JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Bi-Directional Routing through low cost regions (BDR) - A Two-way Graph Traversal Methodology

Anirban Chakraborty
Assistant Professor
Department of Computer Science,
Barrackpore Rastraguru Surendranath College, Barrackpore, Kolkata – 120, India

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.

***Index Terms* - Two-way searching, influencing parameters, land acquisition problem, digitized map, optimal, complete**

I. 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. Started from a particular Source node as fed by the user, which is considered to be at depth zero, the proposed technique expands the child node of that node having the lowest cost. Next the child node of that expanded node with lowest cost is expanded and the process continues until the Destination node fed by the user is reached. Once a path is obtained to reach Destination Node, the method next expands the other nodes, starting from depth zero, which have still not been visited, to examine if there exist any better path.

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 II focuses on preliminary concepts, methodology of the proposed mechanism been mentioned in section III, followed by an illustration in section IV. Results been discussed in section V, comparison of results in section VI and finally concluding remarks are there in section VII.

II. 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.

III. METHODOLOGY

The methodology starts with incorporating a fed digitized map, considering it as a connected graph. Both the Source node S and Destination node D ε V (i.e. S, D ε V). The traversing is done through edges ε E. The procedure of graph traversal is guided by an adjacency matrix, which depicts, from a node which nodes are reachable in one jump. For the present purpose, if node ‘i’ is connected directly with node ‘j’ by an edge, then the Cell (i,j) of the adjacency matrix contains 1, else it contains 0. Moreover Cell (i, i) always contains 0, to signify that no loop is allowed here. It could be mentioned that, a loop is an edge, which connects a node with itself. The technique uses a number of data structures for its implementation. Two different lists are maintained here to discriminate among visited and un-visited nodes. A SETTLED list is used for holding all the nodes that has been expanded and visited. A FRONTIER list is there for holding the nodes those not been explored. Another one list (just an array) COST is maintained to store the cost of the current path which is being traversed. The procedure takes two nodes as input, SOURCE and the DESTINATION nodes, being fed by the user. These two nodes could be pointed out just by clicking onto a digitized map, through a proper Graphical User Interface. The present method incorporates a two-way traversal technique. Which means one searching starts from the Source node and proceed towards Destination, whereas another one proceed towards Source, starting its journey from Destination node. Thus there exists two different concepts regarding the term ‘Source’ and ‘Destination’. For the first type of traversal, the user fed ‘Source’ and ‘Destination’ coincides with ‘Source’ and ‘Destination’ of the method itself. However, for the second type of traversal, which starts from user pointed ‘Destination’ and proceeds towards user pointed ‘Source’, the method treats its ‘Source’ as user pointed ‘Destination’ and user pointed ‘Source’ is the ‘Destination’ for the method. In the following discussion while illustrating the methodology, the terms ‘Source’ and ‘Destination’ points the ‘Source’ and ‘Destination’ encountered by the method, not the actual ‘Source’ and ‘Destination’ being fed by the user. Starting from the SOURCE node, all the children nodes (these are the nodes, those have an existing edge from the source node. In other words, in the adjacency matrix, for the row of source node, the columns corresponding to these child nodes, contains 1) are added to the FRONTIER list. This 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, as illustrated above in the preliminary section.

Taking predefined percentages from all of these influencing factors, along with the distance; the COST is determined.

Next the child node with the least cost is explored and is added to the SETTLED list of the SOURCE and the cost of the node is initialized with the cost of the path from the parent to the child. The COST variable is also initialized with this determined cost. Now the children of this current SETTLED node are added to the FRONTIER list and the node with the least cost in the FRONTIER list is expanded and as well as added to the SETTLED list, as it has been visited. The Cost of reaching a node is also calculated. If A is the parent of a node B, then Cost of reaching B is, Cost of reaching A from Source plus cost of reaching A to B. Parent of a node is updated, only if a new low cost path is found. In other words, if there exists two paths, Path 1 and Path 2 to reach B, where X is the parent of B in path 1 and Y is the parent of B in Path 2, then finally B will indicate the node, which is Parent in the less cost Path, among these two. For example, if Path 2 is less costly, then B will indicate Y as its Parent.

