Solid Waste Management using Shortest Path Algorithm

Gulab Singh[1], Brajesh Singh[2], Shubham Rathi[3], Saurabh Haris[4]
[1]*M.Tech (Environment Science and Engineering),* [1, 2, 3, 4] *Civil Engineering Department, HBTI Kanpur (UP)*
[1]gulabsingh.js@gmail.com, 2 brajeshenvirotech@gmail.com, [3]right.shubhamrathi@gmail.com,
[4]100rabh.haris@gmail.com

Abstract— **Solid Waste Management is a big concern all over the world. Solid waste is generally exist in two forms, one is garbage (Bio-degradable waste) and second one is rubbish (Non-bio degradable waste). Garbage is basically a mixture of kitchen waste coming from house hold and rubbish is the mixture of all type of waste like paper, glass, plastic etc. Solid waste generated in daily life and overcome only by three ways Reduce, Reuse and Recycle (3'R Principals).This paper presents a shortest path (SP) model for the solid waste management (SWM) for Kanpur city. This model is also applicable for any type of route network having range among positive integers. There is a code written in c programming language for this model to find out shortest route. This model will help in finding the shortest path among the node, also find the visiting route means shortest path is calculated by visiting which nodes in the network. This model is helpful for finding the shortest path and shortest route for one-way, two-way and also for there is no direct link among the nodes.**

Keywords*— Solid Waste Management, Shortest Path, Disjktra's Algorithm, Shortest Route, Time Management, Cost Minimization,Garbage,Rubbish,3'R Principle, Bio-degradable, Non-Bio-degradable*

1. INTRODUCTION

Solid waste is a sensitive issue which concerns about serious environmental problems in today's world. The present situation of direct dumping of the waste without proper inspection and separation leaves a serious impact of environmental pollution causing a tremendous growth in health related problems [4]. "Domestic, industrial and other wastes, whether they are of low or medium level wastes, they are causing environmental pollution and have become perennial problems for mankind." If this situation is not handled in a proper manner within time then it would lead to worse consequences on a global level. There has been awareness regarding waste management amongst many countries [5].

The 3Rs principle — reduce, reuse, recycle — has been promoted worldwide to hold increasing municipal solid waste (MSW) problems [6]. Based on the 3Rs principle, many programs are implemented with the cooperation of the governmental and private sectors from social, technological, economic, public health and political perspectives. In addition to achieve the goal of waste reduction, the establishment of a cost-effective waste management system with an adequate scale of operation, providing sufficient capacity of MSW management services, is also an important target for developing a sustainable material-recycling society [11].

Solid Waste is generally classified into two parts

(a) Garbage (bio-degradable organic waste)

(b) Rubbish (non- bio-degradable waste)

The state of solid waste management in Kanpur was no different from most other Indian cities. Kanpur Nagar Nigam (KNN) had the responsibility for collecting, transporting and disposing of the solid waste generated in the city, estimated at about 1500 tons per day. There were numerous collection centers in the city, more than 400 of which were open dumps [1]. A fleet of 132 vehicles and 3000 safai karmacharis were supposed to collect and transport the city garbage and dump it at an "Authorized" site a few kilometers away from the city. This they did at an annual cost of Rs 42 crore, which has now come down to about half [17].

India, the world's second highest populated country after China with population of 1.21 billion (census 2011) already containing 17.5% of the world's population, is a land of physical, climatic, geographic, ecological, social, cultural and linguistic diversity [18]. The annual rate of growth of urban population in India is 3.35% (Census of India, 2011). The proportion of population living in urban areas has increased from 17.35% in 1951 to 31.2% in 2011(Census, 2011). No doubt, India has achieved multifaceted socio-economic progress during last 64 years of its independence. However, in spite of heavy expenditure by Civic bodies, Management of Municipal Solid Wastes (MSW) continues to remain one of the most neglected areas of urban development in India [15].

2 OBJECTIVE

Specifically, the following objectives have been identified:

a) To minimize the cost of transportation of solid waste
b) To reduce the time taken by municipal vehicle from collection point to disposal point.

