# Segunda avaliação de Pesquisa Operacional

Gabriel de Jesus Pereira

## Questão 1

```python
import pulp
from pulp import LpMaximize, LpMinimize, LpVariable, LpProblem, lpSum, LpStatus
```

```python
prob_min = LpProblem('model', LpMinimize)
```

```python
N = ['S', 'A', 'B',
     'C', 'D', 'E',
     'F', 'G', 'H',
     'T']

C = {'S': {'A': 9, 'B': 8, 'C': 12},
     'A': {'B': 8, 'D': 10},
     'B': {'D': 12, 'E': 9},
     'C': {'E': 10, 'F': 15},
     'D': {'G': 8},
     'E': {'G': 11, 'H': 14, 'F': 10},
     'F': {'H': 5},
     'G': {'T': 12},
     'H': {'T': 8}
     }

# C = {
#     1: {2: 16, 5: 13},
#     2: {3: 12},
#     3: {5: 9, 4: 20},
#     5: {2: 4, 6: 14},
#     6: {3: 7, 4: 4}}
# D = {node: 0 for node in N} #definindo o lado direito das restrições

E = [(i,j) for i in N for j in N if i in C.keys() if j in C[i].keys()]
x = LpVariable.dicts('x', E,  lowBound = 0, cat='Binary')
```

```python
prob_min += (9 * x['S', 'A'] + 8 * x['S', 'B'] + 12 * x['S', 'C'] +
             8 * x['A', 'B'] + 10 * x['A', 'D'] + 12 * x['B', 'D'] +
             9 * x['B', 'E'] + 11 * x['E', 'G'] + 8 * x['D', 'G'] +
             12 * x['G', 'T'] + 14 * x['E', 'H'] + 8 * x['H', 'T'] +
             10 * x['C', 'E'] + 15 * x['C', 'F'] + 10 * x['E', 'F'] +
             5 * x['F', 'H'])

prob_min += x['S', 'A'] + x['S', 'B'] + x['S', 'C'] == 1
prob_min += x['S', 'A'] - x['A', 'B'] - x['A', 'D'] == 0
prob_min += x['A', 'B'] + x['S', 'B'] - x['B', 'D'] - x['B', 'E'] == 0
prob_min += x['S', 'C'] - x['C', 'E'] - x['C', 'F'] == 0
prob_min += x['A', 'D'] + x['B', 'D'] - x['D', 'G'] == 0
prob_min += x['D', 'G'] + x['E', 'G'] - x['G', 'T'] == 0
prob_min += x['B', 'E'] + x['C', 'E'] - x['E', 'G'] - x['E', 'H'] - x['E', 'F'] == 0
prob_min += x['C', 'F'] + x['E', 'F'] - x['F', 'H'] == 0
prob_min += x['E', 'H'] + x['F', 'H'] - x['H', 'T'] == 0
prob_min += x['G', 'T'] + x['H', 'T'] == 1
```

```python
status_min = prob_min.solve()

print("Valor da função objetivo:", prob_min.objective.value())

print(f'Status\n{LpStatus[status_min]}\n')

print('Valor ótimo das variáveis')
for v in prob_min.variables():
    print(v.name, '=', v.varValue)
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /home/cowvin/.local/lib/python3.10/site-packages/pulp/solverdir/cbc/linux/64/
At line 2 NAME          MODEL
At line 3 ROWS
At line 15 COLUMNS
At line 96 RHS
At line 107 BOUNDS
At line 124 ENDATA
Problem MODEL has 10 rows, 16 columns and 32 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 39 - 0.00 seconds
Cgl0004I processed model has 0 rows, 0 columns (0 integer (0 of which binary)) and 0 elements
Cbc3007W No integer variables - nothing to do
Cuts at root node changed objective from 39 to -1.79769e+308
Probing was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cu
Gomory was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cut
Knapsack was tried 0 times and created 0 cuts of which 0 were active after adding rounds of c
Clique was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cut
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were active after addin
FlowCover was tried 0 times and created 0 cuts of which 0 were active after adding rounds of
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active after adding rounds of
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after adding rounds of c

Result - Optimal solution found

Objective value:                39.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.00

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Valor da função objetivo: 39.0
Status
Optimal

Valor ótimo das variáveis
x_('A',_'B') = 0.0
x_('A',_'D') = 1.0
x_('B',_'D') = 0.0
x_('B',_'E') = 0.0
x_('C',_'E') = 0.0
x_('C',_'F') = 0.0
x_('D',_'G') = 1.0
x_('E',_'F') = 0.0
x_('E',_'G') = 0.0
x_('E',_'H') = 0.0
x_('F',_'H') = 0.0
x_('G',_'T') = 1.0
x_('H',_'T') = 0.0
x_('S',_'A') = 1.0
x_('S',_'B') = 0.0
x_('S',_'C') = 0.0
```

