# K-SETS OF A GRAPH AND VULNERABILITY OF COMMUNICATION NETS

by

P. Karivaratharajan * and K. Thulasiraman **

The concept of the $k$-vulnerability of communication nets is introduced and a procedure to design $k$-invulnerable communication nets is given. In the course of this study several important properties of the $k$-sets of a graph are discussed. Also the problem of generating the $k$-sets is elucidated. Illustrative examples are worked out.

(Continued from the Matrix and Tensor Quarterly, December 1974.)

*Lemma 1*: Each tip vertex or the vertex adjacent to it will be present in every $L$-set of a graph $G_T$.

*Proof*: Consider any tip vertex $v_i$ and let the vertex adjacent to $v_i$ be denoted by $v_j$. We prove the lemma by contradiction.

Let there be an $L$-set which contains neither $v_i$ nor $v_j$. Consider the set $\{L \cup v_i\}$. Since $v_j$ is not present in $L$, $v_i$ is not adjacent to any vertex in $L$. Hence the set $\{L \cup v_i\}$ constitutes an independent set of vertices. This contradicts the fact that $L$ is a maximal set of independent vertices. Hence the lemma.

*Theorem 3*: (*a*) For every tip vertex $v_i$ of a graph $G_T$, there exists an $L_{max}$-set containing $v_i$.

(*b*) For a graph $G_T$, with $|V| > 2$, there exists an $L_{max}$-set containing all the tip vertices.

*Proof*: (*a*) Consider an $L_{max}$-set, $L'$, which does not contain a tip vertex $v_i$. Let $v_j$ be adjacent to $v_i$. It follows from Lemma 1 that $v_j \in L'$. We next show, by construction, the existence of an $L_{max}$-set containing $v_i$.

Consider the set $S = \{(L' - v_j) \cup v_i\}$. $S$ constitutes a set of independent vertices, since $v_i$ is adjacent to only $v_j$ and $v_j \notin S$. Further $|S| = |L'|$. Hence $S$ is an $L_{max}$-set and it contains $v_i$.

(*b*) By repeating successively the procedure followed in (*a*) for all the tip vertices not present in $L'$, one can construct an $L_{max}$-set containing all the tip vertices of $G_T$.

## GENERATION OF $K_{min}$-SETS FOR A GRAPH $G_T$

We present in this section an algorithm for the generation of a $K_{min}$-set of a graph $G_T$.

We first rename the graph $G_T$ as $G_1$ and then obtain the graph $G_{i+1}$ from $G_i$ recursively as follows.

Let $v_{ia}$ be a tip vertex of graph $G_i$, with $v_{ib}$ adjacent to $v_{ia}$. Let $V_{ia}$ denote the set of all the tip vertices of $G_i$ adjacent to $v_{ib}$. We obtain the graph $G_{i+1}$ from $G_i$

* The Department of Electrical Engineering, Indian Institute of Technology, Madras, India.
** The Computer Centre, Indian Institute of Technology, Madras, India.

by first stripping $v_{ib}$ and then removing all the isolated vertices which result as a consequence.

The process of generation of $G_i$'s will terminate after $n$ steps when $G_{n+1}$ is a null graph.

We now observe the following.

(a) The vertices of $V_{ia}$ are independent.

(b) If $L_i$ is an $L$-set for $G_i$ such that $V_{ia} \in L_i$ then $(L_i - V_{ia})$ is an $L$-set for $G_{i+1}$.

(c) $\{L_i \cup V_{(i-1)a}\}$ is an $L$-set for $G_{i-1}$;

(d) $G_n$ is a star-tree; and

(e) $V_{na}$ is the $L_{max}$-set for $G_n$.

*Lemma 2*: (a) If $L_{i,max}$ is an $L_{max}$-set for $G_i$ such that $V_{ia} \in L_{i,max}$ then the set $(L_{i,max} - V_{ia})$ is an $L_{max}$-set for $G_{i+1}$.

(b) If $L_{(i+1),max}$ is any $L_{max}$-set for $G_{i+1}$ then $\{L_{(i+1),max} \cup V_{ia}\}$ is an $L_{max}$-set of $G_i$.