This searching process continues, until the 'Destination / Goal' node is reached. A new searching is then initiated from the 'Source' to consider all the nodes in the FRONTIER list (which are unexplored) and are explored one by one, with a try to reach the 'Destination node. The Cost of the path is saved. When all the possible paths are encountered, the path with the least cost from 'SOURCE' to 'DESTINATION' is taken as the final optimal path (which forms the actual output). The actual distance of the Source to the Destination is calculated by the cumulative distance of the Source to the Destination, encountering all the intermediate nodes in the path. To make the searching procedure faster, the idea of a two way search or a bidirectional procedure has been incorporated. These two searches are initiated at two different points, continue in opposite direction and are executed in a parallel manner. The origin of one search is from the user defined SOURCE and proceeds to the user defined DESTINATION (i.e. initiated from SOURCE and will try to reach at DESTINATION), while the other search starts from DESTINATION and continues towards the SOURCE (i.e. search initiated from the DESTINATION will try to reach the SOURCE.

Both the SOURCE node initiated and the DESTINATION node initiated searching techniques maintain their own FRONTIER list (the list of the nodes that are GENERATED but are still not PROCESSED), SETTLED list (a list of all the PROCESSED node) and a PARENT list (a list of the PARENT nodes of both the GENERATED and the PROCESSED nodes).

The procedure terminates when any of the two searching processes is able to reach at its desired GOAL successfully or during continuing the process, both of them have met each other in the midway (which one is earlier). A check has to be made here to see if the first node generated by both the searching procedures are same or not. If it is same, then it could be inferred that both the searching procedures have met in the mid-way.

This methodology could be implemented as follows.
The searching process FS, initiated from the SOURCE will first generate a node say M from the SOURCE and check whether M is in the SETTLED list of the searching procedure initiated from the DESTINATION say FD. If M is not found in SETTLED list of FD then no action is taken and the searching method continues. But if, M was found in the SETTLED list of MD then the searching procedures is said to have meet in the mid-way and the GOAL could be found from the PARENT list of FD. This meeting criterion also remains valid for the searching procedure FD, that was initiated from the DESTINATION.

IV. ILLUSTRATION

To illustrate the method discussed above, let us consider an weighted graph, as shown in figure 1.

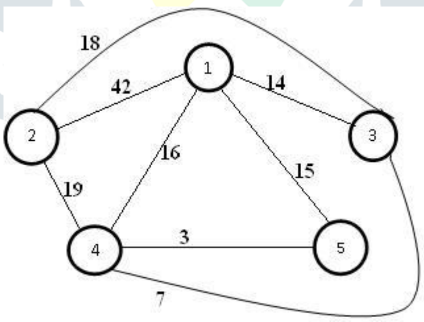


Fig. 1: A weighted graph - To illustrate the BDR Method

Here the weight of each path depicts the Euclidean distance between two end-nodes. Here five nodes been taken into consideration. An adjacency matrix is used to keep the information of the adjacencies of a region. For graph shown in figure 1, the corresponding adjacency matrix is given in table 1.

Table 1: Adjacency Matrix (corresponding to figure 5.1) Nodes Adjacency

Nodes	Adjacency				
	1	2	3	4	5
1	0	1	1	1	1
2	1	0	1	1	0

3	1	1	0	1	0
4	1	1	1	0	1
5	1	0	0	1	0

As already mentioned, during the execution of the searching method in search of an optimal path between two given nodes, in addition to the node-to-node distance, some other influencing factors are considered to determine the cost. In connection to the concept of GIS, if each node corresponds to a region of a input map, then the properties of the regions also plays a major role in determination of the path. Let two such influencing parameters, Parameter A and Parameter B have been considered and the cost of a path is determined by summing up the values taken 50% from path weight (node-to-node distance) and 25% from each of the parameters considered. Table 2 depicts the values of these parameters for each of the nodes. A high sentinel parametric value is offered to sidetrack a node. For example, if the nodes correspond to the lands of a land bank, then the land containing the heritage building should be sidetracked, as this land could not be acquired anyhow for new infrastructural development.

