International Journal of Applied Science and Mathematics
Volume 3, Issue 1, ISSN (Online): 2394-2894

Algorithmic Aspects of Geodetic Sets in Graphs

Dr. A. Anto Kinsley
Department of Mathematics, St. Xavier's (Autonomous) College, Palayamkottai - 627 002, India
Email: antokinsley@yahoo.com

K. Karthika
Department of Mathematics, St. Xavier's (Autonomous) College, Palayamkottai - 627 002, India
Email: karthika2490@gmail.com

Abstract **– Let *G* be a connected graph of order $p \geq 2$. We study about the geodetic sets of *G*. We define link vector and we prove a theorem to develop an algorithm to find the geodetic sets. First we develop algorithms to find adjacency matrix, distance matrix, closed interval between any two vertices of *G* and to find the link vector of the closed interval. Then we present an algorithm to check whether a given set of vertices is a geodetic set. Finally we design algorithm to find the minimum geodetic set of *G*. *Mathematics Subject Classification:* 68R10, 05C85**

Keywords **– Distance Matrix, Geodesic Distance, Geodetic Set, Graph Algorithms.**

I. Introduction

The geodetic number of the graph was introduced in [2], [5] and further studied by [4]. There are interesting applications of geodetic number concepts to the problem of designing the route for a shuttle and network. The different other areas that apply geodetic number concepts are telephone switching centers, facility location, distributed computing, image and video editing, neural networks and data mining.

By a graph $G = (V, E)$, we mean a finite, undirected, connected graph without loop or multiple edges [6]. We assume that $|V| = n$ throughout this paper. Before we present the algorithm, we give a brief description of the computation of link vector [1] of the graph, which are used to design algorithms. To design algorithms we studied the book [7] by Wlif. In this paper, we study a binary operation Λ which is defined by A. Anto Kinsley, S. Somasundaram, and C. Xavier in [1]. We use this operation and prove some result to develop algorithms to check whether a given set of vertices is a geodetic set and to find the minimum geodetic set of *G*. In all algorithms we consider a graph with distance matrix as input values.

In this section, some definitions and important results on geodetic sets [3] - [5] are studied.

Definition 1.1

Given two vertices *u* and *v* of a simple connected graph $G = (V, E)$, the distance $d(u, v)$ is the length of a shortest *u*-*v* path in *G*. A *u*-*v* path of length $d(u, v)$ is called a *u*-*v* geodesic. A vertex *w* is said to lie on a *u*-*v* geodesic *P* if *w* is a vertex of *P* including the vertices *u* and *v*.

Definition 1.2

For a set *S* of vertices, let the closed interval $I[S]$ of *S* be the union of the closed intervals $I[u, v]$ over all the pairs of vertices *u* and *v* in *S*. A set of vertices *S* is called *geodetic set* if $I[S] = V(G)$ and the minimum cardinality of the geodetic set is the *geodetic number* and is denoted by $g(G)$. A geodetic set of cardinality $g(G)$ is called a *minimum geodetic set* (or) *g- set* of *G*.

Observation 1.3

If G is a non- trivial connected graph of order $n \geq 2$, then, $2 \leq g(G) \leq n$.

Example 1.4

For a graph given in Figure 1, the minimum geodetic sets are $\{a, f\}$ and $\{a, h\}$. The cardinality of the minimum geodetic set of *G* is 2.

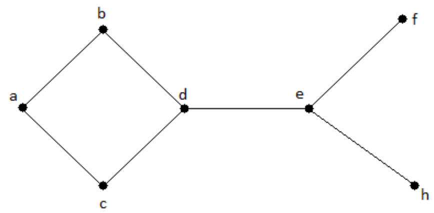

Fig. 1. A Graph

II. Link Vectors

In this section we define link vectors [1] and we will use the concept in the algorithms.

Definition 2.1

Characterize each closed interval as an *n*-tuple. Each place of *n*-tuple can be represented by a binary 1 or 0. Call this *n*-tuple as a *link vector*. Denote $LV(I) = I'$. Put 1 if the vertex belongs to the closed interval otherwise 0. If all the co-ordinate of the link vector is equal to 1 then it is called as *full*. Denote $I[(1)]$.

