Algorithms For Finding Liar Domination Number For Fuzzy Path And Fuzzy Cycle

S. Roseline Mary S. Ruban Raj J. Maria Joseph

Abstract: We define the Liar's Domination Set for fuzzy graphs using the membership value of strong arcs as follows. Let $G =< \sigma, \mu >$ be a fuzzy graph on V. Let the node t be dominated by the node s if $t \in N[s]$, where $N[s] = \{s\} \cup \{r \in V(G)/\mu^{\infty}(r,s) = \mu(r,s)\}$. $D \subseteq V(G)$ is said to be Liar's Domination Set if it satisfies the following two conditions. Every $t \in V(G)$is dominated by at least two nodes in D. For Every two nodes $t, s \in V, t + s$ is dominated by least three nodes in D. The minimum fuzzy cardinality of Liar's Domination Set is known as Liar's Domination Number. In this paper we have given the algorithms for finding Liar's Domination number of Fuzzy paths and Fuzzy Cycles.

Index Terms : Fuzzy Path, Fuzzy Cycle, Fuzzy Line Graph, Liar Domination Set, Liar Domination Number, Strong Fuzzy Graph, Weakest arc.

1 INTRODUCTION

Fuzzy graph theory plays vital role in various fields like clustering analysis, database theory, network analysis, Information theory, etc. [1] Fuzzy model can be used in problems handling uncertainty to get more accurate and precise solutions. [2] In 1975 Rosenfeld presented the notion of fuzzy graph and some fuzzy analogues of graph theoretical concepts such as Paths, Cycle and Connectedness. In 1977, Batacharya [3] and in 1989 Bhutani [4] explored the concept of Fuzzy Automorphism groups. In 1993, Moderson [5] introduced the Concept of fuzzy line graphs and developed its basic properties. Moderson and Nair [6] discussed cycles and co-cycles of fuzzy graphs. In 1998 and 2004, A. Somasundaram and S. Somasundaram [7] discussed the concepts of domination in fuzzy graphs. A. Nagoorgani and Ahamad [8] investigated strong and weak domination in Fuzzy Graphs. A. Nagoorgani and V. T. Chandrasekaran [9] introduced domination in fuzzy graphs using Strong Arc. P.J. Slater and M. L. Roden [10] and [11] introduced the notion of liar's dominating set in graph theory. B. Balandra and B.Canoy [12] characterized the p-liar's dominating sets in the composition of graphs. B.S. Banda and S. Paul [13] strengthened the complexity results of liar's domination decision problem. B. Balandra and B. Canoy [14] studied the concept of liar's domination in the join, carona and lexicographic product of graphs. Durgan and Altundag [15] have found liar domination number for middle graphs of some specific graphs. Let G be a fuzzy graph which can be used to model a system with each node t in $V(G)$ representing an area of the facility such as a room, hallway, or ventilation duct. Likewise, fuzzy graph G can indicate a computer network where each node $t \in V(G)$ indicates a processor. Edges of G can connect nodes indicating adjacent areas of the systems or processors with direct edges. A system or a processor will be recognized with the node that indicates it. Systems are subject to having an "intruder" such as a fire, saboteur, or thief that must be detected and have its location precisely recognized.

• S. Roseline Mary, Research Scholar of St. Joseph's College (Affiliated to Bharadhidasan University), Trichy.
• S. Ruban Raj, Associate Professor, St. Joseph's College (Affiliated to Bharadhidasan University), Trichy.
• J. Maria Joseph, Assistant Professor, St. Joseph's College (Affiliated to Bharadhidasan University), Trichy.

In this method an intruder must be located in a multiprocessor system. In general, it is assumed that the probable locations for one intruder are all of the nodes in $V(G)$. $N(t) = \{s \in V \,|\mu^{\infty}(t,s) = \mu(t,s), where\; t \in V(G)\}$ is called the neighbourhood of t and $N[t] = N(t) \cup \{t\}$. It is assumed that a protection device placed at a node s can detect the existence of an intruder exactly when the intruder is in $N[s]$. When a protection device at node s can identify the presence of an intruder at s or at a node in $N(s)$, but which node in $N(s)$ cannot be located, then one is involved in having a locating-dominating set. This set is called liar's domination set. That is, liar's domination set provides a single fault tolerant protection placement set where a location of fault device could be recognized. We have introduced liar's domination for fuzzy graphs [16]. In this paper we have given algorithms for finding liar's domination number for fuzzy paths and fuzzy cycles.

