International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Cost Effective Guided Routing (CEGR) Technique – A bi-directional searching method towards the solution of land acquisition problem

Anirban Chakraborty

Department of Computer Science, Barrackpore Rastraguru Surendranath College, Barrackpore, Kolkata -120, W.B., India& College

---***---

Abstract - In this paper a two-way searching methodology has been proposed, where simultaneously two searching is being carried out, one from Start node to Destination node and other in reverse direction. The proposed methodology is optimal, complete and runs with a complexity of $O(n^{m/2})$. The proposed algorithm has successfully been implemented for solution of land acquisition problem, a burning issue of the country now-a-days, where while selection of path in addition to distance, many other influencing parameters been judged. This methodology takes as input a digitized land bank. To evaluate the efficiency, the methodology been compared with many popular graph traversal algorithms.

Key Words: Two-way searching, influencing parameters, land acquisition problem, digitized map, optimal, complete

1. INTRODUCTION

In this work a technique for optimal route tracing has been proposed. The digitized map of a land bank has to be fed as an input. The methodology considers the regions of the input digitized map as the set of nodes of a graph G=(V,E) and the paths existing between those regions are treated as the set of edges E. The main objective of the proposed technique is to find a route with the shortest weightage between two chosen nodes—Source and Destination. While fulfilling this main objective, the motivation behind the work is to create an efficient method of goal node search which is both complete and optimal unlike the popular Goal Node Searching Algorithms like Depth first search [1] algorithm, Generate and test algorithm [2] , Hill Climbing problem [3], Steepest Ascent Hill Climbing [4] , Best First Search algorithm [5]. Moreover the proposed approach should not need any apriori mention of the depth limit like the Depth Limited Search [5] algorithm and this will not lead to an unsuccessful search when the depth limit is chosen too shallow. Moreover a two-way search has initiated to make the process faster.

Most of the Path Searching Techniques are unguided and thus often goes far away from goal. To overcome this problem, in this proposed mechanism, help of some heuristic value could be taken, which will indicate whether the search path is being deviated or not. To prevent this deviation, direct Euclidean Distance from a node to the destination, could be taken as the heuristic value of that node, because the direct/Euclidean Distance always gives the minimum value between two points.

This proposed searching strategy not only considers the distance between two nodes, but it actually considers some weightage factors. For example, when it is targeted to construct a road-ways then considering only the distance is not sufficient, but also the land types through which the road be constructed, should be taken into consideration, as it plays an important role in such construction. Fertility of the soil, hardness of the soil etc. plays crucial role for choosing a land from architectural point of view.

In India, although the main usage of land is for cultivation, but in the recent past few decades as population increases by leaps and bounds, so to meet the growing needs of this growing population, the means of transportation like — buses, cars, rails, etc. are also increasing very rapidly. To cater the need of the people for transportation, new construction of roads or rail routes are needed and obviously it is required to acquire land of enough large amount, for construction of such roads, highways, railroad etc.; as these constructions need a large area. This method is termed as “Land Acquisition”. However during land acquisition, the authority faces a large agitation from the farmers. In order to minimize such agitation and anxiety, only less fertile or barren lands could be chosen for acquisition. Although the presented idea seems to be quiet a fascinating solution to the problem, but when a huge geographical area is under consideration, it is quiet impossible to manually consider all the possible routes from source to destination and cover the fertile lands as less as possible for acquisition. Moreover, there should be a balance between the distance traversed and type of land acquired. In other words, moving only through less fertile lands, if it is required to traverse a lot, then again the cost of contraction will become very high. For the acquisition of lands for new infrastructural development purpose, in addition to judge the fertility of lands (as acquisition of less fertile/ barren lands causes little people agitation), it is also required to take into account of some other important characteristics and parameters of lands that serves as prerequisites for construction of roadways/ railway tracks on a land. The aforesaid problem could be realized by feeding the values of necessary parameters, influencing the selection of lands for acquisition.

Applying the proposed goal searching methodology, the least cost path between the two designated end-points (i.e. points between which the new track has to be constructed) is found out, with proper consideration of all the influencing factors, by adorning suitable weightage to them.

