Dimensionnement de canalisations

```
%pip install gurobipy
from gurobipy import Model, GRB, LinExpr, multidict
#import utils_display
import pandas as pd
import matplotlib.pyplot as plt
```

Données du problème

```
pipe = ['C1','C2']   # les canalisations a installer
diam_max = [40, 60]  # diametre maximal (en cm) pour chaque canalisation
cost = [3, 2]        # cout (en euros/cm) d'installation en fonction du diametre pour chaque canalisation
flow = [3, 5]        # debit (en unite/cm) en fonction du diametre pour chaque canalisation
budget = 180         # budget max (en euros) d'installation

VERBOSE = False
```

Modèle mathématique

$$
\begin{aligned}
& max \sum_p Flow_p * diam_p \\
& s.t.: \sum_p Cost_p * diam_p \leq Budget \\
& \quad 0 \leq diam_p \leq MaxDiam_p \qquad \forall p
\end{aligned}
$$

```
m = Model('canalisations')
```

Variables de décisions

```
diam = m.addVars(pipe, lb=0, ub=diam_max, vtype=GRB.CONTINUOUS, name="D") # diametre pour chaque canalisation
m.update()
cost_diam = LinExpr(cost, diam.values())  # cout d'installation pour la solution diam (somme ponderee des variables par les cou
flow_diam = LinExpr(flow, diam.values())  # debit pour la solution diam (somme ponderee des variables par les debits)
```

Contraintes et objectif

```
ctbudget = m.addConstr(cost_diam <= budget)
m.setObjective(flow_diam, GRB.MAXIMIZE)
```

Optimisation

```
m.optimize()
```

```
m.display()
m.write("myfirst.lp")
```

Résultats

```
m.printStats()
```

```
# caractéristiques de la solution
result = {'obj': m.objVal, 'lb': m.objBound, 'time': m.runtime}
print(result)
```

```
# installation optimale
xdict = {c: diam[c].x for c in pipe}
print(f"diametres a installer: {xdict}")

costval = cost_diam.getValue()
flowval = flow_diam.getValue()
print(f"cout = {costval}, debit = {flowval}")
```