CS F402 Computational Geometry

Programming Project
Euclidean K-Supplier Problem

Raj Kashyap Mallala
2017A7PS0025H

L Srihari
2017A7PS1670H

T Naga Sai Bharath
2017A7PS0209H

June 2020

1 Problem Statement

Given a set $\boldsymbol{F}$ of $\boldsymbol{m}$ points and another set $\boldsymbol{C}$ of $\boldsymbol{n}$ points in the plane, the objective is to open a set $\boldsymbol{F}' \subseteq \boldsymbol{F}$ of $\boldsymbol{k}$ facilities such that the maximum distance of any client in $\boldsymbol{C}$ to its nearest open facility in $\boldsymbol{F}$' is minimized. The distances between the points satisfy two properties: symmetry (i.e. d(u,v) = d(v,u) for all $u, v \in \boldsymbol{F} \cup \boldsymbol{C}$) and triangle inequality (i.e $d(u, v) + d(v, w) \geq d(u, w)$ for all $u, v, w \in \boldsymbol{F} \cup \boldsymbol{C}$)

Let $d(f, \boldsymbol{F}')$ = minimum distance from point f to g where g is any point in $\boldsymbol{F}$'. Then,

$$d(f, \boldsymbol{F}') = min\, d(f, g) : g \in \boldsymbol{F}' \tag{1}$$

The given problem can be expressed as,

minimize $max\ d(f,\boldsymbol{F}')$, where

$f \in \boldsymbol{C}$ and $\boldsymbol{F}' \subseteq \boldsymbol{F}$ such that $|\boldsymbol{F}'| = \boldsymbol{k}$

2 Introduction

Multiple applications of the euclidean K-supplier problem can be found. One obvious application is in the domain of facility location in supply - chain markets and businesses, commonly observed in large supermarket chains. Here the consumer outlets form the set $\boldsymbol{C}$ and locations of potential intermediate suppliers form the set $\boldsymbol{F}$.
Other applications include opening hospitals in a city with limited capacities,

opening of centres for the disposal of obnoxious substances, setting up servers in a network, data mining and information retrieval such as data clustering.

3 About the paper in concern

4 Results

The $(1+\sqrt{3})$ approximation Euclidean k-Supplier algorithm proposed in [1] has two major sub-routines, namely, An algorithm for finding maximum matching in general graphs [2]) and An optimal algorithm for approximate nearest neighbor searching in fixed dimensions, which are by themselves well researched in the algorithmic domain (Sunil Arya and David M. Mount [4]). In this programming assignment, the first major subroutine on Maximum Matching, also known the MV algorithm, has been programmed from the scratch based on the algorithmic exposition given by Paul A. Peterson and Michael C. Loui [3]. The second major subroutine on Approximate nearest neighbors has been implemented with the aid of the ANN library provided for public use by the authors themselves [5]. The complete coding was done in C plus plus programming language. As the performance and correctness of the subroutines are crucial for efficient working of the main algorithm, we provide the results of unit tests on these sub routines first. The unit tests are followed by results of the complete algorithm.

4.1 Unit test on component Algorithms

4.1.1 Maximum Matching Algorithm (MicaliVazirani.cpp)

4.1.2 Approximate nearest neighbour search

References

[1] Viswanath Nagarajan, Baruch Schieber, and Hadas Shachnai *The Euclidean k-Supplier Problem*. Integer Programming and Combinatorial Optimization. IPCO 2013.

[2] Silvio Micali, Vijay V. Vazirani *An algorithm for finding maximum matching in general graphs*. 21st Annual Symposium on Foundations of Computer Science (sfcs 1980)

[3] Paul A. Peterson, Michael C. Loui *The general maximum matching algorithm of micali and vazirani*. Algorithmica 3, 511–533 (1988). https://doi.org/10.1007/BF01762129

[4] Sunil Arya, David M. Mount *Approximate nearest neighbor queries in fixed dimensions*. SODA '93: Proceedings of the fourth annual ACM-SIAM symposium on Discrete algorithms, January 1993

[5] Sunil Arya, David M. Mount *ANN: A Library for Approximate Nearest Neighbor Searching.* http://www.cs.umd.edu/ mount/ANN/