Finding Nearest Facility for Multiple Customers using Voronoi Diagram

Ravi Agarwal
Computer Science & Engineering Department
Thapar University
Patiala, India
raviagar25@gmail.com

Dr. Deepak Garg
Computer Science & Engineering Department
Thapar University
Patiala, India
dgarg@thapar.edu

Abstract— **Searching for a facility near to many customers is such a problem that even an approximately correct answer can save lot of labor, time and money. A possible solution for decision makers to reach facility approximately nearer to all customers has been discussed. It took time of order O(n log n) as it is based on voronoi diagram of order O(n log n) and its own time is of linear order. The proposed algorithm tackled the problem of finding the nearest facility for multiple customers by considering two criteria. The first one was minimizing the aggregate distances i.e. sum of total distances covered by all the customers. The second one was minimizing the maximum difference i.e. the difference between the farthest customer and the nearest customer. The approach given here has used Plane sweep algorithm or Fortune's algorithm for voronoi diagram construction algorithm as its base algorithm because it is one the most efficient algorithm known for computing voronoi diagram.**

Keywords— facility location, Plane sweep or Fortune's algorithm, voronoi diagram

I. Introduction

In everyday life people come across many times with situations like finding a place which is near to everyone who wants to gather for a particular reason. For example, there are four friends who live in different parts of a city. They decided to watch a movie running in all theatres in the city. So they all now plan to meet at a theatre. But the question is which one? Which of the theatre they should select so that it is almost at equal distance from every friend? Two situations arise when finding the solution of this condition. In first case they may come with a solution as a theatre which is nearer to three of them but farther from fourth friend i.e. nearer to many but farther for remaining. No doubt in this way they can minimize the total distance covered by all the friends but at the cost of few friends covering major distance, which is not fair. In that condition the friend farther from selected theatre may decide not to come. But they don't want that their friends take a decision like this. So they want to find the theatre which is at a reasonable distance from every friend's location. In the second case, they want to select a theatre such that the difference between the distances travelled by any two friends is minimum. But in this case they may select a theatre far from everyone that is at a distance roughly equal for every friend; hence the difference between distances travelled by any two friends is minimized. So this is also not the solution we are looking for.

Hence a solution is required which gives a facility [1] (theatre in above example) which is near to every query point (friends in above example). This paper deals with the problem as stated above to find a point (object) in space which is approximately at equal distance from multiple query points [2].

Bin Yi+ and Rongheng Li [3] considered one kind of uncapacitated facility location problem which is termed as k-product uncapacitated facility location problem with no-fixed costs (k-PUFLPN). The problem can be defined as follows: There is a set of demand points, where clients are located and a set of potential sites, where facilities of unlimited capacities can be set up. K different kinds of products are there. Each client needs to be supplied with k different kinds of products by a set of k different facilities and each facility can be set up to supply only a distinct product, with no fixed cost. A non-negative cost of shipping goods is there in between each pair of locations. These costs are assumed to be symmetric and also satisfy the triangle inequality. A set of facilities, which are to be opened, and their designated products is to be selected and has to find an assignment for each client within a set of k facilities to minimize the sum of the shipping costs. An approximation algorithm was proposed with a performance guarantee of (3/2) k -1 for the k-PUFLPN.

The competitive facility location problems has been investigated in many papers and been a subject of interest for many researchers. In most of the papers competitive location model for two competitors are given. Shiode *et al.* [4] done it for three competitors, three companies that are in mutual competition to each other and intend to locate their facility on linear market. It is known that Nash equilibrium solution does not exist for location problem of three or more competitive facilities. The demands are continuously distributed on the market and facilities are located in some specific order of sequence A, B and C. Stackelberg equilibrium solution for three competitive facilities are considered. It considered the decision problems of three stages. In the first stage problem it considers the facility location for A so that it is optimal with respect to B and C. In the second stage problems it finds the optimal location of facility B with respect to facility C by using the information related to facility A. In final stage problem it finds the optimal location of the facility C by utilizing the information stored in the facilities A and B. This model has been represented as three stage decision problems.

