

Roteiro Atividade Avião

Para que o projeto funcione, antes de tudo, devemos instalar a biblioteca `from inspyred import ec`, é ela que faz a magia do código, para isso, existem diversas maneiras, se estiver usando VS Code, pode dar `Ctrl + Shift + ;`, ou pode ser feito pelo CDM, com ele aberto, cole esse código -> `pip install inspyred`

Caso esteja usando Colab, crie no inicio, uma nova função cédula, e cole esse código `!pip install inspyred`, e rode, feito isso, o código já esta pronto para ser rodado

A natureza do problema do avião consiste em um cálculo de maximização para encontrar o lucro máximo possível, respeitando o valor aceitável pelo avião.

O trabalho consiste em 12 variáveis

Cargas 1, 2 3 e 4, com a posição, Frente, Meio e Atras do aviao

```
carga1_F # frente  
carga1_M # meio  
carga1_A # atras
```

```
carga2_F # frente  
carga2_M # meio  
carga2_A # atras
```

```
carga3_F # frente  
carga3_M # meio  
carga3_A # atras
```

```
carga4_F # frente  
carga4_M # meio  
carga4_A # atras
```

O cálculo do peso do avião vem em seguida. O problema nos dá essas informações da capacidade do avião, lembrando que não pode passar da capacidade, a resolução é a seguinte:

Compartimento	Capacidade de peso (tonelada)
Dianteiro (D)	10
Central (C)	16
Traseiro (T)	8

```
# Peso maximo

# Frente
aux1 = np.maximum(0, float((carga1_F + carga2_F + carga3_F +
    carga4_F) - 10000)) / float(10000 / num_aux)

# Meio
aux2 = aux3 = np.maximum(0, float(
    (carga1_M + carga2_M + carga3_M + carga4_M) - 16000)) / float(16000 / num_aux)

# Atras
aux3 = np.maximum(0, float((carga1_A + carga2_A + carga3_A + carga4_A) - 8000)
    ) / float(8000 / num_aux)
```

Com cálculo do peso pronto, passamos para o cálculo do volume, no qual o problema nos dá as seguintes limitações

Compartimento	Capacidade volumétrica (m^3)
Dianteiro (D)	6800
Central (C)	8700
Traseiro (T)	5300

Assim definimos o cálculo para o volume:

```
# Volume Maximo

# Frente
aux4 = np.maximum(0, float((0.480 * carga1_F + 0.650 * carga2_F +
    0.580 * carga3_F + 0.390 * carga4_F) - 6800)) / float(6800 / num_aux)

# Meio
aux5 = np.maximum(0, float((0.480 * carga1_M + 0.650 * carga2_M +
    0.580 * carga3_M + 0.390 * carga4_M) - 8700)) / float(8700 / num_aux)

# Atras
aux6 = np.maximum(0, float((0.48 * carga1_A + 0.65 * carga2_A +
    0.58 * carga3_A + 0.39 * carga4_A) - 5300)) / float(5300 / num_aux)
```

Feito isso, passamos para o cálculo de proporção, tanto de carga quanto de volume, isso será relevante no final do código, portanto, deve-ser realizado da seguinte maneira

```
# Maximo peso por proporção
# Frente Maximo
aux7 = np.maximum(0, float((carga_dianteira / soma_total_cargas) - (10000 / peso_max))
    ) / float((10000 / peso_max) / num_aux)

# Meio Maximo
aux8 = np.maximum(0, float((carga_meio / soma_total_cargas) -
    (16000 / peso_max))) / float((16000 / peso_max) / num_aux)

# Atras Maximo
aux9 = np.maximum(0, float((carga_traseira / soma_total_cargas) -
    (8000 / peso_max))) / float((8000 / peso_max) / num_aux)

# Maximo de peso por carga
# Carga 1 Maximo
aux10 = np.maximum(0, float((carga1_F + carga1_M + carga1_A) - 18000)
    ) / float(18000 / 13)
# Carga 2 Maximo
aux11 = np.maximum(0, float((carga2_F + carga2_M + carga2_A) - 15000)
    ) / float(15000 / 13)
# Carga 3 Maximo
aux12 = np.maximum(0, float((carga3_F + carga3_M + carga3_A) - 23000)
    ) / float(23000 / 13)
# Carga 4 Maximo
aux13 = np.maximum(0, float((carga4_F + carga4_M + carga4_A) - 12000)
    ) / float(12000 / 13)

fit = fit - [aux1 + aux2 + aux3 + aux4 + aux5 + aux6 + aux7 +
    aux8 + aux9 + aux10 + aux11 + aux12 + aux13]

return fit
```