The organization of the paper is as follows: Section 2 focuses on preliminary concepts, methodology of the proposed mechanism been mentioned in section 3, followed by an illustration in section 4. Results been discussed in section 5, comparison of results in section 6 and finally concluding remarks are there in section 7.

2. Preliminaries

In GIS anchored graph traversal, it is important to decide the cost of the path joining two points. Although distance plays a crucial role, but knowing only the distance is not sufficient for determining cost from architectural point of view. During construction of new rail-roads/ highways between two points, land acquisition is needed. There exists a number of influencing factors which plays crucial role for determining which lands are beneficial to acquire in terms of cost and architectural point of view. Among the many existing factors, only three have been considered. These are — Fertility of the land, Water content of the soil and Distance; as discussed below.

• Fertility of the land
Fertility of the lands plays a very crucial role for choosing a land for acquisition. It is a general appeal that, if only the barren lands are acquired, then it is very safe from the view point of people/ farmer agitation. On the other hand, acquisition of very fertile lands not only causes financial loss of the farmers and very bad impact on society for a agriculture dependent country like India, but also the compensation amount needed is very large, which in turn increases the overall cost of the construction.

• Water content of the soil
The softness of the land is measured based on the water content of the land. A land on the basis of water content can be marshy, swamp or hard. The hard lands are suitable most from architectural point of view.

• Distance
The distance between the source and destination is always an important point to consider. Obviously, one should always like to traverse least distance. But while doing so, balance should be made between acquiring less fertile lands and harder lands; otherwise construction cost will be larger and social problem will take birth.

3. METHODOLOGY

The Route Tracing Scheme proposed here is capable of producing an optimal solution. This optimality is based on considering not only total distance traversed from Source to Destination, but also includes some influencing factor. These influencing factors varies with concerned application area. For example while choosing an Optimal Route during construction of new railway tracks, type of land is a major influencing factors. To discuss the present methodology, 75% weightage is given to the influencing factor and rest 25% to the distance. However, these values are for demonstration purpose only, for real world applications, these values will be judged from the application areas. In this Cost Effective Guided Routing Technique, always discrimination is made between the regions already been visited/explored and those which are remaining. The Cost of reaching a node is decided in a cumulative way. This means, if Node B is the Child of Node A, then the Cost of reaching Node B is equal to, Cost of reaching A from Source Node plus Cost of reaching A to B. In other words, all the intermediate paths, between source node, to the presently considered node have to be considered. This GIS enabled technique takes a raster map as input, which is digitized and is composed of many small regions. Each region is taken as the node of a graph. The properties of each of these regions are considered as their associated attribute values. For determination of the path between two nodes, these associated attribute values play an important role. However, for the sake of simplicity, only one such parameter (attribute value) is considered here. Two regions are said to be adjacent, if there exists any direct path between these two. An adjacency matrix keeps track of this record of adjacency. Moreover, here exists no loop-edge, means any edge which connects a node to itself. The main motivation behind the technique is to select only the regions with low associated attribute value (decided by the value of the parameter) for construction of the path, while keeping in mind not to make the length of the path very much. To make a balance between these two, while calculating the value of cost factor of selecting any particular region, 75% weightage is given to the associated attribute value and rest 25% is given to the distance traversed. These factors are for illustrations only. Depending upon the situation, these factors vary and if need arises more number of parameters could easily be accommodated.

For keeping track of the regions been already visited and regions those are still remaining, two arrays, named OpenSet and CloseSet are used. A step-wise illustration of the procedure is as follows:

Step 1: Every regions (vertices) are given unique (auto-generated) Identification Number (ID). The user points out the Source Region (starting node for traversal), as well as Destination Region (end of traversal procedure).

Step 2: Initially the OpenSet only contains the region ID of the Source Region, as being fed by user. Each time, for any region, two values are being considered, one is, say d value, signifying the distance traversed till reaching this intermediate region while traversing from the Source (i.e. how much been traversed) and the next one, say e, is the Euclidean distance from this intermediate point to the destination one (i.e. how much remaining to be traversed). The required functional value f is the 25% of summation of these two (i.e. $f(x)=d(x)+e(x)$) plus 75% obtained from associated attribute value of the region.

Thus initially for the Source Region, its d value is set to 0 and its e value is as obtained from graph (Euclidean distance from the Source to Destination). CloseSet is made empty initially.