With multiple transport alternatives, Yosuke Takano *et al.* [5] presented a modeling and optimization of facility location and distribution planning problems. The problem is to determine an optimal facility location with respect to management strategy, by using huge physical distribute on data. Two types of problems are considered. The first one is a facility location problem with transportation from depot to customer and a direct transportation from factory to customer simultaneously. Large size problem is solved efficiently by applying Lagrangian relaxation. The second one is the competitive facility location problem with multiple competing companies. With this distribution profit's effectiveness is shown by collaborative decision making.

Environmental regulations are forcing companies to comply with environmental policies so as to control carbon emission. It is required for companies to green their supply chains. One way to do this is extending the supply chain to collection and recovery of products in closed loop configuration. Profitable reverse logistics to restore the recovered product can be used so as to resell it in primary or secondary market. Ali Diabat *et al.* [6] have introduced a multi-echelon multi-commodity facility location problem with a trading price of carbon emissions and a cost of procurement. If carbon cap is higher than the total emission, then company gains but if carbon cap is less than the total emission, then company might incur cost.

This work deals with finding the facility that is near for many customers [7]. It considered two criteria for obtaining a facility as result. The first criteria being the aggregate distance i.e. the sum of distances of all the customers from the facility [8]. The second criterion is the maximum difference i.e. the difference between the distance of farthest customer and nearest customer from the facility. Here an approach is discussed which minimizes the first criteria and tries to get the best possible minimum value for second criteria. The results obtained by performing according to proposed method came out as it was expected. In all test cases, the first criteria is satisfied with second criteria satisfied up to the best possible value for given discrete locations of facilities and customers. Voronoi diagram has been used to solve the problem. The complexity of the proposed method is found to be O(n log n) as the complexity of construction of voronoi diagram using Fortune's algorithm [9] is O(n log n) and its own complexity is of linear order. This work can be used as the reference for further research in this domain.

The study is organized as follows. Section II formulates the problem in mathematical form. In section III, preliminaries of the technique used, has been discussed. In this section basic concept and properties of voronoi diagram have been discussed. Proposed algorithm is presented in section IV. Working of algorithm is explained in this section with the help of flow chart. Section V shows experimental results and section VI concludes this study and states future works.

II. PROBLEM FORMULATION

First Let $F\{f_1,f_2 \ldots\ldots f_m\}$ is a set of facilities located at different places and $C\{c_1,c_2 \ldots\ldots c_n\}$ is a set of n customer residing at different location and dist(x,y) means distance between x and y.

(i)Aggregate Distance, $AD(f_j,C) = \sum dist(f_j,c_k), \forall c_k \in C$

(ii)Maximum difference, $MD(f_j,C) = MAX[dist(f_j,c_k)] - MIN[dist(f_j,c_k)], \forall c_k \in C$

The aim here is to find a facility f_j that minimizes the maximum difference and aggregate distance i.e. which minimizes the functions (i) and (ii) both.

III. VORONOI DIAGRAM

The name voronoi was coined after the name of a Russian mathematician Georgy Feodosevich Voronoy. He did initial work on voronoi structure. It is also called by many other names like Dirichlet tessellations, Wigner-Seitz zones, Thiessen polygons and Domains of actions, most of which are the names of early researchers on this construct in different fields of science.

The Voronoi diagram of n sites in a plane and its dual, the Delaunay triangulation, are considered to be the most important constructions or technique in the area of computational geometry as well as in some other fields of vision and biology, archeology, computer-aided design, chemistry, geography, pattern recognition, physics, etc. Because of practical and theoretical usefulness of Voronoi diagrams (and Delaunay triangulation), their characteristic features, properties as well as algorithms to construct these diagrams are extensively studied and covered in standard text books of the field, and in numerous papers.