Após as verificações do avião, a função

```
def solution_evaluation(carga1_F, carga1_M, carga1_A, carga2_F,
carga2_M, carga2_A, carga3_F, carga3_M, carga3_A, carga4_F, carga4_M,
carga4_A):
```

É chamada, passando todas as doze variáveis nela, essa função consiste na impressão do código com algumas verificações a mais

A impressão da Carga leva a seguinte formação:

```
print("\n ===== \n")
print("\n [ PESO INDIVIDUAL ] \n")

print("FRENTE 1:", carga1_F)
print("MEIO 1:", carga1_M)
print("ATRAS 1:", carga1_A)
total_carga1 = float(carga1_F + carga1_M + carga1_A)
print("# TOTAL CARGA 1 #", (total_carga1))

print("\nFRENTE 2:", carga2_F)
print("MEIO 2:", carga2_M)
print("ATRAS 2:", carga2_A)
total_carga2 = float(carga2_F + carga2_M + carga2_A)
print("# TOTAL CARGA 2 #", (total_carga2))

print("\nFRENTE 3:", carga3_F)
print("MEIO 3:", carga3_M)
print("ATRAS 3:", carga3_A)
total_carga3 = float(carga3_F + carga3_M + carga3_A)
print("# TOTAL CARGA 3 #", (total_carga3))

print("\nFRENTE 4:", carga4_F)
print("MEIO 4:", carga4_M)
print("ATRAS 4:", carga4_A)
total_carga4 = float(carga4_F + carga4_M + carga4_A)
print("# TOTAL CARGA 4 #", (total_carga4))

cargas_total = float(total_carga1 + total_carga2 +
                     total_carga3 + total_carga4)

print("\n[ PESO TOTAL ]: ", cargas_total)

print("\n- FRENTE TOTAL:", carga1_F + carga2_F + carga3_F + carga4_F)
print("- MEIO TOTAL :", carga1_M + carga2_M + carga3_M + carga4_M)
print("- ATRAS TOTAL:", carga1_A + carga2_A + carga3_A + carga4_A)
```

O volume fica:

```
# PESO X VOLUME
volume_frente = float((carga1_F * 0.480)+(carga2_F * 0.650) +
                      (carga3_F * 0.580) + (carga4_F * 0.390))
volume_meio = float(
    (carga1_M * 0.480) + (carga2_M * 0.650) + (carga3_M * 0.580) + (carga4_M * 0.390))
volume_atras = float(
    (carga1_A * 0.480)+(carga2_A * 0.650) + (carga3_A * 0.580) + (carga4_A * 0.390))

print("\n [ VOLUME INDIVIDUAL ]")
print("- FRENTE TOTAL : ", volume_frente)
print("- MEIO TOTAL : ", volume_meio)
print("- ATRAS TOTAL : ", volume_atras)
```

Proporção:

```
print("\n [PROPORCAO ] ")
print("FRENTE: ", round(
    ((carga1_F + carga2_F + carga3_F + carga4_F) / cargas_total),3))
print("MEIO: ", round(
    ((carga1_M + carga2_M + carga3_M + carga4_M) / cargas_total),3))
print("ATRAS: ", round(
    ((carga1_A + carga2_A + carga3_A + carga4_A) / cargas_total),3))

lucro_carga1 = float(0.310 * carga1_F + 0.310 * carga1_M + 0.310 * carga1_A)
lucro_carga2 = float(0.380 * carga2_F + 0.380 * carga2_M + 0.380 * carga2_A)
lucro_carga3 = float(0.350 * carga3_F + 0.350 * carga3_M + 0.350 * carga3_A)
lucro_carga4 = float(0.285 * carga4_F + 0.285 * carga4_M + 0.285 * carga4_A)
```

Por fim, a mais importante é a impressão do lucro, a meta a ser batida é de R\$ 12.151,56. valor apresentado no problema repassado

A impressão do lucro, recebe o lucro da carga 1, carga 2, carga 3 e carga 4, com isso, a impressão da soma total, recebendo as 4 variáveis.