3.METHODOLOGY

Dijkstra's algorithm, conceived by computer scientist Edsger Dijkstra in 1956 and published in 1959, is a graph search algorithm that solves the single-source shortest path problem for a graph with non-negative edge path costs, producing a shortest path tree. This algorithm is often used in routing and as a subroutine in other graph algorithms. For a given source vertex (node) in the graph, the algorithm finds the path with lowest cost (i.e. the shortest path) between that vertex and every other vertex [3].

The worst-case running time for the Dijkstra's algorithm on a graph with n nodes and m edges is $O(n^2)$ because it allows for directed cycles. It even finds the shortest paths from a source node s to all other nodes in the graph. This is basically $O(n^2)$ for node selection and $O(m)$ for distance updates. While $O(n^2)$ is the best possible complexity for dense graphs, the complexity can be improved significantly for sparse graphs [3].

4. C CODE FOR THIS MODEL

```
/* Program of shortest path between two node in graph using Djikstra algorithm */
#include<stdio.h>
#include<conio.h>
#define MAX 10
#define TEMP 0
#define PERM 1
#define infinity 9999
struct node
{
        int predecessor;
        int dist; /*minimum distance of node from source*/
        int status;
};
int adj[MAX][MAX];
int n;
void main()
{
        int i,j;
        int source,dest;
        int path[MAX];
        int shortdist,count;
        clrscr();
        create_graph();
        printf("The adjacency matrix is :\n");
        display();
        while(1)
        {
                printf("Enter source node(0 to quit) : ");
                scanf("%d",&source);
                printf("Enter destination node(0 to quit) : ");
                scanf("%d",&dest);
                if(source==0 || dest==0)
                exit(1);
                count =
        findpath(source,dest,path,&shortdist);
                if(shortdist!=0)
                {
                        printf("Shortest distance is : %d\n",
shortdist);
                        printf("Shortest Path is : ");
                        for(i=count;i>1;i--)
                                printf("%d->",path[i]);
                        printf("%d",path[i]);
                        printf("\n");
                }
                else
                        printf("There is no path from
source to destination node\n");
        }/*End of while*/
}/*End of main()*/
create_graph()
{
        int i,max_edges,origin,destin,wt;
        printf("Enter number of vertices : ");
        scanf("%d",&n);
        max_edges=n*(n);
        for(i=1;i<=max_edges;i++)
        {
                printf("Enter edge %d(0 0 to quit) : ",i);
                scanf("%d %d",&origin,&destin);
                if((origin==0) && (destin==0))
                break;
                printf("Enter weight for this edge : ");
                scanf("%d",&wt);
```

```
		if( origin > n || destin > n || origin<=0 || destin<=0)
		{
			printf("Invalid edge!\n");
			i--;
		}
		else
			adj[origin][destin]=wt;
	}/*End of for*/
}/*End of create_graph()*/
display()
{
	int i,j;
	for(i=1;i<=n;i++)
	{
		for(j=1;j<=n;j++)
		printf("%3d",adj[i][j]);
		printf("\n");
	}
}/*End of display()*/
int findpath(int s,int d,int path[MAX],int *sdist)
{
	struct node state[MAX];
	int i,min,count=0,current,newdist,u,v;
	*sdist=0;
	/* Make all nodes temporary */
	for(i=1;i<=n;i++)
	{
		state[i].predecessor=0;
		state[i].dist = infinity;
		state[i].status = TEMP;
	}

	/*Source node should be permanent*/
	state[s].predecessor=0;
	state[s].dist = 0;
	state[s].status = PERM;
	/*Starting from source node until destination is found*/
	current=s;
	while(current!=d)
	{
		for(i=1;i<=n;i++)
		{
			/*Checks for adjacent temporary nodes */
			if ( adj[current][i] > 0 && state[i].status == TEMP )
			{
				newdist=state[current].dist + adj[current][i];
				/*Checks for Relabeling*/
				if( newdist < state[i].dist )
				{
	state[i].predecessor = current;
					state[i].dist = newdist;
				}
			}
		}/*End of for*/
		/*Search for temporary node with minimum distand make it current node*/
		min=infinity;
		current=0;
		for(i=1;i<=n;i++)
		{
