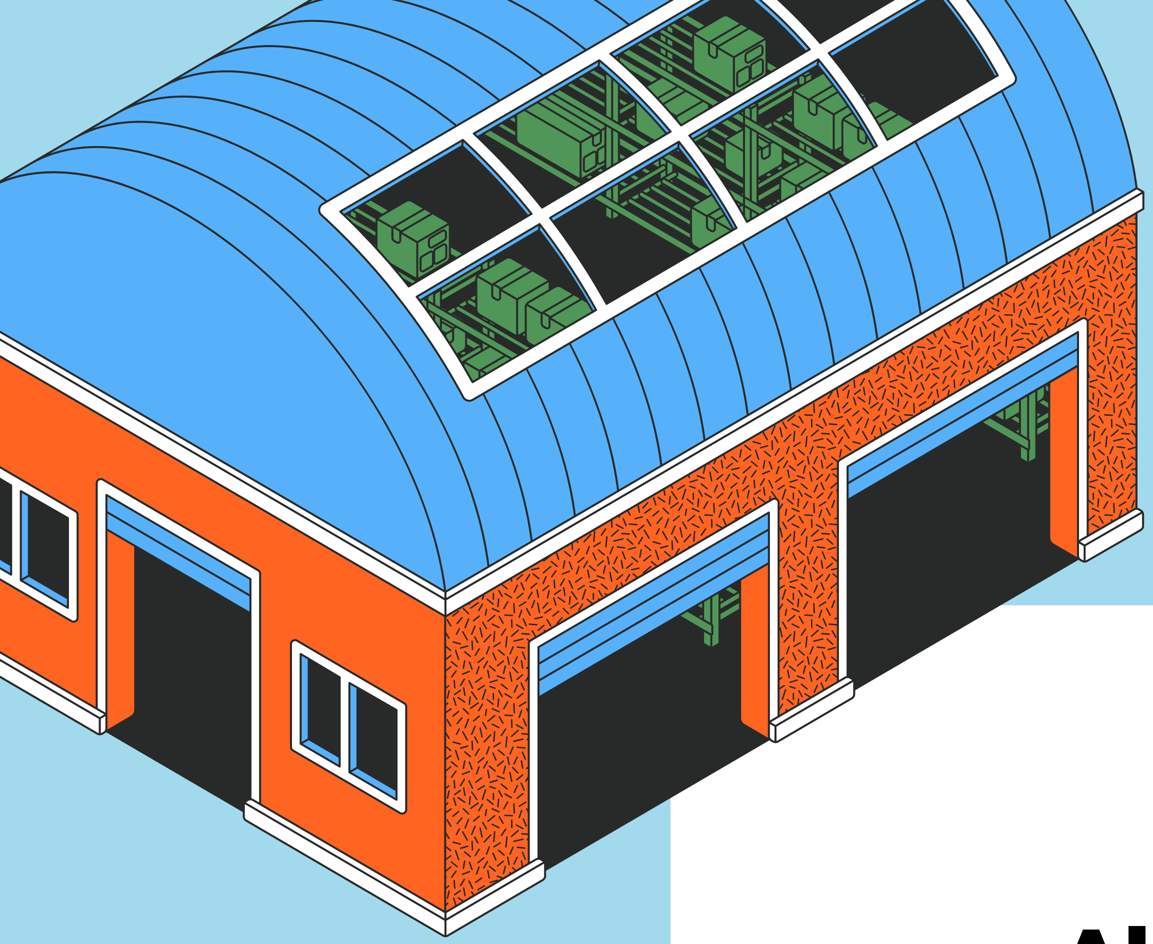
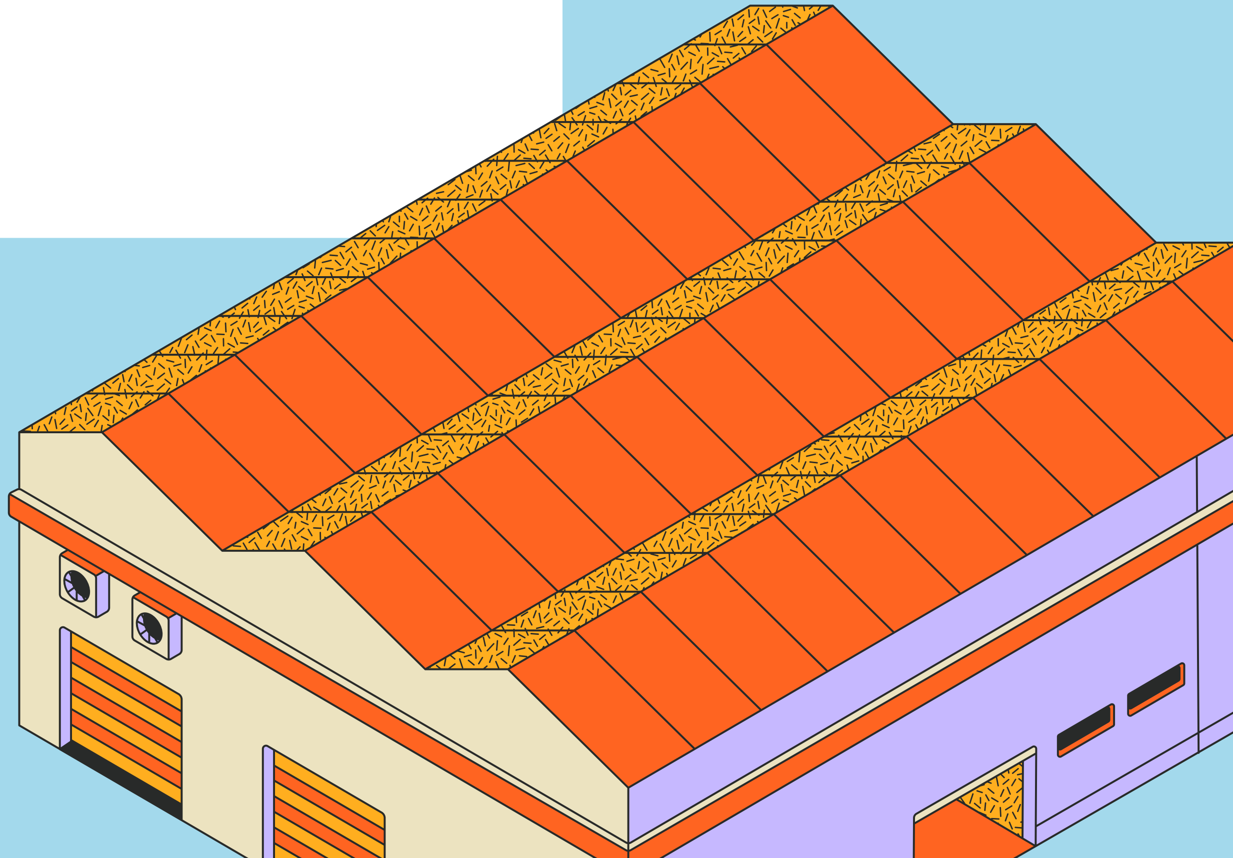


SELECTED TOPICS ON OPTIMIZATION

Aldo Maciel Pardo Cardenas - 1993746
Carlos Yair Hernandez Hernandez - 1952660
Saúl Emmanuel Carreón Alanis - 1901555

