TECH4 Home Exam Solution

November 15, 2025

Part A

Task 1 - Model Formulation

Electrical Stations Location Problem

As an expert I am asked where to build the electrical stations in such a way that each neighbrouhood will be connected to one station and the total cost will be minimized.
I need to serve 20 neighbrouhoods. And there are 10x10 = 100 possible locations where to build electrical stations. Total cost has two parts, the fixed installation cost and a variable cost per km of wiring. Fixed costs can be calculated by summing the respective installation costs of each built station. The variable cost will be computed by multiplying the total distance of wiring by the unti cost.

Parameters

Let us have:

$\mathcal{I}$	Set of neighbrouhoods
$\mathcal{J}$	Set of possible electrical stations' locations
f_j	Fixed cost of building station $j \in \mathcal{J}$.
d_{ij}	Distance from neighbrouhood $i \in \mathcal{I}$ to electrical station $j \in \mathcal{J}$
var_cst	Cost of electrical wiring per km

Decision Variables

Introduce two sets of binary variables:

$$x_j := \begin{cases} 1 & \text{if station } j \text{ is built,} \\ 0 & \text{otherwise,} \end{cases} \quad \forall j \in \mathcal{J} \tag{1}$$

$$y_{ij} := \begin{cases} 1 & \text{if neighbrouhood } i \text{ is served at station } j, \\ 0 & \text{otherwise,} \end{cases} \quad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{2}$$

The Integer Linear Optimization model:

$$
\begin{aligned}
\min_{x,y} \quad & \sum_{j \in \mathcal{J}} f_j x_j + var\_cst \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} d_{ij} y_{ij} && && (3)\\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} y_{ij} = 1, && \forall i \in \mathcal{I} \quad \text{(every neighbrouhood is served)} && (4)\\
& y_{ij} \leq x_j, && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \quad \text{(station built before use)} && (5)\\
& x_j \in \{0, 1\}, && \forall j \in \mathcal{J} && (6)\\
& y_{ij} \in \{0, 1\}, && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} && (7)
\end{aligned}
$$

Task 2 - Electrical stations' location

It is recommended to built two electrical stations located at $E2$ and $G7$.
Total cost is : $8,810.97

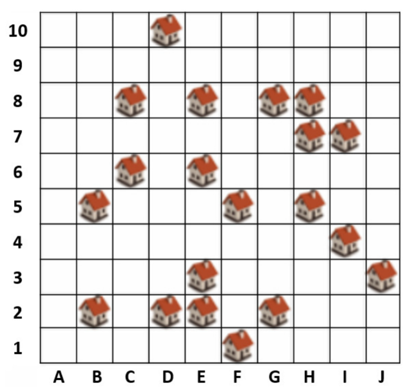

Figure 1: Station locations to serve all the neighbrouhoods

Task 3 - Only one station

In case when we wish to allow only for 1 station to exist, the model from task 1 hold itself. We need to add new constraint, that sum of all the built stations is 1.

$$\sum_{j \in \mathcal{J}} x_j = 1 \tag{8}$$

Integer Linear Model - exactly one station to built

$$\min_{x,y} \quad \sum_{j \in \mathcal{J}} f_j x_j + var\_cst \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} d_{ij} y_{ij} \tag{9}$$

$$\text{s.t.} \quad \sum_{j \in \mathcal{J}} y_{ij} = 1, \qquad \forall i \in \mathcal{I} \quad \text{(every neighbrouhood is served)} \tag{10}$$

$$\sum_{j \in \mathcal{J}} x_j = 1 \qquad \text{(exactly one station is built)} \tag{11}$$

$$y_{ij} \leq x_j, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \quad \text{(station built before use)} \tag{12}$$

$$x_j \in \{0, 1\}, \qquad \forall j \in \mathcal{J} \tag{13}$$

$$y_{ij} \in \{0, 1\}, \qquad \forall i \in \mathcal{I}, \forall j \in \mathcal{J} \tag{14}$$

Recommended to built station at location $F4$.
Total cost is $8,840.19. Slightly more than when building two stations.