Table 2: Values of the influencing parameters of the nodes (corresponding to figure 1)

Influencing Parameters	**Regions**				
	1	2	3	4	5
Parameter A	100	0	5	5	0
Parameter B	5	0	0	5	0

Let Source node is 2 and Destination node is 5. Thus a least cost path is found between 2 and 5. The FRONTIER List could be implemented by using a Priority Queue, where the priority is determined by the Cost. Lesser is the Cost, more is its Priority. There exists a PARENT List for keeping track of the parent of a node. As already been stated, here two parallel search procedures are executing, first one initiated from Source and proceed towards Destination; whereas another being started from Destination, proceeds towards Source. In the following discussion, for each iteration the outcome obtained by both these procedures would be pointed separately.

Iteration 1: The content of FRONTIER List, PARENT List, SETTLED List, as well as the structure of the Search Trees obtained after Iteration 1, are discussed below separately for both
Source and Destination initiated searching techniques.

<u>A: Search initiated by Source Node:</u>

As 2 is the Source, so this node is first considered and all of its children (having direct path from 2) are placed into the FRONTIER List. From Adjacency matrix (Table 1) it is obvious that the children of '2' are '1' , '3' and '4'. As node '2' has been explored, so it is placed at SETTLED List. Now the calculation of Cost of each path is done as follows.

• For the Edge (2, 1) Cost is = 0.50 × 42 (Distance Between node 2 and 1) + 0.25 × 100 (Value of Parameter A [Table 5.1])+0.25 × 5 (Value of Parameter B [Table 1])= 47.5

• For the Edge (2, 3) Cost is = 0.50 × 18 (Distance Between node 2 and 3) + 0.25 × 5 (Value of Parameter A [Table 5.1])+0.25 × 0 (Value of Parameter B [Table 1])=10.25

• For the Edge (2, 4) Cost is = 0.50 × 19 (Distance Between node 2 and 4) + 0.25 × 5 (Value of Parameter A [Table 1]) + 0.25 × 5 (Value of Parameter B [Table 1])=12
Thus at this stage the array Cost and PARENT holds the following values.

• Cost [1] = 47.5 (Cost of reaching node 1 from Source 2) and PARENT [1] = 2 (From which node, 1 has reached)

• Cost [3] = 10.25 (Cost of reaching node 3 from Source 2) and PARENT [3] = 2 (From which node, 3 has reached)

• Cost [4] = 12 (Cost of reaching node 4 from Source 2) and PARENT [4] = 2 (From which node, 4 has reached)

The search tree, at this stage takes the form shown in figure 2

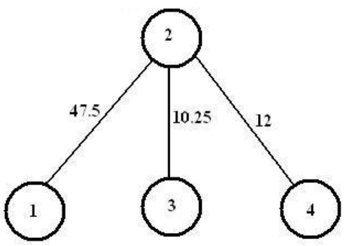


Fig. 2: Search Tree Constructed in BDR Method (Source Initiated Search): Iteration I

From this Iteration 1 (Fig. 2), it is obvious that as Cost of reaching node 3 from Source 2 is least, i.e. Cost[3] is minimum, so now node 3 has to be explored. Thus 3 is placed at SETTLED List and other two nodes 1 and 4 are placed at FRONTIER List. All the children (having direct path from 3) of node 3 are placed into the FRONTIER List.

B: Search initiated by Destination Node:

As 5 is the Starting node here, so this node is first considered and all of its children (having direct path from 5) are placed into the FRONTIER List. From Adjacency matrix (Table 5.1) it is obvious that the children of '5' are 1 and 4. As node 5 has been explored, so it is placed at SETTLED List. Now the calculation of Cost of each path is done as follows.