*Proof*: Let $L_{i,max}$ be an $L_{max}$-set of $G_i$ such that $V_{ia} \in L_{i,max}$. Let $L_{(i+1),max}$ be any $L_{max}$-set of $G_{i+1}$. Further let

$$|L_{i,max}| = l_{i,max}$$

Since $(L_{i,max} - V_{ia})$ is an $L$-set of $G_{i+1}$, we have

$$l_{i,max} - |V_{ia}| < l_{(i+1),max} \tag{1}$$

Also since $\{L_{(i+1),max} \cup V_{ia}\}$ is an $L$-set for $G_i$ we have

$$l_{(i+1),max} + |V_{ia}| < l_{i,max} \tag{2}$$

From (1) and (2) we get

$$l_{i,max} - l_{(i+1),max} = |V_{ia}|$$

This completes the proof of the lemma.

*Theorem 4*: The set $\{v_{1b}, \ldots, v_{nb}\}$ is a $K_{min}$-set for $G_T$.

*Proof*: Consider the set $\{\bigcup_{i=1}^{n} V_{ia}\}$. As we have noted earlier $V_{na}$ is an $L_{max}$-set for $G_n$. If then follows from Lemma 2(b) that $\{V_{na} \cup V_{(n-1)a}\}$ is an $L_{max}$-set for $G_{n-1}$. Proceeding in this way we get that $\{\bigcup_{i=1}^{n} V_{ia}\}$ is an $L_{max}$-set for $G_1$ which by definition is the same as $G_T$.

Hence $\{V - \bigcup_{i=1}^{n} V_{ia}\} = \{v_{1b}, \ldots, v_{nb}\}$ is a $K_{min}$-set for $G_T$.

Based on theorem 4, an algorithm to generate a $K_{min}$-set of $G_T$ may be stated as follows.

**Algorithm 1:**

*Step 1*: Let $G_1 = G_T$

*Step 2*: $S_0 = \phi$, where $\phi$ is the null set. Let $i = 0$

*Step 3*: Replace $i$ by $i+1$. Identify a tip vertex $v_{ia}$ of $G_i$. Let $v_{ib}$ be the vertex adjacent to $v_{ia}$

(i) Set $S_i = \{S_{i-1} \cup v_{ib}\}$

(ii) Find $G_{i+1}$ from $G_i$ by first stripping the vertex $v_{ib}$ and then removing all the isolated vertices which result as a consequence.

*Step 4*: If $G_{i+1}$ is a null graph go to Step 5; otherwise return to Step 3.

*Step 5*: $K_{\min} = S_i$

*Example 2*: We illustrate the algorithm by determining a $K_{\min}$-set for the graph $G_T = G_1$ shown in Fig. 3($a$).

Choosing $v_1$ as $v_{1a}$ we get $v_{1b} = v_2$. The graph that results after stripping $v_2$ is shown in Fig. 3($b$). After removing the isolated vertices from the graph shown in Fig. 3($b$), $G_2$ is obtained as shown in Fig. 3($c$). Choosing successively $v_{2a} = v_6$, $v_{3a} = v_{11}$, $v_{4a} = v_8$ and $v_{5a} = v_3$, we get the graphs $G_3$, $G_4$ and $G_5$ as shown in Figs. 3($d$), ($e$) and ($f$) respectively.

We also see that $v_{2b} = v_5$, $v_{3b} = v_{12}$, $v_{4b} = v_9$ and $v_{5b} = v_4$. Thus the set

$$\{v_{1b}, v_{2b}, v_{3b}, v_{4b}, v_{5b}\} = \{v_2, v_5, v_{12}, v_9, v_4\}$$

is a $K_{\min}$-set for $G_T$.

It may be easily verified, following the procedure described above, that $\{v_1, v_5, v_{11}, v_9, v_3\}$ is also a $K_{\min}$-set for $G_T$.

## DESIGN OF MINIMUM EDGE $V$-VERTEX GRAPHS HAVING A SPECIFIED $K_{\min}$

We give, in this section a new proof of an algorithm given for the construction of minimum edge $v$-vertex graphs having a prescribed $k_{\min}$. We also obtain a lower bound on the number of edges of a $v$-vertex graph having a $k_{\min} = k$.

*Theorem 5*: Let G be a $v$-vertex graph with $l_{\max} = l$. Let G be in $l$ parts $G_i$, $i = 1, \ldots, l$. Let $V_i$ denote the set of vertices in $G_i$. If $G'$ is any $(v+1)$-vertex graph such that $l'_{\max} = l$ and $G \subset G'$, then $G'$ will have at least $(|E| + \min_i\{|V_i|\})$ edges. Further there exists a graph $G'$ with $l'_{\max} = l$ and $|E'| = |E| + \min_i\{|V_i|\}$.