2 THEOREM 1

For all fuzzy paths of order n, $4k + 3 \leq n \leq 4k + 6, where\; k = 1,2,3,\ldots, \gamma(P_n) \leq \left\lceil \frac{3}{4}(n + 1) \right\rceil$

2.1 Proof

Let $v_1, v_2 \ldots v_{(n-1)}, v_n$ be the vertices of P_n $N[v_1] = \{v_1, v_2\}$. Since $|N[v_i] \cap D| \geq 2, \forall v_i \in V, v_1, v_2$ must be in D. $N[v_3] = \{v_2, v_3, v_4\}$. Since $|N[v_i] \cup N[v_j] \cap D| \geq 3$, for every two vertices of V, v_3 must be in D. Therefore $D = \{v_1, v_2, v_3\}$. Similarly we can include $\{v_n, v_{(n-1)}, v_{(n-2)}\} \in D$. Therefore $D = \{v_1, v_2, v_3, v_n, v_{(n-1)}, v_{(n-2)}\}$. v is the vertex of $V_k\; where\; V_k = \{v_k, k = 4,8, \ldots (n-3)\}$ v_k can be non-member of D if $v_{(k-1)}, v_{(k-2)}, v_{(k+1)}, v_{(k+2)} \in D$, since $|N[v_k] \cap D| \geq 2$ and $|N[v_i] \cup N[v_j] \cap D| \geq 3$ where $k' = 1,2 \ldots k - 2, k - 1, k + 1, k + 2$. Therefore $v_k \notin D$.

By using the definition, if $v_{(k+1)}, v_{(k+2)}$ are members of D, then $v_{(k+3)}$ must be member of D. Continuing this process one can identify that, When $k = 1, 4k + 3 \leq n \leq 4k + 6$, maximum one vertex is non-member of D. When $k = 2, 4k + 3 \leq n \leq 4k + 6$, maximum two vertices are non-member of D. When $k = 3, 4k + 3 \leq n \leq 4k + 6$, maximum three vertices are non-member of D and so on. It follows that $\gamma(Pn) \leq \left\lceil \frac{3}{4}(n + 1) \right\rceil$

2.2 Example

Consider a Fuzzy Path having 23 nodes. It is shown in the F

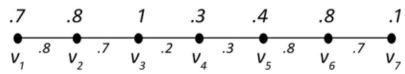

Fig. 3. *Example for Strong Fuzzy Paths*

$$LD\ set = \{v_1, v_2, v_3, v_4, v_5, v_7, v_8, v_9, v_{11}, v_{12}, v_{13}, v_{15}, v_{16}, v_{17}, v_{19}, v_{20}, v_{21}, v_{22}, v_{23}\}$$

$$\gamma = 1 + 0.1 + 0.2 + 0.4 + 0.7 + 0.9 + 0.1 + 0.7 + 0.5 + 0.3 + 0.2 + 0.7 + 0.5 + 0.4 + 0.7 + 0.2 + 0.1 + 0.7 + 0.9 = 9.3$$

2.3 Corollary

For fuzzy line graphs of all fuzzy paths, $4k + 3 \leq n \leq 4k + 6$ where $k = 1,2,3\ldots$, liar's Domination number $\gamma \leq \left\lceil \frac{3}{4} n \right\rceil$.

2.4 Proof

Fuzzy line graph of a fuzzy path P_n is the fuzzy path having $n - 1$ vertices. From the previous theorem, $\gamma(P_{(n-1)}) \leq \left\lceil \frac{3}{4} n \right\rceil$. Therefore, liar's domination number of fuzzy line graphs of fuzzy path is $\leq \left\lceil \frac{3}{4} n \right\rceil$.

2.5 Proposition:

For fuzzy Cycle $C_n, \gamma(C_n) \leq \left\lceil \frac{3}{4} n \right\rceil$

2.6 Example

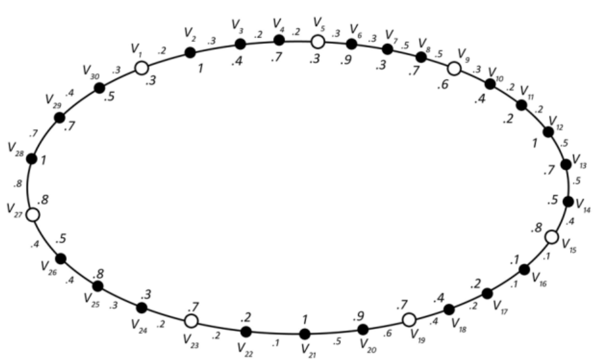

Fig. 2. Example for Fuzzy Cycle

$\gamma = 13.4$

3 THEOREM 2

All fuzzy paths are strong.

