THE ANNALS OF “DUNĂREA DE JOS” UNIVERSITY OF GALAȚI
FASCICLE V, TECHNOLOGIES IN MACHINE BUILDING, ISSN 1221-4566
2010

The Use of Dijkstra's Algorithm in Waste Management Problem

Vlad Gogoncea,Gabriel Murariu, Lucian Georgescu
Faculty of Sciences, Physics, Chemistry and Environment Department, „Dunărea de Jos” University of Galati, Romania

ABSTRACT

Waste disposal facilities, commonly called landfill sites, were originally located on the outskirts of centers of population, due to the lack of large scale transport to carry the waste away. The development of any landfill is expensive and the only return being on how much waste can be stored in the landfill void. The most economical landfill requires that the void is filled to its maximum capacity. Starting from the idea that a GIS can be used to keep a record of filling and transportation parameters, in this paper we present a particular approaching algorithm based on a Dijkstra minimum cost method. In the final a specific application for the city case of Galați is dealt with.

KEYWORDS: waste disposal, Dijkstra minimum cost method

1. Introduction

The disposal of garbage in the world is a problem that continues to grow with the development of the industrialized nations and the growth of population. Since the beginning of time people have needed to find a way of disposing of their trash. In 18th century, England and France, carters were paid by individuals to carry trash and discard it on the outskirts of towns. Disposal in open pits became routine and Benjamin Franklin initiated the first municipal cleaning program in Philadelphia in 1757. Since then, we have come a long way and have developed types of waste that cannot simply be dumped into a hole.

There are many different methods of disposing of waste. Landfill is the most common and probably accounts for handling of more than 90 percent of the nation's municipal garbage even though Landfills have been proven contaminants of drinking water in certain areas. It is the most cost effective method of disposal, with the collection and transportation accounting for 75 percent of the total cost. In a modern landfill, garbage is spread thin, compacted layers covered by a layer of clean earth. Pollution of surface water and groundwater is minimized by lining and contouring the fill, compacting and planting the uppermost cover layer, diverting drainage, and selecting proper soil in sites not subject to flooding or high groundwater levels.

Waste disposal in Romania has changed dramatically over the last two decades, both with respect to its legislation and its public perception. For a long time, this was considered as 'out of sight, out of mind' and to be somebody else's problem.

In our days, many modern services are called to give a supportive help to solve society’s problems. Because of this, for the waste disposal problem, in many countries the GIS applications could be a solution. In the simplest terms, GIS is the merging of cartography and database technology. GIS systems are used in cartography, remote sensing, land surveying, utility management, photogrammetry, geography, urban planning, emergency management, navigation, and localized search engines.

Starting from the idea that a GIS can be used to keep a record of filling and transportation parameters, in this paper, we present an approaching algorithm based on a Dijkstra algorithm approaching method.

2. Description of the Used Approaching

In the present paper, we present our preliminary results obtained for the waste disposal problem for two different areas of Galati - the first one is included in the older area of the city – in the central part of the town, meanwhile the second one is included in the newer zone – in the I. C. Frimu residence part. The applied method is based on the Dijkstra minim cost algorithm approach [1–3], relying on the existing waste disposal facilities and the city hall company capacities. This algorithm is often used in routing [2].

Generally speaking, for a given source vertex (node) in the graph, the above algorithm finds the path with lowest cost (i.e., the shortest path) between that vertex and every other vertex. It can also be used for finding costs of shortest paths from a single vertex to a single destination vertex by stopping the algorithm once the shortest path to the destination vertex has been determined [4 -5].

II. 1. **The algorithm description**

Dijkstra’s algorithm assigns for a given oriented graph structure some initial cost values and will try to improve them step by step. The main threads are the following

1. Assign every node a charge value. Set it to zero for our initial node and to infinity for all other nodes.

2. Mark all nodes as unvisited. Set initial node as current.

3. For current node, consider all its unvisited neighbors and calculate their tentative expenses. If this weight is less than the previously recorded cost (infinity in the beginning, zero for the initial node), overwrite the value.

4. When we are done considering all neighbors of the current node, mark it as visited.

A visited node will not be checked ever again; its cost recorded now is final and minimal.

5. If all nodes have been visited, it up. Otherwise, set the unvisited node with the smallest cost (from the previous node) as the next "current node" and have to continue from step 3 further.