Step 3: Until a Destination node is found, the following procedure is repeated:

If there are no region ID on OpenSet, failure is reported. Otherwise, the ID on OpenSet with the lowest calculated f value is picked, designated as BESTNODE. It is removed from OpenSet and is placed on ClosedSet. If BESTNODE is a goal node, then the procedure halts and a solution is reported. Otherwise, all the regions which are adjacent to the BESTNODE (may be designated as SUCCESSOR), are placed at OpenSet. For each SUCCESSOR, the following is done:

I. The path from SUCCESSOR pointing back to BESTNODE is memorised. This backwards link will make it possible to recover the path once a solution is found.

II. If a SUCCESSOR is just same as the region which is already present in the OpenSet (which signifies that the node has previously been generated, but was not processed), then that SUCCESSOR is labeled as OLD. From the given graph it is simple to add OLD into the existing list of BESTNODE's successors. Now its the time to decide that if it is needed to update the parent link of node OLD for pointing out the BESTNODE. It is required only when the path just evaluated to point SUCCESSOR is less costlier than the existing current best path pointing to OLD (because SUCCESSOR and OLD are actually the same node). Thus its the time to examine, which path is more cheaper, to reach to OLD via its current parent or to SUCCESSOR via BESTNODE and is being done by comparing their f values. If f(OLD) is less costlier (or having same cost), then no action is taken. However, if f(SUCCESOR) is cheaper, then OLD's parent link is reset to point to BESTNODE and f (SUCCESSOR) is recorded.

III. If any region in the ClosedSet, has same value as the SUCCESSOR (i.e., it has already been generated and processed), then at this step the SUCCESSOR is skipped and the node with next least f value is chosen and the previous steps are continued.

IV. If the SUCCESSOR was not already on either OpenSet or ClosedSet, then it is put on OpenSet, f(SUCCESSOR) is computed using the procedure mentioned in Step I and Step II is repeated.

4. ILLUSTRATION:

The technique discussed over here, has been illustrated with the help a weighted graph represented in figure 1.

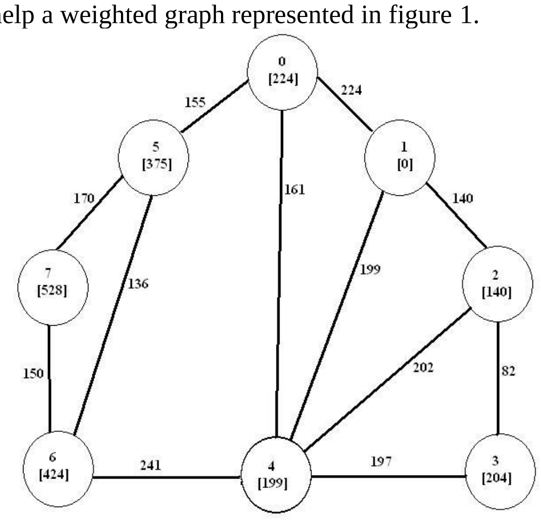


Fig. 1: Sample graph for illustrating CEGR Method

In figure 1, all the digitized regions are denoted by the nodes of the graph, en-marked as circle. The weight of an edge is just the Euclidean Distance between two end nodes of that edge. Considering a general map, each region after digitization serves as the nodes of the graph and therefore the weight of the edge connecting them, is just the centroid-to-centroid distance of them. Every node (circle) has two values, depicted inside the circle. The one-digit number is the unique-identification number of the node. As here 8 nodes are considered, so they have been numbered 0 to 7. During digitization of the regions, these ID numbers are generated automatically. Immediately after digitization, a Distance Matrix is created, holding the Euclidean distance between every nodes. If there are 'n' nodes, then this Distance Matrix is of order 'n × n'. Each Cell (i, j) denotes the Euclidean distance between Node i and j. As the graph contains no loop, so Cell (i, i) = 0.

Let 5 is the Source Node and 1 is the Destination Node. Practically, these two nodes are being fed by user through mouse click onto a user friendly Graphical User Interface. Each circle holds a 3 digit number (mentioned inside bracket), which denotes the Euclidean Distance from that node to the Destination node. The value is readily obtained from the previously generated Distance Matrix.