			if(state[i].status == TEMP && state[i].dist < min)
			{
				min = state[i].dist;
				current=i;
			}
		}/*End of for*/
		if(current==0) /*If Source or Sink node is isolated*/
		return 0;
		state[current].status=PERM;
	}/*End of while*/
	/* Getting full path in array from destination to source */
	while( current!=0 )
	{
		count++;
		path[count]=current;
		current=state[current].predecessor;
	}
	/*Getting distance from source to destination*/
	for(i=count;i>1;i--)
	{
		u=path[i];
		v=path[i-1];
		*sdist+= adj[u][v];
	}
	return (count);
}/*End of findpath()*/
```

```
OUTPUT:
Enter no of vertices 6
Edge weight 1 to 1 (0 if no edge): 0
Edge weight 1 to 2 (0 if no edge): 29
Edge weight 1 to 3 (0 if no edge): 7
Edge weight 1 to 4 (0 if no edge): 9
Edge weight 1 to 5 (0 if no edge): 12
Edge weight 1 to 6 (0 if no edge): 10
Edge weight 1 to 7 (0 if no edge): 6
Edge weight 1 to 8 (0 if no edge): 15
Edge weight 2 to 1 (0 if no edge): 29
Edge weight 2 to 2 (0 if no edge): 0
Edge weight 2 to 3 (0 if no edge): 6
Edge weight 2 to 4 (0 if no edge): 14
Edge weight 2 to 5 (0 if no edge): 5
Edge weight 2 to 6 (0 if no edge): 5
Edge weight 2 to 7 (0 if no edge): 11
Edge weight 2 to 8 (0 if no edge): 13
Edge weight 3 to 1 (0 if no edge): 7
Edge weight 3 to 2 (0 if no edge): 6
Edge weight 3 to 3 (0 if no edge): 0
Edge weight 3 to 4 (0 if no edge): 11
Edge weight 3 to 5 (0 if no edge): 7
Edge weight 3 to 6 (0 if no edge): 7
Edge weight 3 to 7 (0 if no edge): 9
Edge weight 3 to 8 (0 if no edge): 9
Edge weight 4 to 1 (0 if no edge): 9
Edge weight 4 to 2 (0 if no edge): 14
Edge weight 4 to 3 (0 if no edge): 11
Edge weight 4 to 4 (0 if no edge): 0
Edge weight 4 to 5 (0 if no edge): 11
Edge weight 4 to 6 (0 if no edge): 11
Edge weight 4 to 7 (0 if no edge): 4
Edge weight 4 to 8 (0 if no edge): 13
Edge weight 5 to 1 (0 if no edge): 12
Edge weight 5 to 2 (0 if no edge): 5
Edge weight 5 to 3 (0 if no edge): 7
Edge weight 5 to 4 (0 if no edge): 11
Edge weight 5 to 5 (0 if no edge): 0
Edge weight 5 to 6 (0 if no edge): 3
Edge weight 5 to 7 (0 if no edge): 9
Edge weight 5 to 8 (0 if no edge): 8
Edge weight 6 to 1 (0 if no edge): 10
Edge weight 6 to 2 (0 if no edge): 5
Edge weight 6 to 3 (0 if no edge): 3
Edge weight 6 to 4 (0 if no edge): 9
Edge weight 6 to 5 (0 if no edge): 3
Edge weight 6 to 6 (0 if no edge): 0
Edge weight 6 to 7 (0 if no edge): 12
Edge weight 6 to 8 (0 if no edge):15
Edge weight 7 to 1 (0 if no edge): 6
Edge weight 7 to 2 (0 if no edge): 11
Edge weight 7 to 3 (0 if no edge): 9
Edge weight 7 to 4 (0 if no edge): 4
Edge weight 7 to 5 (0 if no edge): 9
Edge weight 7 to 6 (0 if no edge): 12
Edge weight 7 to 7 (0 if no edge): 0
Edge weight 7 to 8 (0 if no edge): 9
Edge weight 8 to 1 (0 if no edge): 15
Edge weight 8 to 2 (0 if no edge): 13
Edge weight 8 to 3 (0 if no edge): 9
Edge weight 8 to 4 (0 if no edge): 13
Edge weight 8 to 5 (0 if no edge): 8
Edge weight 8 to 6 (0 if no edge): 15
Edge weight 8 to 7 (0 if no edge): 9
Edge weight 8 to 8 (0 if no edge): 0
The adjacency matrix is:
0        29       7        9
         12       10       6
         15
29       0        6        14
         5        5        11
         13
7        6        0        11
         7        7        9
         9
9        14       11       0
         11       11       4
         13
12       5        7        11
         0        3        9
         8
10       5        3        9
         3        0        12
         15
6        11       9        4
         9        12       0
         9
15       13       9        13
         8        15       9
         0

Shortest distance is:
12       13       7        9
         12       10       6
         15
13       10       6        14
         5        5        11
         13
7        6        10       11
         7        7        9
         9
9        14       11       8
         11       11       4
         13
12       5        6        11
         6        3        9
         8
10       5        3        9
         3        6        12
         11
6        11       9        4
         9        12       8
         9
15       13       9        13
         8        11       9
         16
```