Definition 2.2

Let *G* be a graph. Let ρ be the set of all *LV* of *G*. Define a binary operation $\vee: \rho \times \rho \rightarrow \rho$ by $(v_1, v_2, \ldots, v_k) \vee (u_1, u_2, \ldots, u_k) = (w_1, w_2, \ldots, w_k)$ where $w_i = \max\{v_i, u_i\}$. Now we generalize this idea for more than two *LV*'s. Operation on any number of *LV*s by $\vee$ can be followed by pair wise. For any $I_i \in \rho$ $(1 \leq i \leq 4)$, $I'_1 \vee I'_2 \vee I'_3$ means $(I'_1 \vee I'_2) \vee I'_3$ or $I'_1 \vee (I'_2 \vee I'_3)$.
$I'_1 \vee I'_2 \vee I'_3 \vee I'_4$ means $(I'_1 \vee I'_2) \vee (I'_3 \vee I'_4)$ and so on.

Example 2.3

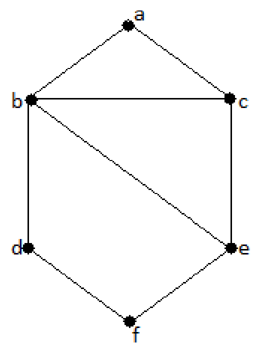

Fig. 2. A graph *G*

Consider the graph $G = (V, E)$ shown in Figure 2. Arrange the vertices in the following order: a, b, c, d, e, f. The closed interval of $\{a, f\}$, $\{c, d\}$ and $\{b, e\}$ are

I_1: a, b, c, e, f
I_2: c, b, d
I_3: b, e

The corresponding link vectors are,

$I_1' = (1,1,1,0,1,1)$
$I_2' = (0,1,0,1,0,0)$
$I_3' = (0,1,0,0,1,0)$

In the above example, we see that
$I'_1 \vee I'_2 = (1,1,1,1,1,1)$ is full.

It yields that $\{a, f, c, d\}$ is the geodetic set of G.

Now we turn our attention to design algorithms for geodetic concepts. We prove an important theorem, which will be used in the main algorithm.

Theorem 2.4

Let G be a graph with n vertices. Then $\bigvee_{i=1}^{r} I'_i$ is full, where r is the number of closed interval obtained between each pair of vertices of S if and only if $S = \{v_1, v_2, \ldots, v_k\}$ is a geodetic set.

Proof

Suppose that $\bigvee_{i=1}^{r} I'_i$ is full. Now I_i' is the link vector LV of I_i. If the j^{th} coordinate of I_i' is 0 means that v_j is not in the I_i. By definition of LV, j^{th} coordinate of $I_p' \vee I_q'$ is 1 means that v_j is in I_p or I_q or both. That is, v_j is in at least one of the closed interval I. Since $\bigvee_{i=1}^{r} I'_i$ is full (that is, $\bigvee_{i=1}^{r} I'_i = (1,1,\ldots\ldots,1) = I([1])$), every vertex of $V(G)$ of G belongs to at least one closed interval obtained by every pair of vertices of S. Thus $S = \{v_1, v_2, \ldots, v_k\}$ is a geodetic set of G.

Conversely, suppose S is the geodetic set of G. Then $\bigcup_{x,y\in S} I[x,y] = V(G)$. We know that $I[x,y]$ consists of all vertices lying in the x-y geodesic for some $x, y \in S$. Since S is geodetic, j^{th} coordinate of any one of $I'_1, I'_2, \ldots\ldots, I'_r$ is 1 and so j^{th} coordinate of $I'_1 \vee I'_2 \vee \ldots\ldots \vee I'_r = 1$. Hence every coordinate of $I'_1 \vee I'_2 \vee \ldots\ldots \vee I'_r = 1$. Hence $\bigvee_{i=1}^{r} I'_i$ is full.

