

# wind\_turbine\_maintenance\_optimization

November 19, 2025

## 1 Wind Turbine Maintenance Optimization

There are 7 wind turbines, which all need to be maintained once during the week. There are two maintenance teams: maintenance team 1 and maintenance team 2. There is no difference between the two maintenance teams.

The maintenance teams only work on workdays, i.e. from Monday to Friday. It takes one maintenance team a full day to maintain one wind turbine. Due to different locations of each wind turbine and the weather of the date, the maintenance costs are different. The costs are the costs are the same for both maintenance teams.

The costs are stated in the following table:

turbine	Mon	Tue	Wed	Thu	Fri
1	10	11	12	13	14
2	12	14	16	18	20
3	17	18	17	18	17
4	20	19	18	17	16
5	22	22	22	22	33
6	24	23	22	23	23
7	9	6	8	7	9

Formulate an integer linear model to minimize the maintenance cost (and eventually solve it numerically).

A suggested integer programming formulation is as follows:

Sets: -  $\mathcal{L} := \{i \mid i \in \{1, \dots, 7\}\}$ : the set of wind turbines, -  $\mathcal{M} := \{j \mid j \in \{\text{Mon}, \dots, \text{Fri}\}\}$ : the set of different days, -  $\mathcal{N} := \{k \mid k \in \{1, 2\}\}$ : the set of two maintenance teams.

To simplify the problem, we add a parameters  $c_{ij}$ ,  $i \in \mathcal{L}, j \in \mathcal{M}$ , are the maintenance costs for different wind turbines at each day.

The decision variables are:

$$x_{i,j,k} = \begin{cases} 1 & \text{if maintenance team } k \in \mathcal{N} \text{ maintain wind turbine } i \in \mathcal{L} \text{ at day } j \in \mathcal{M}, \\ 0 & \text{otherwise} \end{cases}$$

Model:

Minimize

$$\sum_{i \in \mathcal{L}} \sum_{j \in \mathcal{M}} \sum_{k \in \mathcal{N}} c_{ij} x_{ijk}$$

subject to

$$\begin{cases} \sum_{j \in \mathcal{M}} \sum_{k \in \mathcal{N}} x_{ijk} = 1 & i \in \mathcal{L} \\ \sum_{i \in \mathcal{L}} x_{ijk} \leq 1 & k \in \mathcal{N}, j \in \mathcal{M}, \\ x_{ijk} \in \{0, 1\} & i \in \mathcal{L}, j \in \mathcal{M}, k \in \mathcal{N} \end{cases}$$

```
[ ]: cost = {
    1: {"Mon": 10, "Tue": 11, "Wed": 12, "Thu": 13, "Fri": 14},
    2: {"Mon": 12, "Tue": 14, "Wed": 16, "Thu": 18, "Fri": 20},
    3: {"Mon": 17, "Tue": 18, "Wed": 17, "Thu": 18, "Fri": 17},
    4: {"Mon": 20, "Tue": 19, "Wed": 18, "Thu": 17, "Fri": 16},
    5: {"Mon": 22, "Tue": 22, "Wed": 22, "Thu": 22, "Fri": 33},
    6: {"Mon": 24, "Tue": 23, "Wed": 22, "Thu": 23, "Fri": 23},
    7: {"Mon": 9, "Tue": 6, "Wed": 8, "Thu": 7, "Fri": 9}
}

turbines = list(cost.keys())
days = ["Mon", "Tue", "Wed", "Thu", "Fri"]
teams = [1, 2]
```

```
[1]: import pulp as pl

# --- Set di indici ---
turbines = range(1, 8) # 1..7
days = ["Mon", "Tue", "Wed", "Thu", "Fri"]
teams = [1, 2]

# --- Parametri dei costi c_ij ---
# tabella dei costi per turbine x giorni
cost_table = {
    1: {"Mon": 10, "Tue": 11, "Wed": 12, "Thu": 13, "Fri": 14},
    2: {"Mon": 12, "Tue": 14, "Wed": 16, "Thu": 18, "Fri": 20},
    3: {"Mon": 17, "Tue": 18, "Wed": 17, "Thu": 18, "Fri": 17},
    4: {"Mon": 20, "Tue": 19, "Wed": 18, "Thu": 17, "Fri": 16},
    5: {"Mon": 22, "Tue": 22, "Wed": 22, "Thu": 22, "Fri": 33},
    6: {"Mon": 24, "Tue": 23, "Wed": 22, "Thu": 23, "Fri": 23},
    7: {"Mon": 9, "Tue": 6, "Wed": 8, "Thu": 7, "Fri": 9},
}

# --- Modello ---
prob = pl.LpProblem("Wind_Turbine_Maintenance", pl.LpMinimize)
```

```

# --- Variabili  $x_{\{i,j,k\}}$  binarie ---
x = pl.LpVariable.dicts(
    "x",
    (turbines, days, teams),
    lowBound = 0,
    upBound = 1,
    cat = pl.LpBinary
)

# --- Funzione obiettivo:  $\min \sum_{\{i,j,k\}} c_{ij} * x_{ijk}$  ---
prob += pl.lpSum(
    cost_table[i][j] * x[i][j][k]
    for i in turbines
    for j in days
    for k in teams
)

# --- Vincoli ---

# 1) Ogni turbina è mantenuta esattamente una volta nella settimana
for i in turbines:
    prob += pl.lpSum(x[i][j][k] for j in days for k in teams) == 1,
    <f"one_maintenance_turbine_{i}"

# 2) Ogni squadra lavora su al massimo una turbina al giorno
for j in days:
    for k in teams:
        prob += pl.lpSum(x[i][j][k] for i in turbines) <= 1,
        <f"capacity_team_{k}_{j}"

# --- Risoluzione ---
prob.solve(pl.PULP_CBC_CMD(msg=0))

print("Status:", pl.LpStatus[prob.status])
print("Costo totale ottimo:", pl.value(prob.objective))

# Stampa della soluzione
for i in turbines:
    for j in days:
        for k in teams:
            if pl.value(x[i][j][k]) == 1:
                print(f"Turbina {i} mantenuta il {j} dalla squadra {k}")

```

```

Status: Optimal
Costo totale ottimo: 105.0
Turbina 1 mantenuta il Mon dalla squadra 1

```

Turbina 2 mantenuta il Mon dalla squadra 2  
Turbina 3 mantenuta il Fri dalla squadra 1  
Turbina 4 mantenuta il Fri dalla squadra 2  
Turbina 5 mantenuta il Thu dalla squadra 1  
Turbina 6 mantenuta il Wed dalla squadra 1  
Turbina 7 mantenuta il Tue dalla squadra 1