• For the Edge (5, 1) Cost is = 0.50 × 15 (Distance Between node 5 and 1) + 0.25 × 5 (Value of Parameter A [Table 5.1])+0.25 × 100 (Value of Parameter B [Table 1])= 33.75

• For the Edge (5, 4) Cost is = 0.50 × 3 (Distance Between node 2 and 4)+0.25 × 5 (Value of Parameter A [Table 5.1])+0.25 × 5 (Value of Parameter B [Table 1])=4

Thus at this stage the array Cost and PARENT holds the following values.

• Cost [1] = 33.75 (Cost of reaching node 1 from Starting Node 5) and PARENT [1] = 5 (From which node, 1 has reached)

• Cost [4] = 4 (Cost of reaching node 4 from Starting Node 5) and PARENT [4] = 5 (From which node, 4 has reached)
The search tree, at this stage takes the form shown in figure 3

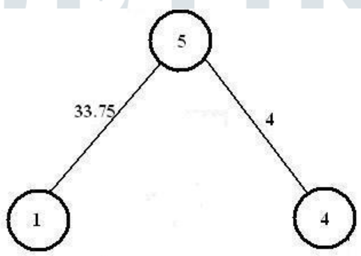


Fig. 3: Search Tree Constructed in BDR Method (Destination Initiated Search): Iteration I

From this Iteration 1 (Fig. 3), it is obvious that as Cost of reaching node 4 from Starting node 5 is least, i.e. Cost[4] is minimum, so now node 4 has to be explored. Thus 4 is placed at SETTLED List and other node 1 is placed at FRONTIER List. All the children (having direct path from 4) of node 4 are placed into the FRONTIER List.

Iteration 2: The content of FRONTIER List, PARENT List, SETTLED List, as well as the structure of the Search Trees obtained after Iteration 2, are discussed below separately for both Source and Destination initiated searching techniques (However the process terminates before Destination Initiated Searching begins at Iteration 2).

A: Search initiated by Source Node:

As obvious from the outcome of Iteration 1, Node 3 is to be explored now. From Adjacency matrix (Table 5.1) it is obvious that the children of '3' are '1' , '2' and '4'. Reaching back to node 2 (which is actually the start node) creates a loop type path, hence it is omitted. Now the calculation of Cost of each of rest of the paths are done as follows.

• For the Edge (3, 1) Cost is = 0.50 × 14 (Distance Between node 3 and 1) + 0.25 × 100 (Value of Parameter A [Table 5.1])+0.25 × 5 (Value of Parameter B [Table 5.1])= 33.25

• For the Edge (3, 4) Cost is = 0.50 × 7 (Distance Between node 3 and 4) + 0.25× 5 (Value of Parameter A [Table 5.1])+0.25 × 5 (Value of Parameter B [Table 1])=6

As the process of calculating cost is the cumulative one, which means cost of reaching a node is equal to, cost of reaching its parent from Starting node plus cost of reaching parent node to the current node. Thus at this stage the array Cost and PARENT holds the following values.

• Cost [1] = 10.25 (Cost of reaching node 3 from the starting node 2, which is the parent of current node 1) + 33.25 (Cost of reaching from 3 to 1) = 43.5 and PARENT [1] = 3 (From which node, 1 has reached)

• Cost [4] = 10.25 (Cost of reaching node 3 from the starting node 2, which is the parent of current node 4) + 6 (Cost of reaching from 3 to 4) = 16.25 and PARENT [4] = 2 (From which node, 4 could be reached with less cost)

It should be noted that, as the path from Source node 2 to 1, is cheaper via 3 (direct path from 2 to 1 costs 47.5, but via 3 it costs 43.5), so the PARENT of 1 has updated to 3.

But in case of node 4, as the direct path from 2 costs less, than via 3, (direct path from 2 to 4 costs 12, but via 3 it costs 16.25), so the PARENT of 4 has not been changed anyhow, it holds it older PARENT 2.
The search tree, at this stage takes form shown in figure 4.

