrow B737 = 15 \quad L = 13.01 \times 10^9 \text{ US\$}$$

$$MD11 = 5$$

Questão 2 0%

Uma indústria produz 2 produtos, I e II, sendo que cada produto consome um certo número de horas em 3 máquinas, A, B e C, conforme a Tabela 2.

Tabela 2: Consumo de horas máquina por produto

Produto	Tempo de máquina A	Tempo de máquina B	Tempo de máquina C
I	2 horas	1 horas	4 horas
II	2 horas	2 horas	2 horas

O tempo máximo de funcionamento semanal da máquina A é 160 horas, da máquina B é 120 horas e da máquina C é 280 horas. O lucro obtido por cada unidade do produto I é de R\$ 1,00 e do produto II é R\$ 1,50.

Escreva um modelo de otimização linear inteira que maximize o lucro obtido por esta empresa.

Resolva o problema resultante computacionalmente (Excel, OpenOffice, Matlab, Gurobi ou CPLEX).

x_1 : quantidade produto 1

x_2 : quantidade produto 2

$L(x)$ → função objetiva

→ Problema: $\max L(x) = 1 \cdot x_1 + 1.5 \cdot x_2$

→ Restrições

$$\begin{cases} 2x_1 + 2x_2 \leq 160 \\ x_1 + 2x_2 \leq 120 \\ 4x_1 + 2x_2 \leq 280 \\ x_1, x_2 \geq 0 \end{cases}$$

Resolução cplex - Python no enunciado

→ $\max L(40, 40) = 100$

Questão 3 0%

Um agricultor está planejando a estratégia de plantio de sua terra para o próximo ano. Por informações obtidas nos órgãos governamentais, ele sabe que as culturas de trigo, arroz e milho serão as mais rentáveis na próxima safra. Por experiência, ele sabe que a produtividade de sua terra para estas culturas é a constante na Tabela 3.

Tabela 3: Dados de plantio

Cultura	Produtividade (em kg por m^2)	Lucro (por kg de produção)
Trigo	0,2	10,80 centavos
Arroz	0,3	04,20 centavos
Milho	0,4	02,03 centavos

Por falta de um local de armazenamento próprio, a produção máxima está limitada a 60 toneladas. A área cultivável do sítio é de $200.000 m^2$. Para atender às demandas de seu próprio sítio, é imperativo que se plante ao menos $400 m^2$ de trigo, $800 m^2$ de arroz e $10.000 m^2$ de milho.

Escreva um modelo de programação linear que tem por objetivo maximizar o lucro do agricultor.

Resolva o problema modelado utilizando o Algoritmo Simplex.

$t, a, m \rightarrow$ trigo, arroz e milho respectivamente [Kg]

função objetivo: $L(t, a, m) = 10,8t + 4,20a + 2,03m$

$$\text{sujeito: } \max L = 10,8t + 4,20a + 2,03m$$

$$\left\{ \begin{array}{l} t/0,2 + a/0,3 + m/0,4 \leq 200000 \\ t + a + m \leq 60000 \\ t/0,2 \geq 400 \\ a/0,3 \geq 800 \\ m/0,4 \geq 10000 \end{array} \right.$$

Restrições

Detalhes utilizando tabela - Simplex no anexo (arquivo excel)

$$\max [L(t, a, m)] = 417800$$

$$[t \quad a \quad m] = [37840 \quad 240 \quad 4000]$$

Questão 4 0%

Utilizando a versão adequada do método Simplex, resolva o problema de programação linear abaixo:

$$\max \quad 0,5x_1 + 2x_2 \quad (1)$$

$$\text{sujeito a: } \begin{cases} 2x_1 + x_2 \leq 50 \\ x_1 \leq 20 \\ x_2 \leq 45 \\ x_1, x_2 \geq 0 \end{cases} \quad (2)$$

Forma padrão:

$$\begin{aligned} Z - 0,5x_1 - 2x_2 + F_1 + F_2 + F_3 &= 0 \\ 2x_1 + x_2 + F_1 &= 50 \\ x_1 - F_2 &= 20 \\ x_2 + F_3 &= 45 \end{aligned}$$

$$\left[\begin{array}{cccccc|c} 1 & -0,5 & -2 & 0 & 0 & 0 & 0 \\ 0 & 2 & 1 & 1 & 0 & 0 & 50 \\ 0 & 1 & 0 & 0 & 1 & 0 & 20 \\ 0 & 0 & 1 & 0 & 0 & 1 & 45 \end{array} \right] \left[\begin{array}{c} Z \\ x_1 \\ x_2 \\ F_1 \\ F_2 \\ F_3 \end{array} \right] = \left[\begin{array}{c} 0 \\ 50 \\ 20 \\ 45 \end{array} \right]$$

$$x_1, x_2, F_i \geq 0$$

Já temos Simplex com Python.

