Algorithm for Obnoxious Facility Location Problem

Rattan Rana, Deepak Garg
Department of Computer Science & Engineering
Thapar University, Patiala (India)
Email: rattanrana77@gmail.com, dgarg@thapar.edu

Abstract

This study presents an endeavor to determine an optimal solution for the obnoxious facility location problem. Obnoxious materials are those which may cause harm to the health of human beings and pollute the environment. The obnoxious material management is a significant environmental issue. Since the social economical growth encourages to establishment of industrialization. While the waste material of industry or such plants necessitate a careful placement to avoid any kind of damage to the inhabitant of concerned area. During the last few decades, a significant work has been done in operation research, graph theory and computation complexity for the placement of desirable facilities due to increasing demand and production of goods. But the mentioned problem could not get a considerable attention of researchers. Hence this study provides a simplified approach to solve the obnoxious facility location problem on network for 1-center (only). 1-center means, to allocate only one center in a given region.

Keywords: *obnoxious material, obnoxious facility, location problem, waste material management,*

1. Introduction

Facility location problem is a most significant and sub problem of commerce optimization problem. A facility may be an outlet, workstation, selling point, hospital, school, police station, emergency care center, fire station, and warehouse. The foremost objective of facility location is to facilitate the inhabitants of any region with basic amenities with in minimum possible distance. Facility location problems emerged as a challenge for both public and private sectors. Since it is the moral responsibility of any state government to provide all basic facilities to their citizens like hospitals, schools, colleges, ration depots, and fire stations etc.

Similarly it is essential for all big and small production houses to reach up to last user of their goods.

Location of such facilities is very significant issue and needs to consider impact of various relevant parameters (such as distance, population and access time) on a location of facility. Broadly we can categories these facilities in two categories like desired facilities and undesired facilities, since all kind of facilities have some effect on quality life. Desired facilities are those which are desired by inhabitants to be placed in closer areas such as schools, hospitals, company outlets. Whereas, obnoxious facilities are those, which are never desired to be placed nearby areas by the inhabitants such as garbage dumps, and chemical plants etc., due to their adverse effects.

Obnoxious facility location problem deals with the proper placement of such materials which are preferred to be placed far from the populated area to prevent the inhabitants from health related issues as caused by such materials. There is a wide list of obnoxious materials such as waste dumps, nuclear power plants, chemical plants, electricity power plants, waste released by industry, airports, corrosive substances, gas plants, flammable liquids and solids, oxidizing substance, radioactive material, poisonous and infectious substances. If such materials are located closer to any populated area it may be dangerous for the life of mankind.

Keeping in view all adverse effects of the obnoxious facilities over the environment and population, it is crucial to locate these facilities away from the populated area. This study focuses on to develop an algorithm that can provide an optimal solution for obnoxious facility location problem. The subsequent section provides an intact vision to the recent work done till now to solve the problem.

2. Related Work

Although, the facility location problem has been remain a dominating research area for the scholars of operation research, graph theory and computational complexity. The obnoxious facility location problem has got some attention in 1970. When the modern industrialization was growing massively but the environmental issues were not addressed, Then it was realized how to reduce the adverse impact of industrialization on environment and society.

First of all Goldman et al [1] have addressed semi undesired facilities and highlighted all related issues and presented a model for noxious facility location problem. Thereafter Church et al. [2] has proposed a model for obnoxious facility location problem and suggested to locate such facilities away from the populated area.

Shobrys [3] has presented an approach for storage of nuclear fuel. Shobrys has given a combined approach that considered the location as well as routing problem of nuclear fuel. Since, it may be very risky to transport such kind of materials among the populated areas. Therefore this study has given a special attention to location and routing of nuclear fuel to reduce the risk and cost involved in transportation.

Caruso et al. [4] has developed a model for urban solid waste management and implemented using heuristic approach.

Several models were presented to summarize the core components of the location problem in literature. Erkut et al. [5] has provided an illustrative study of location models containing *Continuous, Discrete* and *Network* Location models. This study states that in case of continuous location model, facilities can be located in some *d*-dimensional space while discrete location model shows that the facilities can be located at some specified points. On the other hand network location model states that the facilities can be located on network. They have also considered about forbidden zones, which represents restricted sites that can not be candidate site for a facility.