A. Definition and Properties of Voronoi diagram

Let $P = \{p_1, p_2 \ldots p_n\}$ represents a set of n distinct points in any plane then these points can be considered as sites for voronoi diagram. The voronoi diagram of set $P\{p_1,p_2 \ldots p_n\}$ can be defined as the division of plane into n cells or region, , one for each site in P set, with the property that a point q lies in the cell corresponding to a site p_i if and only if the $dist(q,p_i) < dist(q, p_j)$ for each $p_j \in P$ with $j \neq i$. In other words, the voronoi diagram is the division of space in such a way so that any point that lies in the region of a site has this site as the nearest site as compared to others [10]. Sometimes the meaning of voronoi diagram is taken as the subdivision of space showing only vertices and edges. $V(p_i)$ is used to represent the cell of site p_i, said as voronoi cell of p_i. To understand the structure of complete voronoi diagram it is required to study the structure of single voronoi cell first.

- The bisector between two points p and q is defined as the perpendicular bisector of pq. The bisector between two points divides the planes into two equal halves. The half-plane that contains the p in it is represented by h(p,q) and the half plane that contains the point q in it is represented by h(q,p). A point r lies in the half plane of p iff dist(r,p) < dist(r,q).
- A voronoi cell is created as a result of area bounded by the perpendicular bisectors between a site and

every other site in the plane. But as an observation it can be showed that not every bisector define the edges of a voronoi cell. In other word a voronoi cell is created by the intersection of the half planes which contains the site in it. This thing can be stated in formal way as

$$V(p_i) = \bigcap_{1 \le j \le n, j \ne i} h(p_i, p_j)$$

- According to above property, in extreme case the voronoi cell is made up of n-1 half plane intersection. Thus $V(p_i)$ is a region constructed as the intersection of n-1 half planes. Hence it is convex in shape that is bounded by at most (n-1) vertices and at most (n-1) edges.

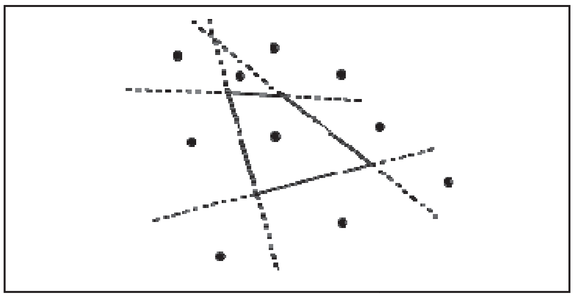

Fig. 1. Single cell of voronoi diagram

- Every edge in voronoi diagram is a straight line. It may be a line segment or half line (bounded from one side and free from other). Sometimes infinite lines also represent an edge of a voronoi diagram but that is a special case where all the sites are collinear and other edges are also infinite lines.
- It can be put in the form of formal statement as property of voronoi diagram: Let P$\{p_1,p_2 \ldots p_n\}$ be a set of points in the plane representing as sites. The voronoi diagram will have all the edges as infinite parallel edges if all the sites are collinear otherwise the voronoi diagram will have one (in case when there are only three sites) or more than one intersection point (vertices) and the edges will be either line segments or half infinite line. In this case the voronoi diagram will be a connected diagram.

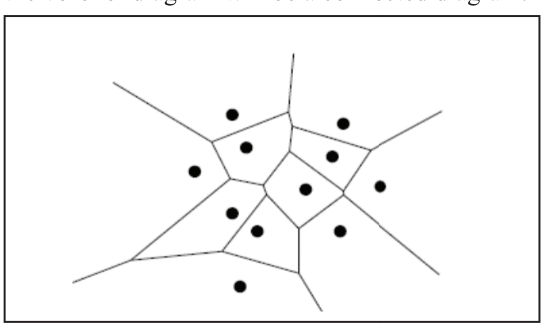

Fig. 2. A typical voronoi diagram

- In a voronoi diagram, for sites (n) $\geq$ 3 the number of vertices is at most 2n-5 and the number of edges is at most 3n-6.
- Edges are the part of perpendicular bisectors between two sites. The numbers of bisectors are quadratic in nature but the complexity of voronoi diagram is linear. Hence not all the bisectors define the edges of the voronoi diagram and hence not all the intersection of these bisectors defines the vertices of voronoi diagram.