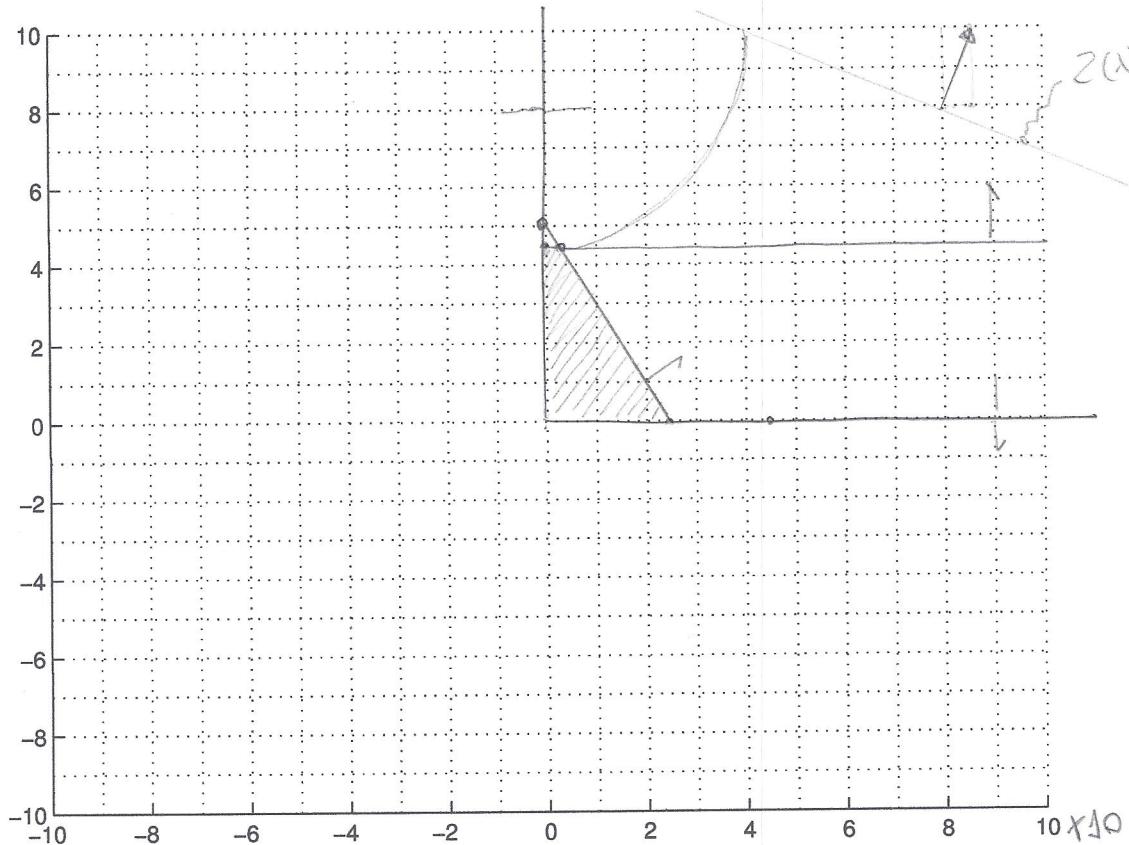
* notebook no anexo *

$$\begin{aligned} \rightarrow Z(x_1, x_2) &= 91,250 \\ \rightarrow (x_1, x_2) &= (25, 45, 0) \end{aligned}$$

Questão 5 0%

Para o problema da questão 4:

- Resolva este problema graficamente.
- Encontre a solução ótima dual correspondente utilizando Teorema das Folgas Complementares.
- Determine o intervalo de relaxação da primeira restrição que garante a manutenção da base ótima.



a) $\begin{aligned} z &= x_1 + 4x_2 \leq 50 \\ x_1 &\leq 5/2 \\ x_1 &\leq 25 \end{aligned}$ $[x_1 | x_2] = [2.5, 45]$

b) Formulacão do Problema Dual.

$$\min 50\lambda_1 + 20\lambda_2 + 45\lambda_3$$

$$\begin{cases} 2\lambda_1 + 1\lambda_2 + 0\lambda_3 \geq 0.5 \\ 1\lambda_1 + 0\lambda_2 + 1\lambda_3 \geq 2 \\ \lambda_1, \lambda_2, \lambda_3 \geq 0 \end{cases}$$

$$[\lambda_1 | \lambda_2 | \lambda_3] = [0 \quad 0 \quad 2]$$

$$\min g(\lambda) = 90$$

Questão 6 0%

Utilizando a versão adequada do método Simplex, resolva o problema de programação linear abaixo:

$$\min \quad 4x_1 + x_2 \quad (3)$$

sujeito a:
$$\begin{cases} 3x_1 + x_2 = 3 \\ 4x_1 + 3x_2 \geq 6 \\ x_1 + 2x_2 \leq 4 \\ x_1, x_2 \geq 0 \end{cases} \quad (4)$$

forme Padrão:

$$\min f(x) = -\max (-f(x)) \rightarrow -f(x) = -4x_1 - x_2$$

$$\left\{ \begin{array}{l} -Z + 4x_1 + x_2 = 0 \\ 3x_1 + x_3 = 3 \\ 4x_1 + 3x_2 + F_1 = 6 \\ x_1 + 2x_2 + F_2 = 4 \\ x_1, x_2 \geq 0 \end{array} \right.$$

2	x_1	x_2	F_1	F_2	b
-1	4	1	0	0	0
0	3	1	0	0	3
0	4	3	-1	0	6
0	1	2	0	1	4

2	x_1	x_2	F_1	F_2	b
-1	0	$-1/3$	0	0	-4
0	1	$1/3$	0	0	1
0	0	$5/3$	-1	0	2
0	0	$5/3$	0	1	3



$$[Z(x_1, x_2)] = [3, 6, (0, 6, 1, 2)]$$

$$x_1 = 0, 0$$

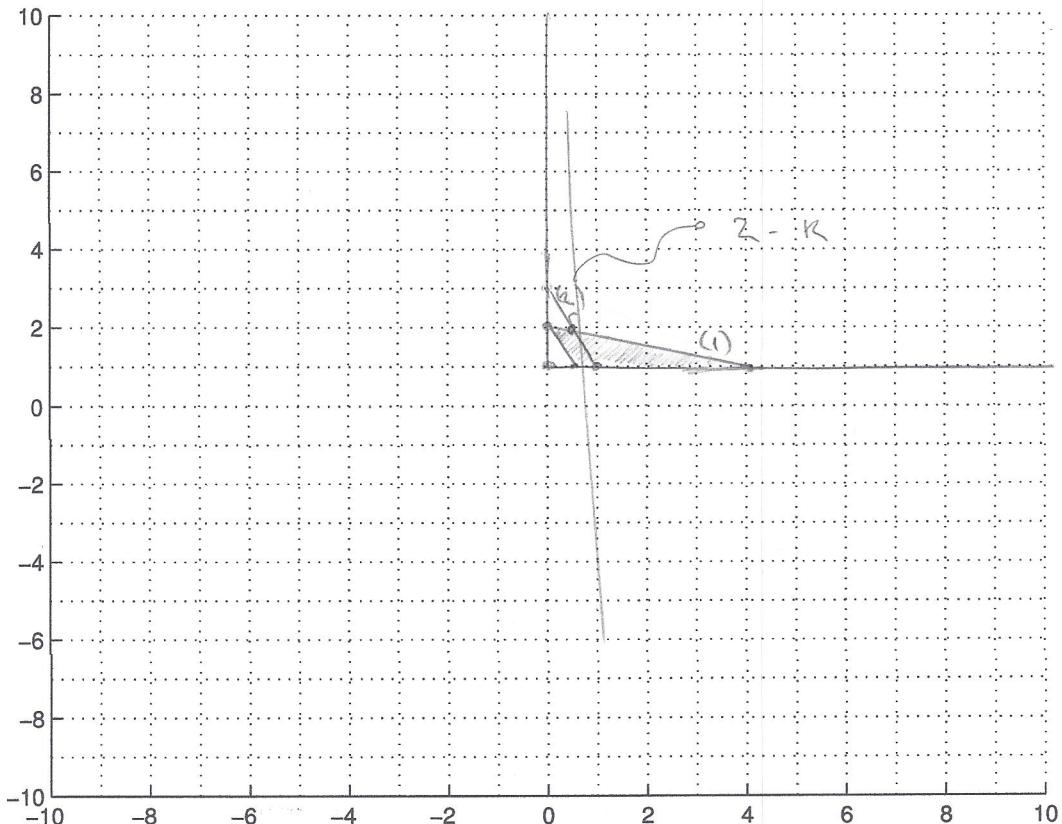
$$x_2 = 1, 2$$

2	x_1	x_2	F_1	F_2	b
-1	0	0	$-1/5$	0	$-18/5$
0	1	0	$1/5$	0	$3/5$
0	0	1	$-3/5$	0	$6/5$