3.1 Proof:

fuzzy path is defined as the sequence of nodes $u_1, u_2, \ldots u_n$ such that $\mu(u_i, u_{i+1}) > 0, \forall i = 1,2, \ldots n - 1$. A fuzzy graph is said to be strong if all of its edges are strong edges. An arc is said to be strong, if $\mu^\infty(u, v) = \mu(u, v)$, where $u, v \in V$. Since $\mu^\infty(u_i, u_{i+1}) = \mu(u_i, u_{i+1})$ for all $u_i = 1,2 \ldots n - 1$ in a fuzzy path, all fuzzy paths are strong.

3.2 Example:

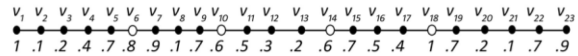

Fig. 1. *Example for Fuzzy Path*

$\mu^\infty(v_i, v_{i+1}) = \mu(v_i, v_{i+1}) \ldots$

Therefore, this fuzzy path is strong.

4 THEOREM 3

All fuzzy cycles are strong.

4.1 Proof:

A fuzzy cycle is a path with $v_0 = v_n, n \geq 3$ and having more than one weakest arc, weakest arc is the arc having least membership value. If there exist more than one weakest in a fuzzy cycle, then all of its edges are strong.

Therefore, all fuzzy cycles are strong.

4.2 Example:

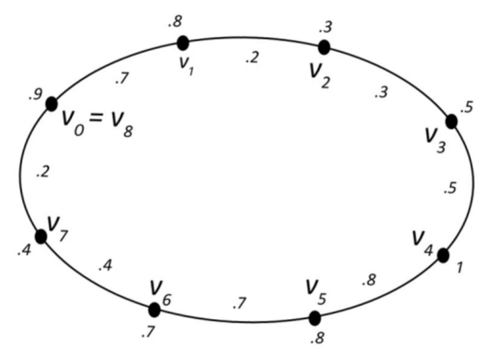

Fig. 4. *Example for Strong Fuzzy Cycle*

$$\mu^\infty(v_i, v_{i+1}) = \mu(v_i, v_{i+1})\ \forall\ v_i \in V, i = 1,2, \ldots 7$$
$$\text{Here } \mu^\infty(v_1, v_2) = \mu^\infty(v_1, v_2) = 0.2$$

This fuzzy cycle is strong. The following are the algorithms of finding liar's domination numbers of fuzzy path and fuzzy cycle. This algorithm is verified using python programming language and the outputs are also shown below the algorithms.

Algorithm 1: Liar Domination Number for a Fuzzy Path

```
Input     : Membership values for Nodes in a Fuzzy Path (FP)
Output : Fuzzy Liars Domination Numbers
begin
P_set = input from user
LD_set : Liar Domination Set
NLD_set : Non Member of LD set
i = 0
size = Size(P_set)
procedure sum_Vmember(num1, list1, n)
    while num1 <= Size – 4 do
        n = n + list1[num1]
        num1 = num1 + 4
        return (n)
```

```
procedure add_LD(num)
   while num <= Size – 4 do
      LD_set.append(P_set[num])
      num = num + 4
procedure add_NLD(num)
   while num <= Size – 4 do
      NLD_set.append(P_set[num])
      num = num + 4
if Size >= 7 then
   LD_set.append(P_set[i])
   LD_set.append(P_set[i+1])
   LD_set.append(P_set[i+2])
   LD_set.append(P_set[Size-1])
   LD_set.append(P_set[Size-2])
   LD_set.append(P_set[Size-3])
   ch1 = i + 3
   ch2 = i + 4
   ch3 = i + 5
   ch4 = i + 6

   Total1 = P_set[ch1]
   Total2 = P_set[ch2]
   Total3 = P_set[ch3]
   Total4 = P_set[ch4]

   Total1 = sum_Vmember(ch1, P_set, Total1)
   Total2 = sum_Vmember(ch2, P_set, Total2)
   Total3 = sum_Vmember(ch3, P_set, Total3)
   Total4 = sum_Vmember(ch4, P_set, Total4)
   if (Total1 > Total2) and (Total1 > Total3) and (Total1 > Total4) then
      add_NLD(ch1)
      add_LD(ch2)
      add_LD(ch3)
      add_LD(ch4)
   elseif (Total2 > Total1) and (Total2 > Total3) and (Total2 > Total4)
then
      add_NLD(ch2)
      add_LD(ch1)
      add_LD(ch3)
      add_LD(ch4)
   elseif (Total3 > Total1) and (Total3 > Total2) and (Total3 > Total4)
then
      add_NLD(ch3)
      add_LD(ch1)
      add_LD(ch2)
      add_LD(ch4)
   else
      add_NLD(ch4)
      add_LD(ch1)
      add_LD(ch2)
      add_LD(ch3)
elseif Size >= 3 and Size <=6 then
   for i in P_set do
      LD_set.append(i)
else
   print("Path must have 3 or more than 3 Vertices")
print("your LD set is:",LD_set)
print("your NLD set is:",NLD_set)
```