Labbe [6] have presented a voting approach to solve the obnoxious facility location problem on network. Labbe presented a comparison between the anti Condorcet points and anti-median points. Karkazis et al [7] has proposed an algorithm for location of facilities causing atmospheric pollution in plane. The objective of this algorithm was to minimize the sum weighted risk factors for each vertex summed over all possible wind directions.

Giannikos [8] have presented a multi objective programming discrete model for locating treatment sites and routing of hazardous waste. Ben-Moshe et al [9] has proposed an algorithm for *k*-facilities, *n*-demand node and *m* regions. The objective of this algorithm was to maximize the minimum distance between demand nodes and facility. Cappanera [10] has proposed a model known as Obnoxious Facility Location and Routing (OFLR) model. He has implemented this using the Branch and Bound method. Chabini [11] have provided a study of all to one dynamic shortest paths problem. Chabini's algorithm has proven an optimal run time complexity that equals to the complexity of problem.

3. A Simplified Procedure to Allocate Obnoxious Facility Location

This section endow with the comprehensive methodological details of the presented approach. It is assumed that the facility to be located on network. Let us suppose that we have a network of nodes that can be represented as a graph $G= (V, E)$, V is a set of vertices (nodes) and E is as set of edges (path) that connects different vertices. Each edge is having a weighted distance d_{ij} (distance from vertex i to j) $\forall\ i,j \in V$.

$V = \{v_1, v_2, \ldots\ldots\ldots v_n)$

$E = \{e_1, e_2, \ldots\ldots\ldots e_m)$

The primary objective of the proposed approach is to determine candidate nodes where obnoxious facility can be provided on given network. Assuming that, we have to allocate single obnoxious facility location on any given network. Distance is a crucial parameter that may be highly significant for deciding the location of facility. Therefore, first of all it is indispensable to find out such nodes those are located at highest distance to all other present nodes in network. This can be attained by using the distance matrix. This distance matrix, help us to find out the maximum and minimum distance of each row. Suppose we have *D* as distance matrix of $[n \times n]$ order. We are also having two arrays *Max*[] and *Min*[] that represents maximum and minimum distance values respectively.

$$D = \begin{vmatrix} d_1^1 & d_2^1 & d_3^1 & d_4^1 \ldots\ldots\ldots\ldots & d_n^1 \\ d_1^2 & d_2^2 & d_3^2 & d_4^2 \ldots\ldots\ldots\ldots & d_n^2 \\ . & . & . & . & . \\ . & . & . & . & . \\ d_1^n & d_2^n & d_3^n & d_4^n \ldots\ldots\ldots\ldots & d_n^n \end{vmatrix}$$

Let $\alpha_1, \alpha_2, \alpha_3, \ldots\ldots\alpha_n$ be the maximum distance value elements and $\beta_1, \beta_2, \beta_3, \ldots\ldots\beta_n$ be the minimum distance value (but not zero) elements of 1^{st}, 2^{nd} ,3^{rd} …………n^{th} rows respectively. Therefore, we have two sub sets Max_d and Min_d having all the elements with highest distance value and lowest distance values respectively.

$$Max_d = \{\alpha_1, \alpha_2, \alpha_3, \ldots\ldots\alpha_n \}$$

$Min_d = \{ \beta_1, \beta_2, \beta_3, \ldots\ldots\beta_n \}$

Initialized both the arrays as shown below

$Max[] = Max_d$

$Min[] = Min_d$

Thereafter we have to find a pair $\alpha i\ \beta i$, where αi is having the maximum value in *Max*[] and βi is maximum distance value in *Min*[].

Now let us have an example to understand the procedure opted to implemented presented approach. Suppose we have a network of nodes as shown in Figure 1. A, B, C, D, and E are the connected nodes and the weighted distance between nodes is mentioned above each edge.

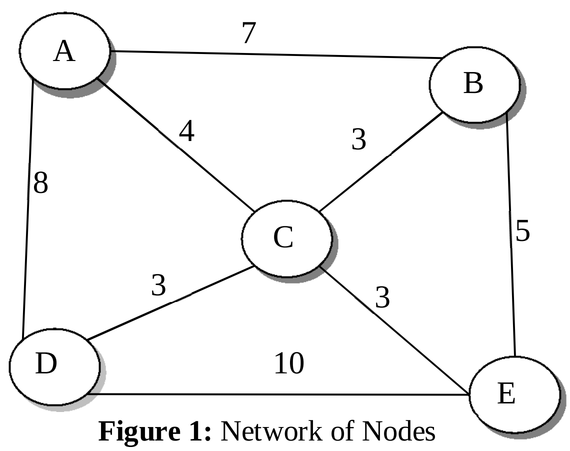

Figure 1: Network of Nodes

Table 1 presents the distance matrix for the network shown in Figure 1.

