

Estimation of all-terminal network reliability using press forward GA with stochastic selection method

^[1] Anuradha, ^[2] A.K. Solanki, ^[3] Harish Kumar Taluja

^{[1][3]} Department of Computer Engineering & Engineering, Meerut Institute of Engineering & Technology, Meerut (India)

^[2] Department of Computer science & Engineering, Bundelkhand Institute of Engineering & Technology, Jhansi (India)

^[1] anutaluja@gmail.com, ^[3] harishtaluja@gmail.com

^[2] solankibiet13@gmail.com

Article History: Received: 10 January 2021; Revised: 12 February 2021; Accepted: 27 March 2021; Published online: 20 April 2021

Abstract: Network reliability is an important measure of how well the network meets its design aim and mathematically is the probability that a network will perform communication efficiently for at least a given period of time. The reliability analysis is used for determining the probability of correct functioning of a system in terrain conditions. Estimating the all-terminal network reliability is an NP-hard problem, the press forward Genetic Algorithm is also an alternative to solve the problem of all-terminal network reliability. This paper presents a reliability evaluation method by using a feed-forward genetic algorithm with a random selection of nodes and links i.e., estimate the network reliability such that the flow is not less than a given threshold. To verify the efficiency of an algorithm, various sets of runs are applied to identify the most reliable network and the algorithm is tested on the network size of 2,4,6,8, -----256 nodes with link $N(N-1)/2$. The results of the algorithm are obtained for all-terminal reliability problem based on close analysis of the complexity of the network. Our assumption for reliability evaluation of node 1 to node N, with link reliability of 50%. Another result of the proposed GA approach is the significant reduction in computational time.

Index Terms— ANN, CRR, DNC, GA, MCS, press forward.

I. INTRODUCTION

The computer network can be called a stochastic network, containing nodes and links. The design of a reliable computer network is a significant problem due to complete failure, limited failure, and alimentation requirements. Network performance is evaluated by routing of data packets along the network path from source to destination. Network links are functional with their associated probability and successful exchange of information is to determine the probability of having at least one functional link between source and destination. Generally, designers may be interested in the flow of the network from one node to another. A transportation network (computer network) can be modelled as N nodes, L links, assumes that each link and node have an associated probability of failure and success and the reliability is calculated as the probability that a computer network is operational for transmitting and receiving data [7-8]. The network design must be in such a way that components are placed at their best-fit position so that it minimizes the cost while reaching a criterion that indicates performance. The performance criteria may be transmission delay, throughput, secure communication, and reliability. Therefore, network reliability is described as the probability that k units of a network can successfully transmit data from S_n to T_n . Consider a computer network as Graph $G(V, E)$ that comprises a set of nodes and links represented by V and E . This paper is focused on all-terminal network reliability can be described as probability that node pair $(v_i, v_j \in V)$ communicate with each other. Each link must have a probability of failure (P_f).

$$P_f(i) = \frac{n_{failure(i)}}{Ot_{all(i)}} \quad \{\text{Node Failure}\}$$

$$P_f(ij) = \frac{n_{failure(e_{ij})}}{Ot_{all(e_{ij})}} \quad \{\text{Edge Failure}\}$$

Destination or Target may be one or more than one, the main emphasis of the article is to evaluate reliability between source and destination. Network Reliability occurs only when there exists at least one path between source and destination nodes without any failure link or with a link probability $\geq$ threshold value ($L_p \geq T_h$). The main design approach focuses on different strategies such as maximizing network reliability, commodity allocation, network topology selection and flow

assignment resource allotment [1,2,4,5,6]. The optimal design problem can be devised as a combinatorial problem where the selection of devices plays an important role in maximizing reliability or minimizing cost. The reliability definition is depending on which nodes are operational and contribute in reliability evaluation [1,2,3,4]. For instance, a node may fail during a terrain condition and affect the performance, reliability of the network. The overall reliability measurement of the growing size computer network is NP- hard problem, so it requires more computation effort for reliability calculation with growing network size. Nicholson and Du (1994) suggest some methods to improve reliability are as follows:

1. Replace weak bridges (provide an alternate path with respect to articulation vertex).
2. Constructing new links.
3. Use emergency air-ferry services in case of failure (Protection path).
4. Regular monitoring of traitors' components.
5. Regular preventive maintenance.