*Proof*: Let $v_x$ denote the only vertex not present in G, but present in $G'$. We will denote by $R_i$, the subset of vertices of $V_i$ adjacent to $v_x$. Since for $G'$, $l'_{\max} = l$ it is necessary that for some $j$ and $k$, all the vertices of $(V_k - R_k)$ should be adjacent to all the vertices of $(V_j - R_j)$. Hence $G'$ will have $(|E| + n)$ edges where

$$n = (|V_k| - |R_k|)(|V_j| - |R_j|) + \sum_{i=1}^{l} R_i$$

$$= (|V_k| - |R_k| - 1)(|V_j| - |R_j| - 1) - 1 + |V_k| + |V_j| + \sum_{\substack{i=1 \\ i\neq j,k}}^{l} R_i$$

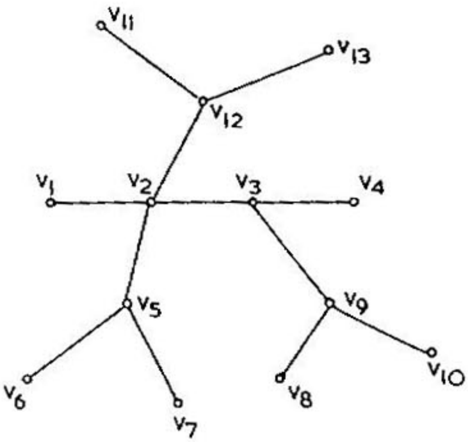


Fig. 3($a$): Graph $G_1$

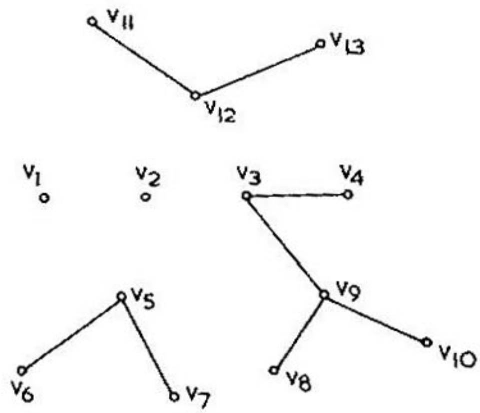


Fig. 3($b$)

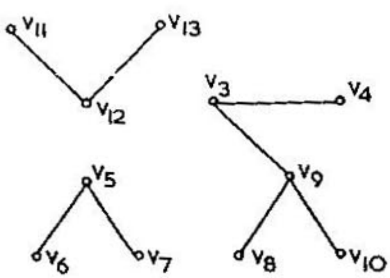


Fig. 3($c$): Graph $G_2$

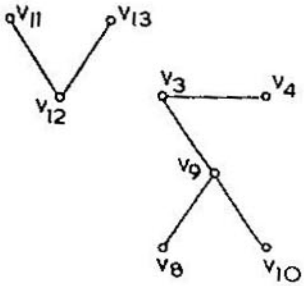


Fig. 3($d$): Graph $G_3$

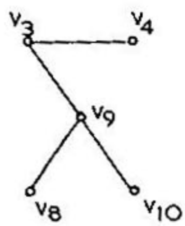


Fig. 3($e$): Graph $G_4$

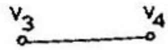


Fig. 3($f$): Graph $G_5$

Assuming, without loss of generality, that $V_k \geqslant V_j$, we note that $n$ will be minimum when

$$R_i = 0 , \quad i = 1, \ldots , l, \ i \neq j$$

$$R_j = |V_j|$$

The minimum value $n_m$ will be equal to $|V_j|$

Hence
$$\begin{aligned} |E'| &\geqslant |E| + n_m \\ &\geqslant |E| + |V_j| \\ &\geqslant |E| + \operatorname*{Min}_i \{ |V_i| \} \end{aligned}$$

Further the graph $G' = (V', E')$ with $V' = \{V \cup v_x\}$ and $E' = \{E \cup (v_x, V_i)\}$ where $|V_i| = \operatorname*{Min}_j \{ |V_j| \}$ is a $(v+1)$-vertex graph with $l'_{max} = l$.

We now give the algorithm [3] to construct a $v$-vertex minimum edge graph having a specified $k_{min} = k = v - 1$.

Algorithm 2:

*Step 1*: $i = 0$. Let $G^{\circ\circ} = (E^\circ, V^\circ)$ be an $l$-vertex graph with $l^0_{max} = l$. ($G^\circ$ will contain only isolated vertices.) Designate the vertices of $G^\circ$ by $v_1, \ldots, v_l$.

