Home Exam TDA251

Tomas Alander, tomasal@student.chalmers.se, MPALG

2024-08-28

Introduction

This report addresses the challenge of mapping n nodes to a set S of p grid points in a 2-dimensional plane, where $p \geq n$. Each pair of nodes $\{i, j\}$ is associated with a positive weight w_{ij}, representing the importance of placing those nodes close to each other. The objective is to assign each node to a unique grid point in S such that the sum of the weighted Manhattan distances between all node pairs is minimized.

Given a complete graph $G(V, E)$, where the nodes are connected by positive weighted edges representing their relationships, the number of different mappings grows with the size of the problem: $\frac{p!}{(p-n)!}$. This problem quickly becomes computationally infeasible as n increases. It is known to be NP-complete, indicating that finding an exact solution in polynomial time is impossible unless P = NP. Therefore, this report focuses on developing and analyzing polynomial-time algorithms that aim to provide approximations to the optimal solutions.

Grid

Given n nodes, we define k as $k = \lceil \sqrt{n} \rceil$. Let S be a rectangular grid in a 2-dimensional plane with integer Cartesian coordinates, where $|S| = k^2$. The set S is defined as a square grid such that k represents both the height and the width of S, meaning that S contains $k \times k$ grid points. Each point in S is uniquely identified by its integer coordinates (x, y), where $0 \leq x, y < k$.

This grid serves as the placement space for the n nodes, where each node must be assigned to a unique grid point within S. The choice of $k = \lceil \sqrt{n} \rceil$ ensures that the grid has sufficient capacity to fit all nodes, while keeping it square and as compact as possible.

Grid shape

The choice of a square grid S is motivated by the desire to minimize the maximum distance between any two points. In a rectangular grid, the farthest distance between any two grid points occurs along the diagonal. The total Manhattan distance is the sum of the horizontal and vertical distances. Due to the inequality of arithmetic and geometric means, a square has the smallest perimeter among all rectangles of equal area, resulting in a maximum distance of $2(k-1)$ in a $k \times k$ grid.

Bounds Analysis

In evaluating approximation algorithms, both lower and upper bounds on a solution are established to evaluate a solution to some optimum one.

Lower Bound

The lower bound provides a baseline for evaluating the performance of approximation algorithms by establishing the minimum possible weighted distance that any optimal solution could achieve.

Theoretical Optimal Distribution of Weighted Edges

An optimal solution strives to minimize the weighted sum of all edge distances. Let OPT be the theoretical optimal solution, and let $W = \{w_1, w_2, \ldots, w_m\}$ be the set of weights corresponding to all the edges in G, sorted in descending order. The goal of OPT is to minimize the weighted distance by assigning the heaviest weights to the smallest possible distances in S. This process ensures that the most significant edges contribute minimally to the total cost.

In the opposite case, the heaviest edges would be assigned to the grid points with the farthest distance, which can occur in some random placements of nodes onto the grid.

Grid Properties

Let d be the distance between a pair of grid points in the same row or column, where $d < k$ and $d \in \mathbb{N}$. The grid S consists of k rows and k columns. Each row and column contain $k - d$ unique pairs of points with a distance d between them. Specifically, the grid S contains $2k(k-1)$ pairs of grid points with a distance of 1.

For diagonal pairs (x_i, y_j) and (x'_i, y'_j) where $x_i \neq x'_i$ and $y_j \neq y'_j$, let d_x and d_y represent the absolute differences in their x-coordinates and y-coordinates, respectively:

$$d_x = |x_i - x'_i|, \quad d_y = |y_j - y'_j|$$

With $1 \leq d_x, d_y \leq k-1$. The number of such diagonal pairs can then be expressed as:

$$2 \times \sum_{d_x=1}^{k-1} \left((k - d_x) \times \sum_{d_y=1}^{k-1} (k - d_y) \right)$$

Summing over all pairs that differ in both the x-axis and y-axis, including those with swapped coordinates, such as (y_j, x_i) and (y'_j, x'_i), is done by using symmetry and multiplying by a factor of 2.