As stated in the methodology, apart from distance, while evaluating path cost, the associated attribute value of the region has also to consider. Let this parameter is denoted by 'Cost of Region'. Less is the cost, better is the choice. Table 1 reflects the priority value of each regions. A region which should be avoided during determination of path (for example a land having some heritage construction and thus could not be acquired for new constructions) is given a very high cost parameter value, say 100.

Table1: Values of the Cost Parameters of the nodes (corresponding to figure 1)

Cost Parameter	Regions							
	0	1	2	3	4	5	6	7
	100	0	0	5	6	0	0	0

The outcome of each iteration is described below.

Iteration 1: As Node 5 is the Starting node, so at beginning, it is the only node and thus placed in OpenSet. When a node is placed at OpenSet its Parent is also recorded in the ParentList, to recover the path using Back Tracking later. However, as 5 is the Source, so its Parent is given a Sentinel Value (a negative number). Thus, OpenSet and CloseSet has entries at this stage, as shown in table 2.

Table 2: OpenSet and CloseSet Contents: Iteration 1

OpenSet	5	-	-	-	-	-	-	-
CloseSet	-	-	-	-	-	-	-	-

The ParentList takes the form as shown in table 3) after Iteration 1.

International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 07 Issue: 08 | August - 2023 SJIF Rating: 8.176 ISSN: 2582-3930

Table 3: ParentList Content: Iteration 1

Node	5
Parent	-9999

Iteration 2: Now the best node from OpenSet (i.e. the only node) 5 is placed at CloseSet. All the adjacent nodes (having direct edge) of 5, i.e. 0, 7 and 6, are added to the OpenSet. The Parents of these 3 nodes are also recorded in the ParentList. Thus now the OpenList and the CloseSet takes the form shown in table 4.

Table 4: OpenSet and CloseSet Contents: Iteration 2

Open Set	0	6	7	-	-	-	-	-
Close Set	5	-	-	-	-	-	-	-

After Iteration 2, the content of the ParentList is shown in table 5.

Table 5: ParentList Content: Iteration 2

Node	5	0	6	7
Parent	-9999	5	5	5

Iteration 3: For each node in the OpenSet, Cost is calculated. As already mentioned, for cost Calculation 25% weightage is given to the distance for arriving at the node and 75% weightage is given to the nodes own Cost value, as obtained from Table 5.3. Thus the following costs are obtained.

• Cost of Node 0 :

– f(0) = d(0) [How much traversed from Source] + e(0)[Distance from Destination] = 155 + 224 (From Fig. 5.6) = 379
– Total cost of 0 = f(0) × 25% + Cost value of Node 0 (From Table 5.3) × 75% = 379 × 25% + 100 × 25% = 169.75

• Cost of Node 6 :

– f(6) = d(6) [How much traversed from Source] + e(6)[Distance from Destination] = 136 + 424 (From Fig. 5.6) = 560
– Total cost of 6 = f(6) × 25% + Cost value of Node 6 (From Table 5.3) × 75% = 560 × 25% + 0 × 25% = 140

• Cost of Node 7 :

– f(7) = d(7) [How much traversed from Source] + e(7)[Distance from Destination] = 170 + 528 (From Fig. 5.6) = 698
– Total cost of 7 = f(7) × 25% + Cost value of Node 7 (From Table 5.3) × 75% = 698 × 25% + 0 × 25% = 174.5

Thus the Costs obtained are

• Cost(0)= 169.75
• Cost(6)= 140
• Cost(7)= 174.5

Comparing among these 3, it is obvious that Node 6 gives the minimum Cost, so now it is placed at CloseSet and its neighbors 7 and 4 are to be placed at OpenSet (7 is already there). Thus now the OpenSet and the CloseSet takes the form depicted in table 6.

Table 6: OpenSet and CloseSet Contents: Iteration 3

OpenSet	0	7	4	-	-	-	-	-
CloseSet	5	6	-	-	-	-	-	-

While determining the Parent of 7, there is needed to calculate the cost of both the paths.