The order is conceptually simple: at each iteration, the algorithm creates a set of intersections consisting of every unmarked node that is directly connected to a marked crossroads; this will be your set of considered intersections. From that set of considered intersections, find the closest junction to the destination (making this a greedy algorithm) and highlight it and mark the street to that intersection, draw an arrow with the direction, then repeat it. In each stage, mark just one new intersection. When you get to the destination, follow the obtained path backwards.

In order to succeed in reaching such an aim, it a consistent approach stratagem has to be built. Our approach is based on the adjacency matrix for an oriented graph problem.

The adjacency matrix of a simple graph is a matrix with rows and columns labeled by graph vertices, with an 1 or 0 in position (i, j) according to whether node i and j is adjacent or not.

II. 2. **The software approaching**

The software approach starts with the data base designing. It is easy to perceive that a complete data base structure is required, and that successive improvements on the information’s structure lead to different results. For a consistency of the study, our revise was focused on two different areas of our city.

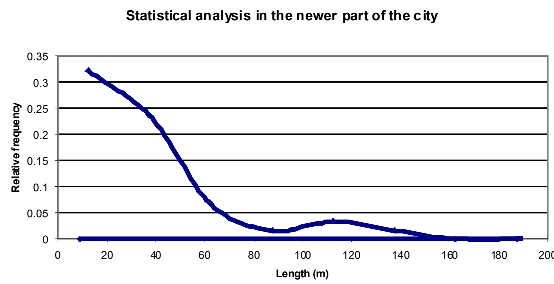

Fig 1. *The statistical analysis of the graph arc length in the older part of the city*

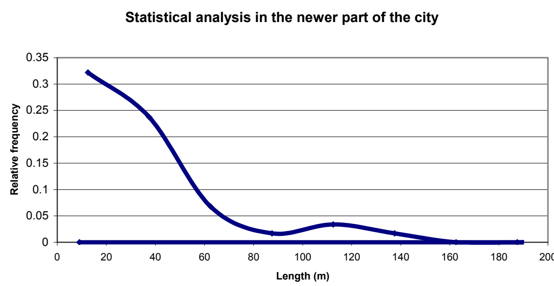

Fig 2. *The statistical analysis of the graph arc length in the newer part of the city.*

The first chosen one includes aging streets, with small widths and small distances between two cross street points (Fig.1). In this area, the relative frequencies of short lanes are almost of 50 percent magnitude (Fig. 4).

On the other hand, the second chosen area includes a newer part of our town (Fig. 2). In this quarter, the relative frequency is different, the street being longer and of significant widths (Fig.5).

Our used data base included, in preliminary stages, street elements as thickness. After a series of model runs, the future data base will include additional records with a series of supplementary information as blacktop condition and even the existence of parked cars among (Fig. 3). Our software build tool allow to obtain the adjacency matrix.

Start index	Street Name	Stop index	Width	Sense	Quality state (5/5)	Parked cars	Pointer reference 1	Pointer reference 2	Distance
1	Strada Egalitatii	2	5	0	4	-1	1	2	88.10221337
2	Strada Egalitatii	3	5	0	4	-1	2	3	53.15072906
3	Strada Cuza	4	4	-1	3	0	3	4	56.22277119
4	Strada Cuza	5	4	-1	3	0	4	5	29.41088234
5	Strada Cuza	6	4	-1	3	1	5	6	113.0176977
5	Strada Universitatii	7	5	0	4	-1	5	7	23.76972865
7	Strada Garii	8	5	0	4	0	7	8	116.7261753
7	Strada Garii	9	6	0	3	0	7	9	135.0037037
9	Strada Mihai Bravu	10	5	-1	3	0	9	10	59.05929224
10	Strada Basarabiei	11	7	0	4	1	10	11	34.20526275
11	Strada Basarabiei	12	7	0	4	1	11	12	45.96737974
12	Strada Basarabiei	13	7	0	4	1	12	13	28.28427125
13	Strada Basarabiei	14	7	0	4	1	13	14	11.3137085
14	Strada Basarabiei	15	7	0	4	1	14	15	35.35533906
15	Strada Basarabiei	16	7	0	4	0	15	16	39.05124838
16	Strada Basarabiei	17	7	0	4	0	16	17	33.9705755
17	Strada Basarabiei	18	7	0	4	0	17	18	40.45985665
18	Strada Basarabiei	19	7	0	4	0	18	19	41.03656906
19	Strada Basarabiei	20	7	0	4	-1	19	20	33.24154028
20	Strada Basarabiei	21	7	0	4	0	20	21	21.40093456