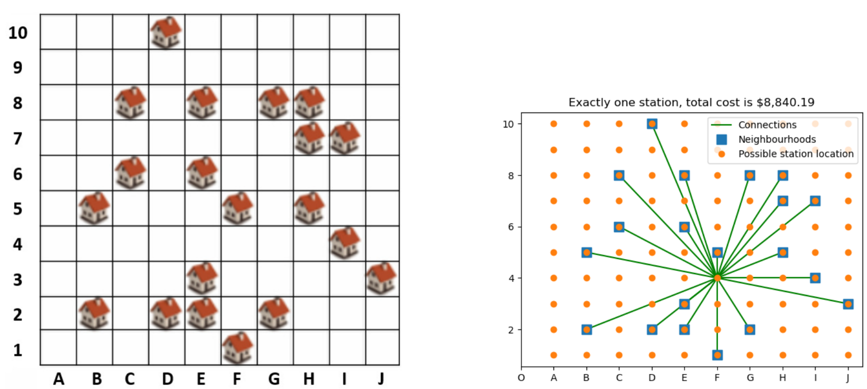

Figure 2: One station to serve all

Task 4 - Fraction of neighbrouhoods to connect

For case when only fraction of neighbrouhoods is served we need to relax the constraint (4) from model in Task 1 and allow the value 0 as well. As we now do not require that every neighbrouhood was served, but that each neighbrouhood was served at most once. And we will intruduce the new parameter b.

$b \in \{0, 1, \ldots, |\mathcal{I}|\}$ minimum number of neighborhoods to serve

$$\sum_{j \in \mathcal{J}} y_{ij} \leq 1 \quad \forall i \in \mathcal{I} \quad \text{(each neighbrouhood is served at most once)} \tag{15}$$

$$\sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} y_{ij} \geq b \qquad \text{(at least } b \text{ customers served)} \tag{16}$$

Integer Linear Model - at least b neighbrouhoods connected

$$\begin{aligned}
\min_{x,y} \quad & \sum_{j \in \mathcal{J}} f_j x_j + var_cst \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} d_{ij} y_{ij} && && (17) \\
\text{s.t.} \quad & \sum_{j \in \mathcal{J}} y_{ij} \leq 1, && \forall i \in \mathcal{I} && (18) \\
& \sum_{i \in \mathcal{I}} \sum_{j \in \mathcal{J}} y_{ij} \geq b && && (19) \\
& y_{ij} \leq x_j, && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} && (20) \\
& x_j \in \{0, 1\}, && \forall j \in \mathcal{J} && (21) \\
& y_{ij} \in \{0, 1\}, && \forall i \in \mathcal{I}, \forall j \in \mathcal{J} && (22)
\end{aligned}$$

When fraction of neighborhoods is to be served the following results were found:

Lower bound	Total cost	Station location
14	\$5,846.71	$G5$
16	\$6,683.28	$G5$
18	\$7,683.28	$G5$
20	\$8,810.97	$G7$, $E2$

Table 1: Connection fraction of neighborhoods costs and station location.