IV. Algorithm Proposed

The approach of finding the optimal facility is divided into two algorithms. First is finding the voronoi diagram of facilities using any voronoi diagram construction algorithm. In this work, Fortune's algorithm is used to calculate the voronoi diagram. Second computing the location of optimal facility by the proposed algorithm taking voronoi diagram of facilities Vor(F) and set of locations of customers C as input. The second algorithm uses the result of first algorithm to give output as a facility at optimal distances from all customers.

A. Proposed Algorithm

NEARFACILOC(Vor(F) , C)

Input: voronoi diagram of facilities Vor(F), set of customer's locations, C

Output: facility f_o optimal for all customers

1. Give as input voronoi diagram, Vor(F) of facilities and locations of customers C.
2. Create voronoi diagram for customers at different location by using VORONOIDIAGRAM(C). // to compute voronoi diagram of given input Fortune's algorithm VORONOIDIAGRAM() as given in Mark de Berg [6] is used.//
3. Find all the vertices of this voronoi diagram created in step 2.
4. Find the coordinates of these vertices.
5. Calculate the mean, M, of these vertices.
6. Locate this mean in the original voronoi diagram of facilities.
7. Find in which region or cell this mean, M, lies.
8. Find the facility (site) of the above selected region or cell.

B. Working of Proposed Algorithm

The proposed algorithm takes voronoi diagram of facilities as computed by using Fortune's sweep line algorithm as one of the input, the customer's locations as being the other input of this algorithm. First of all, by treating customer's location as sites, voronoi diagram of customer's location is computed using the Fortune's sweep line algorithm. The voronoi diagram is calculated in optimal time of O(n log n) which is the time complexity of Fortune's algorithm. Now in this computed voronoi diagram of customers locations, all the vertices are located.

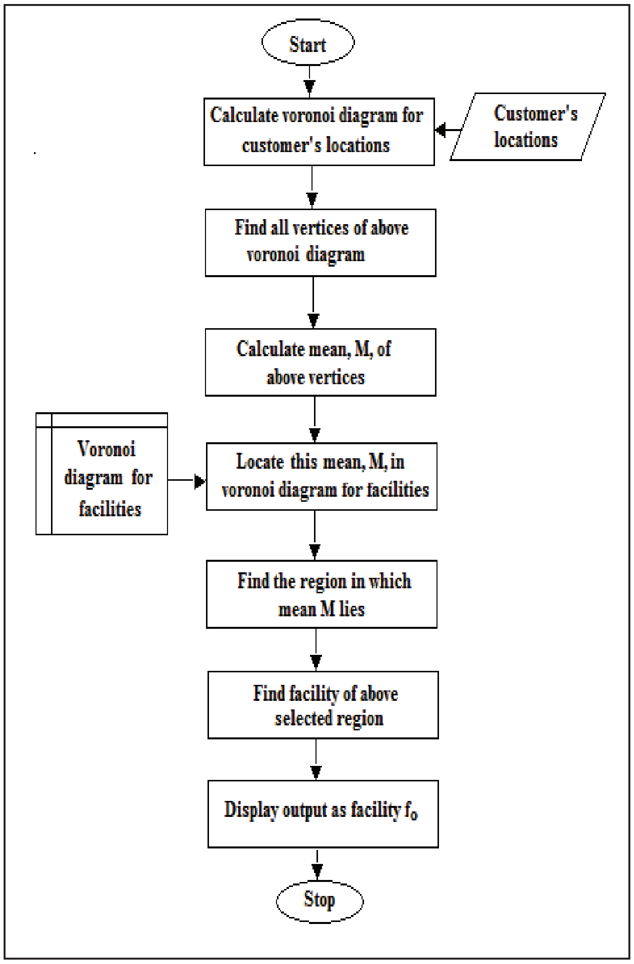

Fig. 3. Flowchart of proposed algorithm

By using these vertices, average or mean of vertices is calculated. After calculating mean of new voronoi diagram of customers, the old voronoi diagram of facilities is required. The old voronoi diagram is stored in main memory and can be used as and when required. Then the calculated mean of new voronoi diagram is located in old voronoi diagram. When located, the region of old voronoi diagram in which mean lies, is selected. Then the facility as a site of this selected region is found. This facility is the required output. This working can be shown in the form of flowchart given in Fig. 3. This facility will minimize the aggregate distance i.e. the sum of the distances of all the customers from this facility. Also this facility selected will be the one which tries to minimize the maximum difference i.e. the difference of the distance between the maximum distance and the minimum distance, by always keeping the first minimization criteria at priority (minimizing aggregate distance).