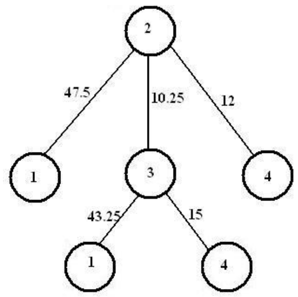


Fig. 4: Search Tree Constructed in BDR Method (Source Initiated Search): Iteration II

Now the nodes 2 and 3 have already been visited and are in the SETTLED list. From the search tree in figure 5.4 it could be concluded that the cheapest node in the FRONTIER List is node 4; so node 4 is expanded and the path from 2 to 3 is now changed to 2 to 4, because 2 is the PARENT of 4.

It is observed at this stage that, node 4 is already in the SETTLED list of the Destination Initiated Search Process, generated at end of Iteration 1. Thus it could be inferred that, both the search process, one initiated at source and other by Destination met at a mid-way at node 4. So the process halts.

Here, as the process halts by finding a node (Node 4) from both of the SETTLED List, so a successful searching of the Goal node is announced. After successful halt of the searching process, now it is the time to find out the path and the Cost of Traveling. The total Cost is found by a sum of the Cost of node 4, resulted by both the process. Hence the total cost is = Cost [4] (Resulted from Searching initiated from Source) + Cost [4] (Resulted from Searching initiated from Destination) = 12 + 4 = 16.

Similarly the entire path is found by a method of Backtracking through PARENT list, resulted by both the process. Here, PARENT[4]=2, i.e. the Source (Resulted from Searching initiated from Source) and PARENT[4]=5; i.e. the Destination, actually starting point of the Second Searching (Resulted from Searching initiated from Destination). Thus the path reduces to 2→ 4 → 5. The final search tree is as shown in figure 5.

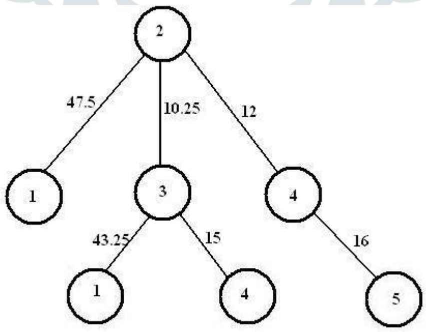


Fig. 5: Final Search Tree Constructed in BDR Method

This is the cheapest path between 2 and 5.

V. 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.

As already discussed, in the present methodology, a two way search, one initiated from source and other from destination, is incorporated. At each step, the node which could be reached by traversed lowest cost is explored. In figure 6, the centroid of all the digitized lands are presented using small circles for better understanding.

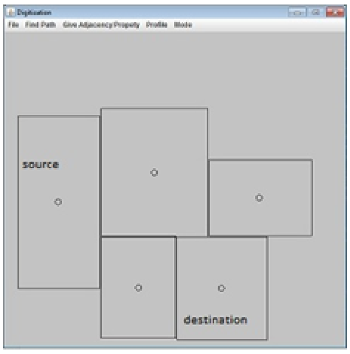


Fig. 6: Digitized lands with centroids en-marked thereon

Next the adjacencies are fed and in this step the main region, whose adjacency is being fed, is shown by red dot and its adjacent regions are by blue dots, as shown in figure 7.

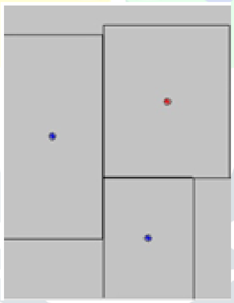
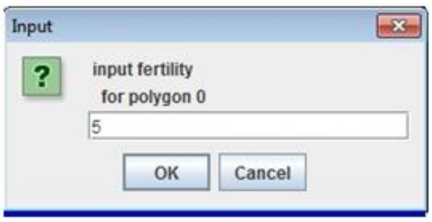
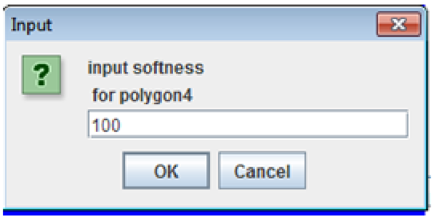


Fig. 7: Feeding Adjacencies of lands

The values of various parameter, like fertility of the land, its softness etc. are to be fed. For this purpose user has to click onto the designated region and through proper Graphical User Interface, the values are entered one after another, as depicted in figure 8.