P - CENTER PROBLEM



INTRODUCTION

TSO

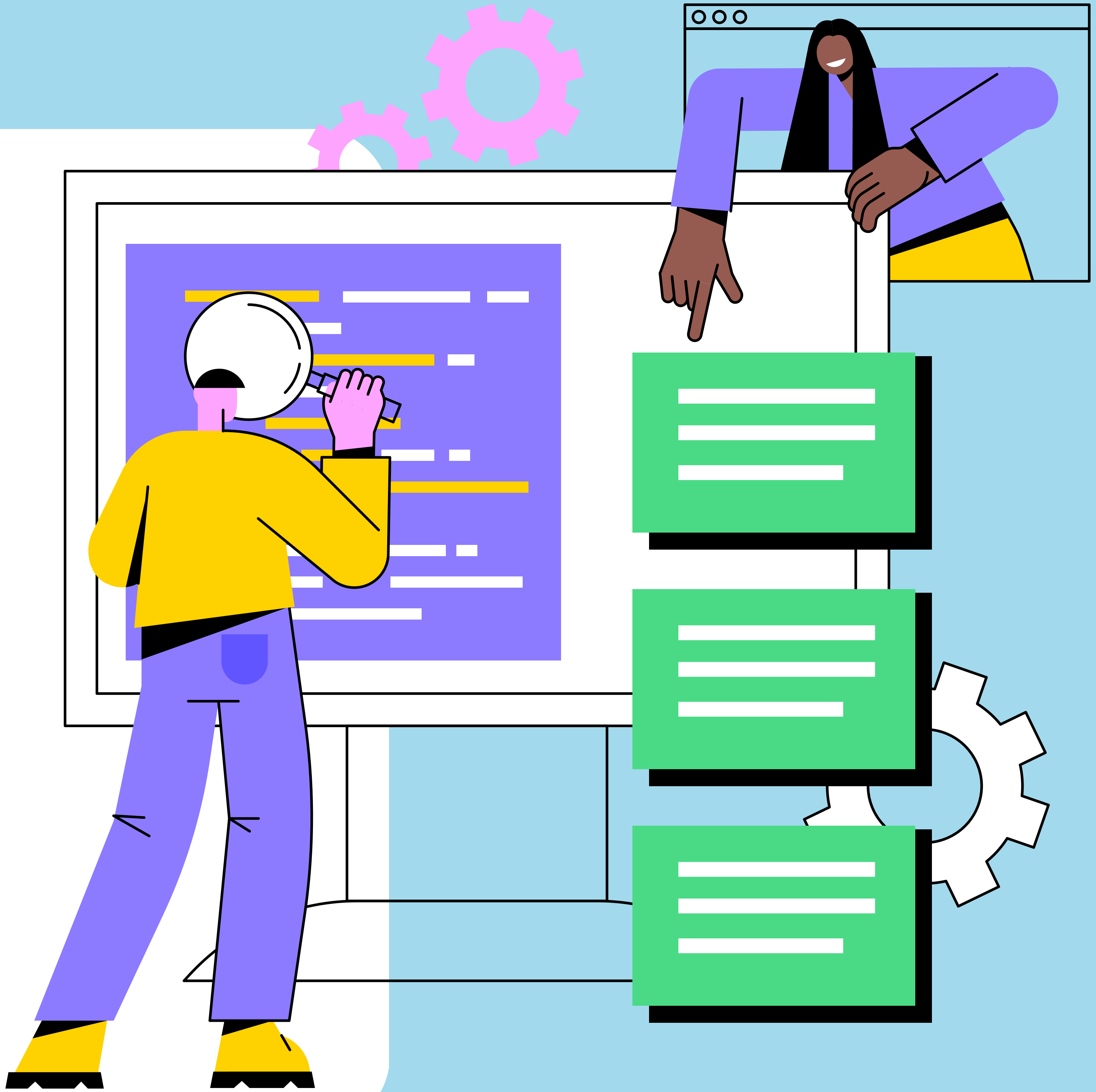
Selected Optimization Topics focus on analyzing optimization and simulation problems in professional practice. This involves finding alternative solutions using heuristic methods or effective combinatorial algorithms.

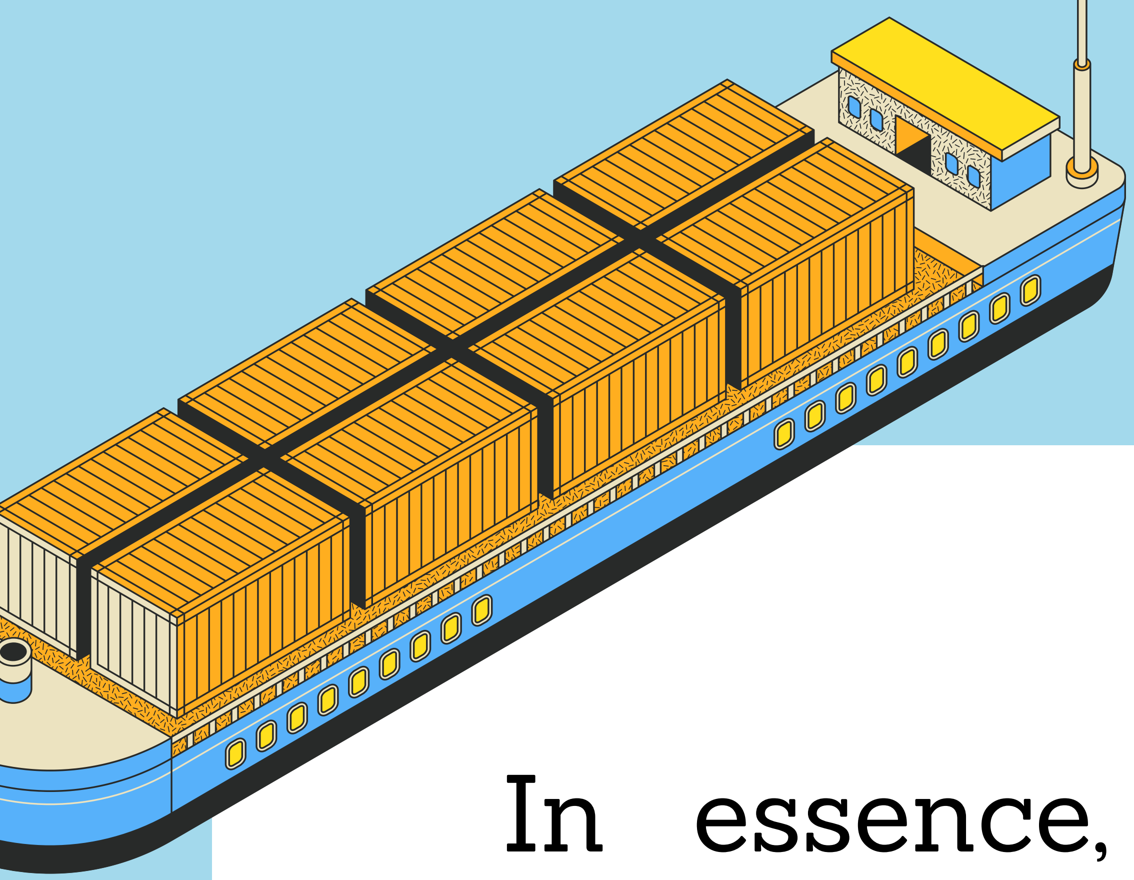
P-CENTER PROBLEM

The p-center problem is a fundamental problem in the field of location science. In this problem, we are given customer demand points and potential facility locations, and our goal is to choose p of these locations to open facilities in a way that minimizes the maximum distance between any customer demand point and its nearest open facility.

Given a set of n demand points (customers) in the plane, along with a weight associated with each demand point, the objective is to find p new facilities in the plane that minimize the weighted maximum Euclidean distance between each demand point and its nearest new facility.

