Algorithmic Aspects of k-Geodetic Sets in Graphs

Dr. A. Anto Kinsley¹* K. Karthika¹

¹ Department of Mathematics, St. Xavier’s (Autonomous) College, Palayamkottai, Tamilnadu, India.

Abstract: Let G be a connected graph of order \( p \geq 2 \). We study about the geodetic sets and k-geodetic sets of G. We study link vectors and prove a theorem to develop an algorithm to find the k-geodetic sets. Initially we study algorithms to find the closed interval between any two vertices of G and to find its link vectors. In this paper we present two algorithms to check whether a given set of vertices is a k-geodetic set and to find the minimum k-geodetic set of G.

MSC: 68R10, 05C85.

Keywords: Graph, geodetic set, k-geodetic set, graph algorithms.

© JS Publication.

1. Introduction

The concepts of a geodetic set and the geodetic number of a graph \([3, 4]\) were introduced by Harary et al., and further studied by several authors. Sergio Bermudo et al., studied about the relation between geodetic and k-geodetic sets in arbitrary graph. Before we present the algorithm, we give a brief description of the computation of link vector of the closed interval of the graph those are involved in our algorithm. By a graph \( G = (V, E) \), we mean a finite, undirected, connected graph without loop or multiple edges \([5]\). 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 \([8]\). In this paper, we study a binary operation \( \vee \) \([1]\) and prove some important results. This operation \([1]\) is used to develop algorithms to check whether a given set of vertices is a k-geodetic set and find the minimum k-geodetic set of G.

In this section, some basic definitions and important results on k-geodetic sets \([6, 7]\) are given.

Definition 1.1. Let G be connected graph of order \( p \geq 2 \). For an integer \( k \geq 1 \), a vertex \( v \in V \) of G is k-geodominated by a pair \( x, y \in V \) if v lies on an \( x-y \) geodesic of G and \( d(x, y) = k \). A subset \( S \subseteq V \) is a k-geodetic set if each vertex \( v \in V/S \) is k-geodominated by some pair of vertices of S. The minimum cardinality of a k-geodetic set of G is the k-geodetic number of G and it is denoted by \( g_k(G) \) and that set is called as minimum k-geodetic set.

Example 1.2. For a graph G shown in Figure 1, the k-geodetic number of a graph G is shown in the Table 1.

<table>
<thead>
<tr>
<th>K</th>
<th>k-geodetic set</th>
<th>( g_k(G) )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>{a, b, c, d, e, f}</td>
<td>6</td>
</tr>
<tr>
<td>2</td>
<td>{a, e, b, f}</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>{a, f}</td>
<td>2</td>
</tr>
</tbody>
</table>

Table 1.

* E-mail: antokinsley@yahoo.com
Algorithmic Aspects of k-Geodetic Sets in Graphs

Figure 1.

Theorem 1.3. For any graph $G$ of order $n$ and maximum degree, $g_k(G) \geq \lceil \frac{2n}{\Delta(\Delta-1)^{k-1}+(k-1)+2} \rceil$.

2. Link Vectors

In this section we briefly study about the definition of link vectors, some results [4] and we will use this concept in the algorithm.

Definition 2.1. Characterize each closed interval as a $n$-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 are equal to 1 then it is called as full. Denote $I[(1)]$.

Definition 2.2. Let $G$ be a graph. Let $\rho$ be the set of all LV of $G$. Define a binary operation $\lor : \rho \times \rho \to \rho$ by $(v_1,v_2,...,v_k) \lor (u_1,u_2,...,u_k) = (w_1,w_2,...,w_k)$ where $w_i = \max\{v_i,u_i\}$. Now we generalize this idea for more than two LVs. Operation on any number of LVs by $\lor$ can be followed by pairwise. For any $I_i \in \rho(1 \leq i \leq 4), I_1 \lor I_2 \lor I_3 \lor I_4$ means $(I_1 \lor I_2) \lor I_3 \lor I_4$ or $I_1 \lor (I_2 \lor I_3) \lor I_4$. $I_1 \lor I_2 \lor I_3 \lor I_4$ means $(I_1 \lor I_2) \lor (I_3 \lor I_4)$ and so on.

Theorem 2.3. Let $G$ be a graph with $n$ vertices. Then $\lor_{i=1}^r I_i$ is full, where $r$ is the number of closed interval obtained between each pair of vertices of $S$ if and only if $S = \{v_1,v_2,...,v_k\}$ is a geodetic set.