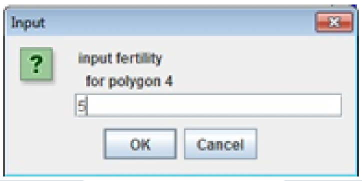
(a) Inserting Fertility Level



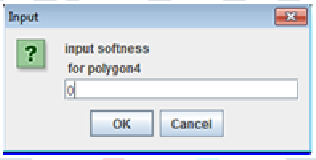
(b) Inserting Softness Level

Fig. 8: Feeding various parameters for each lands

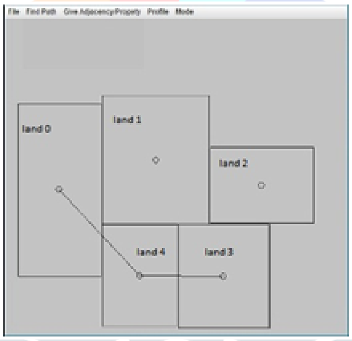
In the digitized land map shown in figure 9c , let Land 0 is the Source and Land 3 is the Destination. Let us consider two different case studies. In the 1st Case Study (Fig. 9), for Land 4, the fertility and softness are given low parametric cost value (5 for fertility value and 0 for softness value), the path is then obtained is through Land 4 (Fig. 9).



(a) Inserting Fertility Level for Land 4



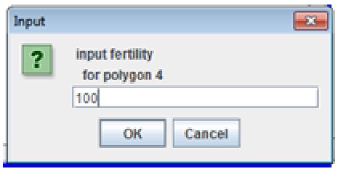
(b) Inserting Softness Level for Land 4



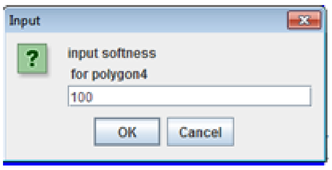
(c) Path obtained from Land 0 to Land 4

Fig. 9: Generation of path between Land 0 and Land 4: Case Study 1

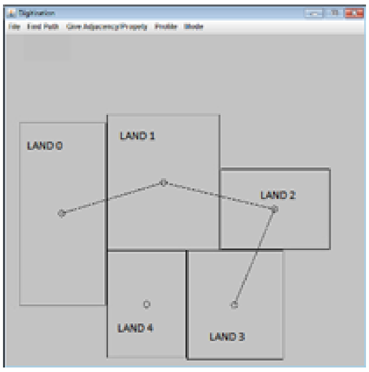
In the 2nd Case Study (Fig. 10), for Land 4, the fertility and softness are given high parametric cost value (100 for fertility value and 100 for softness value), causing the path obtained is through Land 1 and 2, i.e. here Land 4 is sidetracked (Fig. 10).



(a) Inserting Fertility Level for Land 4



(b) Inserting Softness Level for Land 4



(c) Path obtained from Land 0 to Land 4

Fig. 10: Generation of path between Land 0 and Land 4: Case Study 2

Thus it is obvious from figures 3 and 4, that parametric costs of the lands take crucial role in determination of the path.

VI. COMPARISON

For comparing efficiency of the proposed goal node searching methods, the metrices completeness, optimality and time-complexity are taken into account. Table 3 compares the proposed three methods of goal searching, with three existing and very popular goal searching techniques.

Table 3 : 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

VII. CONCLUSION

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 Bi-Directional Routing through low cost regions (BDR) 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 May,2023.

[7] www.tutorialspoint.com/java/index.htm. Online;Accessed on 14 May,2023.

[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.