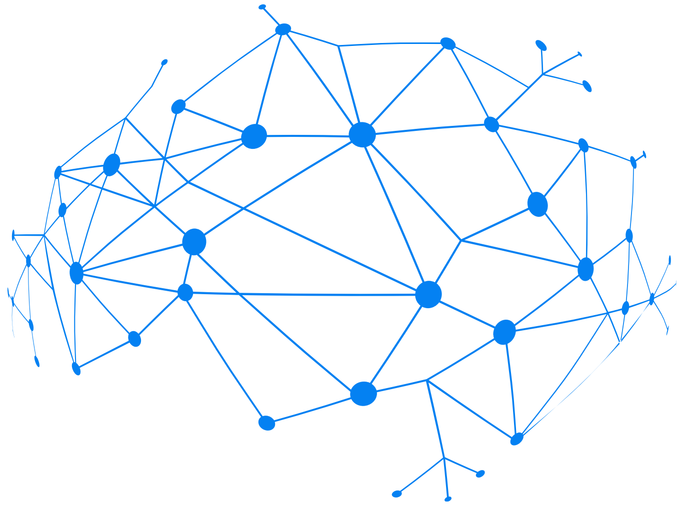
PROBLEM DESCRIPTION





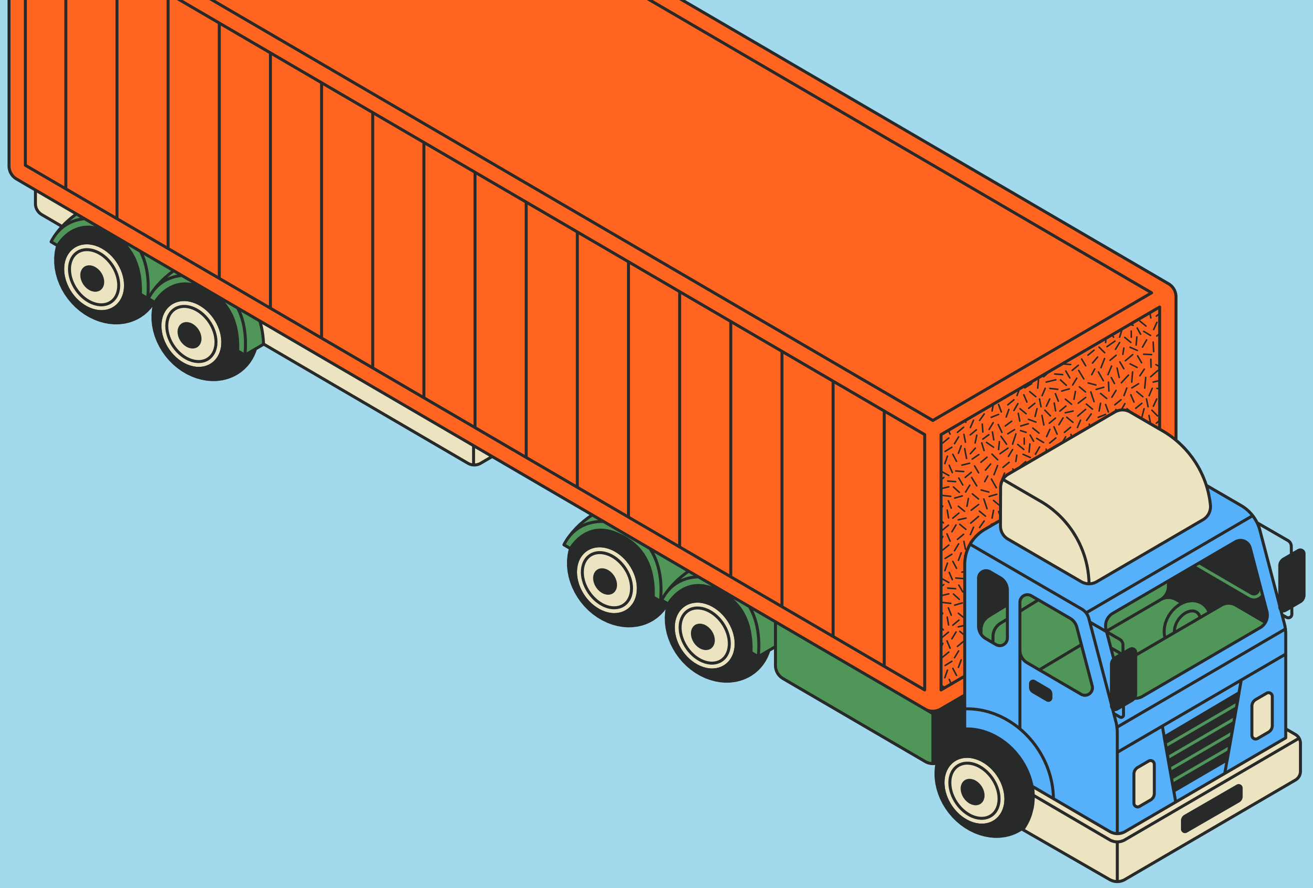
In essence, the task involves selecting P facility locations within a network of nodes (demand points) and edges (distances between nodes) in such a way that the maximum distance any demand point has to its nearest facility (the 'service radius') is minimized.

GIVEN



A network represented by a set of nodes N = {1, 2, ..., n} where each node i represents a demand point or location.

The distance or cost dij between node i and j, representing the distance (or cost) required to serve demand point j from a potential facility located at node i.

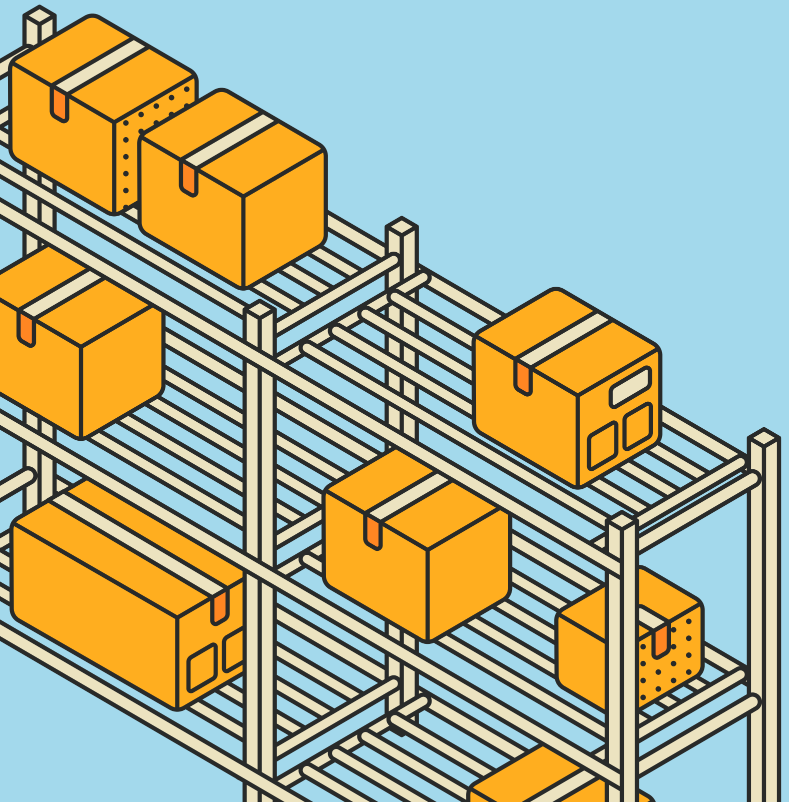


OBJECTIVE

Determine the optimal placement of P facilities among the nodes to minimize the maximum distance (or cost) from any demand point to its nearest facility.

DECISION VARIABLES

Let xi = 1 if a facility is located at node i, and xi = 0 otherwise. Therefore, xi is a binary variable indicating facility location.



MATHEMATICAL MODEL

MINIMIZE

$$max_{i \in N} \{min_{i \in F}\, d_{ii} \; * \; x_i\}$$

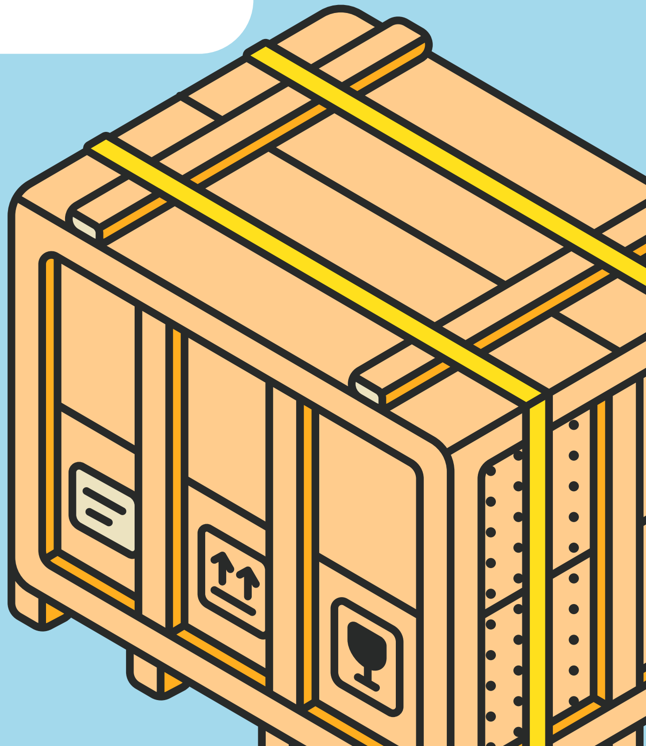
SUBJECT TO

1.Each demand point j must be assigned to exactly one facility:

$$\Sigma_{i \in F}\, x_i = 1 \qquad \forall j \in N$$

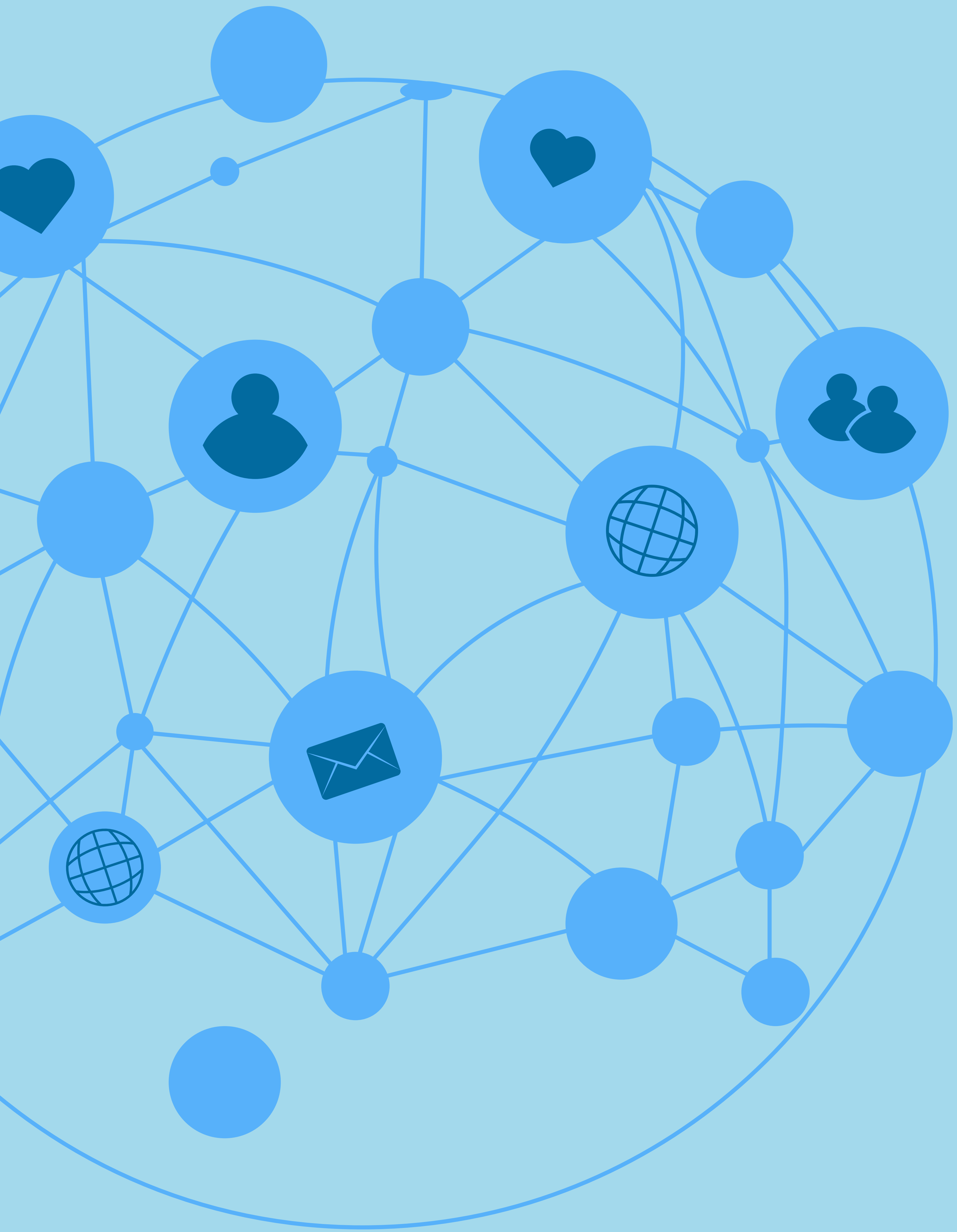
2.The total number of facilities to be located is P:

$$\Sigma_{i \in N}\, x_i = P$$



WHERE

F denotes the set of nodes where facilities are located (|F| = P).



FORMULATION

1

The first constraint ensures that each demand point j is assigned to exactly one facility, meaning that for each j, there exists exactly one i (from F) such that xi = 1, indicating that facility i serves demand point .

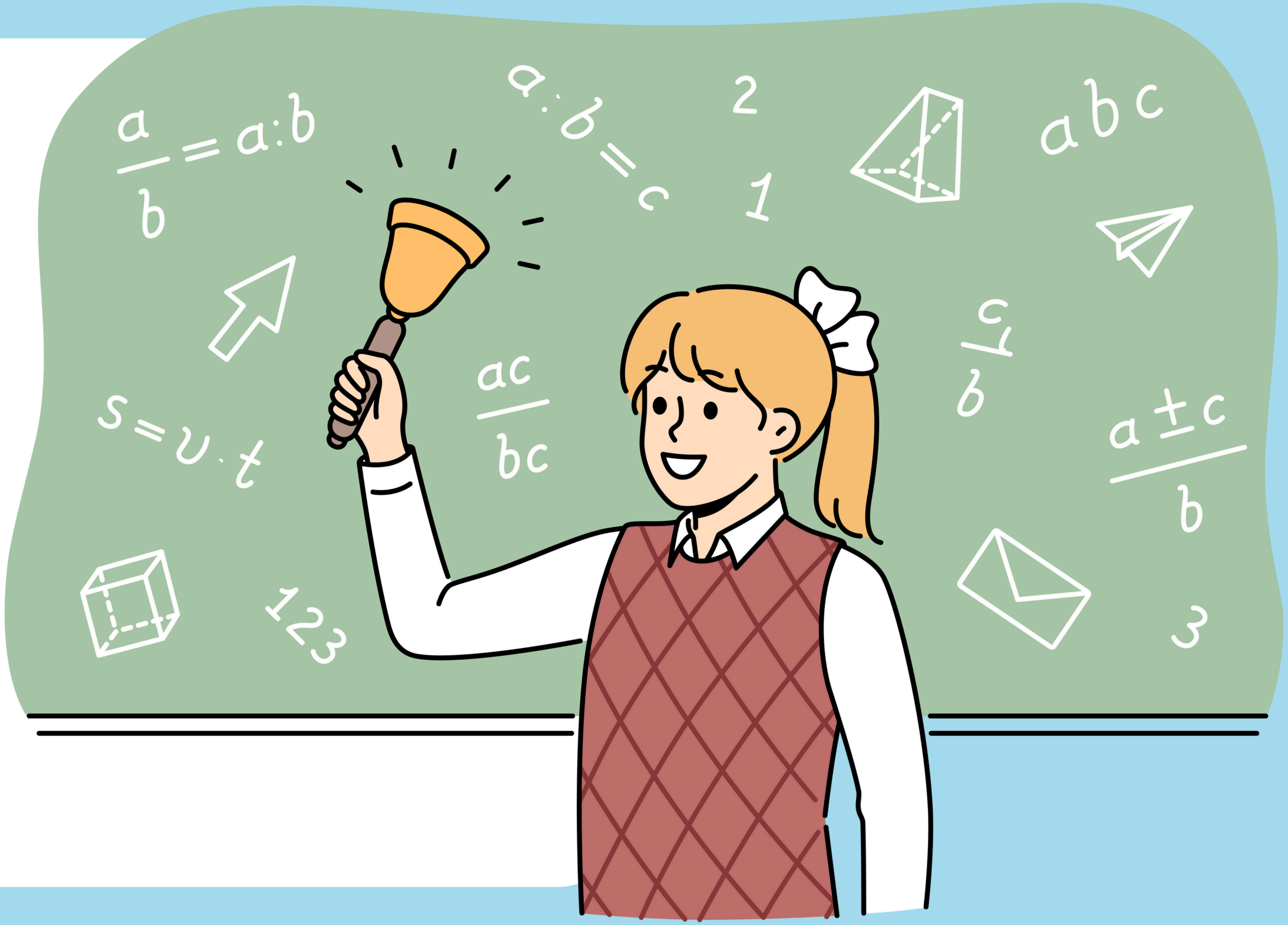
2

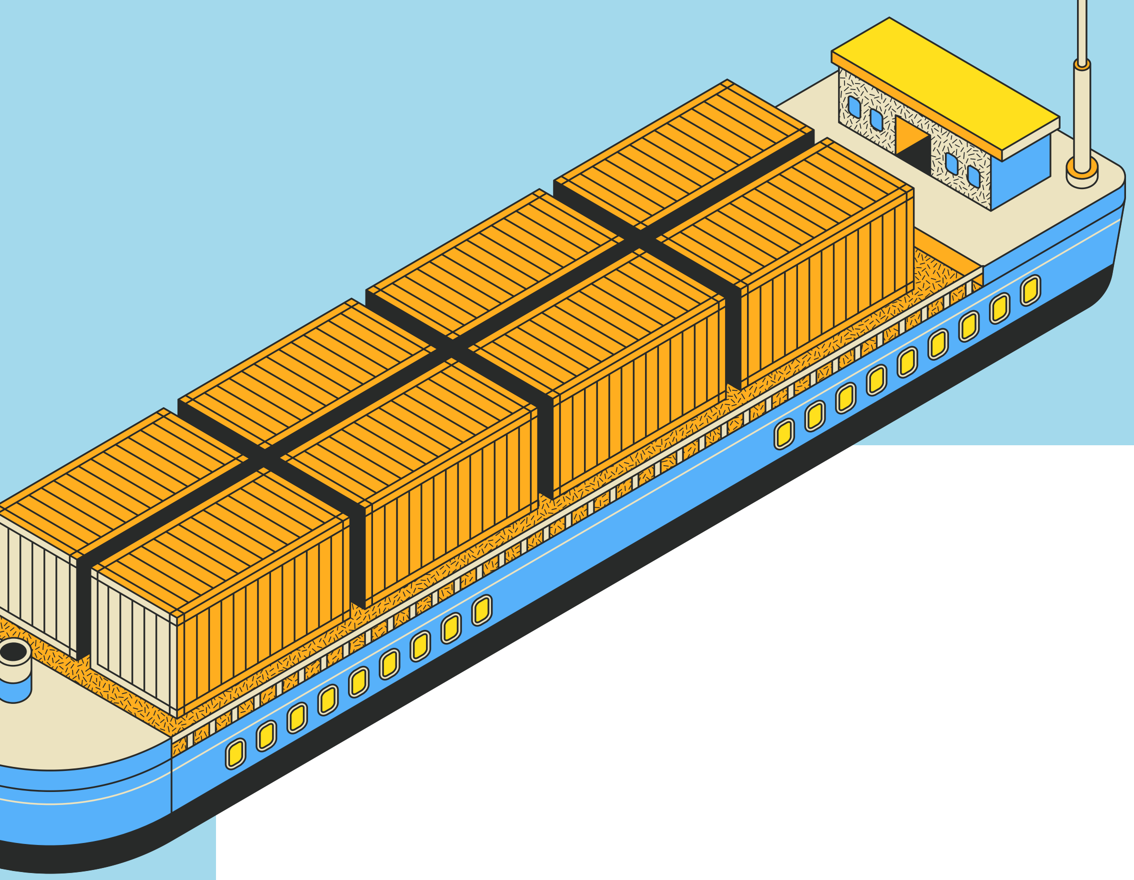
The second constraint dictates that exactly P facilities must be located among all nodes.