Caso houver quebra de regra, cairá nos 3 IF, nele mostra qual das 3 áreas do avião passou da cota

```
print("\n -[LUCROS ]")
print("LUCRO DA CARGA 1:", lucro_carga1)
print("LUCRO DA CARGA 2:", lucro_carga2)
print("LUCRO DA CARGA 3:", lucro_carga3)
print("LUCRO DA CARGA 4:", lucro_carga4)
print("\n[LUCRO TOTAL] ", lucro_carga1 + lucro_carga2 + lucro_carga3 + lucro_carga4)

if (((compartimento_frente/total) >= 0.3) or ((compartimento_frente/total) <= 0.29)):

    print(f"frente passou: {np.round((compartimento_frente/total),4)}")

if (((compartimento_meio/total) >= 0.48) or ((compartimento_meio/total) <= 0.47)):

    print(f"meio passou: {np.round((compartimento_meio/total),4)}")

if (((compartimento_atras/total) >= 0.24) or ((compartimento_atras/total) <= 0.23)):

    print(f"atras passou: {np.round((compartimento_atras/total),4)})")
```

Por fim e não menos importante, a chamada da main, na qual recebe a biblioteca inspyred, é ela que faz toda a magia de rotação de carga.

Trago agora um exemplo funcional da atividade:

Generation	Evaluation	Worst	Best	Median	Average	Std Dev
1344	26128	-11.986039	0.37664645	-3.5474265	-4.1943283	2.89654127
1345	26140	-11.461768	0.37664645	-3.5436938	-4.1852863	2.88668793
1346	26152	-11.425444	0.37664645	-3.5377783	-4.1768255	2.87755001
1347	26164	-11.390073	0.37664645	-3.5337733	-4.1671589	2.86769315

Em execução (3m23s) <cell line: 285> > main() > evolve()

[PESO INDIVIDUAL]

FRENTE 1: 2752.0
MEIO 1: 4193.0
ATRAS 1: 259.0
TOTAL CARGA 1 # 7204.0

FRENTE 2: 2897.0
MEIO 2: 8056.0
ATRAS 2: 2673.0
TOTAL CARGA 2 # 13626.0

FRENTE 3: 1917.0
MEIO 3: 1181.0
ATRAS 3: 2978.0
TOTAL CARGA 3 # 6076.0

FRENTE 4: 1924.0
MEIO 4: 1754.0
ATRAS 4: 1682.0
TOTAL CARGA 4 # 5360.0

[PESO TOTAL]: 32266.0

- FRENTE TOTAL: 9490.0
- MEIO TOTAL : 15184.0
- ATRAS TOTAL: 7592.0

=====

[VOLUME INDIVIDUAL]

- FRENTE TOTAL : 5066.23
- MEIO TOTAL : 8618.08
- ATRAS TOTAL : 4244.99

=====

[PROPORCAO]

FRENTE: 0.294
MEIO: 0.471
ATRAS: 0.235

=====

-[LUCROS]

LUCRO DA CARGA 1: 2233.24
LUCRO DA CARGA 2: 5177.88
LUCRO DA CARGA 3: 2126.6
LUCRO DA CARGA 4: 1527.5999999999997

[LUCRO TOTAL] 11065.32

O melhor dos resultados feitos ate agora, é no valor de R\$ 11.335,27

```
[PROPORCAO ]
FRENTE: 0.294
MEIO: 0.472
ATRAS: 0.235

=====
-[LUCROS ]
LUCRO DA CARGA 1: 2073.59
LUCRO DA CARGA 2: 4015.08
LUCRO DA CARGA 3: 3186.05
LUCRO DA CARGA 4: 2060.549999999997

[LUCRO TOTAL] 11335.27
```

Trago um exemplo de código estourado, na qual imprime qual das cargas passou do cálculo desejado

```
-[LUCROS ]
LUCRO DA CARGA 1: 2058.089999999997
LUCRO DA CARGA 2: 1600.94
LUCRO DA CARGA 3: 3283.0
LUCRO DA CARGA 4: 2799.839999999997

[LUCRO TOTAL] 9741.86999999999
frente passou: 0.3187
atras passou: 0.216
```