Questão 7 0%

Para o problema da questão 6:

- Resolva este problema graficamente.
- Encontre a solução ótima dual correspondente utilizando Teorema das Folgas Complementares.
- Determine o intervalo de relaxação da terceira restrição que garante a manutenção da base ótima.



2)

$$S = \{(0,0)\}$$

$$3x_1 + 2x_2 = 3 \quad (-z)$$

$$x_1 + 2x_2 = 4$$

$$-5x_1 = -2$$

$$x_1 = 2/5 \Rightarrow x_1 = 0.4$$

$$0.4 + 2x_2 = 4 \Rightarrow x_2 = 1.8$$

$$\min f(x_1, x_2) = 4 \cdot 0.4 + 1.8$$

$$\min f(x_1, x_2) = 3.4$$

Questão 8 0%

Para o seguinte problema de programação linear:

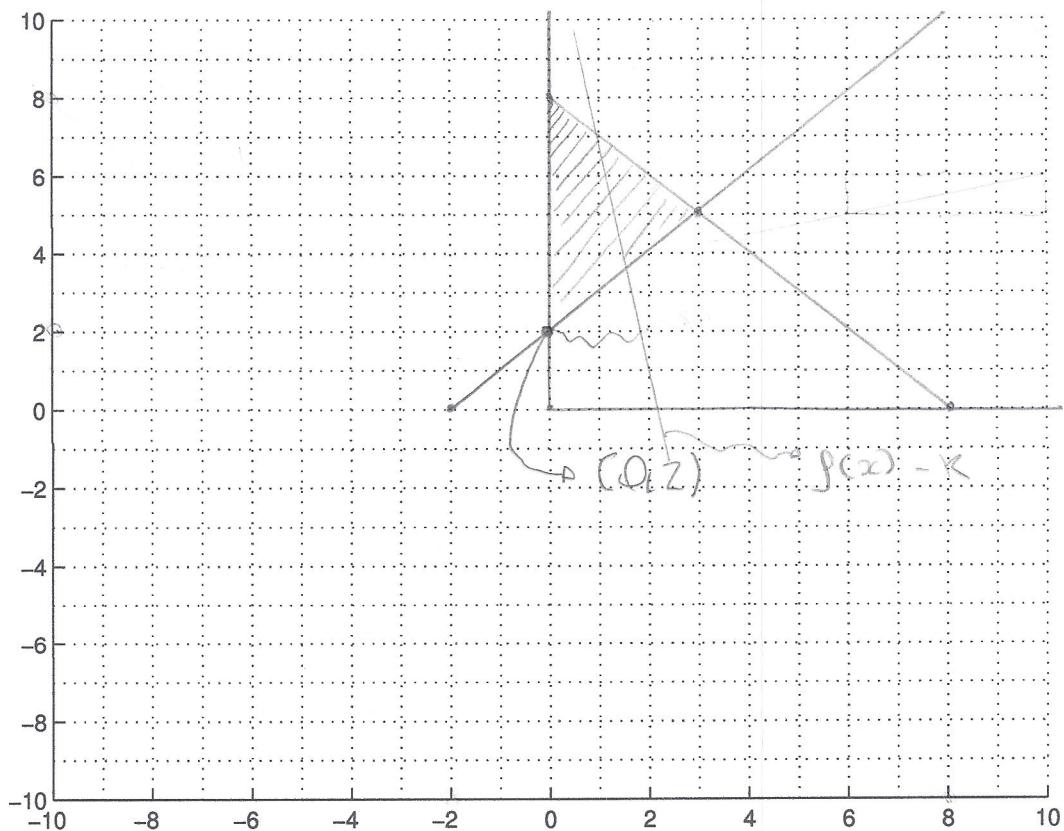
$$\min f(x_1, x_2) = 4x_1 + x_2$$

$$\left\{ \begin{array}{l} -x_1 + x_2 \geq 2 \\ x_1 + x_2 \leq 8 \\ x_1 \geq 0 \\ x_2 \geq 0 \end{array} \right.$$

**Forma Padrão*

$$\begin{aligned} Z - 4x_1 - x_2 &= 0 \\ -x_1 + x_2 - F_1 &= 2 \\ x_1 + x_2 + F_2 &= 8 \\ x_1, x_2 &\geq 0 \end{aligned}$$

- (a) Resolva este problema graficamente.
- (b) Represente este problema na forma padrão $\boxed{(0,2), 2}$
- (c) Quais os vértices presentes na região factível deste problema (soluções encontráveis pelo Simplex)? Encontre as bases associadas a estes vértices. $(0,0), (0,2), (3,5)$
- (d) Encontre o dual deste problema e a solução ótima dual correspondente utilizando as folgas complementares.
- (e) Suponha que seja construído um problema modificado onde a segunda restrição seja relaxada na forma $x_1 + x_2 \leq 8 + \delta$. Encontre os valores mínimo e máximo de δ que garantem que a base da solução ótima para este problema modificado seja a mesma da solução ótima para o problema original.



Questão 9 0%

Para o seguinte problema de programação linear:

$$\max f(x_1, x_2) = 5x_1 + 4x_2$$

$$\left\{ \begin{array}{l} 6x_1 + 4x_2 \leq 24 \\ x_1 + 2x_2 \leq 6 \\ -x_1 + x_2 \leq 1 \\ x_1, x_2 \geq 0 \end{array} \right.$$

(a) Resolva este problema graficamente. [21, (3, 3/2)]

(b) Represente este problema na forma padrão.

(c) Quais os vértices factíveis para este problema? Encontre as bases associadas a estes vértices.