V. Experimental Result

For testing the working of proposed algorithm, it is run for different test cases. It works fine for every test case. Three test cases are given here to show its validity.

A. Test case1

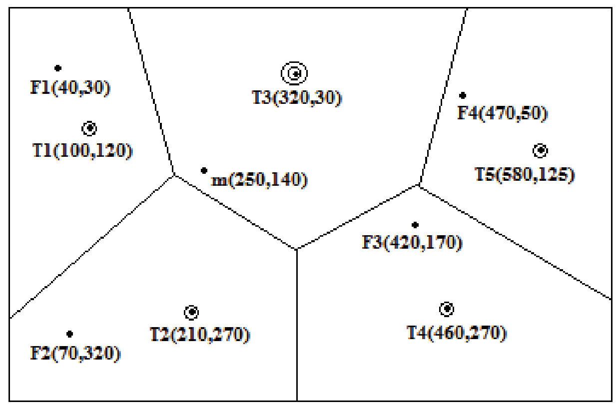

Fig. 4. Test case 1

The box represents area of a city. There are some theatres located in different parts of the city represented as T1 to T5 in Fig. 4. Voronoi diagram is created for these theatres using Fortune's algorithm. There are four friends F1 to F4 who resides at different locations in the city. The proposed algorithm when run for this test case, calculates the mean and finds the region in which this mean lies(Table I). Then it gives theatre of the selected region T3 as the output.

TABLE I. Result of Test Case 1

Facility	**Distance of Customer1, F1 (40,30) D1**	**Distance of Customer2, F2(70,320) D2**	**Distance of Customer3, F3(420,170) D3**	**Distance of Customer4, F4 (470,50) D4**	**Maximum Distance, M1**	**Minimum Distance, M2**	**Aggregate Distance, AD = D1+D2+D3+D4**	**Maximum Difference, MD = M1-M2**
T1(100,120)	108.166	202.238	323.882	376.563	376.563	108.166	1010.849	268.397
T2(210,270)	294.109	148.661	232.594	340.588	340.588	148.661	1015.952	191.927
T3(320,30)	280.000	382.884	172.046	151.328	382.884	151.328	986.258	231.556
T4(460,270)	483.736	393.192	107.703	220.227	483.736	107.703	1204.858	376.033
T5(580,125)	548.293	546.008	166.208	133.135	548.293	133.135	1393.644	415.158

