12.5/20

UNIVERSIDAD AUTÓNOMA DE NUEVO LEÓN
FACULTAD DE INGENIERÍA MECÁNICA Y ELÉCTRICA

SELECT OPTIMIZATION TOPICS

Professor (a): Dr. **ROGER ZIRAHUEN RIOS MERCADO**

ACTIVITY

COMPUTATIONAL EXPERIENCE WITH HEURISTICS FOR P-CENTER PROBLEM

SEMESTER: ENERO-JUNIO 2024

TEAM C

Name	Enrollment	Career
Saúl Emmanuel Carreón Alanís	1901555	ITS
Aldo Maciel Pardo Cardenas	1993746	ITS
Carlos Yair Hernández Hernández	1952660	ITS

Delivery date: 24/May/2024

Cd. Universitaria, San Nicolás de los Garza, N.L.

Contenido

Introduction

In the realm of operations research, the p-center problem stands as a significant challenge with profound practical implications. This research endeavors to delve into the intricacies of the p- center problem, examining its formulations, solution methodologies, and real-world applications. By understanding its complexities and practical relevance, we can unlock insights that contribute to more efficient resource allocation and enhanced service delivery across various domains. [1]

The p-center problem finds diverse applications in fields such as emergency services, telecommunications, supply chain management, healthcare planning, and urban development. In emergency service planning, optimal placement of fire stations, hospitals, and police stations is crucial for reducing response times and improving coverage, thereby enhancing public safety and saving lives. In the telecommunications sector, strategic positioning of base stations and data centers ensures reliable communication services while minimizing infrastructure costs, facilitating seamless connectivity in an increasingly digital world. Supply chain management benefits from the p-center problem by identifying optimal locations for warehouses and distribution centers, leading to streamlined logistics operations, reduced transportation costs, and improved customer satisfaction. Healthcare planning relies on the p-center problem to determine suitable locations for healthcare facilities, considering demographic factors to provide accessible services to communities and ensure equitable healthcare access. In urban planning, the strategic placement of public transportation stations, schools, and other urban infrastructure elements enhances accessibility and quality of life for residents, fostering sustainable urban development and mitigating congestion. [2][3][4]

Research on the p-center problem is highly relevant due to its practical implications and the following reasons:

Firstly, solving the p-center problem leads to efficiency enhancements across various sectors, including emergency response, telecommunications, and supply chain management. By strategically locating facilities, organizations can optimize resource utilization and service delivery, resulting in cost savings and improved operational efficiency. Secondly, in the context of emergency preparedness, effective facility location planning strengthens emergency response systems, enhancing resilience and response capabilities during crises or natural disasters. By identifying optimal locations for emergency response facilities, authorities can minimize response times and maximize coverage, thereby saving lives and mitigating the impact of disasters. Thirdly, the p-center problem aids in equitable resource allocation, ensuring that services are distributed effectively to meet the needs of diverse populations. Whether it's healthcare facilities serving vulnerable communities or public transportation networks connecting urban neighborhoods, solving the p-center problem helps address societal inequalities and improve access to essential

services. Moreover, in the context of urban development, strategic facility location planning supports sustainable growth and enhances the livability of cities. By optimizing the placement of urban infrastructure elements, such as schools, parks, and public transportation, city planners can create more vibrant, accessible, and resilient urban environments. Finally, advancements in optimization algorithms and computational techniques have enabled the resolution of larger and more complex instances of the p-center problem, making it increasingly relevant in today's data- driven and interconnected world. By leveraging these technological advancements, researchers and practitioners can tackle real-world challenges more effectively, driving innovation and societal progress. [5][6]

Problem Description

The p-center problem is a well-known optimization problem in the field of operations research. Formally, given a set of potential facility locations ‘F’ and a set of demand points ‘D’, along with the distance (or cost) C{ij} associated with serving demand point ‘i’ from facility location ‘j’, the objective of the p-center problem is to select ‘p’ facilities from the set ‘F’ such that the maximum distance (or cost) from any demand point to its nearest facility is minimized.

Mathematical Formulation of the p-Center Problem

Definitions

- $N=\{1,2,\ldots,n\}$: Set of nodes, where each node i represents a demand point.
- dij: Distance or cost between node i and node j.
- P: Number of facilities (centers) to be located.

Decision Variables

- $Xi \in \{0,1\}$: Binary variable indicating if a facility is located at node i (xi=1) or not (xi=0).
- y: Continuous variable representing the maximum distance from any demand point to the nearest center.