III. Development of Algorithms

In this section we present six algorithms. Now suppose that a graph G with edge and vertex set is given. We develop a new algorithm to find adjacency matrix and distance matrix. Also we develop the algorithms to find the closed intervals and link vectors which are useful to develop the main algorithm.

Algorithm 3.1

Algorithm to find an adjacency matrix:

Input: A graph $G = (V, E)$ given with its vertex set $V = \{v_1, v_2, \ldots, v_n\}$ and edge set $E = \{e_1, \ldots, e_m\}$.

Output: Adjacency matrix (a_{ij}) of G.

```
for i = 1 to n
   for j = 1 to n
      if vivj ∈ E , then aij = 1
      else aij = 0
   Print (aij).
```

In this algorithm, we have used a nested for loop and so it requires n^2 verifications. Thus it requires $O(n^2)$ cost of time.

Algorithm 3.2

Algorithm to find distance matrix:

Input: A graph $G = (V, E)$ with its adjacency matrix (a_{ij}) and the degree sequence.

Output: Distance matrix (D_{ij}) of G.

```
for i = 1 to n
  for j = i to n
    if (aij = 0 & i = j)  Dij = 0
    else if (aij = 1 & i ≠ j)  Dij = 1
    else if (aij = 0 & i ≠ j)  count = 1
       for l = 1 to deg (vi)
        begin
          for k = 1 to n
            if (aik = 1 & k = j)  Dij = count + 1
            else if (akl = 1 & l ≠ j)
                  begin
        count =  count + 1
                  i = k
        end
          Dij (l) = count
          end
      Dij = min {Dij (l)}
for j = 2 to n
  for i = 1 to j-1
    Dji = Dij
Print (Dij).
```

In this algorithm, we have nested a for loop, so it requires n^4 verifications. Also we have another nested for loop for substitution, which requires n^2 substitution. Thus it requires $O(n^4)$ cost of time.

Next we develop an algorithm to find the closed interval between any two vertices of G.

Algorithm 3.3

Algorithm to find $I[v_i, v_j]$:

Procedure closed-interval $I[S_2]$

Input: A graph $G = (V, E)$ with its distance matrix and a subset $S_2 = \{v_i, v_j\}$ of V.

Output: $I[v_i, v_j]$

```
Let I [vi, vj] = {vi}
find nbh {vi}
  if  d (nbh(vi ) , vj ) = d(vi ,vj) – 1
    I [vi, vj ] = I [vi, vj ] ∪ {nbh(vi )}
    vi = nbh(vi )
```

Here the algorithm collects the neighborhood of each vertex. That is, it works in deg (v_i) number of times to find the neighborhood of v_i. That is, totally it works in $2m$ times. Thus it requires $O(m)$ cost of time.

Next we develop an algorithm to find the link vector of the closed interval $I[S]$.

Algorithm 3.4

Algorithm to find the link vector $I'[S_2]$:

Procedure link vector $I'[S_2]$

Input: A graph $G = (V, E)$ and a 2-subset S_2 of V with its closed interval $I[S_2]$.

Output: The link vector $I'[S_2]$

```
LV: (x1, x2, . . . , xn)
for i = 1 to n
```

if $v_i \in I[S_2]$ then put $x_i = 1$
else $x_i = 0$

Here the algorithm takes n verifications. That is, it works $O(n)$ cost of times.

Next we develop an algorithm to check whether the given set S of vertices is geodetic or not.

Algorithm 3.5

Geodetic set confirmation algorithm:

Procedure geodetic [S]

Input: A graph $G = (V, E)$ of order > 2 and a subset $S = \{v_1, v_2, \ldots, v_k\}$ of vertices.

Output: S is a geodetic set or not.

Step 1: Find all the 2-subsets S_2 of S

{There are $\binom{k}{2}$ number of subsets S_2 of S}

Step 2: Take $L \leftarrow (0)$

for $i = 1$ to $\binom{k}{2}$

begin

Closed- interval $I_i[S_2]$

Link vector $I'_i[S_2]$

$L = L \vee I'_i[S_2]$

end

If L is full then the given set S is a geodetic set.

