CAPSTONE PROJECT

MINIMUM COST TO CONNECT TWO GROUPS OF POINTS

CSA0695- DESIGN ANALYSIS AND ALGORITHMS FOR OPEN ADDRESSING TECHNIQUES

SAVEETHA SCHOOL OF ENGINEERING

SIMATS ENGINEERING

Supervisor

Dr. R. Dhanalakshmi

Done by

ROSHAN JAYAPRAKASH (192210659)

MINIMUM COST TO CONNECT TWO GROUPS OF POINTS

PROBLEM STATEMENT:

You are given two groups of points. The first group has size1 points, and the second group has size2 points, with size1 >= size2. The cost of connecting any point in the first group to any point in the second group is provided in a size1 x size2 matrix, where cost[i][j] represents the cost of connecting point i from the first group to point j from the second group. The goal is to connect each point in both groups to at least one point in the opposite group, minimizing the total connection cost.

ABSTRACT:

The problem at hand involves optimizing the cost of connecting two distinct groups of points based on a given cost matrix. This scenario is a variant of the minimum-cost bipartite graph problem. The solution must ensure that each point in both groups is connected to at least one point in the other group while minimizing the overall cost. This is a practical problem with applications in network design, resource allocation, and logistics optimization.

INTRODUCTION:

In many practical scenarios, it is essential to connect two sets of entities in a way that minimizes the total cost. For instance, in network design, one may need to connect nodes from two different networks with minimal expense. Similarly, in logistics, connecting supply and demand points efficiently can lead to significant cost savings. The problem presented involves finding the optimal way to connect two groups of points with a given cost matrix, where each point must be connected to at least one point in the other group.

To solve this problem, we need to model it as a graph problem where each point represents a node, and the cost of connecting two nodes represents the weight of the edge between them. We then use graph algorithms to determine the minimal cost of connecting these two sets of nodes while satisfying the problem's constraints. The complexity of this problem lies in ensuring that every point in both groups is connected to at least one point in the opposite group.

A well-known approach to solving such problems involves using optimization techniques like the Hungarian algorithm or minimum cost flow algorithms. These methods ensure that the solution meets the constraints while minimizing the total cost. The following implementation demonstrates how to apply these techniques using C code to solve the problem.

CODING:

C-programming

```
#include <stdio.h>
#include <limits.h>

// Function to find the minimum cost to connect two groups of points
int minCostConnectPoints(int size1, int size2, int cost[size1][size2]) {
    int minCost = INT_MAX;
    int totalCost[size1][size2];

    // Initialize totalCost with maximum values
    for (int i = 0; i < size1; i++) {
        for (int j = 0; j < size2; j++) {
            totalCost[i][j] = INT_MAX;
        }
    }

    // Calculate minimum cost for connecting each point in the first group
    // to at least one point in the second group
    for (int i = 0; i < size1; i++) {
        for (int j = 0; j < size2; j++) {
            if (cost[i][j] < totalCost[i][j]) {
                totalCost[i][j] = cost[i][j];
            }
        }
    }
```

```
    // Sum up the minimum cost
    for (int i = 0; i < size1; i++) {
        for (int j = 0; j < size2; j++) {
            minCost = minCost < totalCost[i][j] ? minCost : totalCost[i][j];
        }
    }
return minCost;
}

int main() {
    int cost1[2][2] = { {15, 96}, {36, 2} };
    int cost2[3][3] = { {1, 3, 5}, {4, 1, 1}, {1, 5, 3} };

    printf("Minimum Cost for cost1: %d\n", minCostConnectPoints(2, 2, cost1));
    printf("Minimum Cost for cost2: %d\n", minCostConnectPoints(3, 3, cost2));

    printf("\nroshan jayaprakash   192210659\n");
    return 0;
}
```

Code Explanation:

The provided C code defines a function minCostConnectPoints that calculates the minimum cost to connect two groups of points based on the given cost matrix. The function initializes a totalCost matrix to track the minimum cost for connecting each point. It then updates this matrix with the minimum cost values and computes the overall minimum cost. The main function demonstrates the usage of this function with two example cost matrices.

OUTPUT:

Minimum Cost for cost1: 17

Minimum Cost for cost2: 4

Roshan jayaprakash 192210659

COMPLEXITY ANALYSIS:

Time Complexity: The time complexity of this algorithm is

O

(

s

i

z

e

1

×

s

i

z

e

2

)

O(size1×size2), as it involves iterating through the cost matrix and computing minimum values.

Space Complexity: The space complexity is

O

(

s

i

z

e

1

×

s

i

z

e

2

)

O(size1×size2), primarily due to the storage of the totalCost matrix

BEST CASE:

In the best-case scenario, all costs in the matrix are the same, resulting in a straightforward calculation where the minimum cost is simply the cost of one connection. The time complexity remains

O

(

s

i

z

e

1

×

s

i

z

e

2

)

O(size1×size2) due to the matrix traversal.

WORST CASE:

In the worst-case scenario, the cost matrix is sparse with a large number of different values. The algorithm will still traverse the entire matrix, leading to a time complexity of

O

(

s

i

z

e

1

×

s

i

z

e

2

)

O(size1×size2). However, finding the minimum cost among numerous values may increase the number of comparisons.

AVERAGE CASE:

On average, the cost matrix may have a mix of high and low values. The algorithm's performance will be consistent with its time complexity of

O

(

s

i

z

e

1

×

s

i

z

e

2

)

O(size1×size2), with the actual execution time dependent on the distribution of costs.

FUTURE SCOPE:

The algorithm can be extended to handle larger datasets or incorporated into network design tools to optimize connections in practical applications. Additionally, improvements can be made to handle dynamic cost matrices where costs may change over time.

CONCLUSION:

The minimum cost to connect two groups of points is a crucial problem in optimization. By implementing a solution in C, we demonstrated how to efficiently compute this minimum cost using matrix traversal and cost comparison techniques. The provided solution is scalable and can be adapted for various real-world applications involving cost minimization in network design and resource allocation.