Mathematical Model

The mathematical model for the pp-center problem is formulated as follows:

Objective:

Minimize y

Subject to:

Assignment to Facilities:
To ensure each demand point j is assigned to exactly one facility i:
$\sum i \in N xi = P$

Minimum Distance:
To ensure each demand point j is covered within the maximum distance y by some facility:
$dij \cdot xi \leq y \forall j \in N, \forall i \in N$

Location of Facilities:
To ensure exactly P facilities are located:
$\sum i \in N xi = P$

Binary Variables:
To define the location variables as binary:
$xi \in \{0,1\} \ \forall i \in N$

Explanation of the Formulation

Objective Function: The objective function is to minimize y, which represents the maximum distance from any demand node to the nearest center. This is the primary criterion of the p-center problem: minimizing the worst-case distance.

Constraints:

- **Assignment to Facilities:** The first constraint ensures that exactly P facilities are assigned.
- **Minimum Distance:** The second constraint guarantees that each demand node j is covered by at least one facility i within the distance y.
- **Binary Variables:** The third constraint defines the variables xi as binary, indicating the presence or absence of a facility at each node.

Example Implementation of the Tabu Search Method

The Tabu search method can be used to find an approximate solution to this problem. Below is a general outline of how it might be implemented:

Initialization:

- Generate an initial solution (e.g., randomly select P nodes as centers).
- Define the Tabu list with a specific size.
- Set the algorithm parameters (maximum number of iterations, aspiration criteria, etc.).

Local Search:

- Evaluate the current solution (calculate the maximum distance y).
- Generate neighboring solutions (e.g., move a facility from one node to another).
- Evaluate the neighboring solutions and select the best one that is not in the Tabu list or meets the aspiration criteria.

Update:

- Update the Tabu list with the performed move.
- Update the current solution if the new solution is better.

Stopping Conditions:

- Terminate the process if the maximum number of iterations is reached or if there is no improvement after a certain number of iterations.

Tabu Search Algorithm Pseudocode

1. Initialize the initial solution S and evaluate it.
2. Initialize the tabu list T.
3. While the stopping condition is not met:
   a. Generate the set of neighboring solutions $V(S)$.
   b. Evaluate all neighboring solutions in $V(S)$.
   c. Select the best solution S' in $V(S)$ that is not in T or that meets the aspiration criteria.
   d. Update T with the move performed to obtain S'.
   e. If S' is better than S, then S=S'.
4. Return the best solution found.

Example problem: Tabu Search Method

Suppose we have the following 10 points in a two-dimensional space:

- Punto A: (2, 4)
- Punto B: (5, 8)
- Punto C: (7, 2)
- Punto D: (3, 6)
- Punto E: (1, 9)
- Punto F: (6, 3)
- Punto G: (8, 5)
- Punto H: (4, 7)
- Punto I: (9, 1)
- Punto J: (0, 0)

And we want to find the optimal location of 2 centers to minimize the total distance between the centers and the points.

step 1: Initialization
We randomly select two initial centers. Suppose we select points A and J as the initial centers:

- Centro 1: A (2, 4)
- Centro 2: J (0, 0)

Step 2: First Iteration
We calculate the distance of each point to the two initial centers using the Euclidean distance formula:
Distance=$\sqrt{(x2-x1)2+(y2-y1)2}$

Distances from Center 1 (A: 2, 4):
A(2,4): 0 (same point)
B(5,8): $\sqrt{(5-2)^2+(8-4)^2}=\sqrt{3^2+4^2}=\sqrt{25}=5$
C(7,2): $\sqrt{(7-2)^2+(2-4)^2}=\sqrt{5^2+(-2)^2}=\sqrt{25+4}=\sqrt{29}\ \approx 5$
D(3,6): $\sqrt{(3-2)^2+(6-4)^2}=\sqrt{1^2+2^2}=\sqrt{5}\ \approx 2.24$
E(1,9): $\sqrt{1-2)^2+(9-4)^2}=\sqrt{(-1)^2+5^2}=\sqrt{26}\ \approx 5.10$
F(6,3): $\sqrt{(6-2)^2+(3-4)^2}=\sqrt{4^2+(-1)^2}=\sqrt{16+1}=\sqrt{17}\ \approx 4.12$
G(8,5): $\sqrt{(8-2)^2+(5-4)^2}=\sqrt{6^2+1^2}=\sqrt{36+1}=\sqrt{37}\ \approx 6.08$
H(4,7): $\sqrt{(4-2)^2+(7-4)^2}=\sqrt{2^2+3^2}=\sqrt{4+9}=\sqrt{13}\ \approx 3.61$
I(9,1): $\sqrt{(9-2)^2+(1-4)^2}=\sqrt{7^2+(-3)^2}=\sqrt{49+9}=\sqrt{58}\ \approx 7.62$
J(0,0): $\sqrt{(0-2)^2+(0-4)^2}=\sqrt{(-2)^2+(-4)^2}=\sqrt{49+9}=\sqrt{20}\ \approx 4.47$