3

The goal is to find the values of xi(facility locations) that minimize the maximum distance from any demand point to its nearest facility, while satisfying the constraints.

PROBLEM EXAMPLE





Data of problem

1. N: The set of nodes where each node i represents a demand point.
2. dij: The distance between node i and node j.
3. P: Number of distribution centers to be located..

Decision variables.

z: Variable that represents the maximum distance from any city to its nearest distribution center.

xj: Binary variable that indicates whether a distribution center is located in the city j (xj=1xj=1) o no (xj=0xj=0).

yi: Binary variable that indicates whether the city is assigned to a distribution center
(yi=1) o no (yi=0).

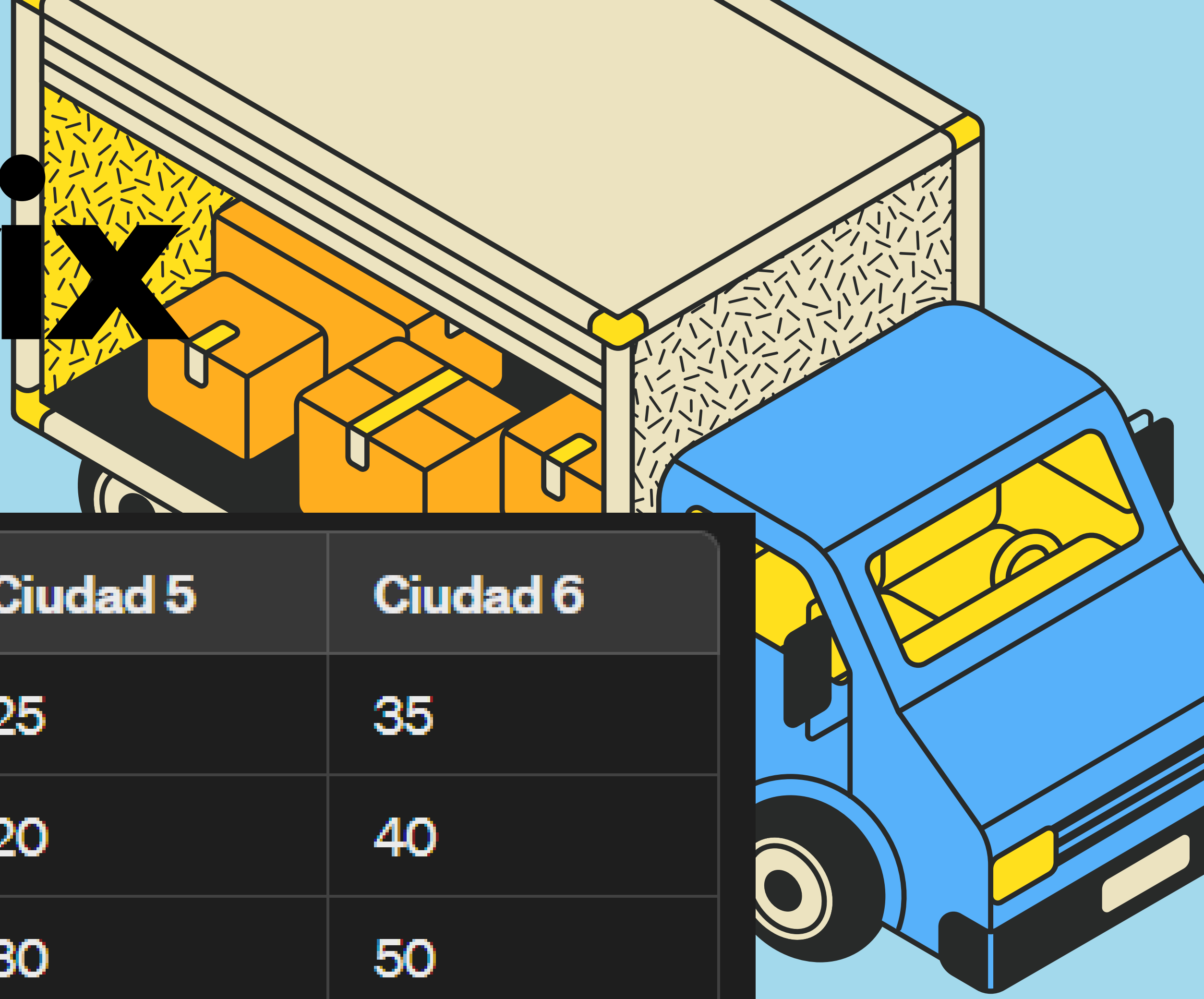
MATHEMATICAL FORMULATION

Minimize z

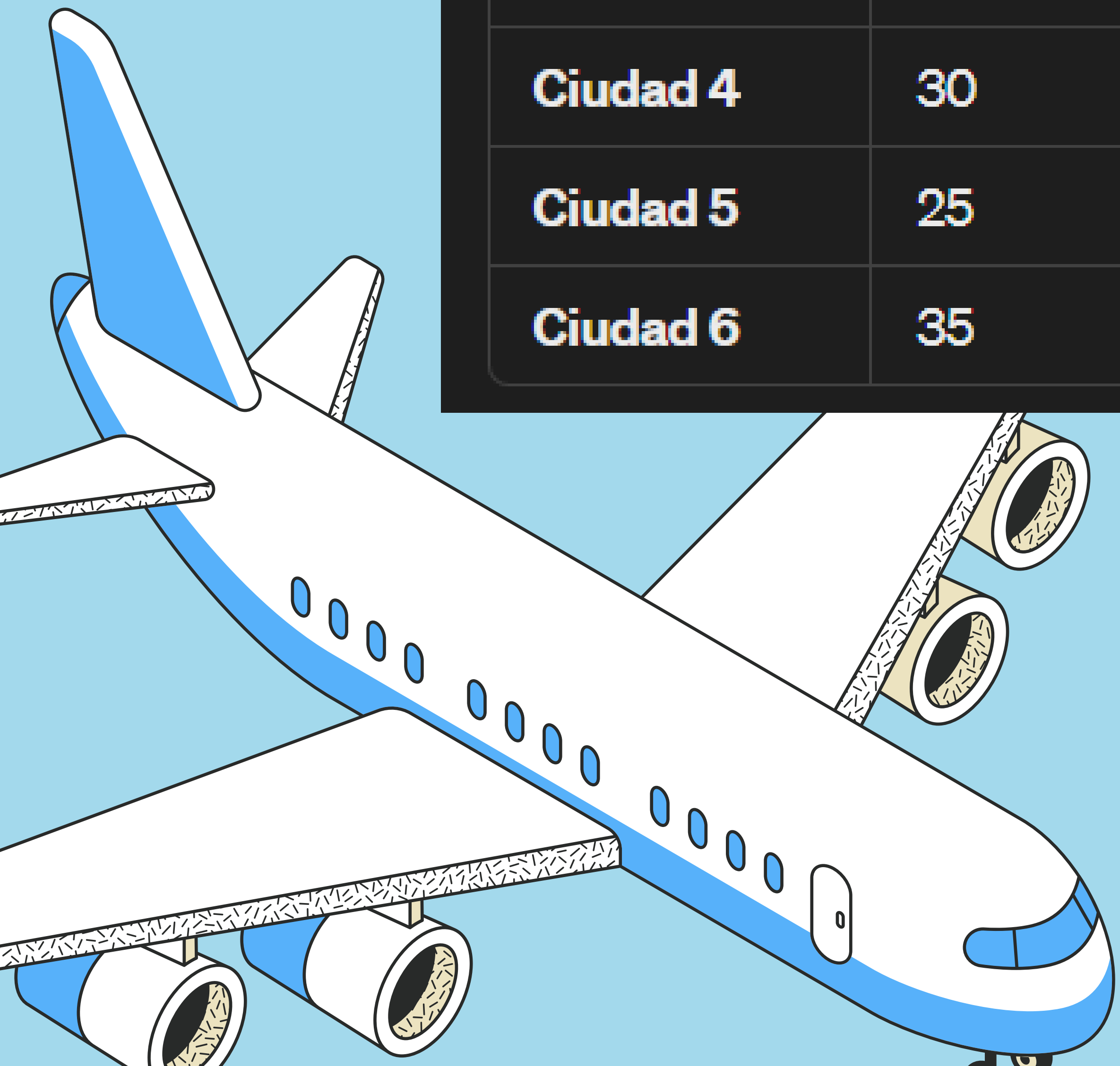
subject to:

- $\sum_{j=1}^{n} x_j = P$ *(Exactly P centers must be located)*
- $\sum_{j=1}^{n} d_{ij} x_j \leqslant z$ *(The distance from each city to the nearest center must be less than or equal to z)*
- $x_j \in \{0,1\} \forall j$
- $y_i \in \{0,1\} \forall i$
- $z \geqslant 0$

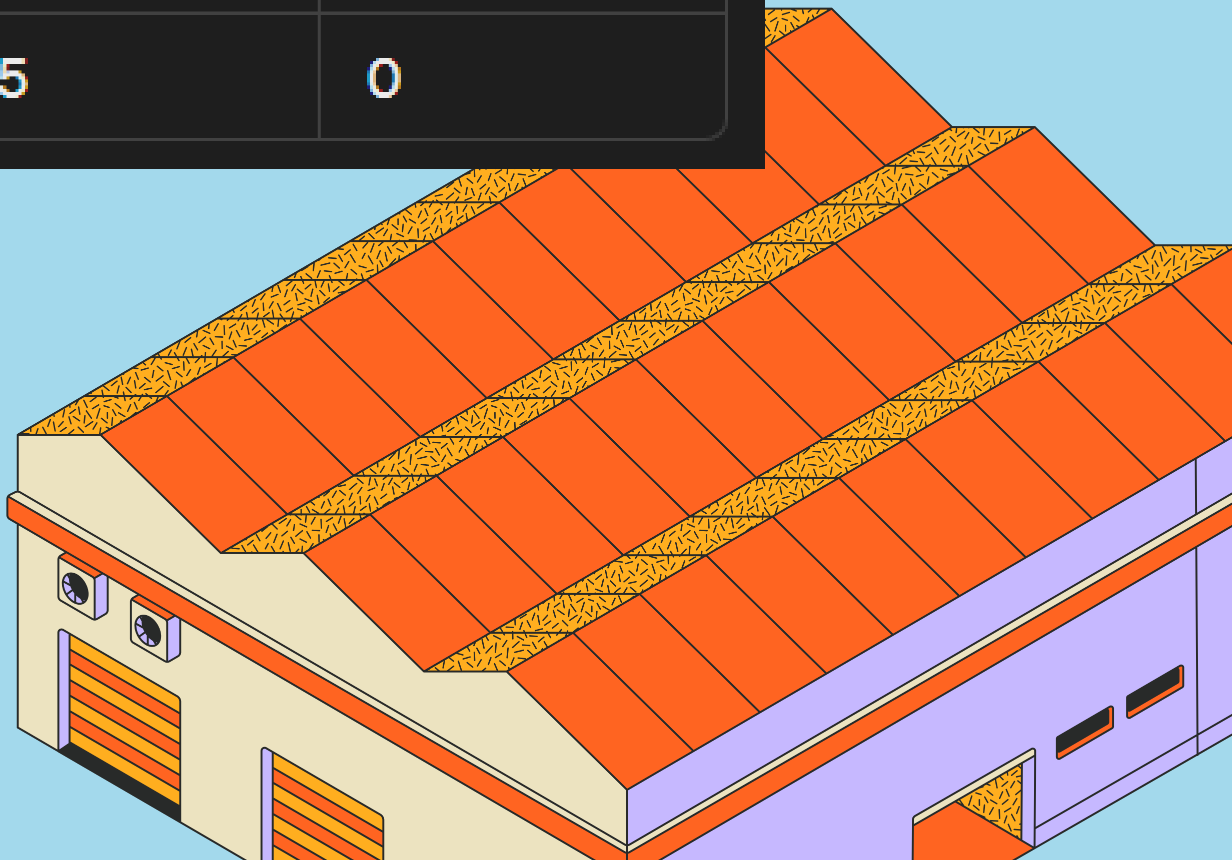
Distance matrix



	Ciudad 1	Ciudad 2	Ciudad 3	Ciudad 4	Ciudad 5	Ciudad 6
Ciudad 1	0	10	20	30	25	35
Ciudad 2	10	0	15	35	20	40
Ciudad 3	20	15	0	25	30	50
Ciudad 4	30	35	25	0	15	20
Ciudad 5	25	20	30	15	0	45
Ciudad 6	35	40	50	20	45	0



We want to locate P=2 distribution centers.

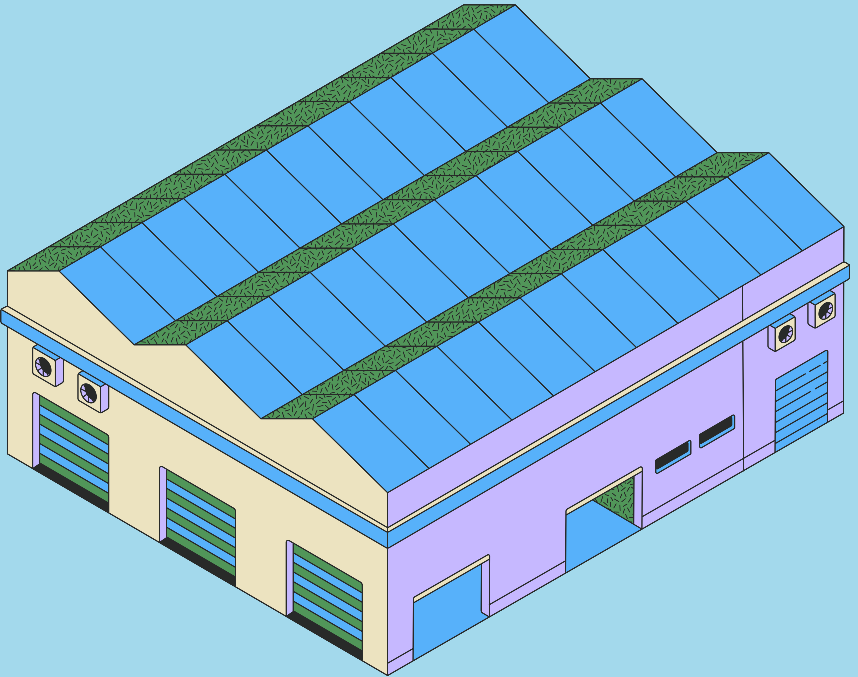


Evaluate all possible combinations of 2 centers

1. Centers in city 1 and city 2
    - City 1: Distance to its nearest center = 0
    - City 2: Distance to its nearest center = 0
    - City 3: Distance to its nearest center = 15 (City center 2)
    - City 4: Distance to its nearest center = 30 (City center 1)
    - City 5: Distance to its nearest center = 20 (City center 2)
    - City 6: Distance to its nearest center = 35 (City center 1)
    - Maximum distance = 35

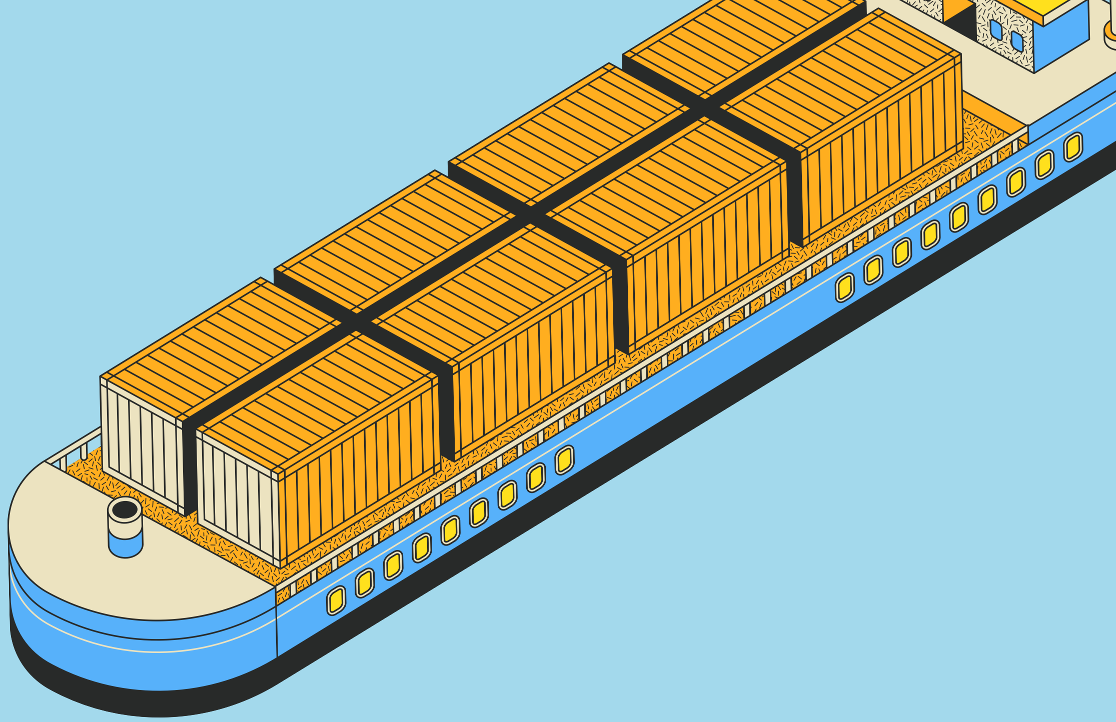
2. Centers in city 1 and city 3
    - City 1: Distance to its nearest center = 0
    - City 2: Distance to its nearest center = 10 (City center 1)
    - City 3: Distance to its nearest center = 0
    - City 4: Distance to its nearest center = 25 (ity center 3)
    - City 5: Distance to its nearest center = 25 (City center 1)
    - City 6: Distance to its nearest center = 35