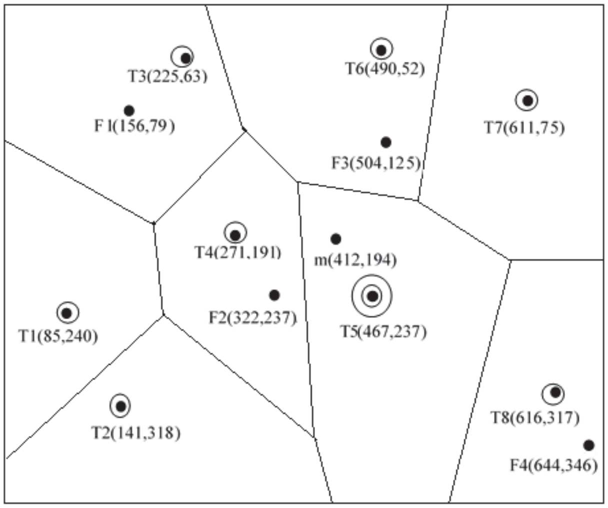

Fig. 5. Test case 2

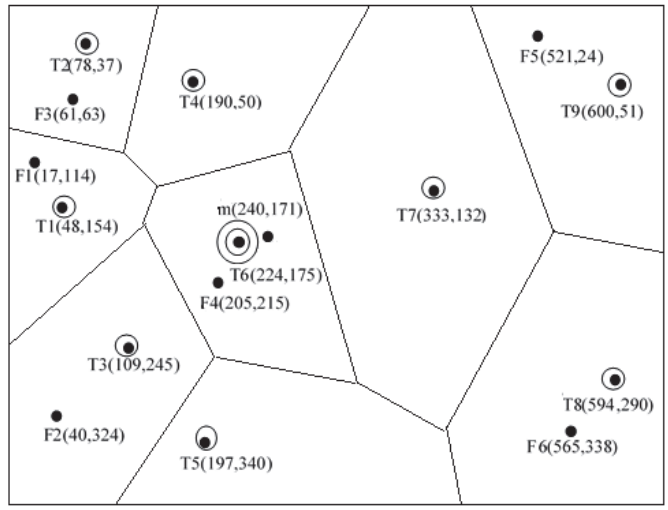

Fig. 6. Test case 3

TABLE I. RESULT OF TEST CASE 2

Facility	Distance of Customer 1, F1(156,79) D1	Distance of Customer 2, F2(322,237) D2	Distance of Customer 3, F3(504,125) D3	Distance of Customer 4, F4(644,346) D4	Maximum Distance, M1	Minimum Distance, M2	Aggregate Distance, AD = D1+D2+D3+D4	Maximum Difference, MD = M1-M2
T1 (85,240)	175.960	237.019	434.495	568.961	568.961	175.960	1416.435	393.001
T2 (141,318)	239.470	198.298	411.118	503.779	503.779	198.298	1352.665	305.481
T3 (225,63)	70.831	199.211	285.806	505.618	505.618	70.831	1061.466	434.787
T4 (271,191)	160.527	68.680	242.167	403.923	403.923	68.680	875.297	335.243
T5 (467,237)	348.833	145.000	117.953	207.870	348.833	117.953	819.656	230.88
T6 (490,52)	335.090	249.898	74.330	331.892	335.090	74.330	991.210	260.76
T7 (611,75)	455.018	331.308	118.106	273.002	455.018	118.106	1177.434	336.912
T8 (616,317)	517.923	304.690	222.279	40.311	517.923	40.311	1085.203	477.612

Facility T3(320,30), as chosen by proposed method has the minimum aggregate distance from all customers as shown in TABLE I. Also it minimized the maximum difference up to second best possible result. Hence friends will select the theatre T3 so that all have to travel less distance. The facility T5(467,237) is selected in second test case by the proposed method as shown by TABLE II. It gives the minimum aggregate distance from all the available options. Also it gives the minimum of the maximum difference i.e. it minimizes the maximum possible difference between any two customers. Hence the proposed algorithm works as expected. In third test case also the selected facility T6(224,175) is the right choice as it minimizes the aggregate distance and also it minimizes the maximum difference up to third from the minimum which is satisfactory as it is in addition to the minimum of aggregate distances (TABLE III). Hence proposed algorithm works fine in all three test cases including this one.

VI. CONCLUSION

As the facilities are discrete objects [11], it is not always possible to have a site that gives minimum for both the optimizing functions. Hence it always give the minimum aggregate distance with best possible solution for minimum of maximum difference in combination with first optimizing criteria. The time complexity of the proposed algorithm is same as time complexity of Fortune's algorithm because it