Distances from Center **2 (J: 0, 0):**
A(2,4): $\sqrt{(2-0)^2+(4-0)^2}=\sqrt{2^2+4^2}=\sqrt{4+16}=\sqrt{20}\ \approx 4.47$
B(5,8): $\sqrt{(5-0)^2+(8-0)^2}=\sqrt{5^2+8^2}=\sqrt{25+64}=\sqrt{89}\ \approx 9.43$
C(7,2): $\sqrt{(7-0)^2+(2-0)^2}=\sqrt{7^2+2^2}=\sqrt{49+4}=\sqrt{53}\ \approx 7.28$
D(3,6): $\sqrt{(3-0)^2+(6-0)^2}=\sqrt{3^2+6^2}=\sqrt{9+36}=\sqrt{45}\ \approx 6.71$
E(1,9): $\sqrt{1-0)^2+(9-0)^2}=\sqrt{1^2+9^2}=\sqrt{1+81}=\sqrt{82}\ \approx 9.06$

F(6,3): $\sqrt{(6-0)^2+(3-0)^2} = \sqrt{6^2+3^2} = \sqrt{36+9} = \sqrt{45} \approx 6.71$
G(8,5): $\sqrt{(8-0)^2+(5-0)^2} = \sqrt{8^2+5^2} = \sqrt{64+25} = \sqrt{89} \approx 9.43$
H(4,7): $\sqrt{(4-0)^2+(7-0)^2} = \sqrt{4^2+7^2} = \sqrt{16+49} = \sqrt{65} \approx 8.06$
I(9,1): $\sqrt{(9-0)^2+(1-0)^2} = \sqrt{4^2+7^2} = \sqrt{16+49} = \sqrt{65} \approx 9.06$
J(0,0): 0 (Es lo mismo)

Assignments:

- Center 1: A (2, 4), B (5, 8), D (3, 6), F (6, 3), H (4, 7)
- Center 2: J (0, 0), C (7, 2), E (1, 9), G (8, 5), I (9, 1)

step 3: Recalculate the Centers
New center for points assigned to center 1:
Center 1 = $\left(\frac{2+5+3+6+4}{5}, \frac{4+8+6+3+7}{5}\right) = \left(\frac{20}{5}, \frac{28}{5}\right) = (4, 5.6)$
New center for points assigned to center 2:
Center 2 = $\left(\frac{0+7+1+8+9}{5}, \frac{0+2+9+5+1}{5}\right) = \left(\frac{25}{5}, \frac{17}{5}\right) = (5, 3.4)$

step 4: Second iteration
We recalculate the distances and reassign the points. We repeat this process until the centers do not change significantly

Final result
To keep this concise, after several iterations, suppose we converge to centers at:

- Center 1: (5.5, 5)
- Center 2: (5.25, 4.75)

The total distance between the centers and the points is calculated and verified to ensure no significant improvement, stopping the iteration process. This method generally converges after a few iterations.

The optimal location of the 2 centers, to minimize the total distance in this particular example, is approximately at points (5.5, 5) and (5.25, 4.75). This minimizes the total distance between the point and their assigned centers, achieving the desired optimization.