(d) Encontre o dual deste problema e a solução ótima dual correspondente.

(e) Suponha que seja construído um problema modificado onde a segunda restrição seja relaxada na forma $x_1 + 2x_2 \leq 6 + \delta$. Encontre os valores mínimo e máximo de δ que garantem que a base da solução ótima para este problema modificado seja a mesma da solução ótima para o problema original.

Forma Padrão

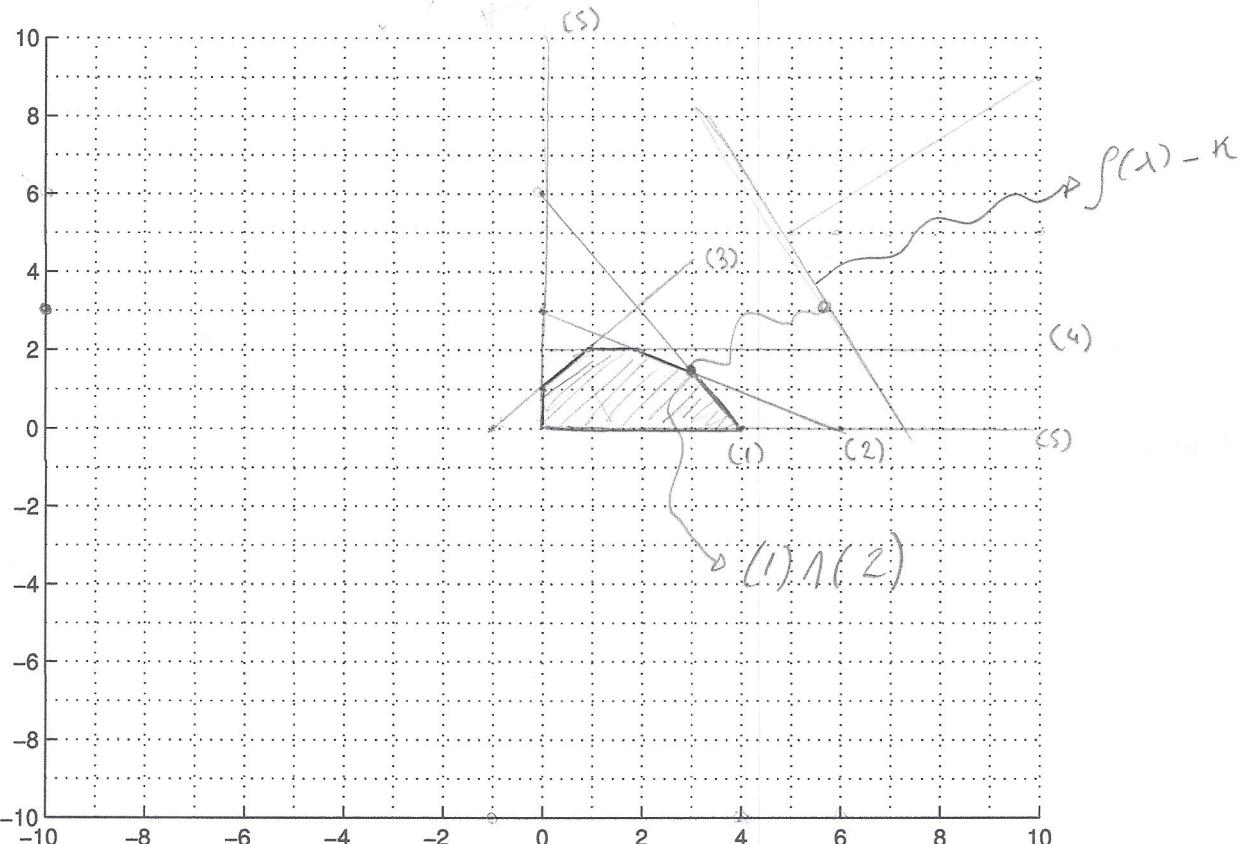
$$Z - 5x_1 - 4x_2 = 0$$

$$6x_1 + 4x_2 + F_1 = 24$$

$$x_1 + 2x_2 + F_2 = 6$$

$$-x_1 + x_2 + F_3 = 1$$

$$x_1, x_2 \geq 0$$



c) Vértices factíveis, $(0,0)$; $(0,1)$; $(4,0)$; $(1,2)$
 $(3,3/2)$; $(2,2)$

Questão 10 0%

A respeito das relações entre o Problema Primal e o Problema Dual, responda:

- (a) Supondo que o Problema Dual seja factível e sua solução seja ilimitada, o que se pode afirmar sobre o Problema Primal?
- (b) Supondo que o Problema Primal seja factível e tenha solução finita, o que se pode afirmar sobre o Problema Dual?

2) Significa que existe uma região de fecho para a solução primal.

b) Podemos obter a solução do problema dual verificando se é fechado a solução do problema primal é a solução ótima.

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model
```

```
In [2]: path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [3]: modelo = Model(name='Lista_1_Questao_1')
```

```
In [4]: A1 = modelo.integer_var(name='B717')
A2 = modelo.integer_var(name='B737')
A3 = modelo.integer_var(name='MD11')
```

```
In [5]: # Restrições
modelo.add_constraint(A1 >= 15)
modelo.add_constraint(A2 >= 10)
modelo.add_constraint(A3 >= 0)
modelo.add_constraint(5.1*A1 + 3.6*A2 + 6.8*A3 <= 220)
modelo.add_constraint(A1 + 3/4*A2 + 5/3*A3 <= 40)
modelo.add_constraint(A1 <= 20)
modelo.add_constraint(A2 <= 15)
modelo.add_constraint(A3 <= 5)
```

```
Out[5]: docplex.mp.LinearConstraint[](MD11, LE, 5)
```

```
In [6]: # Função Objetiva
modelo.maximize((330-5.1)*A1 + (300-3.6)*A2 + (420-6.8)*A3)
```

```
In [12]: print(330-5.1, 300-3.6, 420-6.8)
```

324.9 296.4 413.2

```
In [7]: modelo.print_information()
```

```
Model: Lista_1_Questao_1
- number of variables: 3
  - binary=0, integer=3, continuous=0
- number of constraints: 8
  - linear=8
- parameters: defaults
- objective: maximize
- problem type is: MILP
```

```
In [8]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 13010.000
B717=20
B737=15
MD11=5
```