TABLE II. RESULT OF TEST CASE 3

Facility	Dist of Cust 1, F1 (17,114) D1	Dist of Cust 2, F2 (40,324) D2	Dist of Cust 3, F3 (61,63) D3	Dist of Cust 4, F4 (205,215) D4	Dist of Cust 5, F5 (521,24) D5	Dist of Cust 6, F6 (565,338) D6	Max Dist, M1	Min Dist, M2	Agg. Dist, AD = D1+D2+D3+D4 +D5+D6	Max Diff, MD = M1-M2
T1 (48,154)	50.606	170.188	91.924	168.434	490.540	548.767	548.767	50.606	1520.459	498.161
T2 (78,37)	98.234	289.505	31.064	218.662	443.191	572.512	572.512	31.064	1653.168	541.448
T3 (109,245)	160.078	104.890	188.223	100.578	467.531	465.387	467.531	100.578	1486.687	366.953
T4 (190,50)	184.459	312.372	129.653	165.680	332.020	472.831	472.831	129.653	1597.015	343.178
T5 (197,340)	288.922	157.813	308.586	125.256	452.584	368.005	452.584	125.256	1701.166	327.328
T6 (224,175)	215.801	236.764	197.770	44.283	333.182	377.955	377.955	44.283	1405.755	333.672
T7 (333,132)	316.512	350.304	280.615	152.555	216.813	310.258	350.304	152.555	1627.057	197.749
T8 (594,290)	603.245	555.042	579.326	396.164	275.835	56.080	603.245	56.080	2465.692	547.165
T9 (600,51)	586.394	623.000	539.133	427.693	83.486	289.126	623.000	83.486	2548.832	539.514

uses Fortune's algorithm as the base algorithm. It spent constant amount of time on computing the mean. Hence overall complexity of proposed method is O (n log n). The proposed algorithm can be implemented using hybrid techniques which is the combination of techniques to find the nearest facility. Also voronoi construction can be done using an approach specific to the situation. It may work much faster when implemented in distributed environment where local system find out their local optimal and then using these local solutions a global optimal solution can be obtained.

REFERENCES

[1] Takano, Y.; Nishi, T.; Inuiguchi, M., "Facility location and distribution planning with multiple transport alternatives," SICE Annual Conference (SICE), 2011 Proceedings of, vol., no., pp.967, 972, 13-18 Sept. 2011.

[2] Ling Hu; Wei-Shinn Ku; Bakiras, S.; Shahabi, C., "Spatial Query Integrity with Voronoi Neighbors," Knowledge and Data Engineering, IEEE Transactions on , vol.25, no.4, pp.863,876, April 2013.

[3] Bin Yi; Rongheng Li, "Approximation algorithm for the k- product uncapacitated facility location problem," Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on, vol.5, no., pp.602, 605, 9-11 July 2010.

[4] Shiode, S.; Kuang-Yih Yeh; Hao-Ching Hsia, "On optimal location for three competitive facilities," Computers and Industrial Engineering (CIE), 2010 40th International Conference on, vol., no., pp.1,5, 25-28 July 2010.

[5] Takano, Y.; Nishi, T.; Inuiguchi, M., "Facility location and distribution planning with multiple transport alternatives," SICE Annual Conference (SICE), 2011 Proceedings of, vol., no., pp.967, 972, 13-18 Sept. 2011.

[6] Diabat, A.; Abdallah, T.; Al-Refaie, A.; Svetinovic, D.; Govindan, K., "Strategic Closed-Loop Facility Location Problem With Carbon Market Trading," Engineering Management, IEEE Transactions on , vol.60, no.2, pp.398,408, May 2013.

[7] Drezner, Z., H.W. Hamacher edt., "Facility Location: Applications and Theory", Springer, New York, 2002.

[8] Zheng Hong-Zhen; Huang Jun-Heng; Zhan De-Chen, "Facility Location Optimization Methods Based on Aggregate and Disperse Moves," Fuzzy Systems and Knowledge Discovery, 2007. FSKD 2007. Fourth International Conference on, vol.4, no., pp.669, 673, 24-27 Aug. 2007.

[9] S. Fortune, "A sweepline algorithm for Voronoi diagrams". Algorithmica, 2:153–174, 1987.

[10] de Berg, M. van Kreveld, M. Overmars, M. Schwarzkozpf, "Computational Geometry: Algorithms and Applications". Springer-Verlag (2000).

[11] Shishebori, D.; Mahnam, M.; Nookabadi, A.S., "An efficient approach to discrete Multiple Different Facility Location Problem," Service Operations and Logistics, and Informatics, 2008. IEEE/SOLI 2008. IEEE International Conference on, vol.2, no., pp.2519, 2524, 12-15 Oct. 2008.