      Maximum distance = 35



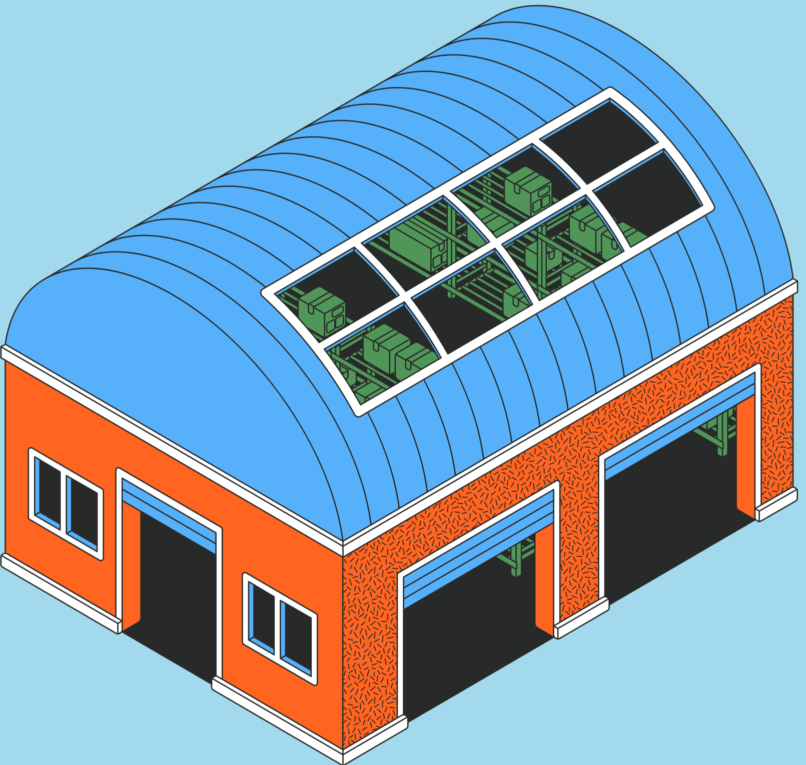
3.Centers in city 1 and city 4

- City 1: Distance to its nearest center = 0
- City 2: Distance to its nearest center = 10 (City Center 1)
- City 3: Distance to its nearest center = 20 (City Center 1)
- City 4: Distance to its nearest center = 0
- City 5: Distance to its nearest center = 15 (City center 4)
- City 6: Distance to its nearest center = 20 (City center 4)
- Maximum distance = 20



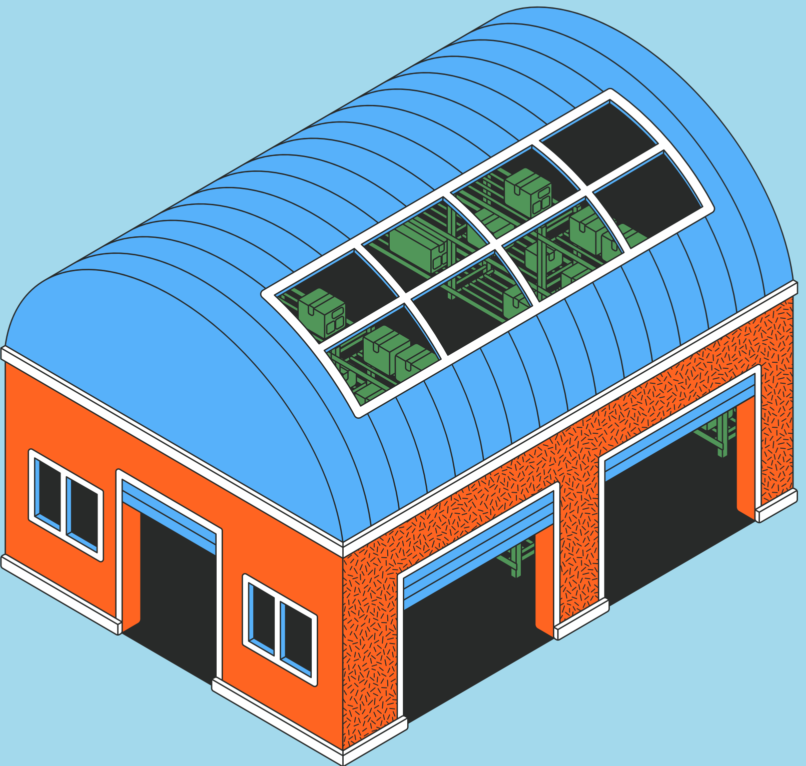
4. Centers in city 1 and city 5

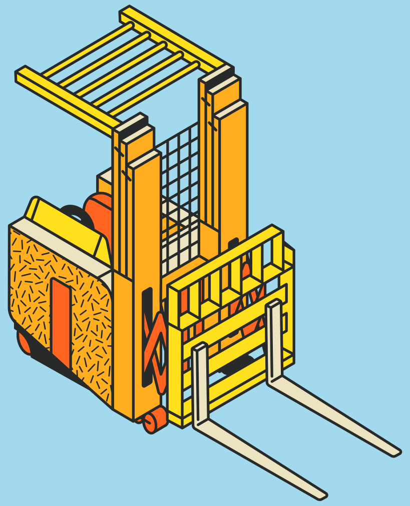
- City 1: Distance to its nearest center = 0
- City 2: Distance to its nearest center = 10 (City center 1)
- City 3: Distance to its nearest center = 20 (City center 1)
- City 4: Distance to its nearest center = 15 (City center5)
- City 5: Distance to its nearest center = 0
- City 6: Distance to its nearest center = 35 (City center 1)
- Maximum distance = 35



5. Centers in city 1 and city 6
    - City 1: Distance to its nearest center = 0
    - City 2: Distance to its nearest center = 10 (City center 1)
    - City 3: Distance to its nearest center = 20 (City center 1)
    - City 4: Distance to its nearest center = 20 (City center 6)
    - City 5: Distance to its nearest center = 25 (City center 1)
    - City 6: Distance to its nearest center = 0
    - Maximum distance = 25

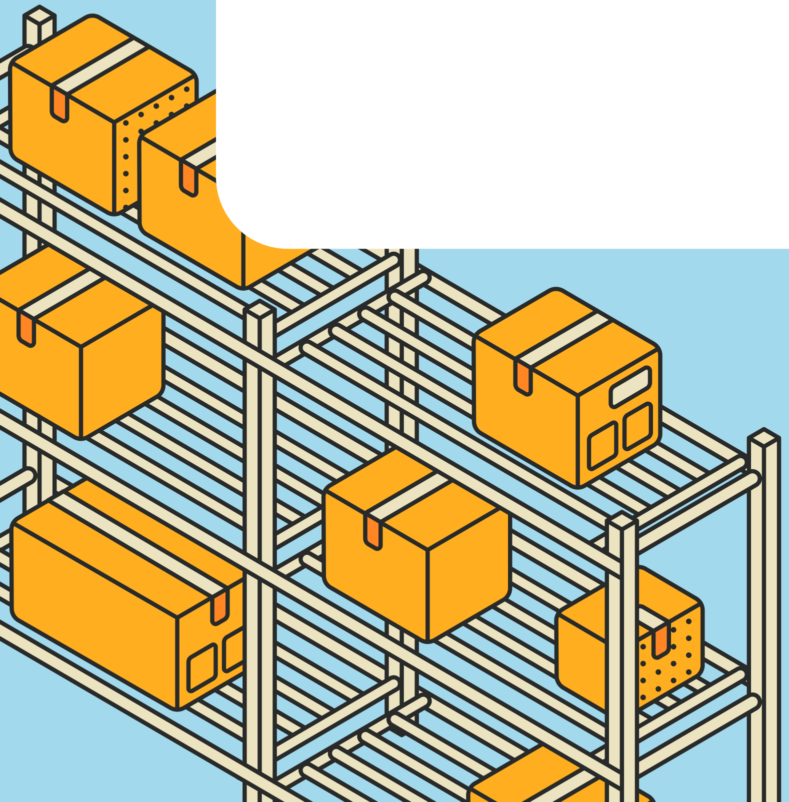
6. Centers in city 2 and 3
    - City 1: Distance to its nearest centre = 10 (City center 2)
    - City 2: Distance to its nearest centre = 0
    - City 3: Distance to its nearest center = 0
    - City 4: Distance to its nearest center = 25 (City center 3)
    - City 5: Distance to its nearest center = 20 (City center 2)
    - City 6: Distance to its nearest center = 40 (City center 2)
- Maximum distance = 40