• For the path to 7 through 5, Cost is 174.5

• For the path to 7 through 6,

– f(7 through 6) = d(6) [How much traversed from Source] + e(4) [Distance from Destination]
– Now d(7 through 6) = Cost upto node 6 (Parent of 7) from Source + Cost of traveling from 6 to 7 = 140 + 150 = 290
– e(4) = 528 (From Table 1)

• Thus f(7 through 6) = 290 + 528 = 818
• Thus Total cost of 7 (through 6) = 818 × 25% + 0 × 75% = 204.5
Thus, the Cost of reaching 7 is less through 5, than through 6. Hence, 5 is treated as Parent of 7. Thus the ParentList have the content shown in table 7 after Iteration 3.

Table 7: ParentList Content: Iteration 3

Node	5	0	6	7	4
Parent	-9999	5	5	5	6

Iteration 4: For each node in the OpenSet, Cost is calculated, by giving 25% weightage to the distance for arriving at the node and 75% weightage is given to the node's own Cost value, as obtained from table 5.3. Thus the following costs are obtained.

• Cost of Node 0 = 169.75 (From iteration 3)

• Cost of Node 7 = 174.5 (From iteration 3)

• Cost of Node 4:

– f(4) = d(4) [How much traversed from Source] + e(4)[Distance from Destination] – Now d(4) = Cost upto node 6 (Parent of 4) from Source + Cost of traveling from 6 to 4 = 140 (From Iteration 3) + 241 (From Fig. 1) = 381

– e(4) = 199 (From Fig. 1)

– Thus f(4) = 381 + 199 = 580

– Total Cost of Node 4 = 580 × 25% + 6 × 75% = 145 + 4.5 = 149.4

Comparing among these 3, it is obvious that Node 4 gives the minimum Cost, so now it is palced at CloseSet and its neighbors, Node 0, 1, 2, and 3 are placed at OpenSet (Node 0 is already there). Thus now the OpenSet and the CloseSet takes the form shown in table 8.

Table 8: OpenSet and CloseSet Contents: Iteration 4

OpenSet	0	7	1	2	3	-	-	-
CloseSet	5	6	4	-	-	-	-	-

While determining the Parent of 0, it is needed to calculate the cost of both the paths.

• For the path to 0 through 5, Cost is 169.75 (From iteration 3)

• For the path to 0 through 4,

– f(0 through 4) = d(0) [How much traversed from Source] + e(0) [Distance from Destination]
– Now d(0 through 4) = Cost upto node 4 (Parent of 0) from Source + Cost of traveling from 4 to 0 = 149.4 (From Iteration 4) + 161 = 310.4

– e(0) = 224 (From Fig. 1)

• Thus f(0 through 4) =310.4 + 224 = 534.4

• Thus Total cost of 0 (through 4) = 534.4 × 25% + 100 (From Table 5.3) × 75% = 133.6 + 75 = 208.6

So, the Cost of reaching 0 is less through 5, than through 4. Hence, 5 is treated as Parent of 0. Thus the ParentList have the content shown in table 9 after Iteration 4.

Table 9: ParentList Content: Iteration 4

Node	5	0	6	7	4	1	2	3
Parent	-9999	5	5	5	6	4	4	4

Iteration 5: For each node in the OpenSet, Cost is calculated, by giving 25% weightage to the distance for arriving at the node and 75% weightage is given to the node's own Cost value, as obtained from table 5.3. Thus the following costs are obtained.

• Cost of Node 0 = 169.75 (From iteration 4)

• Cost of Node 7 = 174.5 (From iteration 4)

• Cost of Node 1:

– f(1) = d(1) [How much traversed from Source] + e(1)[Distance from Destination]

– Now d(1) = Cost upto node 4 (Parent of 1) from Source + Cost of traveling from 4 to 1 = 149.4 (From Iteration 4) + 199 (From Fig. 1) = 348.4

– e(1) = 0 (From Fig. 1)[As 1 the Destination node]

– Thus f(1) = 348.4 + 0 = 348.4

– Total Cost of Node 1 = 348.4 × 25% + 0 × 75% = 87.1 + 0 = 87.1

• Cost of Node 2:

– f(2) = d(2) [How much traversed from Source] + e(2)[Distance from Destination]

– Now d(2) = Cost upto node 4 (Parent of 2) from Source + Cost of traveling from 4 to 2 = 149.4 (From Iteration 4) + 202 (From Fig. 1) = 351.4