*Step 2*: Construct $G^{i+1} = (E^{i+1}, V^{i+1})$ from $G^j = (E^i, V^i)$ as follows

$$V^{i+1} = V^i \cup v_{l+i+1} \qquad \text{and}$$

$$E^{i+1} = E^i \cup \{(V_{l+i+1}, V^i_k)\} \quad , \quad |V^i_k| = \operatorname*{Min}_j \{ |V^i_j| \}$$

where $V^i_j$ denotes the set of vertices in the $j$th part of $G^i$.

*Step 3*: If $i+1$ equals $k$ go to step 4; otherwise replace $i$ by $i+1$ and return to step 2.

*Step 4*: $G^k$ is the required $v$-vertex minimum edge graph with $k_{min} = k$.

*Theorem 6*: $G^k$ is a $v$-vertex minimum edge graph with $k_{min} = k$.

*Proof*: It follows from Step 2 of Algorithm 2 that if $G^i$ is in $l$ parts with each part complete then $G^{i+1}$ will also have the same property. In view of the choice of $G^\circ$, each $G^i$ constructed as in the algorithm, will be in $l$ parts with each part complete. Further for all $G^i$, $l_{max} = l$.

In view of theorem 5, the $(l+i+1)$-vertex graph $G^{i+1}$ will have minimum number of edges if $G^i$ has minimum number of edges. Since $G^\circ$ is an $l$-vertex graph with minimum number of edges $G^1$ is an $(l+1)$-vertex graph with minimum number of edges. It then follows by induction that $G^k$ is a $v$-vertex minimum edge graph with $k_{min} = k$.

We next proceed to calculate the number of edges in a minimum edge $v$-vertex graph with $k_{\min} = k$. Let

$$v = lr + q \qquad 0 \leqslant q < l \tag{3}$$

where $v$, $l$, $r$ and $q$ are integers.

Then in $G^k$, $(l - q)$ of the complete parts will have $r$ vertices and the remaining $q$ parts will have $(r + 1)$ vertices. Since $G^k$ is a $v$-vertex minimum edge graph with $k_{\min} = k$, it follows that the minimum number of edges required for constructing any $v$-vertex graph having $k_{\min} = k$ is given by

$$\begin{aligned} e_{\min} &= \frac{r(r-1)}{2}(l-q) + \frac{(r+1)r}{2}q \\ &= \frac{(v-q)(v+q-l)}{2l} \end{aligned} \tag{4}$$

If $q = 0$, then

$$e_{\min} = \frac{v(v-l)}{2l} = \frac{vk}{2(v-k)} \tag{5}$$

A minimum edge connected $v$-vertex graph having a $k_{\min} = k$ can be obtained by adding to $G^k$ the edges $(v_1, v_i)$, $i = 2, \ldots, l$.

*Example 3*: It is required to get a minimum edge 14-vertex graph having a $k_{\min} = 10$. We first obtain $l = 4$. $q$ and $r$ can be obtained using (3) as $q = 2$ and $r = 3$. The required 14-vertex graph $G^{10}$ will have 4 complete parts, two of which have 4 vertices each and the remaining two have 3 vertices each. This graph is shown in Fig. 4.

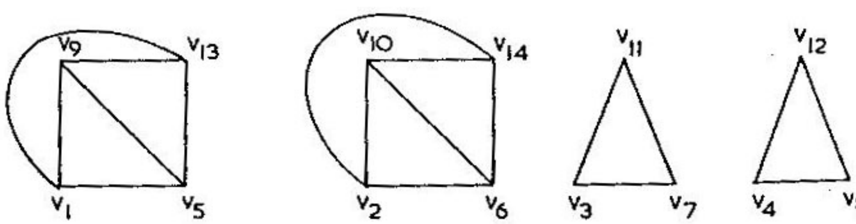


Fig. 4

It may be seen that when $k_{\min} = k < \left[\frac{v}{2}\right]$, $l = v - k \geqslant v - \left[\frac{v}{2}\right] = \left[\frac{v}{2}\right]'$ $\therefore$ Hence $r = 1$ and $q = k$. Then in the minimum edge $v$-vertex graph $G^k$, $k < \left[\frac{v}{2}\right]$, $k$ complete parts will have 2 vertices each and the remaining $(l - k)$ parts will have only one vertex each. The graph obtained by adding to $G^k$ the edges $(v_1, v_i)$, $i = 2, \ldots, l$, will be in the form of a tree.