Com base nos resultados do caminho mínimo, o percurso mais curto começa em S, segue para A, de A para D, de D para G, e finalmente de G para T.

## Questão 2

```python
x = LpVariable.dicts('x', E,  lowBound = 0)

prob_max = LpProblem('model', LpMaximize)
prob_max += x['G', 'T'] + x['H', 'T']

prob_max += x['S', 'A'] - x['A', 'B'] - x['A', 'D'] == 0
prob_max += x['S', 'B'] + x['A', 'B'] - x['B', 'D'] - x['B', 'E'] == 0
prob_max += x['S', 'C'] - x['C', 'E'] - x['C', 'F'] == 0
prob_max += x['A', 'D'] + x['B', 'D'] - x['D', 'G'] == 0
prob_max += x['B', 'E'] + x['C', 'E'] - x['E', 'G'] - x['E', 'H'] - x['E', 'F'] == 0
prob_max += x['E', 'F'] + x['C', 'F'] - x['E', 'H'] == 0
prob_max += x['D', 'G'] + x['E', 'G'] - x['G', 'T'] == 0
prob_max += x['E', 'H'] + x['F', 'H'] - x['H', 'T'] == 0

for i in N:
  for j in N:
    if (i,j) in E:
      prob_max += x[i,j] <= C[i][j]
```

```python
status = prob_max.solve()

# print("Valor da função objetivo:", prob.objective.value())

print(f'Status\n{LpStatus[status]}\n')
print('Valor ótimo das variáveis')
for v in prob_max.variables():
    print(v.name, '=', v.varValue)
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /home/cowvin/.local/lib/python3.10/site-packages/pulp/solverdir/cbc/linux/64/
At line 2 NAME          MODEL
At line 3 ROWS
At line 29 COLUMNS
At line 75 RHS
At line 100 BOUNDS
At line 101 ENDATA
Problem MODEL has 24 rows, 16 columns and 43 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve 7 (-17) rows, 11 (-5) columns and 21 (-22) elements
Perturbing problem by 0.001% of 1 - largest nonzero change 9.4905782e-07 ( 9.4905782e-05%) -
0  Obj -0 Dual inf 1.9999962 (2)
6  Obj 19.999958
Optimal - objective value 20
After Postsolve, objective 20, infeasibilities - dual 0 (0), primal 0 (0)
Optimal objective 20 - 6 iterations time 0.002, Presolve 0.00
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Status
Optimal

Valor ótimo das variáveis
x_('A',_'B') = 1.0
x_('A',_'D') = 6.0
x_('B',_'D') = 0.0
x_('B',_'E') = 9.0
x_('C',_'E') = 0.0
x_('C',_'F') = 3.0
x_('D',_'G') = 6.0
x_('E',_'F') = 0.0
x_('E',_'G') = 6.0
x_('E',_'H') = 3.0
x_('F',_'H') = 5.0
x_('G',_'T') = 12.0
x_('H',_'T') = 8.0
x_('S',_'A') = 7.0
x_('S',_'B') = 8.0
x_('S',_'C') = 3.0
```

Pelo valor da função objetivo, o valor do fluxo máximo que chega ao destino T é de 20 unidades.