Otherwise S is not a geodetic set.

In this algorithm, step 2 will work in $\frac{k(k-1)}{2}$ times. Next part of step 2 is the algorithm Closed- interval $I_i[S_2]$ and Link vector $I'_i[S_2]$ and hence this part will work with $2m + n$ verifications. Total cost of time is $O(k^2(m + n)$, where k is the cardinality of the given vertex subset and m is the number of edges in G. But in this step the given vertex acts as a root and all other vertices are approached through a spanning tree. Therefore there are $n + (n\text{-}1)$ verifications needed, since $m = n\text{-}1$ for a tree. That is, $O(n^2(2n\text{-}1))$, that is $O(n^3)$. Thus this algorithm requires $O(n^3)$ cost of time.

Finally we develop the main algorithm to find the minimum geodetic set of a graph G.

Algorithm 3.6

Minimum geodetic set algorithm:

Procedure g-set[S]

Input: A graph $G = (V, E)$ with its vertex set $V(G) = \{v_1, v_2, \ldots, v_n\}$ of vertices.

Output: S_j with g(G) vertices.

Step 1: Take $k \leftarrow 2$

Step 2: Take all the subsets S_j $\left(1 \le j \le \binom{n}{k}\right)$ of $V(G)$ with k vertices.

Step 3: for $j = 1$ to $\binom{n}{k}$

begin

geodetic [S_j]

if yes then stop and print S_j is a minimum geodetic set.

end

Step 4: Otherwise take $k = k + 1$ and return to step 2.

In this algorithm, by the for loop it takes all the 2^n subsets and each time it calls the procedure geodetic [S], its complexity is $O(n^2)$. Hence this algorithm is *NP*-complete.

IV. Conclusion

In this paper we have established algorithms to check whether a given set is geodetic or not and to find a g-set of G, whose complexities are $O(n^3)$ and *NP*- complete respectively.

V. Acknowledgement

We thank God and our parents for their support. The author would like to thank the referee for his valuable suggestions.

References

[1] A. Anto Kinsley, S. Somasundaram, C. Xavier, An algorithm to find the strong domination number of a graph, 34th Annual Convention of the Computer Society of India, Tata McGraw-Hill Publishing Company limited, New Delhi, (2000), 255-262.
[2] F. Buckley, F. Harary, Distance in Graphs, Addison- Wesley, Redwood City, CA (1990).
[3] Byung Kee Kim, The geodetic number of a graph, J. Appl. Math. & Computing Volume 16 (2004), No. 1 - 2, pp. 525 – 532.
[4] G. Chartrand, F. Harary, H. Swart, and P. Zang, Geodomination in graphs, Bull. Inst. Appl. Volume 31 (2001), pp. 51 – 59.
[5] Frank Harary, Emmanuel Loukakis, Constantine Tsouros, The Geodetic number of a graph, Math. Computing Modeling, Volume 17(1993), No. 11, pp. 89 – 95.
[6] F. Harary, Graph Theory, Addison- Wesley, New York, 1972.
[7] H. S. Wilf, Algorithms and Complexity, Prentice – Hall International, Inc., U. S. A (1986)

Author's Profile

Dr. A. Anto Kinsley M.Sc., M. Phil., M. Tech., Ph.D.
Associate Professor in Mathematics in St. Xavier's College (Autonomous), Palayamkottai. He has 30 years teaching experience. He published 13 research articles in National and International Journals and he completed two UGC minor research projects. His area of specializations are Graph Algorithms, Flow Networks. Distance in Graphs, Central structures in Graphs. Domination and Distance domination in graphs, Star Graphs, Convexity in Graphs, DNA computing.

K. Karthika
Research scholar in Department of Mathematics in St. Xavier's College (Autonomous), Palayamkottai. She received his M. Sc in Aditanar College of Arts and Science, affilicated to MS University, Tirunelveli and M. Phil in St. Xavier's College (Autonomous), Palayamkottai. Area of Specialization is graph theory.