*Example 4*: Let $v = 19$, $k_{\min} = k = 7$. Hence $k < \left[\frac{v}{2}\right] = 9$. The minimum edge 19-vertex graph $G^T$ is shown in Fig. 5($a$). We make this graph connected by adding edge $(v_1, v_i)$ $i = 2, \ldots, 12$. The new graph which is a tree is shown in Fig. 5($b$).

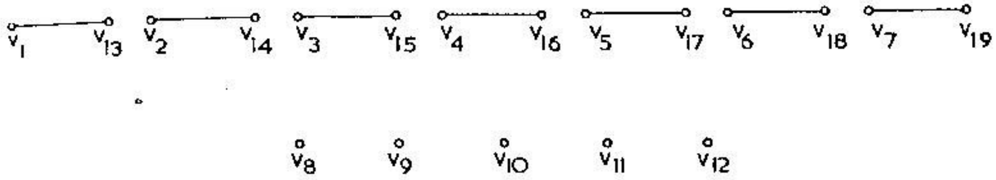


Fig. 5($a$)

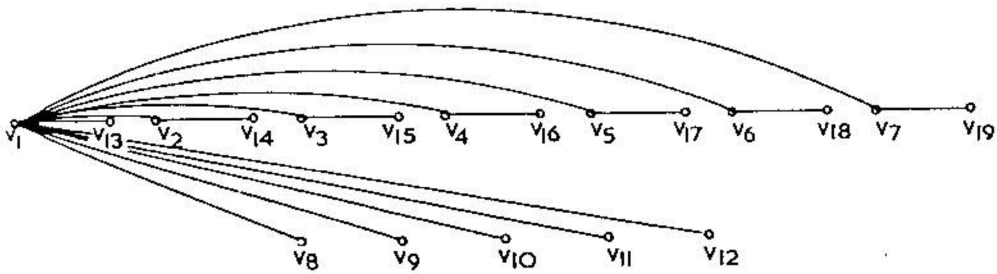


Fig. 5($b$)

## $k$-VULNERABILITY AND DESIGN OF OPTIMALLY $k$-INVULNERABLE COMMUNICATION NETS

In this section we, first, establish an upper bound for $k_{\min}$ for a $v$-vertex, $e$-edge graph. We then define $k$-vulnerability index of communication nets and give a procedure based on algorithm 2 of the last section to design $v$-vertex $e$-edge optimally $k$-invulnerable nets.

*Theorem 7*: For a $v$-vertex, $e$-edge graph

$$k_{\min} \leq \left[\frac{2ev}{2e+v}\right]$$

*Proof*: Let $x = \left[\frac{2ev}{2e+v}\right]^*$ and $y = v - x$. We then get

$$\frac{2ev}{2e+v} \leqslant x = v - y \quad .$$

The above inequality reduces to the following

$$v^2 - vy - 2ey \geqslant 0 \tag{6}$$

We prove the theorem by contradiction.

For a $v$-vertex, $e$-edge graph, let $k_{min} = k = x + s$, $s > 0$ and $l_{max} = l = v - x - s = y - s$. Using the lower bound for $e$ given in (4) we find that

$$e \geqslant \frac{(v - q)(v + q - y + s)}{2(y - s)} \tag{7}$$

where $v = lr + q$ as in equation (3).

Inequality (7) reduces to the following:

$$v^2 - vy - 2ey + 2es + s(v - q) + q(y - q) \leqslant 0 \tag{8}$$

Since $y \geqslant q$ and $v \geqslant q$ and by (6),

$$v^2 - vy - 2ey \geqslant 0$$

It can be seen from (8) that the assumption $k_{min} = \left[\frac{2ev}{2e + v}\right]^* + s$, $s \geqslant 0$ leads to a contradiction, and therefore

$$k_{min} < \left[\frac{2ev}{2e + v}\right]^*$$

or

$$k_{min} \leqslant \left[\frac{2ev}{2e + v}\right]$$

Hence the theorem.

*Definition 3: k-vulnerability index $k_v$*

The $k$-vulnerability index $k_v$ of a graph is defined as equal to the $k_{min}$ of the graph.