## Questão 3

```
prob_trans = pulp.LpProblem("Problema_de_Transbordo", pulp.LpMinimize)

# Definindo as variáveis de decisão
x_1_3 = pulp.LpVariable('x_1_3', lowBound=0, cat='Continuous')
x_1_4 = pulp.LpVariable('x_1_4', lowBound=0, cat='Continuous')
x_1_5 = pulp.LpVariable('x_1_5', lowBound=0, cat='Continuous')
x_2_3 = pulp.LpVariable('x_2_3', lowBound=0, cat='Continuous')
x_2_4 = pulp.LpVariable('x_2_4', lowBound=0, cat='Continuous')
x_2_5 = pulp.LpVariable('x_2_5', lowBound=0, cat='Continuous')
x_3_6 = pulp.LpVariable('x_3_6', lowBound=0, cat='Continuous')
x_3_7 = pulp.LpVariable('x_3_7', lowBound=0, cat='Continuous')
x_3_8 = pulp.LpVariable('x_3_8', lowBound=0, cat='Continuous')
x_4_6 = pulp.LpVariable('x_4_6', lowBound=0, cat='Continuous')
x_4_7 = pulp.LpVariable('x_4_7', lowBound=0, cat='Continuous')
x_4_8 = pulp.LpVariable('x_4_8', lowBound=0, cat='Continuous')
x_5_6 = pulp.LpVariable('x_5_6', lowBound=0, cat='Continuous')
x_5_7 = pulp.LpVariable('x_5_7', lowBound=0, cat='Continuous')
x_5_8 = pulp.LpVariable('x_5_8', lowBound=0, cat='Continuous')
```

```
prob_trans += (16 * x_1_3 + 10 * x_1_4 + 12 * x_1_5 +
               15 * x_2_3 + 14 * x_2_4 + 17 * x_2_5 +
               6 * x_3_6 + 8 * x_3_7 + 10 * x_3_8 +
               7 * x_4_6 + 11 * x_4_7 + 11 * x_4_8 +
               4 * x_5_6 + 5 * x_5_7 + 12 * x_5_8)

prob_trans += x_1_3 + x_1_4 + x_1_5 == 300
prob_trans += x_2_3 + x_2_4 + x_2_5 == 300
prob_trans += x_3_6 + x_4_6 + x_5_6 == 200
prob_trans += x_3_7 + x_4_7 + x_5_7 == 100
prob_trans += x_3_8 + x_4_8 + x_5_8 == 300
prob_trans += x_1_3 + x_2_3 - x_3_6 - x_3_7 - x_3_8 == 0
prob_trans += x_1_4 + x_2_4 - x_4_6 - x_4_7 - x_4_8 == 0
prob_trans += x_1_5 + x_2_5 - x_5_6 - x_5_7 - x_5_8 == 0
```

```
prob_trans.solve()

# Status da solução
status = pulp.LpStatus[prob_trans.status]

# Valores ótimos das variáveis
solution = {v.name: v.varValue for v in prob_trans.variables()}

# Custo total mínimo
total_cost = pulp.value(prob_trans.objective)

print(f"Status: {status}")
# print(f"Solução: {solution}")
print(solution)
print(f"Custo total mínimo: {total_cost}")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /home/cowvin/.local/lib/python3.10/site-packages/pulp/solverdir/cbc/linux/64/
At line 2 NAME          MODEL
At line 3 ROWS
At line 13 COLUMNS
At line 59 RHS
At line 68 BOUNDS
At line 69 ENDATA
Problem MODEL has 8 rows, 15 columns and 30 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Presolve 8 (0) rows, 15 (0) columns and 30 (0) elements
0  Obj 0 Primal inf 1200 (5)
8  Obj 12400
Optimal - objective value 12400
Optimal objective 12400 - 8 iterations time 0.002
Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Status: Optimal
{'x_1_3': 0.0, 'x_1_4': 0.0, 'x_1_5': 300.0, 'x_2_3': 0.0, 'x_2_4': 300.0, 'x_2_5': 0.0, 'x_
Custo total mínimo: 12400.0
```