Factor influencing the system reliability are shown in Fig 1 and 2 based on the properties of network components which are further divided into various part based on measurement.

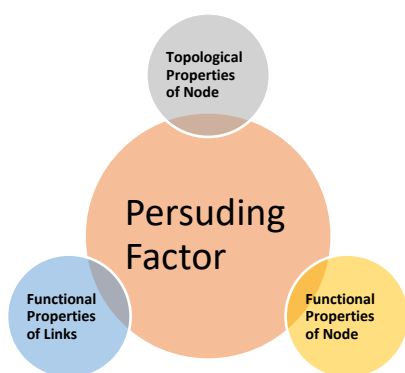
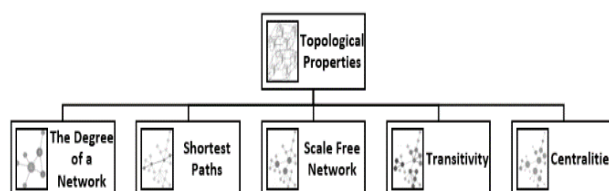
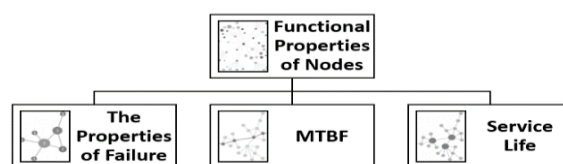


Fig I: Influencing factors on network reliability



(a)



(b)

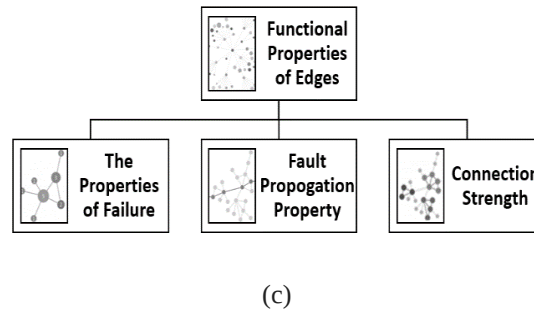


Fig II (a, b, c): Influencing factors on network reliability based on measurement.

Table I: Comparison of network reliability measures.

	Failure Propagation	Property of Nodes	Property of Edges	Connectivity	Threshold of System Reliability
Terminal Reliability	N	N	N	Y	N
Efficiency Reliability	N	N	Y	Y	N
System Reliability	Y	Y	Y	Y	Y

Reliability measures considering both connectivity and capacity called capacity related reliability (CRR), requires some techniques for reliability calculation. This technique may include path set or cut set evaluation with building a branching tree with disjoint tree nodes [1-5, 11-17]. The problem has been studied in this paper includes branch-and-bound and heuristic methods. This paper focuses on the press forward genetic algorithm approach for reliability evaluation of all-terminal networks using a random selection method [1]. The approach used in this paper includes a network of the size of 256 nodes. The article is organized as follows, Section 2 introduces the problem statement, Section 3 introduces various reliability evaluation methods, Section 4 proposes solution algorithm that contains initial population, genetic algorithm parameters, fitness function and proposed pseudocode, Section 5 includes results and conclusion and future aspect are introduced in section 6.

Importance: Reliability estimation is an important factor for the designing of computer network and this problem is well studied and presented by various researchers [26]. We considered a computer network of our college which is similar to a transit network consists of more than 1500 nodes and situated in 7 different blocks. This network is used for the experimental purpose in this paper. A similar result will be used for a big network of a city like ARPANET. In our paper we had use functional probability $\Pr(l)$ and failure probability is $P_f = 1 - \Pr(l)$.