Fig 3. *The used data base structure for preliminary algorithm implementation.*

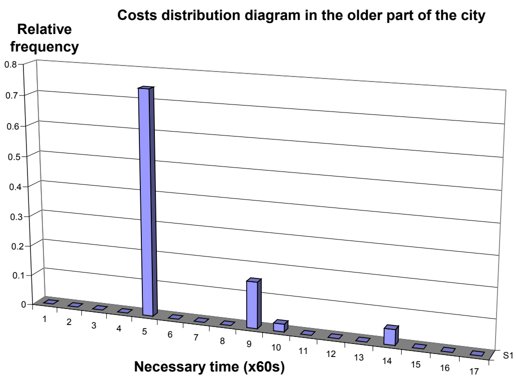

Fig 4. *Costs distribution diagram in the older part of the city*

3. Results

The considered cost function is based on an empirical model, by the following formula.

$$W_{i \to f}(k) = \sum_{l} \left(\frac{d_l}{v_l(k)} + t_l \right) \qquad (1)$$

where we consider $W_{i \to f}(k)$ is the weight function associated to path from vertex $i \to j$, d_l is the l - the arc length from this path, $v(k)$ is the velocity for moving in the k - th time interval of the day, meanwhile the t_i is the corresponding time for technological operations in the i - th path point.

In figures 4 and 5, our assessments are presented for cost functions for the two considered areas, for the same time interval of the day. It is easy to observe that the two distributions are different, fact that could be explained by the dissimilar structure of the areas ‘streets. Our future study will include a systematic analysis, which will include the influence of the traffic data.

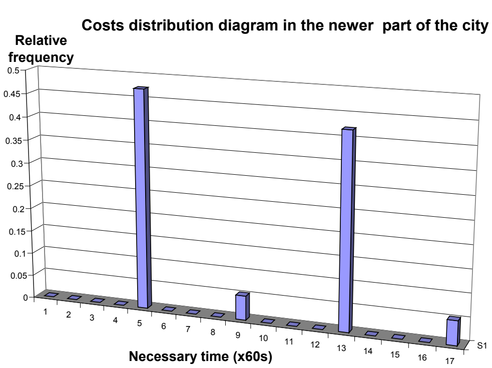

Fig 5. *Costs distribution diagram in the newer part of the city*.

4. Simulations Results and Discussions

GIS data represent real objects (such as roads, land use, elevation, trees, waterways, etc.) with digital data determining the mix. Real objects can be divided into two abstractions: discrete objects (e.g., a house) and continuous fields (such as rainfall amount, or elevations). Traditionally, there are two broad methods used to store data in a GIS for both kinds of abstractions mapping references: raster images and vector. Points, lines, and polygons are the stuff of mapped location attribute references. A new hybrid method of storing data is that of identifying point clouds, which combine three-dimensional points with RGB information at each point, returning a "3D color image" GIS. Our preliminary approaching method is based on the first strategy in a specific way.

5. Conclusions

We have described a new computer approach method for optimizing the waste disposal practice in the Galati city area. The considered procedure in this paper was based on the Dijkstra algorithm application and, using such approach we made some preliminary advances in reaching the proposed aim [5].

The major disadvantage of the algorithm is the fact that it does a blind search there by consuming a lot of time and wasting of necessary resources.

Another disadvantage is that it cannot handle negative edges. This leads to acyclic graphs and most often cannot obtain the right shortest path.

In practice, this problem is usually solved by heuristic variations of Dijkstra's algorithm, which do not guarantee an optimal result. We report results from a pilot study, in which we focused on the travel time as the only optimization criterion.

In conclusion, we have a method that produces good results that has to be reasonably optimized. Tests still remain to be done for larger graph structures.

Acknowledgment

The work of Vlad Gogoncea was supported by Project SOP HRD - SIMBAD 6853, 1.5/S/15 - 01.10.2008 "

References:

1 **Dijkstra, E. W.,** *A Note on Two Problems in Connection with Graphs*, Numerische Math. 1, pp. 269-271, 1959.

2. **Kiena, S.**, *Dijkstra's Algorithm*, §6.1.1 in Implementing Discrete Mathematics: Combinatorics and Graph Theory with Mathematica. Reading, MA: Addison-Wesley, pp. 225-227, 1990..

3 . **Whiting, P. D., Hillier, J. A.,** *A Method for Finding the Shortest Route through a Road Network*, Operational Res. Quart. 11, pp. 37-40, 1960

4. **Puşcasu, Gh., Palade, V., Stancu, Al.** *Classical and Smart process leading systems*