– e(2) = 140 (From Fig. 5.6)
– Thus f(2) = 351.4 + 140 = 491.4

– Total Cost of Node 2 = 491.4 × 25% + 0 × 75% = 122.85 + 0 = 122.85

• Cost of Node 3:

– f(3) = d(3) [How much traversed from Source] + e(3)[Distance from Destination]
– Now d(3) = Cost upto node 4 (Parent of 2) from Source + Cost of traveling from 4 to 3 = 149.4 (From Iteration 4) + 197 (From Fig. 1) = 346.4

– e(3) = 204 (From Fig. 5.6)

– Thus f(2) = 346.4 + 204 = 550.4

– Total Cost of Node 3 = 550.4 × 25% + 5 × 75% = 137.6 + 3.75 = 141.35

Comparing among these five costs (Cost of members of OpenSet), it is obvious that Node 1 gives the minimum Cost, so now it is placed at ClosedSet. As 'Node 1' is the Destination node and it is at ClosedSet, successful announcement of the searching procedure is done and the procedure halts.

The path is now constructed using backtracking from the ParentList obtained finally, as shown in table 9. This required minimum cost path (from Destination to source) is :

• 1 (Destination) → 4 (as 4 is the Parent of 1)

• 4→ 6 (as 6 is the Parent of 4)

• 6→ 5 [Source] (as 5 is the Parent of 6)

Thus the final path from Source to Destination is 5→ 6→ 4→ 1.

To make the search procedure even faster, a two-way searching strategy may be adopted. Two parallel searching continues here, one initiated from source, as illustrated above and other initiated from the Destination. A set of same but separate data structures (OpenSet, CloseSet and ParentList) are kept by destination initiated search method also. When this two-way searching is adopted, everything will run same, except two searching will run in parallel and the termination criteria would become as follows (process halts when anyone among the three occurs).

- If Source initiated process reaches at Destination node
- If Destination initiated process reaches at Source node
- If both the procedure meets at mid-way. This occurs when ClosedSet of both the process holds some same node

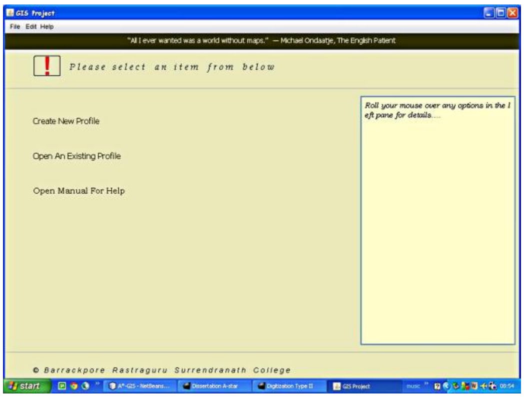
5. RESULTS

For determination of least cost path between two designated points at first the entire map should be digitized using any digitization tool and for all of the constituent small lands, the associated attribute values are also have to be fed. In addition to the identification information of the lands, the values of the influencing factors, as depicted below, are also to fed as attribute data for the constituent lands. The adjacency information of the lands are also stored. Two lands are said to be adjacent if they share some common geographic boundary line. This information is needed, because after arriving at any land, the next move is at any of its adjacent and so on. As discussed in section 1, while finding the least cost path between two designated points, the cost is judged on the basis of several influencing factors. Although there exists a huge number of influencing factors from the architectural point of view, but for the present purpose only three have been considered—Fertility of the land, Water content of the soil and distance.

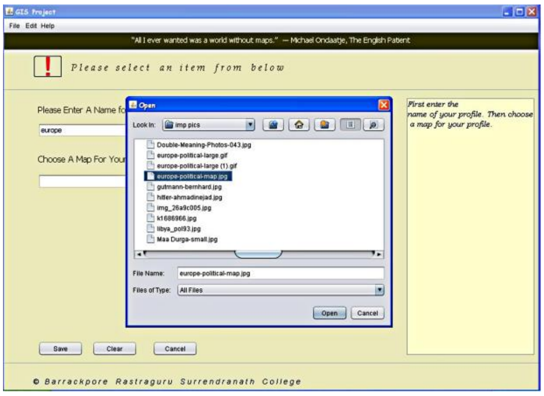
During implementation, parametric values are assigned with each of the influencing factors. Lower is the parametric value, cheaper is the cost and suitable is the land for acquisition. As less fertile lands are always tried to acquire for minimizing both cost and people agitation, thus the parametric cost of the land is increased with its fertility. A very low value, say 0 is associated with barren lands, 10 with one crop lands and so on, to be decided by the implementer. A very high sentinel value, say 999 is associated with a land which could not be acquired anyhow, for example, land possessing any heritage construction. In terms of water contents, as hard soils (minimum water content) are most preferable, so the parametric cost of the land increases with its water content. This means, hard lands possess lowest parametric cost, because they are good choice for construction and swampy lands are the worst most choice, hence possess highest parametric cost. Here also just like parameter fertility, suitable parametric values under certain scale are assigned.