Weighted Distribution Lower Bound

Let d_1 be the number of adjacent grid pairs in S such that the distance between them is 1: $d_1 = 2k(k-1)$. Let d_2 represent all grid pairs in S that have a distance equal to or less than 2. Summing up all diagonal, vertical, and horizontal pairs, we get: $d_2 = 2[(k-1)(k-1) + k(k-2)] + d_1$.

By examining how the weights are distributed across different distances in S, we can adjust the lower bound according to the distribution of weights in W.

$$\sum_{i=1}^{d_2} w_i \overset{?}{<} \sum_{i=d_2+1}^{m} w_i$$

If the majority of the weights correspond to grid pairs with a distance greater than or equal to 3, then the lower bound is given by:

$$\text{Lower Bound} = \sum_{i<j} w_{ij} \times 2$$

If the majority is not found among the weights with a distance of ≥ 3, we check if the majority is found among the weights with a distance greater than or equal to 2:

$$\sum_{i=1}^{d_1} w_i \overset{?}{<} \sum_{i=d_1+1}^{m} w_i$$

If the majority of the weights have a distance ≥ 2, then the lower bound is given by:

$$\text{Lower Bound} = \sum_{i<j} w_{ij} \times \frac{2}{3}$$

Otherwise, if the theoretical OPT manages to distribute the majority of the weights to pairs with the shortest distances, then the lower bound is:

$$\text{Lower Bound} = \sum_{i<j} w_{ij} \times 1$$

Upper Bound

An upper bound defines the theoretical maximum weighted distance that any solution may incur. We assume the worst-case scenario, where each node pair is placed at the maximum possible distance in the grid. For a square grid of size $k \times k$, the maximum Manhattan distance between any two grid points is $2(k-1)$. And we can set the upper bound as:

$$\text{Upper Bound} = \sum_{i<j} w_{ij} \times 2(k-1)$$

This bound provides an upper limit, ensuring that no solution can exceed this weighted distance.

Weighted Distribution Upper Bound

Since the worst case scenario is that for all edges all the heavies weights are given the biggest distance there by maximising the weighted distance. By investigating the weighted distribution we may lower the bound if the majority of weights are located at a distance $k-1$ or lower. The distance less then or equal $k-1$ can be expressed as all the diagonal pairs such that $|x_i - x'_i| + |y_j - y'_j| \leq k-1$:

$$2 \times \sum_{d_x=1}^{k-2} \left((k-d_x) \times \sum_{d_y=1}^{k-1-d_x} (k-d_y) \right)$$

Together with all the horizontal and vertical pairs, we define the distance $d_{\leq k-1}$as:

$$d_{\leq k-1} = 2\left[\sum_{d_x=1}^{k-2}\left((k-d_x)\times\sum_{d_y=1}^{k-1-d_x}(k-d_y)\right)+\sum_{d=1}^{k-1}k(k-d)\right]$$

We can directly determine the number of grid pairs with a distance greater than $k-1$ as:

$$d_{>k-1} = \binom{k^2}{2} - d_{\leq k-1}$$

If the weighted distribution satisfied the inequality:

$$\sum_{i=1}^{d_{>k-1}} w_i \overset{?}{<} \sum_{j=d_{>k-1}+1}^{m} w_j$$

Then it follows that:

$$\frac{1}{2}\left[\sum_{i=1}^{d_{>k-1}} w_i \times 2(k-1) + \sum_{j=d_{>k-1}+1}^{m} w_j \times (k-1)\right] < \sum_{i=1}^{m} w_i \times 2(k-1)$$

And we can set the upper bound as:

$$\text{Upper Bound} = \sum_{i<j} w_{ij} \times \frac{3}{2}(k-1)$$

Randomized Assignment and Expected Distance

We aim to apply a randomized strategy for assigning nodes to grid points and calculate the expected weighted sum of distances between node pairs using the principle of linearity of expectation.