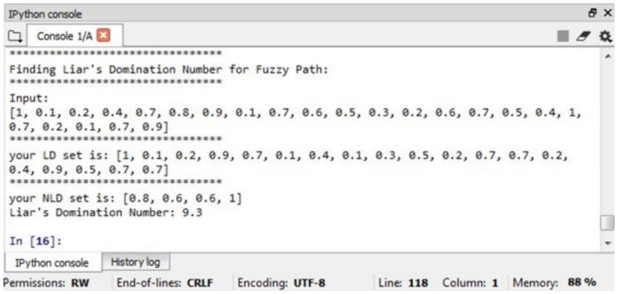

Fig. 5. *Output for Fuzzy Path Algorithm.*

Algorithm 2: Liar Domination Number for a Fuzzy Cycle

```
Input : Membership values for Nodes in a Fuzzy Cycle (FC)
Output : Fuzzy Liars Domination Numbers
Begin
FC_Set = input from user
LD_Set : Liar Domination Set
NLD_Set : Non Member of LD Set
i = 0
size = SIZE(FC_set)
procedure getCount(num1, num2)
i = -1
while num1 != num2 do
    i = i + 1
    num1 = (num1 + 1) % size
return(i)
procedure check_max(list1, N)
for x in list1 do
   if val>= x:
         return False
return True
If size > 3 then
         for i to size do
            Total_FC_set = FC_set[i] + FC_set[(i + 1) % size] +
         FC_set[(i + 2) % size]
         end
         location = Total_FC_set.index(min(Total_FC_set))
         LD_set ← FC_set[location] , FC_set[(location + 1) % size],
         FC_set[(location + 2) % size]
         Front_loc  =  (location + 3) % size
         Back_loc   =  (location - 1) % size
         while getCount(front_loc , rev_loc) != 0, 1, 2, 3 or 4 do
            if FC_set[front_loc] >= FC_set[rev_loc] then
               NLD_set ← FC_set[front_loc]
               LD_set ← FC_set[(front_loc + 1) % size]
               FC_set[(front_loc + 2) % size]
               FC_set[(front_loc + 3) % size]
               Front_loc = (front_loc + 4) % size
            else
               NLD_set ← FC_set[rev_loc]
               LD_set ← FC_set[(rev_loc - 1) % size]
               FC_set[(rev_loc - 2) % size]
               FC_set[(rev_loc - 3) % size]
               rev_loc = (rev_loc - 4) % size
         if getCount(front_loc , rev_loc) == 3 then
               N = FC_set[front_loc] + FC_set[rev_loc]
               temp_set ← FC_set[front_loc + 1]
               FC_set[(front_loc + 2) % size]
               FC_set[(front_loc + 3) % size]
               if check_max(temp_set,N) then
                    LD_set ← FC_set[front_loc]
                    LD_set ← FC_set[rev_loc]
                    NLD_set ← max(temp_set)
                    Temp_set.remove(max(temp_set))
                    LD_set ← remaining nodes of temp_set
               else
                    NLD_set ← FC_set[front_loc]
                    NLD_set ← FC_set[rev_loc]
                    LD_set   ← temp_set
         elseif getCount(front_loc , rev_loc) == 2 then
            temp_set ← FC_set[front_loc]
            FC_set[(front_loc + 1) % size]
            FC_set[(front_loc + 2) % size]
            FC_set[(front_loc + 3) % size]
             NLD_set ← max(temp_set)
             LD_set ← remaining nodes of temp_set
         elseif getCount(front_loc , rev_loc) == 1 then
             temp_set ← FC_set[front_loc]
             FC_set[(front_loc + 1) % size]
             FC_set[(front_loc + 2) % size]
             NLD_set ← max(temp_set)
             LD_set ← remaining nodes of temp_set
         elseif getCount(front_loc , rev_loc) == 0 then
            temp_set ← FC_set[front_loc]
            FC_set[(front_loc + 1) % size]
```