## Questão 4

```
custos = [
    [12, 5, 3, 2],  # Tarefa 1
    [2, 6, 9, 5],  # Tarefa 2
    [6, 2, 3, 2],  # Tarefa 3
    [1, 7, 4, 2]   # Tarefa 4
]

# Definir o número de operários e tarefas
n_operarios = len(custos)
n_tarefas = len(custos[0])


problema = pulp.LpProblem("Problema_de_Designacao", pulp.LpMinimize)

x = pulp.LpVariable.dicts("x", [(i, j) for i in range(n_operarios) for j in range(n_tarefas)

problema += pulp.lpSum([custos[i][j] * x[(i, j)] for i in range(n_operarios) for j in range(

for i in range(n_operarios):
    problema += pulp.lpSum([x[(i, j)] for j in range(n_tarefas)]) == 1

for j in range(n_tarefas):
    problema += pulp.lpSum([x[(i, j)] for i in range(n_operarios)]) == 1

# Resolver o problema
problema.solve()

# Exibir a solução
print("Status:", pulp.LpStatus[problema.status])
```

```python
# Exibir a designação de operários para as tarefas
for i in range(n_operarios):
    for j in range(n_tarefas):
        if pulp.value(x[(i, j)]) == 1:
            print(f"Operário {i+1} designado para Tarefa {j+1}, Custo: {custos[i][j]} horas.

# Exibir o tempo total mínimo
print(f"Tempo total mínimo: {pulp.value(problema.objective)} horas.")
```

```
Welcome to the CBC MILP Solver
Version: 2.10.3
Build Date: Dec 15 2019

command line - /home/cowvin/.local/lib/python3.10/site-packages/pulp/solverdir/cbc/linux/64/
At line 2 NAME          MODEL
At line 3 ROWS
At line 13 COLUMNS
At line 94 RHS
At line 103 BOUNDS
At line 120 ENDATA
Problem MODEL has 8 rows, 16 columns and 32 elements
Coin0008I MODEL read with 0 errors
Option for timeMode changed from cpu to elapsed
Continuous objective value is 9 - 0.00 seconds
Cgl0004I processed model has 8 rows, 16 columns (16 integer (16 of which binary)) and 32 ele
Cutoff increment increased from 1e-05 to 0.9999
Cbc0038I Initial state - 0 integers unsatisfied sum - 0
Cbc0038I Solution found of 9
Cbc0038I Before mini branch and bound, 16 integers at bound fixed and 0 continuous
Cbc0038I Mini branch and bound did not improve solution (0.00 seconds)
Cbc0038I After 0.00 seconds - Feasibility pump exiting with objective of 9 - took 0.00 secon
Cbc0012I Integer solution of 9 found by feasibility pump after 0 iterations and 0 nodes (0.0
Cbc0001I Search completed - best objective 9, took 0 iterations and 0 nodes (0.00 seconds)
Cbc0035I Maximum depth 0, 0 variables fixed on reduced cost
Cuts at root node changed objective from 9 to 9
Probing was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cu
Gomory was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cu
Knapsack was tried 0 times and created 0 cuts of which 0 were active after adding rounds of
Clique was tried 0 times and created 0 cuts of which 0 were active after adding rounds of cu
MixedIntegerRounding2 was tried 0 times and created 0 cuts of which 0 were active after addi
FlowCover was tried 0 times and created 0 cuts of which 0 were active after adding rounds of
TwoMirCuts was tried 0 times and created 0 cuts of which 0 were active after adding rounds o
```

```
ZeroHalf was tried 0 times and created 0 cuts of which 0 were active after adding rounds of

Result - Optimal solution found

Objective value:                9.00000000
Enumerated nodes:               0
Total iterations:               0
Time (CPU seconds):             0.00
Time (Wallclock seconds):       0.00

Option for printingOptions changed from normal to all
Total time (CPU seconds):       0.00   (Wallclock seconds):       0.00

Status: Optimal
Operário 1 designado para Tarefa 3, Custo: 3 horas.
Operário 2 designado para Tarefa 1, Custo: 2 horas.
Operário 3 designado para Tarefa 2, Custo: 2 horas.
Operário 4 designado para Tarefa 4, Custo: 2 horas.
Tempo total mínimo: 9.0 horas.
```