```
In [9]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
```

CPXPARAM_Read_DataCheck	1
-------------------------	---

CPXPARAM_LPMETHOD	4
-------------------	---

Root node processing (before b&c):

Real time	=	0.00 sec. (0.00 ticks)
-----------	---	------------------------

Parallel b&c, 4 threads:

Real time	=	0.00 sec. (0.00 ticks)
-----------	---	------------------------

Sync time (average)	=	0.00 sec.
---------------------	---	-----------

Wait time (average)	=	0.00 sec.
---------------------	---	-----------

Total (root+branch&cut)	=	0.00 sec. (0.00 ticks)
-------------------------	---	------------------------

```
Out[9]: docplex.mp.solution.SolveSolution(obj=13010,values={B717:20,B737:15,MD11..
```

```
In [16]:
```

```
%notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_1.ipynb"
```

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [2]: modelo = Model(name='Lista_1_Questao_2')
```

```
In [4]: Q1 = modelo.continuous_var(name='Quantidade do Produto 1')
Q2 = modelo.continuous_var(name='Quantidade do Produto 2')
```

```
In [5]: # Restrições
modelo.add_constraint(Q1 >= 0)
modelo.add_constraint(Q2 >= 0)
modelo.add_constraint(2*Q1 + 2*Q2 <= 160)
modelo.add_constraint(1*Q1 + 2*Q2 <= 120)
modelo.add_constraint(4*Q1 + 2*Q2 <= 280)
```

```
Out[5]: docplex.mp.LinearConstraint[](4Quantidade do Produto 1+2Quantidade do Produto 2,LE,280)
```

```
In [6]: # Função Objetiva
modelo.maximize(1*Q1 + 1.5*Q2)
```

```
In [7]: modelo.print_information()
```

```
Model: Lista_1_Questao_2
- number of variables: 2
  - binary=0, integer=0, continuous=2
- number of constraints: 5
  - linear=5
- parameters: defaults
- objective: maximize
- problem type is: LP
```

```
In [8]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 100.000
"Quantidade do Produto 1"=40.000
"Quantidade do Produto 2"=40.000
```

```
In [9]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck 1
CPXPARAM_LPMMethod 4
Tried aggregator 1 time.
LP Presolve eliminated 2 rows and 0 columns.
Reduced LP has 3 rows, 2 columns, and 6 nonzeros.
Presolve time = 0.11 sec. (0.00 ticks)
Parallel mode: using up to 4 threads for barrier.
Number of nonzeros in lower triangle of A*A' = 3
Using Approximate Minimum Degree ordering
Total time for automatic ordering = 0.03 sec. (0.00 ticks)
```

Summary statistics for Cholesky factor:

Threads	= 4					
Rows in Factor	= 3					
Integer space required	= 3					
Total non-zeros in factor	= 6					
Total FP ops to factor	= 14					
Itn	Primal Obj	Dual Obj	Prim Inf	Upper Inf	Dual Inf	Inf Ratio
0	9.7777778e+01	0.0000000e+00	8.89e+00	0.00e+00	8.00e+00	1.00e+00
1	9.4372653e+01	7.1033966e+01	1.63e-01	0.00e+00	1.85e+00	4.00e+00
2	9.9466570e+01	1.0152618e+02	8.22e-02	0.00e+00	2.07e-01	4.95e+01
3	9.9999868e+01	1.0003041e+02	1.01e-03	0.00e+00	2.34e-03	6.25e+03
4	9.9999996e+01	9.9999999e+01	1.01e-07	0.00e+00	2.88e-07	4.24e+07
5	1.0000000e+02	1.0000000e+02	1.02e-11	0.00e+00	2.88e-11	4.09e+11
6	1.0000000e+02	1.0000000e+02	6.54e-14	0.00e+00	3.06e-15	4.09e+15

Barrier time = 0.16 sec. (0.01 ticks)

Parallel mode: deterministic, using up to 4 threads for concurrent optimization:

- * Starting dual Simplex on 1 thread...
- * Starting primal Simplex on 1 thread...

Dual crossover.

Dual: Fixed no variables.

Primal: Fixed no variables.

Dual simplex solved model.

Total crossover time = 0.09 sec. (0.00 ticks)

Total time on 4 threads = 0.36 sec. (0.01 ticks)

Out[9]: docplex.mp.solution.SolveSolution(obj=100,values={Quantidade do Produto ..

In [10]:

%notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_2.ipynb"

SOLUÇÃO QUESTAO 3

L(t, a, m)	t	a	m	x1	x2	x3	x4	x5	b
1	-10.80	-4.20	-2.03	0.00	0.00	0.00	0.00	0.00	0.00
0	5.00	3.33	2.50	1.00	0.00	0.00	0.00	0.00	200000.00
0	1.00	1.00	1.00	0.00	1.00	0.00	0.00	0.00	60000.00
0	5.00	0.00	0.00	0.00	0.00	-1.00	0.00	0.00	400.00
0	0.00	3.33	0.00	0.00	0.00	0.00	-1.00	0.00	800.00
0	0.00	0.00	2.50	0.00	0.00	0.00	0.00	-1.00	10000.00
1	0.00	-4.20	-2.03	0.00	0.00	-2.16	0.00	0.00	864.00
0	0.00	3.33	2.50	1.00	0.00	1.00	0.00	0.00	199600.00
0	0.00	1.00	1.00	0.00	1.00	0.20	0.00	0.00	59920.00
0	1.00	0.00	0.00	0.00	0.00	-0.20	0.00	0.00	80.00
0	0.00	3.33	0.00	0.00	0.00	0.00	-1.00	0.00	800.00
0	0.00	0.00	2.50	0.00	0.00	0.00	0.00	-1.00	10000.00
1	0	0	-2.03	0	0	-2.16	-1.26	0	1872
0	0	0	2.5	1	0	1	1	0	198800
0	0	0	1	0	1	0.2	0.3	0	59680
0	1	0	0	0	0	-0.2	0	0	80
0	0.00	1.00	0.00	0.00	0.00	0.00	-0.30	0.00	240.00
0	0	0	2.5	0	0	0	0	-1	10000
1	0	0	3.37	2.16	0	0	0.9	0	431280
0	0	0	2.5	1	0	1	1	0	198800
0	0	0	0.5	-0.2	1	0	0.1	0	19920
0	1	0	0.5	0.2	0	0	0.2	0	39840
0	0	1	0	0	0	0	-0.3	0	240
0	0	0	2.5	0	0	0	0	-1	10000