NLD_set ← max(temp_set)
LD_set ← remaining nodes of temp_set
elseif size == 3 then
LD_set ← FC_set
else
Set Must have minimum 3 vertices
End

4 EQUATIONS

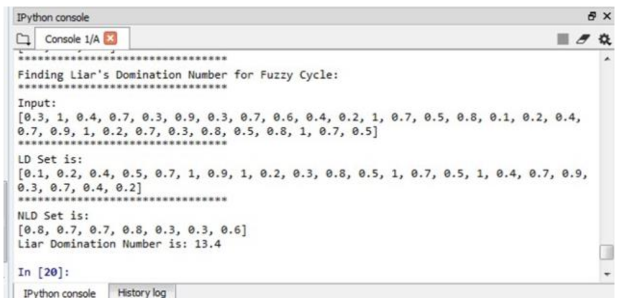

Fig. 6. *Output for Fuzzy Cycle Algorithm.*

5 CONCLUSION

Liar's dominating set is the set which helps to identify and report intruder's location when the intruder presents in a computer network. This is recent and very useful concept to control fault tolerance in a computer network. We have given the algorithms for finding liar's domination number for fuzzy paths and fuzzy cycles. In our next paper we are going to find the upper bound for the liar's domination number of some specific fuzzy graphs.

6 REFERENCES

[1] J. N Moderson., P.S. Nair., Fuzzy graphs and Fuzzy Hypergraphs. Physica, Verlag 2000.
[2] L. A. Zadeh., Fuzzy Sets, Information and Control. 8 (1965), 338 – 353.
[3] Batacharya P., Some Remarks on Fuzzy Graphs. Pattern Recognition Letters, 6, (1987), 297 – 302
[4] Bhutani K. R., On Auto morphism on Fuzzy Graphs. Pattern Recognition Letters 9 (1989), 159 – 162.
[5] J.N. Moderson., Fuzzy Line Graphs. Pattern Recognition Letter, 14 (1993), 391 – 384.
[6] J.N. Moderson and P.F. Nair. Cycles and Co Cycles of Fuzzy Graphs. Information Science, 90, 39 – 49
[7] A somasundaram., S. Somasundaram., Domination in Fuzzy Graphs. I, Pattern Recognition Letters, 19 (1998), 787 – 791.
[8] A. Nagoor Gani., M. Basheer Ahamed., East Asian Mathematical Journal. Strong and Weak Domination in Fuzzy Graphs. Vol. 23. 1-8. 10.13140/2.1.1497.1528
[9] Nagoorgani., V. T. Chandrasekaran., Domination in Fuzzy Graphs Advances in Fuzzy Sets and Systems, 2006, 17 – 26.
[10] P. J. Slater., Liar's domination Networks, 54 (2009), 70–74.
[11] P. J. Slater., M. L. Roden., Liar's domination in graphs. Discrete Mathematics, 309 (2008), 5884–5890.
[12] Carlito B. Balandra., Sergio R. Canoy., Another Look at p-liar's Domination in Graphs. International Journal of Mathematical Analysis, Vol. 10, 201, no. 5, 213 – 221.
[13] B.S. Banda., S. Paul., Liar's Domination in graphs: Complexity and algorithm. Discrete Applied Mathematics, 11 (2013), 1085 – 1092.
[14] Carlito B. Balandra., Sergio R. Canoy., Liar's Domination in Graphs Under Some Operations. Tamkang Journalof Mathematics, Volume 48, Numer 1,45 – 59, March 2017.
[15] Derya Dogan Durgan., Ferhan Nihan Altundag., Liar's Domination in Middle Graphs. Bulletin of the International Mathematical Virtual Institute, ISSN (p) 2303 – 87, ISSN (o) 2303 – 4955, Vol. 7 (2017), 407 – 415.
[16] S. Roseline Mary., S. Ruban Raj., Liar's Domination on Fuzzy Graphs. Journal of Applied Science and Computations, ISSN : 1076 5131.