5. CONCLUSION

The shortest path model is tested on 8 major collection and disposal point of Kanpur city which provides a shortest path among the collection and disposal points. This model also helps in finding the intermediate nodes form which shortest path is obtained. User can simply figure out the route from where shortest path can be obtained.

This model is valid for one way, two way and also for there is no way between the nodes. The complexity of the problem is more when there is one way from one node to another node. This model can be used for effective management of solid waste collection and disposal, and help in reducing time taken to collect and dispose solid waste among various nodes.

REFERENCES

[1] Alphonce Kyessi (2009) "*Tanzania GIS Application in Coordinating Solid Waste Collection: The Case of Sinza Neighbourhood in Kinondoni Municipality, Dar es Salaam City, Tanzania*" Surveyors Key Role in Accelerated Development,pp 1-19

[2] Christos chalkias (2000) "*Optimizing municipal solid waste collection using GIS*"

[3] Dijkstra, E.W. (1959) *"A note on two problems in connexion with graphs*". Numerische Mathematik, 1, 269-271.

[4] Dorigo et al. (1999) "*Ant Algorithms for Discrete Optimization*". Artificial Life, 5(2), 137–172.

[5] SRI, GIS and Mapping Software Support Group (2006) "*ArcGIS Network Analyst: Routing, Closest Facility, and Service Area Analysis*". ttp://www.esri.com/networkanalyst (Accessed on February 10, 2007).

[6] GEOLORE (2003) *Development of a Geographical Information System for the organization and implementation of an integrated waste management at a local and/or regional level*. http://aix.meng.auth.gr/geolore (Accessed on February 10, 2007).

[7] Holland et al. (1975) "*Adaptation in Natural and Artificial Systems*", University of Michigan Press.

[8] Karagiannidis et al. (2006) "*Optimization of urban solid waste collection through GIS use: A part implementation for the Municipalities of Panorama and Sikies*". 21st European Conference for ESRI Users.

[9] Kirkpatrick (1983) "*Optimization by Simulated Annealing, Science*". 220(4598), 671-680.

[10] Katzin (2008)"*Route optimization application in waste collection system*" Vilniaus Gedimino technikos universities, pp 326-334

[11] Karagiannidis et al. (2006) "*Using GIS to assess the local problem of Waste Management in an urban area".*

[12] Lakshumi et al. (2006) "*Optimal Route Analysis for Solid Waste Disposal Using Geographical Information System*".

[13] Ludwig et al. (1968) "*Report on the solid waste problem*". Journal of Sanitary Engineering Div. 94 (2), 355-370.

[14] M.K.Ghose,(2005)" *A GIS based transportation model for solid waste disposal- A case study on Asansol municipality*",science direct, pp 1287-1293.

[15] Modak et al. (1996) "*Optimal regional scheduling of solid waste system. II: Model solution*", ASCE Journal of Environmental Engineering, 122 (9), 793-799.

[16] Mukti Advani (2005) "*Improvement in Transit Service using GIS – Case study of Bhavnagar StateTransport Depot*", Proceedings ESRI National Conference noida, India.

[17] N.P. Thanh (2009) "*GIS application for estimating the current status and improvement on municipal solid waste collection and transport system: Case study at Can Tho city, Vietnam*", Asian Journal on Energy and Environment,pp 108-121.

[18] Nikolaos Kardimas (2007) " *Municipal solid collection of large items optimized with Arc GIS network analyst* "Proceedings 21st European Conference on Modelling and Simulation Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni,pp 1-6

[19] Rahman (2008) "*Suitable sites for urban solid waste disposal using GIS approach in khulana city, Bamgladesh*", Proc. Pakistan Acad. Sci. 45 (1), pp 11-22.