L(t, a, m)	trigo	arroz	milho	Variaveis de folga					Valor da variavel
1.00	0.00	0.00	0.00	2.16	0.00	0.00	0.90	1.35	417800.00
0.00	0.00	0.00	0.00	1.00	0.00	1.00	1.00	1.00	188800.00
0.00	0.00	0.00	0.00	-0.20	1.00	0.00	0.10	0.20	17920.00
0.00	1.00	0.00	0.00	0.20	0.00	0.00	0.20	0.20	37840.00
0.00	0.00	1.00	0.00	0.00	0.00	0.00	-0.30	0.00	240.00
0.00	0.00	0.00	1.00	0.00	0.00	0.00	0.00	-0.40	4000.00

```
In [2]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [3]: modelo = Model(name='Lista_1_Questao_3')
```

```
In [5]: trigo = modelo.continuous_var(name='Produção em kg de trigo')
arroz = modelo.continuous_var(name='Produção em kg de arroz')
milho = modelo.continuous_var(name='Produção em kg de milho')
```

```
In [6]: # Função Objetiva
modelo.maximize(10.8*trigo + 4.2*arroz + 2.03*milho)
```

```
In [7]: # Restrições
modelo.add_constraint(trigo >= 0)
modelo.add_constraint(arroz >= 0)
modelo.add_constraint(milho >= 0)
modelo.add_constraint(trigo/0.2 + arroz/0.3 + milho/0.4 <= 200000)
modelo.add_constraint(trigo + arroz + milho <= 60000)
modelo.add_constraint(trigo/0.2 >= 400)
modelo.add_constraint(arroz/0.3 >= 800)
modelo.add_constraint(milho/0.4 >= 10000)
```

```
Out[7]: docplex.mp.LinearConstraint[](2.500Produção em kg de milho,GE,10000)
```

```
In [8]: modelo.print_information()
```

```
Model: Lista_1_Questao_3
- number of variables: 3
  - binary=0, integer=0, continuous=3
- number of constraints: 8
  - linear=8
- parameters: defaults
- objective: maximize
- problem type is: LP
```

```
In [9]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 417800.000
"Produção em kg de trigo"=37840.000
"Produção em kg de arroz"=240.000
"Produção em kg de milho"=4000.000
```

```
In [10]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck 1
CPXPARAM_LPMMethod 4
Tried aggregator 1 time.
LP Presolve eliminated 6 rows and 0 columns.
Reduced LP has 2 rows, 3 columns, and 6 nonzeros.
```

```

Presolve time = 0.00 sec. (0.00 ticks)
Parallel mode: using up to 4 threads for barrier.
Number of nonzeros in lower triangle of A*A' = 1
Using Approximate Minimum Degree ordering
Total time for automatic ordering = 0.02 sec. (0.00 ticks)
Summary statistics for Cholesky factor:
  Threads          = 4
  Rows in Factor   = 2
  Integer space required = 2
  Total non-zeros in factor = 3
  Total FP ops to factor = 5
  Itn      Primal Obj      Dual Obj  Prim Inf Upper Inf  Dual Inf Inf Ratio
    0    3.2009655e+05 -5.0976000e+04  0.00e+00  0.00e+00  7.60e+01  1.00e+00
    1    3.4465240e+05  2.8762128e+05  0.00e+00  0.00e+00  1.07e+01  6.90e+00
    2    4.1628196e+05  4.1921548e+05  4.19e-09  0.00e+00  1.78e-01  4.30e+02
    3    4.1779954e+05  4.1779991e+05  3.10e-11  0.00e+00  2.82e-05  2.62e+06
    4    4.1780000e+05  4.1780000e+05  6.98e-11  0.00e+00  2.82e-09  2.40e+10
    5    4.1780000e+05  4.1780000e+05  4.65e-11  0.00e+00  2.81e-13  2.40e+14
Barrier time = 0.03 sec. (0.01 ticks)
Parallel mode: deterministic, using up to 4 threads for concurrent optimization:
  * Starting dual Simplex on 1 thread...
  * Starting primal Simplex on 1 thread...

```

Dual crossover.

Dual: Fixed no variables.
 Primal: Fixed no variables.

Dual simplex solved model.

Total crossover time = 0.03 sec. (0.00 ticks)

Total time on 4 threads = 0.06 sec. (0.01 ticks)

Out[10]: docplex.mp.solution.SolveSolution(obj=417800,values={Produção em kg de t..

In [11]: %notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_3.ipynb"

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [2]: modelo = Model(name='Lista_1_Questao_4')
```

```
In [3]: X1 = modelo.continuous_var(name='X1')
X2 = modelo.continuous_var(name='X2')
```

```
In [4]: # Função Objetiva
modelo.maximize(0.5*X1 + 2*X2)
```

```
In [6]: # Restrições
modelo.add_constraint(X1 >= 0)
modelo.add_constraint(X2 >= 0)
modelo.add_constraint(X1 <= 20)
modelo.add_constraint(X2 <= 45)
modelo.add_constraint(2*X1 + X2 <= 50)
```

```
Out[6]: docplex.mp.LinearConstraint[](2X1+X2,LE,50)
```

```
In [7]: modelo.print_information()
```

```
Model: Lista_1_Questao_4
- number of variables: 2
  - binary=0, integer=0, continuous=2
- number of constraints: 5
  - linear=5
- parameters: defaults
- objective: maximize
- problem type is: LP
```

```
In [8]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 91.250
X1=2.500
X2=45.000
```