Table 1: Distance Matrix

	A	**B**	**C**	**D**	**E**
A	0	7	4	8	∞
B	7	0	3	∞	5
C	4	3	0	3	3
D	8	∞	3	0	10
E	∞	5	3	10	0

Now we have to proceed to get the maximum and minimum (more than zero) elements of each row and initialize the *Max*[] and *Min*[].

Max[] = {8, 7, 4, 10, 10};

Similarly *Min*[] = {4, 3, 3, 3, 3};

Then we have applied Sorting technique on *Min*[] to identify the maximum distance value. The index *i* of *Min*[] array which contains the maximum value will decide the maximum value index of *Max*[] array. As we have shown in our example that 4 (that is at index 0, and encircled in Table 1) is the maximum distance value of *Min*[], therefore the value at index 0 in *Max*[] will be considered as maximum value of *Max*[]. Hence we got a distance pair (maximum-8, minimum-4). It reveals that the candidate site for the obnoxious facility is node A (since 8, 4 both distances belongs to node A).

The main objective to provide priority to the maximum distance in *Min*[] array is only to increase the distance between connected nodes and obnoxious facility. In view of the fact that everybody wants that the obnoxious facility should be located as far as possible. Therefore we have tried to maximize the distance between nodes and obnoxious facility. Then the selected maximum distance from *Min*[] array is passed to FLoc() procedure to identify the appropriate node for obnoxious facility.

4. Results and Discussion

It is quite important to remind that the presented approach has considered a case where we have to allocate only one obnoxious facility on given network. Since such facilities are never desired to have massively. Additionally, such facilities are required to be allocated as far as possible. Keeping in view the fact of matter, we have not taken the maximum distance as main parameter to decide the location of facility. In fact the maximum distance among minimum distance values has been considered as major criterion.

This approach intends to increase the maximum distance as well as minimum distance of obnoxious facility from all adjacent nodes.

Algorithm to Allocate Obnoxious Facility

Step 1: **Initialize** int *Max*[], *Min*[];

Step 2: **Sorting** *max*[*i*] && *min*[*i*];

Rowid=0,

For(*i*=0 to *N*)

IF((Maximum<=*Max*[*i*] && *Minimum* <=*Min*[*i*]) || (*Minimum* <=*Min*[*i*]))

Then

Assign Maximum=*Max*[*i*];

Minimum=*Min*[*i*];

Rowid=*i*

End if

End for

More than 50 cases with different numbers of nodes has been undertaken to test the presented approach. The number of nodes covered in a set from 8 to 98. During the computation a few percent relative errors are reported as usual. But the major achievement of algorithm is that it has speed up the overall processing, consequently the time taken by the algorithm is comparatively low, as shown in Table 2. Computation times are expressed in CPU seconds Program executed on dual core 1.6 GHz Microprocessor with 1GB RAM and code is written in Turbo C++. Table 2 presents the execution time variation of a few cases out of total 50 tested cases. It reveals that the execution time increases with the number of nodes. But the increase in execution time is very marginal. During the testing phase we have observed a significant increase and decrease in execution time in a few cases, which usually occurs. In order to verify the execution time for such cases, it has been performed repeatedly.

Table 2: Results Obtained from Presented Approach

Total Nodes	No of Iterations	Time Taken by Algorithm (in CPU seconds)	Average Time (In CPU seconds)
15	1	2.45	2.45
	2	2.46	
	3	2.45	
20	1	3.30	3.30
	2	3.30	
	3	3.30	
25	1	4.50	4.52
	2	4.55	
	3	4.52	
35	1	5.93	5.93
	2	5.93	
	3	5.92	
42	1	6.80	6.80
	2	6.80	
	3	6.80	
57	1	7.99	7.99
	2	7.98	
	3	7.99	
66	1	8.70	8.70
	2	8.71	
	3	8.70	
83	1	10.35	10.36
	2	10.36	
	3	10.36	
98	1	11.26	11.25
	2	11.25	
	3	11.25	

When the consistent time is reported by the algorithm in consecutive execution then it has been noted as final execution time for a particular case (such exceptions are not shown in Table2). But for majority of cases the presented algorithm has shown very consistent behavior as described in Chart 1.