*Definition 4: Optimally k-invulnerable graph*

A $v$-vertex, $e$-edge graph G is said to be *optimally k-invulnerable* if it has the maximum possible $k$-vulnerability index. Written mathematically, a $v$-vertex $e$-edge graph G is said to be optimally $k$-invulnerable if, for G, $k_v = k^*$, with $k^*$ satisfying the following conditions

$$e \geqslant e_0(v, k^*)$$

and

$$e < e_0(v, k) \ , \quad k > k^*$$

where

$$e_0(v, k) = \frac{(v - q)(q + k)}{2(v - k)} \tag{9}$$

with $q$ as defined in (3).

(Note: $e_0(v, k)$ is the minimum number of edges of a $v$-vertex graph having a $k_{min} = k$.)

The following algorithm, based on algorithm 2, may be used to design optimally $k$-invulnerable communication nets having $v$-vertices and $e$-edges.

Algorithm 3:

*Step 1*: Find $k^*$ satisfying the following conditions

$$k^* \leqslant \left[\frac{2ev}{2e+v}\right]$$

$$e \geqslant e_0(v, k^*)$$

and $$e < e_0(v, k) \ , \quad k > k^*$$

*Step 2*: Using Algorithm 2, obtain the minimum edge, $v$-vertex graph $G^{k^*}$ having $k_{min} = k^*$
(Note: Vertex numbering of $G^{k^*}$ as given in Algorithm 2 is retained.)

*Step 3*: Add $(e - e_0(v, k^*))$ edges to $G^{k^*}$ arbitrarily making sure that no edge is added between vertices $v_i$ and $v_j$ whenever $i, j \in \{1, \ldots, l\}$.

*Step 4*: The $v$-vertex, $e$-edge graph $G$ that results from Step 3 is optimally $k$-invulnerable.

*Example 5*: It is required to design an optimally $k$-invulnerable communication network having 14 vertices and 25 edges. From step 1 of algorithm 3, we find $k^* = 10$. The 14 vertex minimum edge graph $G^{10}$ having $k_{min} = 10$ is shown in Fig. 4.

For $v = 14$ and $k^* = 10$, $e_0(v, k^*) = 18$. Adding 7 more edges to the graph shown in Fig. 4, taking the precaution given in Step 3 of Algorithm 3, we get the required optimally $k$-invulnerable graph shown in Fig. 6.

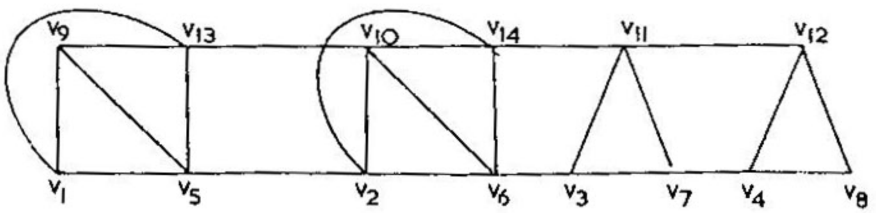


Fig. 6

*Example 6*: In Example 5, we found that

$$k^* = \left[\frac{2ev}{2e+v}\right]$$

We now give an example in which $k^* < \left[\frac{2ev}{2e+v}\right]$. For $v = 28$ and $e = 35$, $\left[\frac{2ev}{2e+v}\right] = 20$.

Since $e_0(v, 20) = 36 > 35$, and $e_0(v, 19) = 30 < 35$, $k^* = 19$ for this case.

## CONCLUSION *

In this paper we have established several results useful in the vulnerability studies of communication nets, based on the properties of $K$-sets of a graph. We have also introduced the concept of $k$-vulnerability of communication nets and have given a procedure to design optimally $k$-invulnerable communication nets. An algorithm is also given for the generation of a $K_{\min}$-set of a graph $G_T$.

We observe that while the results of this paper are useful in vulnerability studies, they also find application in the studies of other aspects of communication nets. For instance, in locating the maintenance and control personnel of a communication system, one may be interested in finding the minimum number of stations from where the personnel will have directs access to all the links. This problem can be handled by finding a $K_{\min}$-set of the graph representing the system. The algorithm 1 discussed in the third section, to generate a $K_{\min}$-set of a graph $G_T$ will find application in an irrigation system whose model is usually in the form of a graph $G_T$.

## REFERENCES

1. *H. Frank and I. T. Frisch* Communication, Transmission and Transportation Network, Addison-Wesley Publishing Company, Reading, Massachusetts, U.S.A. 1971.

2. *F. Harary* Graph Theory, Addison-Wesley Publishing Company, Reading, Massachusetts, U.S.A. 1969.

3. *O. Ore* Theory of Graphs, American Mathematical Society Colloquium Publications, Vol. XXXVIII, 1962.