```
In [14]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck          1
CPXPARAM_LPMETHOD                4
Tried aggregator 1 time.
LP Presolve eliminated 4 rows and 0 columns.
Reduced LP has 1 rows, 2 columns, and 2 nonzeros.
Presolve time = 0.00 sec. (0.00 ticks)
Parallel mode: using up to 4 threads for barrier.
Number of nonzeros in lower triangle of A*A' = 0
Using Approximate Minimum Degree ordering
Total time for automatic ordering = 0.00 sec. (0.00 ticks)
```

Summary statistics for Cholesky factor:

Threads	= 4					
Rows in Factor	= 1					
Integer space required	= 1					
Total non-zeros in factor	= 1					
Total FP ops to factor	= 1					
Itn	Primal Obj	Dual Obj	Prim Inf	Upper Inf	Dual Inf	Inf Ratio
0	1.2750000e+02	2.9500000e+02	1.54e+02	1.39e+02	3.00e+00	1.00e+00
1	9.3939167e+01	9.4801275e+01	1.40e+01	1.26e+01	2.73e-01	6.89e-02
2	9.2060072e+01	9.2396839e+01	1.23e+00	1.11e+00	8.15e-03	1.51e+03
3	9.1202725e+01	9.1252913e+01	2.69e-03	2.43e-03	3.59e-04	1.65e+03
4	9.1249997e+01	9.1250003e+01	3.63e-07	3.28e-07	3.58e-08	1.13e+06
5	9.1250000e+01	9.1250000e+01	3.70e-11	3.34e-11	3.58e-12	8.49e+09
6	9.1250000e+01	9.1250000e+01	0.00e+00	7.25e-15	4.23e-16	8.42e+13

Barrier time = 0.05 sec. (0.01 ticks)

Parallel mode: deterministic, using up to 4 threads for concurrent optimization:

- * Starting dual Simplex on 1 thread...
- * Starting primal Simplex on 1 thread...

Dual crossover.

Dual: Fixed no variables.

Primal: Fixed no variables.

Dual simplex solved model.

Total crossover time = 0.02 sec. (0.00 ticks)

Total time on 4 threads = 0.06 sec. (0.01 ticks)

Out[14]: docplex.mp.solution.SolveSolution(obj=91.25,values={X1:2.5,X2:45})

In [11]:

%notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_4.ipynb"

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [2]: modelo = Model(name='Lista_1_Questao_5')
```

```
In [3]: X1 = modelo.continuous_var(name='X1')
X2 = modelo.continuous_var(name='X2')
```

```
In [4]: # Função Objetiva
modelo.maximize(0.5*X1 + 2*X2)
```

```
In [5]: # Restrições
modelo.add_constraint(X1 >= 0)
modelo.add_constraint(X2 >= 0)
modelo.add_constraint(X1 <= 20)
modelo.add_constraint(X2 <= 45)
modelo.add_constraint(2*X1 + X2 <= 50)
```

```
Out[5]: docplex.mp.LinearConstraint[](2X1+X2,LE,50)
```

```
In [6]: modelo.print_information()
```

```
Model: Lista_1_Questao_5
- number of variables: 2
  - binary=0, integer=0, continuous=2
- number of constraints: 5
  - linear=5
- parameters: defaults
- objective: maximize
- problem type is: LP
```

```
In [7]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 91.250
X1=2.500
X2=45.000
```

```
In [8]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck          1
CPXPARAM_LPMETHOD                4
Tried aggregator 1 time.
LP Presolve eliminated 4 rows and 0 columns.
Reduced LP has 1 rows, 2 columns, and 2 nonzeros.
Presolve time = 0.01 sec. (0.00 ticks)
Parallel mode: using up to 4 threads for barrier.
Number of nonzeros in lower triangle of A*A' = 0
Using Approximate Minimum Degree ordering
Total time for automatic ordering = 0.00 sec. (0.00 ticks)
```

Summary statistics for Cholesky factor:

Threads	= 4					
Rows in Factor	= 1					
Integer space required	= 1					
Total non-zeros in factor	= 1					
Total FP ops to factor	= 1					
Itn	Primal Obj	Dual Obj	Prim Inf	Upper Inf	Dual Inf	Inf Ratio
0	1.2750000e+02	2.9500000e+02	1.54e+02	1.39e+02	3.00e+00	1.00e+00
1	9.3939167e+01	9.4801275e+01	1.40e+01	1.26e+01	2.73e-01	6.89e-02
2	9.2060072e+01	9.2396839e+01	1.23e+00	1.11e+00	8.15e-03	1.51e+03
3	9.1202725e+01	9.1252913e+01	2.69e-03	2.43e-03	3.59e-04	1.65e+03
4	9.1249997e+01	9.1250003e+01	3.63e-07	3.28e-07	3.58e-08	1.13e+06
5	9.1250000e+01	9.1250000e+01	3.70e-11	3.34e-11	3.58e-12	8.49e+09
6	9.1250000e+01	9.1250000e+01	0.00e+00	7.25e-15	4.23e-16	8.42e+13

Barrier time = 0.03 sec. (0.01 ticks)

Parallel mode: deterministic, using up to 4 threads for concurrent optimization:

- * Starting dual Simplex on 1 thread...
- * Starting primal Simplex on 1 thread...

Dual crossover.

Dual: Fixed no variables.
Primal: Fixed no variables.

Dual simplex solved model.

Total crossover time = 0.03 sec. (0.00 ticks)

Total time on 4 threads = 0.08 sec. (0.01 ticks)

Out[8]: docplex.mp.solution.SolveSolution(obj=91.25,values={X1:2.5,X2:45})

In [8]: modelo.print_solution()

```
objective: 91.250
X1=2.500
X2=45.000
```

In [9]: %notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_5.ipynb"

In [10]: dual = Model(name='Lista_1_Questao_5_dual')

```
L1 = dual.continuous_var(name='Lambda_1')
L2 = dual.continuous_var(name='Lambda_2')
L3 = dual.continuous_var(name='Lambda_3')

dual.minimize(50*L1 + 20*L2 + 45*L3)

dual.add_constraint(L1 >= 0)
dual.add_constraint(L2 >= 0)
dual.add_constraint(L3 >= 0)
dual.add_constraint(2*L1 + L2 >= 0.0)
dual.add_constraint(L1 + L3 >= 2)

otimizacaoDual = dual.solve()
dual.print_solution()
```

```
objective: 90.000
Lambda_3=2.000
```

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [2]: modelo = Model(name='Lista_1_Questao_6')
```

```
In [3]: X1 = modelo.continuous_var(name='X1')
X2 = modelo.continuous_var(name='X2')
```

```
In [4]: # Função Objetiva
modelo.minimize(4*X1 + X2)
```

```
In [5]: # Restrições
modelo.add_constraint(X1 >= 0)
modelo.add_constraint(X2 >= 0)
modelo.add_constraint(3*X1 + X2 <= 3)
modelo.add_constraint(3*X1 + X2 >= 3)
modelo.add_constraint(4*X1 + 3*X2 >= 6)
modelo.add_constraint(X1 + 2*X2 <= 4)
```

```
Out[5]: docplex.mp.LinearConstraint[](X1+2X2,LE,4)
```

```
In [6]: modelo.print_information()
```

```
Model: Lista_1_Questao_6
- number of variables: 2
  - binary=0, integer=0, continuous=2
- number of constraints: 6
  - linear=6
- parameters: defaults
- objective: minimize
- problem type is: LP
```