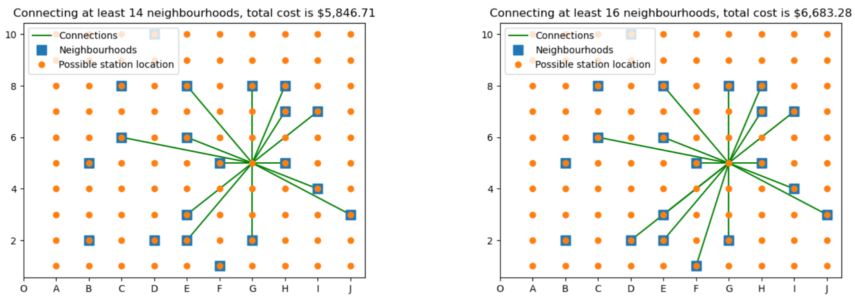

Figure 3: Connect fraction of the neighborhoods - 14, 16

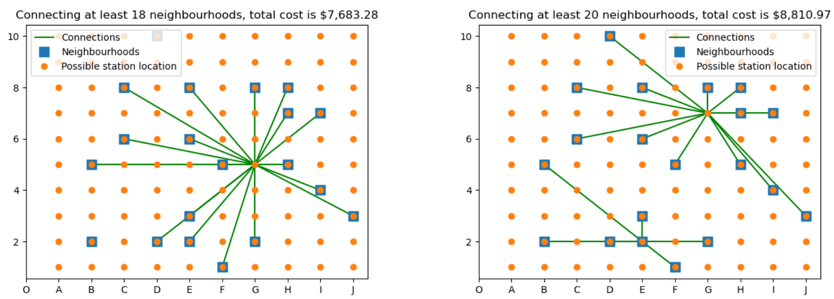

Figure 4: Connect fraction of the neighborhoods - 18, 20

It is interesting to note that the third model to connect at least b neighborhoods, in case of b = 20 (all the neighborhoods), found the same solution as the first model that connects all the neighborhoods. It is one of the indications that models are formulated and implemented correctly.

Note

The solution was inspired by 3.6 Facility location problem from MO book. https://mobook.github.io/MO-book/notebooks/03/06-facility-location.html

Part B

Task 1 - Network Optimization Problem Formulation

Throughout, we use abbreviation for city names. City names will be abbreviated to their first letter except for the cities which names end in $-most$. These will be abbreviated to first letter plus letter M (note that this identifies the cities uniquely).

The network (G, w) is as follows:

- The set of nodes is $V(G) = \{N, E, S, W, NM, EM, SM, WM\}$.
- The set of edges is

$$E(G) = \{ \{NM, N\}, \{NM, W\}, \{N, E\}, \{N, EM\}, \{N, S\}, \{N, W\}, \{E, EM\}, \{E, S\}, \{E, SM\}, \{S, SM\}, \{S, W\}, \{S, WM\}, \{SM, W\}, \{W, WM\} \}.$$

  These edges are *undirected*.
- For each edge $e \in E(G)$, denote by $d(e)$ the distance given in the table and by $t(e)$ the toll revenue recieved in the table. The weight of each edge e is

$$w(e) = 0.5 \cdot d(e) - t(e).$$

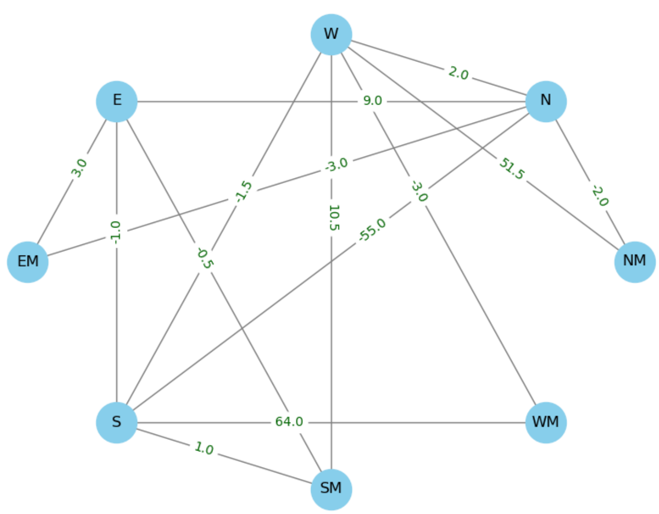

Figure 5: Network visualization.

Problem formulation

We are given a connected, as shown in Figure 5, network with edge costs, and our task is to select the cheapest set of edges that still connect everything. We are solving a minimum spanning tree problem.

We visualized the network to see if it is connected. Another way is to drop the weights, and run BFS on unweighted network and see if number of visited nodes is equal to the number of nodes in the graph.

Input: Undirected connected network $((G, w), N)$, where $w : E(G) \to \mathbb{R}$ is a cost function. And N is deliberately chosen start node. (As we need to connect all cities, we are free to choose randomly).

Task: Find a spanning tree of minimum total cost in (G, w).