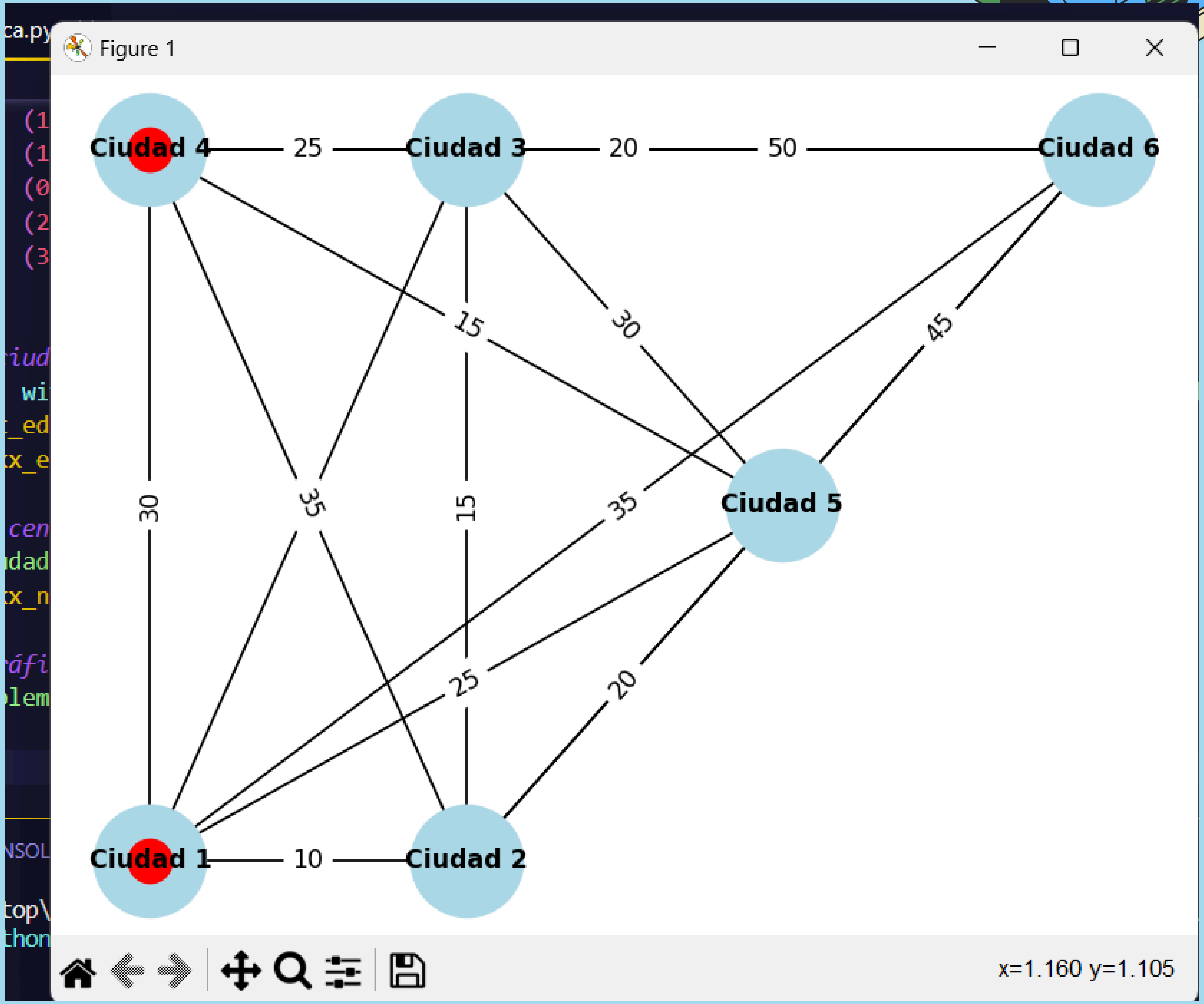




Choosing the best combination

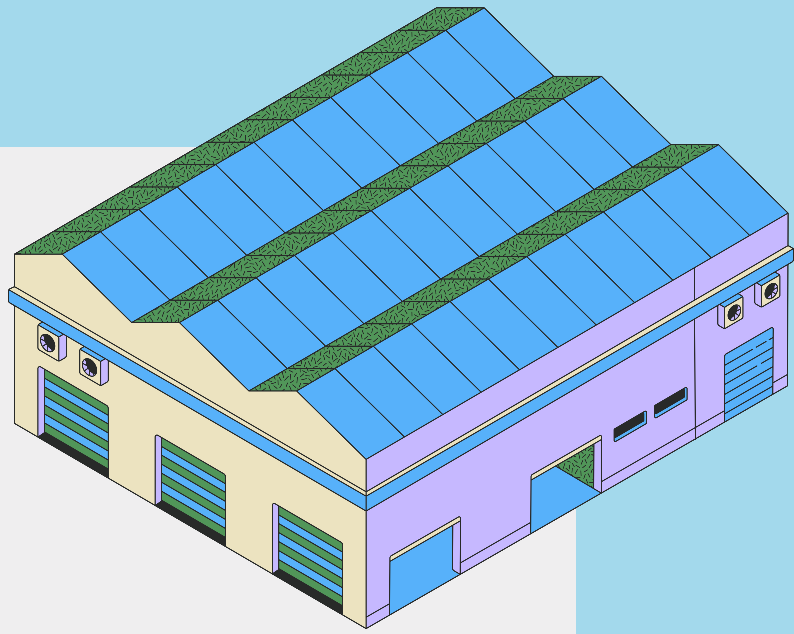
We observe that combining centers in City 1 and City 4 results in the lowest maximum distance, which is 20. Therefore, this is the best combination for our problem.





Code for graph

```
import matplotlib.pyplot as plt
import networkx as nx

# Crear un gráfico vacío
G = nx.Graph()

# Lista de ciudades
cities = ['Ciudad 1', 'Ciudad 2', 'Ciudad 3', 'Ciudad 4', 'Ciudad 5', 'Ciudad 6']

# Añadir nodos
for city in cities:
    G.add_node(city)

# Añadir aristas con las distancias
distances = {
    ('Ciudad 1', 'Ciudad 2'): 10,
    ('Ciudad 1', 'Ciudad 3'): 20,
    ('Ciudad 1', 'Ciudad 4'): 30,
    ('Ciudad 1', 'Ciudad 5'): 25,
    ('Ciudad 1', 'Ciudad 6'): 35,
    ('Ciudad 2', 'Ciudad 3'): 15,
    ('Ciudad 2', 'Ciudad 4'): 35,
    ('Ciudad 2', 'Ciudad 5'): 20,
    ('Ciudad 2', 'Ciudad 6'): 40,
    ('Ciudad 3', 'Ciudad 4'): 25,
    ('Ciudad 3', 'Ciudad 5'): 30,
    ('Ciudad 3', 'Ciudad 6'): 50,
    ('Ciudad 4', 'Ciudad 5'): 15,
    ('Ciudad 4', 'Ciudad 6'): 20,
    ('Ciudad 5', 'Ciudad 6'): 45
}

for (city1, city2), distance in distances.items():
    G.add_edge(city1, city2, weight=distance)

# Posiciones de las ciudades para la gráfica (solo para propósitos visuales)
pos = {
    'Ciudad 1': (0, 0),
    'Ciudad 2': (1, 0),
    'Ciudad 3': (1, 1),
    'Ciudad 4': (0, 1),
    'Ciudad 5': (2, 0.5),
    'Ciudad 6': (3, 1)
}

# Dibujar las ciudades y las aristas
nx.draw(G, pos, with_labels=True, node_size=2000, node_color='lightblue', font_size=10, font_weight='bold')
labels = nx.get_edge_attributes(G, 'weight')
nx.draw_networkx_edge_labels(G, pos, edge_labels=labels)

# Destacar los centros de distribución
centros = ['Ciudad 1', 'Ciudad 4']
nx.draw_networkx_nodes(G, pos, nodelist=centros, node_color='red')

# Mostrar la gráfica
plt.title("Problema de P-centro con P = 2 y 6 ciudades")
plt.show()
```



CONCLUSION

The optimal solution to the P-center problem with 6 cities and 2 distribution centers is to place the centers in cities 1 and 4, resulting in a maximum distance of 20 units from any city to its nearest center.