Expected Distance

In analyzing the performance of algorithms that employ randomization, calculating the expected Manhattan distance between node pairs, weighted by their probabilities, can offer a more accurate measure of the algorithm's result. This approach utilizes the principles of probability to estimate the average outcome, potentially leading to a tighter bound that better reflects the algorithm's typical behavior rather than just its worst-case performance.

Randomized Assignment of Nodes

Given a set of n nodes and a grid S with k^2 points, each node is assigned to a grid point uniformly at random. This random placement ensures that each node is equally likely to be placed at any of the available grid points, with the constraint that only one node can occupy a grid point.

Expected Manhattan Distance on a $k \times k$ Grid

Given n points uniformly distributed on a $k \times k$ grid, where $k = \lceil \sqrt{n} \rceil$, we want to calculate the expected Manhattan distance between two randomly chosen points on the grid.
The Manhattan distance between two points (x_i, y_i) and (x_j, y_j) is defined as:

$$D_{ij} = |x_i - x_j| + |y_i - y_j|$$

Expected Distance in One Dimension

First, we calculate the expected distance between two points along one dimension, the x-axis. The same logic will apply to the y-axis.

Consider a 1-dimensional grid with k integer coordinates $0, 1, \ldots, k-1$. The expected value of the distance between two randomly chosen points on this grid can be computed as follows:

- Let $d = |x_i - x_j|$ define the distance between two points x_i and x_j, where $x_i, x_j \in \{0, 1, \ldots, k-1\}$.
- For each fixed distance d, there are $k - d$ pairs (x_i, x_j) that have this distance.
- The sum of all distances is given by:

$$\sum_{d=1}^{k-1} d \times (k - d)$$

This sum can be expressed as:

$$\sum_{d=1}^{k-1} d \times k - \sum_{d=1}^{k-1} d^2$$

These two sums, the sum of the first $k-1$ integers and the sum of the squares of the first $k-1$ integers, can be calculated as follows:

$$\sum_{d=1}^{k-1} d = \frac{k(k-1)}{2}$$

$$\sum_{d=1}^{k-1} d^2 = \frac{k(k-1)(2k-1)}{6}$$

The total sum of all distances then becomes:

$$\sum_{d=1}^{k-1} \left(d \times k - d^2\right) = \frac{k(k-1)}{2} \times k - \frac{k(k-1)(2k-1)}{6} = k \times \frac{(k^2-1)}{6}$$

By dividing the summed distance by all $\binom{k}{2}$ pairs, we get:

$$\frac{k \times \frac{(k^2-1)}{6}}{\binom{k}{2}} = \frac{k \times \frac{(k^2-1)}{6}}{k \times \frac{(k-1)}{2}} = \frac{k+1}{3}$$

The expected distance in one dimension is then:

$$E[|x_i - x_j|] = \frac{1}{\binom{k}{2}} \sum_{0 \le i < j \le k} |x_i - x_j| = \frac{k+1}{3}$$

Expected Manhattan Distance in Two Dimensions

The expected Manhattan distance on a 2D grid is the sum of the expected distances in the x- and y-dimensions. Since the distribution of x and y coordinates is independent and identical, we have:

$$E[D_{ij}] = E[|x_i - x_j|] + E[|y_i - y_j|] = \frac{k+1}{3} + \frac{k+1}{3} = \frac{2(k+1)}{3}$$

Expected Distance Conclusion

The expected Manhattan distance between two randomly chosen points on a $k \times k$ grid, where $k = \lceil \sqrt{n} \rceil$, is given by:

$$E[D_{ij}] = \frac{2(k+1)}{3}$$

By calculating the distance on the x-axis and y-axis separately, the total expected distance is achieved by adding them together since the Manhattan distance treats distances on each axis independently.