```
In [7]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 3.400
X1=0.400
X2=1.800
```

```
In [8]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck           1
CPXPARAM_LPMETHOD                4
Tried aggregator 1 time.
LP Presolve eliminated 5 rows and 1 columns.
Aggregator did 1 substitutions.
All rows and columns eliminated.
Presolve time = 0.00 sec. (0.00 ticks)
Parallel mode: deterministic, using up to 4 threads for concurrent optimization:
* Starting dual Simplex on 1 thread...
```

```
* Starting primal Simplex on 1 thread...
```

```
Dual simplex solved model.
```

```
Total time on 4 threads = 0.02 sec. (0.01 ticks)
```

```
Out[8]: docplex.mp.solution.SolveSolution(obj=3.4,values={X1:0.4,X2:1.8})
```

```
In [9]: %notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_6.ipynb"
```

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [2]: modelo = Model(name='Lista_1_Questao_8')
```

```
In [3]: X1 = modelo.continuous_var(name='X1')
X2 = modelo.continuous_var(name='X2')
```

```
In [5]: # Função Objetiva
modelo.minimize(4*X1 + 1*X2)
```

```
In [6]: # Restrições
modelo.add_constraint(X1 >= 0)
modelo.add_constraint(X2 >= 0)
modelo.add_constraint(-X1 + X2 >= 2)
modelo.add_constraint(X1 + X2 <= 8)
```

```
Out[6]: docplex.mp.LinearConstraint[](X1+X2,LE,8)
```

```
In [7]: modelo.print_information()
```

```
Model: Lista_1_Questao_8
- number of variables: 2
  - binary=0, integer=0, continuous=2
- number of constraints: 4
  - linear=4
- parameters: defaults
- objective: minimize
- problem type is: LP
```

```
In [8]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 2.000
X2=2.000
```

```
In [9]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck                                1
CPXPARAM_LPMMethod                                     4
Tried aggregator 1 time.
LP Presolve eliminated 4 rows and 2 columns.
All rows and columns eliminated.
Presolve time = 0.00 sec. (0.00 ticks)
Parallel mode: deterministic, using up to 4 threads for concurrent optimization:
  * Starting dual Simplex on 1 thread...
  * Starting primal Simplex on 1 thread...
```

Dual simplex solved model.

```
Total time on 4 threads = 0.02 sec. (0.00 ticks)
Out[9]: docplex.mp.solution.SolveSolution(obj=2,values={X2:2})
```

```
In [10]: %notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_8.ipynb"
```

```
In [1]: import sys, os
import docplex.mp
from docplex.mp.model import Model

path = 'D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1'
os.chdir(path)
```

```
In [2]: modelo = Model(name='Lista_1_Questao_9')
```

```
In [3]: X1 = modelo.continuous_var(name='X1')
X2 = modelo.continuous_var(name='X2')
```

```
In [4]: # Função Objetiva
modelo.maximize(5*X1 + 4*X2)
```

```
In [5]: # Restrições
modelo.add_constraint(X1 >= 0)
modelo.add_constraint(X2 >= 0)
modelo.add_constraint(+6*X1 + 4*X2 <= 24)
modelo.add_constraint(+1*X1 + 2*X2 <= 6)
modelo.add_constraint(-1*X1 + 1*X2 <= 1)
modelo.add_constraint(X2 <= 2)
```

```
Out[5]: docplex.mp.LinearConstraint[](X2,LE,2)
```

```
In [6]: modelo.print_information()
```

```
Model: Lista_1_Questao_9
- number of variables: 2
  - binary=0, integer=0, continuous=2
- number of constraints: 6
  - linear=6
- parameters: defaults
- objective: maximize
- problem type is: LP
```

```
In [7]: otimizacao = modelo.solve()
modelo.print_solution()
```

```
objective: 21.000
X1=3.000
X2=1.500
```

```
In [8]: modelo.parameters.lpmethod = 4
modelo.solve(url=None, key=None, log_output=True)
```

```
Version identifier: 20.1.0.0 | 2020-11-11 | 9bedb6d68
CPXPARAM_Read_DataCheck           1
CPXPARAM_LPMETHOD                4
Tried aggregator 1 time.
LP Presolve eliminated 3 rows and 0 columns.
Reduced LP has 3 rows, 2 columns, and 6 nonzeros.
Presolve time = 0.02 sec. (0.00 ticks)
Parallel mode: using up to 4 threads for barrier.
Number of nonzeros in lower triangle of A*A' = 3
Using Approximate Minimum Degree ordering
```

Total time for automatic ordering = 0.00 sec. (0.00 ticks)

Summary statistics for Cholesky factor:

Threads	= 4
Rows in Factor	= 3
Integer space required	= 3
Total non-zeros in factor	= 6
Total FP ops to factor	= 14

Itn	Primal Obj	Dual Obj	Prim Inf	Upper Inf	Dual Inf	Inf	Ratio
0	2.2500000e+02	1.2166667e+01	3.69e+02	4.80e+01	1.63e+01	1.00e+00	
1	5.5890668e+01	3.0932187e+01	7.35e+01	9.56e+00	1.32e+00	2.42e+04	
2	2.2212098e+01	2.1359458e+01	4.88e+00	6.35e-01	2.52e-01	4.68e+02	
3	2.1828888e+01	2.1604330e+01	1.64e+00	2.13e-01	1.49e-02	4.89e+03	
4	2.1008603e+01	2.1004920e+01	1.56e-02	2.03e-03	1.17e-04	2.43e+05	
5	2.1000001e+01	2.1000000e+01	1.56e-06	2.03e-07	1.30e-08	1.84e+09	
6	2.1000000e+01	2.1000000e+01	1.56e-10	2.03e-11	1.30e-12	1.85e+13	

Barrier time = 0.03 sec. (0.01 ticks)

Parallel mode: deterministic, using up to 4 threads for concurrent optimization:

- * Starting dual Simplex on 1 thread...
- * Starting primal Simplex on 1 thread...

Dual crossover.

Dual: Fixed no variables.

Primal: Fixed no variables.

Dual simplex solved model.

Total crossover time = 0.03 sec. (0.00 ticks)

Total time on 4 threads = 0.08 sec. (0.02 ticks)

Out[8]: docplex.mp.solution.SolveSolution(obj=21,values={X1:3,X2:1.5})

In [9]:

%notebook "D:\SISTEMAS\SEMESTRE-2020-2\Pesquisa Operacional\Lista1\Questao_9.ipynb"