Output: A spanning tree $T \subseteq E$ of G with the minimum possible sum $\sum_{e \in T} w_e$ of edge costs.

Tree will represent the roads that should be built. And minimum possible sum of edge costs is our profit/loss.

Task 2 - Implementation results

Total net profit of the project:

Scenario	Profit
5 years utilization	66
10 years utilization	347
20 years utilization	948.50
5 years, no road Northern Southern	9

Table 2: Total net profit of the project under different scenarios, in million $.

Roads that should be built

5 years utilization
Cities to connect: [('N', 'S', -55.0), ('N', 'EM', -3.0), ('N', 'NM', -2.0), ('S', 'W', -1.5), ('W', 'WM', -3.0), ('S', 'E', -1.0), ('E', 'SM', -0.5)]

10 years utilization
Cities to connect: [('N', 'S', -115.0), ('N', 'EM', -81.0), ('EM', 'E', -32.0), ('N', 'W', -26.0), ('W', 'WM', -53.0), ('S', 'SM', -24.0), ('N', 'NM', -16.0)]

20 years utilization
Cities to connect: [('N', 'EM', -237.0), ('N', 'S', -235.0), ('N', 'E', -114.0), ('N', 'W', -82.0), ('W', 'WM', -153.0), ('S', 'SM', -74.0), ('W', 'NM', -53.5)]