To achieve the least cost path, the user has to fed only the two end points of construction and the present methodology shows the output path graphically. The implementations are done in NetBeans(JAVA) [6] [7]. The implementation requires only flat file system, not any database, which in turn increases portability of the system.

In this technique, while choosing the next node to explore, one heuristic is put, which enables the search to follow the least cost path with minimal deviation from the direct path from source to destination. The procedure needs as prerequisite the creation of profile, selection of land map and digitization using any digitization as reflected in figure 2.



(a) Creation of a new profile



(b) Associating a map with a profile for present/future use

Fig. 2: Prerequisite steps in CEGR Technique

For each of the region, associated attribute data (parametric values like fertility level, softness etc.) are fed, as shown in figure 3.

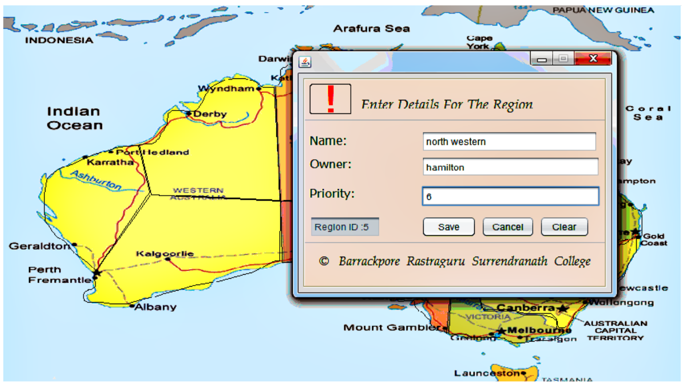
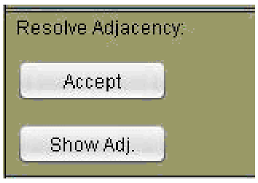


Fig. 3: Data Association in CEGR Technique

The next step is to feed the "Adjacency" information, where the information is saved by clicking "Accept" button and upon clicking "Show Adj." button, two adjacent regions are shown by a connected red line. This has been reflected in figure 4.

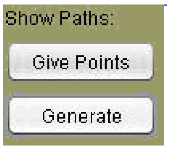


(a) Buttons for feeding or displaying adjacency information

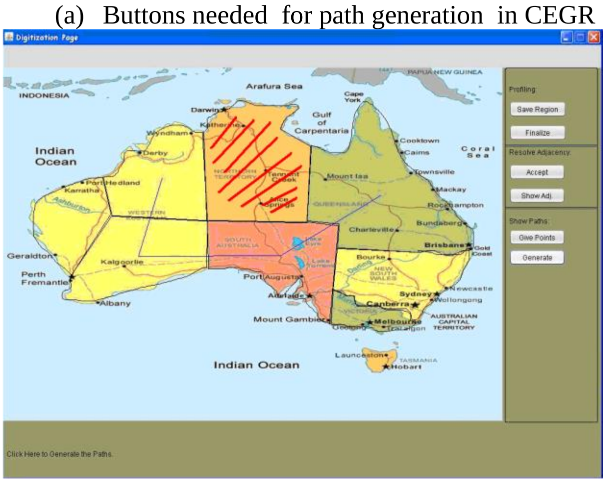
(b) Displaying adjacency

Fig. 4: Displaying adjacency information for the entire land bank

Finally the source and the destination points are fed by clicking "Give Points" button, after which the best suggested path between them is generated and displayed graphically upon clicking onto the "Generate" button, as depicted in figure 5.