Let $E[D_{w_{ij}}]$ represent the expected weighted Manhattan distance between nodes i and j. This is defined as:

$$E[D_{w_{ij}}] = \sum_{i<j} w_{ij} \times \frac{2(k+1)}{3}$$

Algorithm

Let $t \in \mathbb{N}$ be a user-defined variable representing the number of iterations that the algorithm will run. The random assignment of n nodes to a $k \times k$ grid is done in linear time, $O(n)$.
The calculation of the combined weighted sum of distances for the complete graph G is done in:

$$n \times \frac{n-1}{2} \times O(1) \in O(n^2)$$

Each result can be evaluated using upper and lower bounds, as well as the expected weighted distance $E[D_{w_{ij}}]$. By running this algorithm t times and comparing each iteration to the previous minimum weight solution, this approach yields an approximation algorithm that runs in:

$$t \times O(n^2)$$

By running the algorithm multiple times, we may improve the weighted distance by finding a more effective placement of nodes that will lower the total weighted distance.

Analysis

Let ALG be the output from our randomization algorithm. The **Upper Bound** represents the worst-case scenario for the performance of ALG, where nodes are placed at the maximum possible distance from each other.

$$\text{Upper Bound} = \sum_{i<j} w_{ij} \times 2(k-1)$$

The random placement of nodes results in an expected weighted distance given by:

$$E[D_{w_{ij}}] = \sum_{i<j} w_{ij} \times \frac{2(k+1)}{3}$$

The randomization assignment of nodes and the comparison of the Expected weighted distance gives us an average-case performance analysis of ALG, while the **Upper bound** serves as a worst-case scenario.

Worst-case approximation ratio

Let's define $R_{\max}$ as the worst-case approximation ratio between ALG and an optimal solution OPT:

$$R_{\max} = \frac{\text{Upper Bound}}{\text{Lower Bound}}$$

The worst-case scenario refers to the configuration of node placements that results in the maximum possible weighted Manhattan distance. Due to the distribution of weights in W, the approximation ratio may vary. In this worst-case scenario, the upper bound is given by: Upper Bound $= \sum_{i<j} w_{ij} \times 2(k-1)$ and the lower bound is given by: Lower Bound $= \sum_{i<j} w_{ij} \times 1$, which yields a worst-case approximation ratio of:

$$R_{\max} = \frac{\sum_{i<j} w_{ij} \times 2(k-1)}{\sum_{i<j} w_{ij} \times 1} = 2(k-1)$$

Minimum Approximation Ratio

We may lower the approximation ratio by recalculating the bounds. This can led to a more accurate approximation ratio and we can make a better evaluation of any results produced by our algorithm. Let $R_{\min}$ be the lowest possible approximation ratio between or algorithm ALG and an optimal solution OPT. Here, the **Upper Bound** is given by $\sum_{i<j} w_{ij} \times \frac{3}{2}(k-1)$, and the **Lower Bound** is given by $\sum_{i<j} w_{ij} \times 2$. We get:

$$R_{\min} = \frac{\sum_{i<j} w_{ij} \times \frac{3}{2}(k-1)}{\sum_{i<j} w_{ij} \times 2} = \frac{3(k-1)}{4}$$

By analyzing the distribution of weights, we may lower the approximation ratio by more than half, which is useful for a more accurate evaluation. If no information about the weight distribution is available, the worst-case ratio must be assumed.

Randomized approximation ratio

Due to the distribution of weights in W, different randomized approximation ratios can be obtained. However, if the weight distribution isn't considered, the worst-case scenario will apply. Let R_{app} be the randomized approximation ratio between ALG and OPT:

$$R_{app} = \frac{E[D_{w_{ij}}]}{\text{Lower Bound}}$$

The worst-case scenario yields:

$$R_{app} = \frac{2(k+1)}{3}$$

If we consider the weight distribution, we get one of three ratios:

$$R_{app} \in \left\{ \frac{2}{3}(k+1),\ (k+1),\ \frac{1}{3}(k+1) \right\}$$

The randomized approximation ratio R_{app} provides a measure of the expected performance of the algorithm across different scenarios. By considering the expected weighted Manhattan distance and the weight distribution, we can more accurately estimate the quality of the output produced by the algorithm.