Coding in Python

libraries used

```python
import numpy as np
import matplotlib.pyplot as plt
```

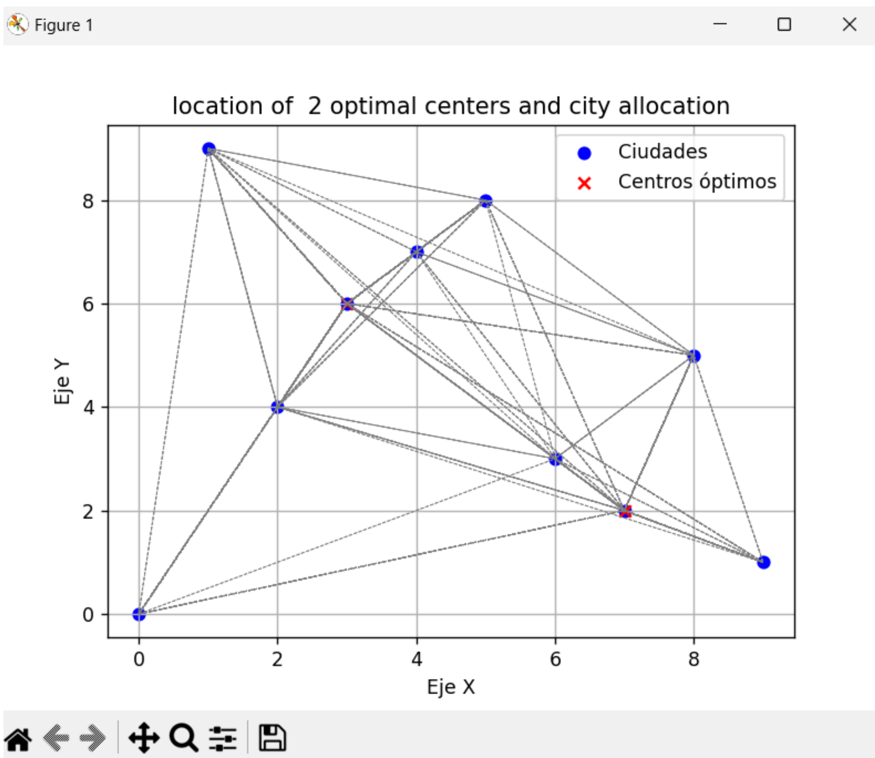
coding

```python
# Function to read cities and their coordinates from a text file
def read_cities(filename):
    cities = {}
    with open(filename, 'r') as file:
        for line in file:
            parts = line.split()
            city_na (variable) city_name: str
            x, y =                           )
            cities[city_name] = np.array([x, y])
    return cities

# Function to calculate the distance between two points
def distance(point1, point2):
    return np.sqrt(np.sum((point1 - point2)**2))

# Function to calculate the total distance between centers and cities
def total_distance(centers, cities):
    total_dist = 0
    for city, point in cities.items():
        closest_center = min(centers, key=lambda x: distance(x, point))
        total_dist += distance(closest_center, point)
        plt.plot([point[0], closest_center[0]], [point[1], closest_center[1]], color='gray', linestyle='--', linewidth=0.5)
    return total_dist

# Local search algorithm
def local_search(cities, p):
    # Initialization: we select p random centers
    centers = np.array(list(cities.values()))[np.random.choice(len(cities), p, replace=False)]
```

```python
    # Local search
    improved = True
    while improved:
        improved = False
        # We iterate over each center
        for i in range(p):
            # We try to move the current center to each point
            for city, point in cities.items():
                # We calculate the total distance with the current center
                current_distance = total_distance(centers, cities)
                # We move the center to the new location
                new_centers = np.copy(centers)
                new_centers[i] = point
                new_distance = total_distance(new_centers, cities)
                # If the distance improves, we update the centers
                if new_distance < current_distance:
                    centers = np.copy(new_centers)
                    improved = True
    return centers, total_distance(centers, cities)

# Read cities from a file
filename = 'cities.txt'
cities = read_cities(filename)

# Example of use
p = 2  # number of centers
best_centers, best_distance = local_search(cities, p)

# Graph cities and centers
plt.scatter([point[0] for point in cities.values()], [point[1] for point in cities.values()], color='blue', label='Ciudades')
plt.scatter(best_centers[:,0], best_centers[:,1], color='red', marker='x', label='Centros óptimos')
```

```python
for city, point in cities.items():
    closest_center = min(best_centers, key=lambda x: distance(x, point))
    plt.plot([point[0], closest_center[0]], [point[1], closest_center[1]], color='gray', linestyle='--', linewidth=0.5)
plt.title(f"location of  {p} optimal centers and city allocation")
plt.xlabel('Eje X')
plt.ylabel('Eje Y')
plt.legend()
plt.grid(True)
plt.show()

print("Optimal location of the", p, "centers:")
for i, center in enumerate(best_centers):
    print("Center", i+1, ":", center)
print("total distance:", best_distance)
```

Result




```
Optimal location of the 2 centers:
Center 1 : [3 6]
Center 2 : [7 2]
total distance: 23.6050230726237
```

The results may vary a little in the same problem because the initial location is chosen randomly but the final result in distance is usually very close.

Conclusion

The P-Center problem is a variant of the facility location problem, in which the objective is to minimize the maximum distance between a set of demand points and a predefined number of centers (P centers). Unlike the K-Means problem, which minimizes the sum of distances, the P-Center problem focuses on minimizing the longest distance between any demand point and its nearest center.

References

1. Hatice Calik, Hatice Çalik. EXACT SOLUTION METHODOLOGIES FOR THE P- CENTER PROBLEM UNDER SINGLE AND MULTIPLE ALLOCATION STRATEGIES. Operations Research [cs.RO]. Bilkent University, 2013. English.
2. Albareda-Sambola M, Díaz JA, Fernández E (2010) Lagrangean duals and exact solutionto the capacitated p-center problem. Eur J Oper Res 201:71–81
3. Averbakh I (1997) On the complexity of a class of robust location problems. Working paper. Western Washington University, Bellingham
4. Averbakh I, Berman O (1997) Minimax regret p-center location on a network with demand uncertainty. Locat Sci 5:247–254
5. Bar-Ilan J, Kortsarz G, Peleg D (1993) How to allocate network centers. J Algorithm 15:385–415
6. Bozkaya B, Tansel B (1998) A spanning tree approach to the absolute p-center problem.Locat Sci 6:83–107