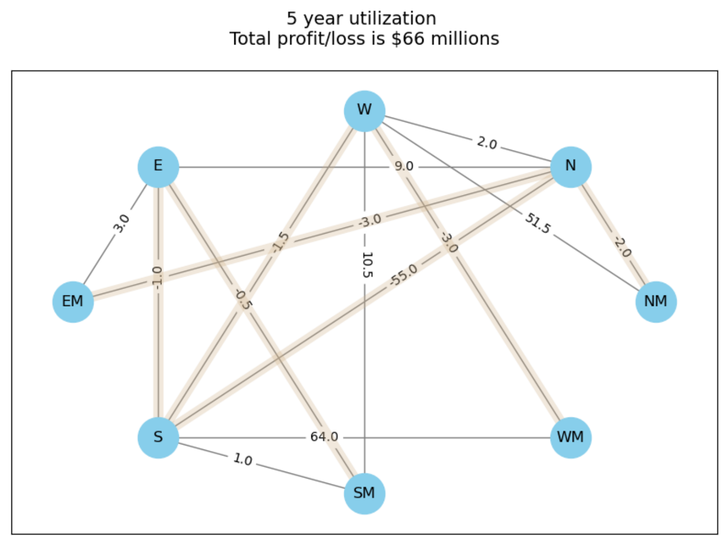

Figure 6: Roads that should be build under 5 years utilization

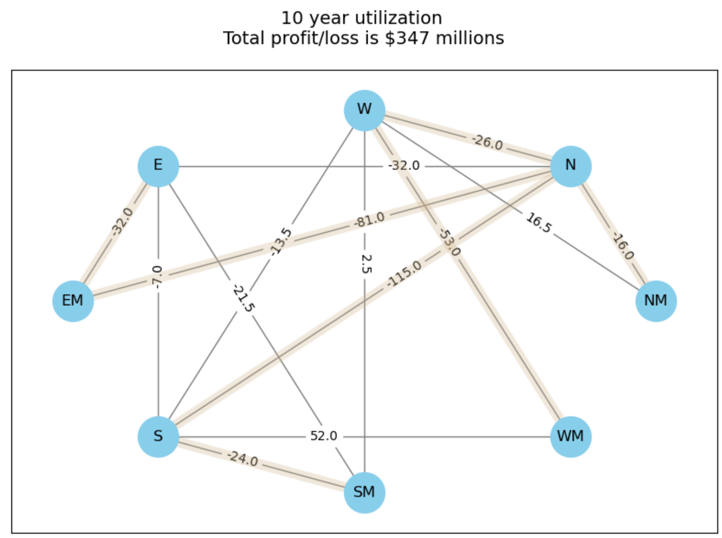

Figure 7: Roads that should be build under 10 years utilization

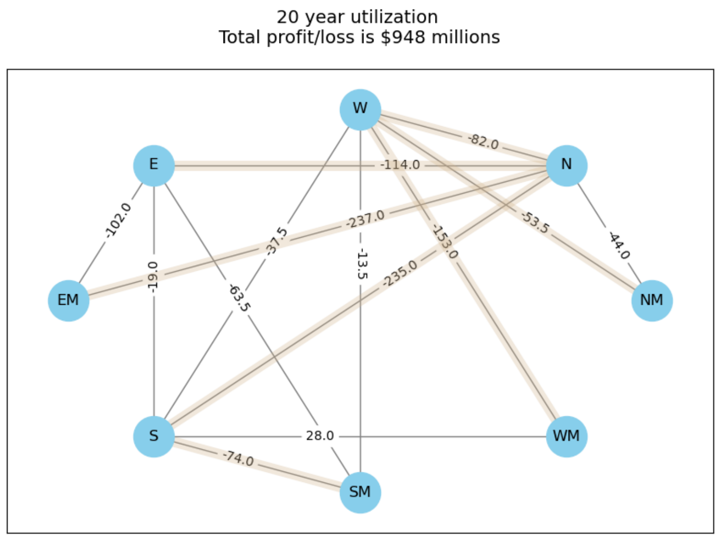

Figure 8: Roads that should be build under 20 years utilization

Task 3 - No road between Northern and Southern

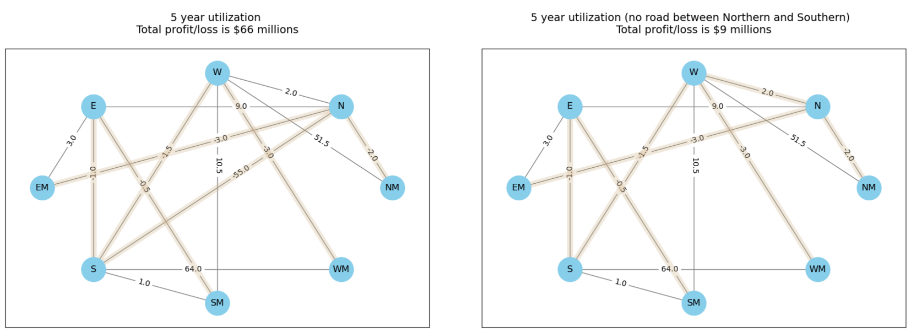

Figure 9: Road to replace Northern-Southern

As can be seen on Figure 9, if the road Northern-Southern is removed, the network is no longer connected. If we imagine that our network is a pearl necklace and we will try to pick it up by node N, we will pick up only the EM-N-NM part. The rest will be left behind because necklace is not connected. There are three ways how to connect the network:

- EM - E with cost 3.0
- W - N with cost 2.0
- N - NM with cost 51.5

As the goal is to minimize cost, edge with the lowest cost should be selected. In this case W - N.

There already is the implementation of finding the minimum spanning tree from connected network from task 2. Edge N-S was removed from the original network. And the implementation was run on this altered network to find new minimum spannig tree for this specific scenario. Found MST, shown on Figure 9 (on the right), confirms our choice.

Implementation Notes

Implementation project is organized to two parts, part A and part B. Each part has several scripts. This is done to have some structure and some clear orientation in such a large project. Each part has scripts that deal with data (tranform, calculate), algorithm or models, visualization scirpt and the solution script. Solution scripts bring all parts together to solve the tasks.

And there is also *env.yml* file with dependencies for easier rerunning, sharing. It contains far more dependencies than needed for this project as it is authors env for tech4 and different side playground projects.

Project structure

```
home exam
 ├── part_A
 │    ├── models_A.py
 │    ├── util_generate_data.py
 │    ├── util_visualize_A.py
 │    └── solution_tasks_A.py
 ├── part_B
 │    ├── prim_alg_B.py
 │    ├── util_generate_data.py
 │    ├── util_visualize_B.py
 │    └── solution_tasks_B.py
 ├── home_exam_report.pdf
 └── env.yml
```