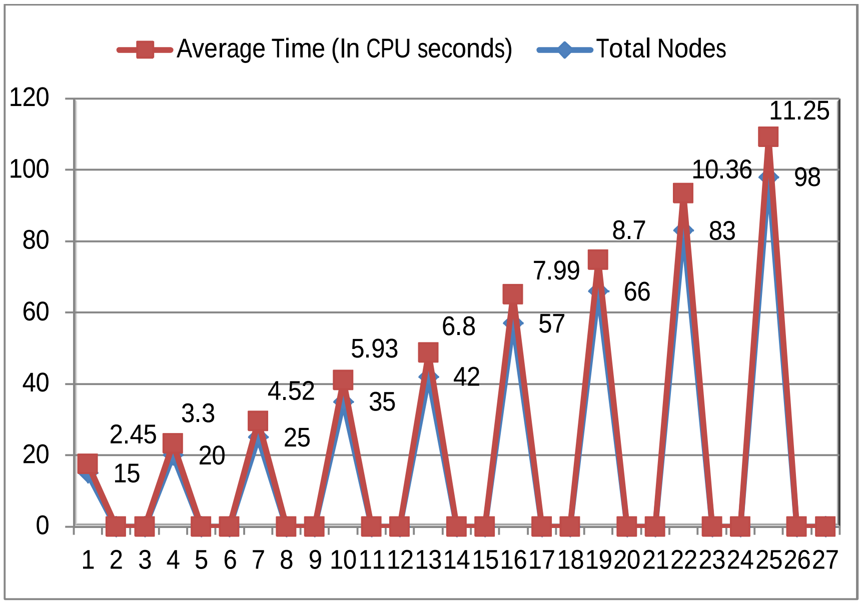

Chart 1: Graphical Presentation of Results Obtained

5. Conclusion

The problem undertaken has a great significance and impact on the structure of modern society as well as environment. Keeping in view all the facts of matter, it is essential to locate the obnoxious facility away from the residential areas for the sake of healthy environment and health of inhabitants. The presented approach provides an effective and simplified method to allocate obnoxious facility locations (for 1-center only). The major achievement of presented approach is that it increases the minimum distance between obnoxious facility and connected nodes. A few related issues such as transportation of obnoxious material and their routing is still an open problem for research.

References

[1] Goldman A.J., Dearing P. M., 1975. “Concepts of Optimal Location for Partially Noxious Facilities”. ORSA Bulletin, 23(1), B-31.

[2] Church R.L., Garfinkel R.S., 1978. “Locating an Obnoxious Facility on a Network”, Transportation Science, 12, 107-118.

[3] Shobrys D., 1981. "A model for the Selection of Shipping Routes and Storage Locations for a Hazardous Substance". Ph.D. Thesis, Johns Hopkins University, Baltimore

[4] Caruso C., Colorni A., Paruccni M., (1986) "The Regional Urban Solid Waste Management System: A modeling approach". European Journal of Operation Research, 70:16-30.

[5] Erkut E., Neuman S., 1989, "Analytical Models for Locating Undesirable Facilities". European Journal of Operational Research, 40, 275–291.

[6] Labbe M., 1990. "Location of an Obnoxious Facility on a Network: a Voting Approach". Networks, 20, 197-207.

[7] Karkazis J., Papadimitriou C., 1992. "A Branch-and-Bound Algorithm for the Location of Facilities Causing Atmospheric Pollution". European Journal of Operational Research, 58, 265-281.

[8] Giannikos I., 1998. "A Multi-Objective Programming Model for Locating Treatment Sites and Routing Hazardous Waste". European Journal of Operational Research, 104, 333-342.

[9] Ben-Moshe B., Katz M. J., Segal M., 2000. "Obnoxious Facility Location: Complete Service with Minimal Harm". International Journal of Computational Geometry, 10(6).

[10] Cappanera P., 2004, "Discrete Facility Location and Routing of Obnoxious Activities". Discrete Applied Mathematics, 133, 3-28.

[11] Chabini I., 2010. "Discrete Dynamic Shortest Path Problems in Transportation Applications: complexity and algorithms with optimal run time.