settle_up_todo

March 1, 2021

1 Settle Up Problem (Dlužníček)

Combinatorial Optimization course, FEE CTU in Prague. Created by Industrial Informatics Department.

1.1 Motivation

You went on a trip with a group of your friends. All of you shared some expenses, and now it is the time to settle all the debts. It is clear that everyone should pay the same amount; however, people are lazy, and so you want to find the solution which minimizes the number of transactions.

1.2 Input

You are given the following:

- A set of people P
- For every person $i \in P$ the cost c_i (i.e., amount of money that i payed)

For the experiments, you may use the following instance:

[1]:

```python
P = set(["A", "B", "C", "D"])
c = {"A": 0, "B": 590, "C": 110, "D": 300}  # c_i is accessed by calling c[i]
sv = sum(c.values())/len(c)  # the settlement value
```

Number sv (the settlement value) is the fair price that every person should pay.

1.3 Output

You should find a list of tuples (x_k, y_k, z_k) representing the transactions: person x_k should pay person y_k z_k euros. The number of transactions (i.e., the length of the list) should be minimized.

For the given instance, the **optimal solution** consists of 3 transactions, namely

```
A -> B: 250.0
C -> D: 50.0
C -> B: 90.0
```

1.4 Exercise

Implement the ILP model of the problem. You can assume that the settlement value is int (or was rounded).

[2]:

```python
import gurobipy as g  # import Gurobi module


# model --------------------------------------------------
m = g.Model()

# - ADD VARIABLES
# TODO

# - ADD CONSTRAINTS
# TODO

# - SET OBJECTIVE
# TODO

# call the solver -------------------------------------------
m.optimize()

# print the solution -----------------------------------------
print('\nSOLUTION:')
# TODO
```

```
Academic license - for non-commercial use only
Optimize a model with 0 rows, 0 columns and 0 nonzeros
Coefficient statistics:
  Matrix range     [0e+00, 0e+00]
  Objective range  [0e+00, 0e+00]
  Bounds range     [0e+00, 0e+00]
  RHS range        [0e+00, 0e+00]
Presolve time: 0.00s
Presolve: All rows and columns removed
Iteration    Objective       Primal Inf.    Dual Inf.      Time
       0    0.0000000e+00   0.000000e+00   0.000000e+00      0s

Solved in 0 iterations and 0.01 seconds
Optimal objective  0.000000000e+00

SOLUTION:
```

1.5 Additional experiments

- experiment with different values of the ‘big M’ constant

- try to generalize the model to work even with float sv (e.g., rounded to 0.01)

[ ]: