

teste_pli

August 22, 2025

```
[1]: import pulp
import pandas as pd

[2]: prof = pd.read_csv('../dados/docentes.csv')
disc = pd.read_csv('../dados/disciplinas.csv')
prefs = pd.read_csv('../dados/preferencias.csv')
conf = pd.read_csv('../dados/matriz_conflitos_disciplinas.csv')

[3]: W = 4

[4]: #Criando os dicionarios a partir dos DataFrames
#Dicionario com carga horária das disciplinas
ch = disc[['id_disciplina', 'carga_horaria']].set_index('id_disciplina').
    ↪to_dict(orient='index')
#Dicionario com carga horária máxima dos professores
ch_max = prof[['id_docente', 'carga_horaria']].set_index('id_docente').
    ↪to_dict(orient='index')
#Dicionario de preferências
pref = prefs.pivot(index='id_docente', columns='id_disciplina',
    ↪values='preferencia').to_dict(orient='index')

[29]: # Cria a instância do problema de maximização
modelo = pulp.LpProblem("Alocacao_Docentes_PLI", pulp.LpMaximize)

[30]: x = pulp.LpVariable.dicts("x", (prof['id_docente'], disc['id_disciplina']),
    ↪cat='Binary')

y = pulp.LpVariable.dicts("y", prof['id_docente'], lowBound=0, cat='Integer')

[31]: soma_preferencias = pulp.lpSum(pref[p][d] * x[p][d] for p in prof['id_docente']
    ↪for d in disc['id_disciplina'])
soma_penalidades = W * pulp.lpSum(y[p] for p in prof['id_docente'])

# Adiciona a função objetivo ao modelo
modelo += soma_preferencias - soma_penalidades, "Funcao_Objetivo_Max_Satisfacao"
```

```
[32]: for d in disc['id_disciplina']:
        modelo += pulp.lpSum(x[p][d] for p in prof['id_docente']) == 1,
        ↪f"Cobertura_Disciplina_{d}"

        # Supondo que 'ch' e 'ch_max' são dicionários com as cargas horárias
        for p in prof['id_docente']:
            modelo += pulp.lpSum(ch[d]['carga_horaria'] * x[p][d] for d in
            ↪disc['id_disciplina']) <= ch_max[p], f"Carga_Horaria_Prof_{p}"

        for p in prof['id_docente']:
            # Soma apenas as alocações onde a preferência é 1
            modelo += pulp.lpSum(x[p][d] for d in disc['id_disciplina'] if pref[p][d]
            ↪== 1) <= y[p], f"Penalidade_Prof_{p}"
```

```
[34]: modelo.solve()
```

```
[34]: 1
```

```
[13]: ch['EST0001_01']['carga_horaria']
```

```
[13]: 30
```