(a) Buttons needed for path generation in CEGR



(b) Displaying path between source and destination

Fig. 10.9: Generation of path using CEGR

In this figure 10.9, the region en-marked with red lines has very high parametric cost, that's why this region has been sidetracked during determination of optimal cost path.

6. COMPARISONS

For comparing efficiency of the proposed goal node searching methods, the metrices completeness, optimality and time-complexity are taken into account. Table 10 compares the proposed three methods of goal searching, with three existing and very popular goal searching techniques.

Table 10: Comparison between popular existing goal node searching techniques with proposed technique

Characteristics	BFS	DFS	Elmasry et al. [8]	**Proposed Methodology**
Restriction on memory usage	No	No	No	No
Two-way search	Generally searched one-way	Generally searched one-way	Generally searched one-way	Two-way parallel search
Is the technique complete?	Yes	No	Yes	Yes
Is the technique optimal?	Yes	No	Yes	Yes
Time Complexity	$O(n^m)$	$O(n^m)$	$O(n^m)$	$O(n^{m/2})$

Where, m = depth of solution within search tree
n = branching factor of search tree

7. CONCLUSIONS

Agriculture is the backbone of many countries like India. But as time passes, increasing population and with that, increasing amount of traffic requires new construction of roads, high-ways etc. To serve better to the huge population of these countries and for accepting the challenge of improvement of the communication system, new roadways/ railway tracks are to construct, which in turn require land acquisition, leading to many agitations in regional/national level. The methodologies proposed in this chapter is an attempt, which suggests constructing a roadway from one point to another, by acquiring less fertile lands as much as possible, expecting to minimize the people (farmers) agitations at least up to a certain extent. From architectural point of view, these techniques suggest the most promising lands, in terms of various parameters like hardness, height etc., as well. But at the same time care has taken to make a trade-off between these parameters and the length of the path. Because in view of choosing the promising lands only, if the length of the path to traverse become very large, then it will in turn again increase the construction cost. To meet the challenge, the proposed method CEGR has been implemented. Here the problem is that, no supervision is there, if the path is diverting from the goal. Thus with an objective to chose the best lands, sometimes the track may go far away from the goal; requiring some extra amount of time. Moreover, there is no restriction on how many nodes will be searched to find the best one, which sometimes requires a very large amount of memory and searching time.

REFERENCES

1.Shimon Even. Graph Algorithms, Chapter- Depth First Search. Cambridge University Press, 2nd edition, 2012.

2. Leslie Pack Kaelbling. Associative reinforcement learning: A generate and test algorithm. The Journal of Machine Learning Research, 15(3):299–319, June 1994.

3. Enver Yucesan Sheldon H. Jacobson. Analyzing the Performance of Generalized Hill Climbing Algorithms. Journal of Heuristics, 10(4):387–405, July 2004.

4. S. Abraham and M. Sanglikar. Diophantine equation solver-a genetic algorithm application. Mathematical Colloquium Journal, 15(3):16–20, 2001.

5. Judea Pearl Rina Dechter. Generalized best-first search strategies and the optimality of A*. Journal of the ACM (JACM), 32(3):505–536, July 1985.

6. www.zetcode.com/tutorials/javaswingtutorial. Online;Accessed on 05 August,2014.

7. www.tutorialspoint.com/java/index.htm. Online;Accessed on 14 August,2014.

8. Amr Elmasry, Torben Hagerup, and Frank Kammer. Space-efficient Basic Graph Algorithms. In Ernst W. Mayr and Nicolas Ollinger, editors, In Conference Proceedings 32nd Symposium on Theoretical Aspects of Computer Science (STACS 2015), pages 288–301, 2015.

BIOGRAPHIES

Dr. Anirban Chakraborty received his graduation degree with Physics Honours from University of Calcutta in 1999, completed his Masters Degree in Computer Application (MCA) from University of North Bengal in 2002, received his M.Phil. Degree in Computer Science from Madurai Kamaraj University in 2009 and received his Ph.D. from University of Kalyani in 2018. He has more than 20 years of teaching experience in College and University level. Presently he is working as Selection Grade Assistant Professor in Computer Science in Barrackpore Rastraguru Surendranath College, Koplkata-120.