3. Development of Algorithms

In this section first we studied algorithms closed-interval $I[S]$ and link vector $I'[S_2]$ [2] which are used to develop an algorithm to find k-geodetic sets. Next we design an algorithm to check whether a given set of vertices is a k-geodetic set and then find the minimum k-geodetic set of $G$.

Algorithm 3.1. Algorithm to find $I[v_i,v_j]$.

Procedure closed-interval $I[S]$.

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

Output: $I[v_i,v_j]$

Let $I[v_i,v_j] = \{v_i\}$

find nbh $\{v_i\}$

if $d(\text{nbh}(v_i),v_j) = d(v_i,v_j) - 1$

$I[v_i,v_j] = I[v_i,v_j] \cup \{\text{nbh}(v_i)\}$

$v_i = \text{nbh}(v_i)$

Here the algorithm collects the neighborhood of each vertex. That is, it works in $\text{deg}(v_i)$ number of times to find the neighborhood of $v_i$. That is, totally it works in $2q$ times, $q$ is the number of edges in $G$. Thus it requires $O(q)$ cost of time.

Next we develop an algorithm to find the link vector of the closed interval $I[S]$.
Algorithm 3.2. 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 $I'[S_2]$ 

$LV : (x_1, x_2, \ldots, x_n)$

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 the following algorithm to check whether the given set $S$ of vertices is $k$-geodetic or not.

Algorithm 3.3. $k$-geodetic set confirmation algorithm

Procedure $k$-geodetic [$S$].

**Input:** A graph $G = (V, E)$ with its distance matrix and a subset $S = \{v_1, v_2, \ldots, v_m\}$ and $k$.

**Output:** $S$ is a $k$-geodetic set or not.

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

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

**Step 2:** for $j = 1$ to $\binom{m}{2}$

check $d_j(S_2) = k$

if all $d_j(S_2) = k$, then take $L \leftarrow (0)$

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

closed interval $I_i[S_2]$

link vector $I'_i[S_2]$

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

If $L$ is full then the given set is a $k$-geodetic set.

Otherwise $S$ is not a $k$-geodetic set.

In this algorithm, step 2 will work in $\frac{m(m-1)}{2}$ times. Next part of step 2 is the Algorithm 3.1 and 3.2 and hence this part will work with $\frac{m(m-1)}{2}(2q + n)$ verifications. Thus this algorithm requires $O(m^2 + m^2(q + n))$ cost of time, where $m$ is the cardinality of the given vertex subset and $q$ 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 $n + (n - 1)$ verifications needed, since $q = n-1$ for a tree. Total cost of time is $O(m^2 + ((n - 1) + n)m^2)$, that is $O(n^2 + (2n - 1)n^2)$, that is $O(n^3)$. Thus this algorithm requires $O(n^3)$ cost of time. Finally we develop an algorithm to find all minimum $k$-geodetic sets of a graph $G$.

Algorithm 3.4. Minimum $k$-geodetic set algorithm.

**Input:** A graph $G = (V, E)$ with $V(G) = \{v_1, v_2, \ldots, v_n\}$ of vertices, its distance matrix, the maximum degree and the value $k$.

**Output:** $S_j$'s with $g_k(G)$ vertices.

**Step 1:** Take $r \leftarrow \left\lceil \frac{2n}{3\Delta - 1 + \frac{2n}{(k-1)^2}} \right\rceil$

**Step 2:** Take all the $\binom{m}{r}$ subsets $S_j$ of $V$ with $m$ vertices.

**Step 3:** for $j = 1$ to $\binom{m}{r}$
begin
k-geodetic \[ S_j \]
if yes then stop and print \( S_j \) is a minimum k-geodetic set.
end

**Step 5:** Otherwise take \( r = r + 1 \) and return to step 2.

In this algorithm, we work on all subsets of \( V \) and hence it will be a NP-complete problem.

## 4. Conclusion

In this paper we studied about the k-geodetic sets on a finite, undirected, connected graph without loop or multiple edges, whose distance and maximum degree are known. We have studied about the link vector of the closed interval of \( G \) and a binary operations \( \lor \). Some important results which play a vital role in the algorithm development are found. Initially we have designed an algorithm to check whether the given set of vertices is a k -geodetic set. Then we have presented an algorithm to find the minimum k-geodetic sets of a graph.

## References