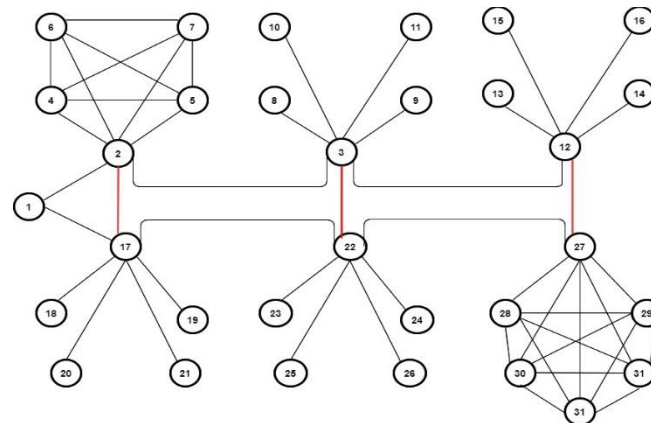


Fig III: Network design of Block 1 main building

Suppose that node 1 wants to send some data to node 8 or node 11, node 1 checks to see if it can send the data through a path in the network that

- (i) Consists only of functional edges and functional nodes without any articulation vertex
- (ii) Starts at node1 and ends at 8 or 11.

If such a path exists, then mode1 transmits the data through network. If node 1 is unable to send the data, it has to check all the available alternate paths. The selection of another path increases the cost, time of transmission and reduces the reliability of the network. Sudden selection of alternate path is more likely to be more expensive. Therefore, from one to another node transmission, the system must record the second shortest path between any specified pair of nodes and set it as an alternate path or protection path as shown in Fig 4 [22] [26][27][28][29].

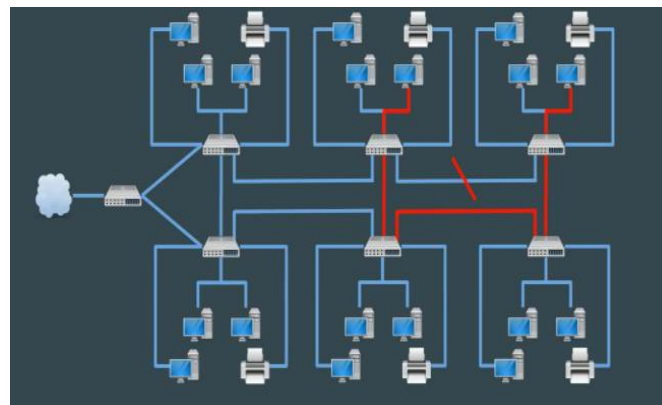


Fig IV: Network design of LAB NO 1

Our algorithm emphasis the reliability evaluation before selecting path 1(preferable path) and after selecting an alternate path whether it is worth the investment. This process continues even after the addition or deletion of any node, encourages us to design more efficient algorithms that can evaluate the reliability of variable size network with press forward technique.

Stochastic selection method having an important feature that they can carry out broad search of the design space and thus avoid local optima. This helps for finding the best reliable solution.

Notation Used: The following notations are used to evaluate the network reliability

1. Network is denoted by a graph G having a set of nodes and links represented by V and E . Nodes and links are of variable size, path is represented by v_i and v_j by an ordered pair (v_i, v_j) where $(v_i, v_j \in V)$ and path length must be ≥ 1 (indicates at least there must be 1 link(edge) is available between any specified pair of vertices) .
2. A network reliability is represented by a set of the following
 $[G, v_s, v_t, Pr(l), Pr_a(l)]$

where

G : Network Graph

v_s : Source Vertex

v_t : Terminal Vertex

$Pr(l)$: Functional Probability

$Pr_a(l)$: Functional Probability of Alternate Path *

* is optional

3. Functional Graph G^f is a sub-graph of G consist of all operable links and nodes.

$[G^f, v_s, v_t, Pr(l) : \text{Functional Graph}]$

4. A hyper tree or tree is represented by G^{fh} means no cycle is found in the network.

$[G^{fh}, v_s, v_t, Pr(l) : \text{Functional Tree Graph}]$

Note: Tree means alternate paths are not available otherwise it forms a cycle

5. C_{ij} : Cost of a link or cost between v_i, v_j
6. $X_{i,j}$: is the decision variable either 0 or 1.
7. $Rel_{(Current)}$: Reliability of Component.
8. fes : Feasibility index which improves component's reliability.
9. I_M : Incident Matrix
10. A_M : Adjacency Matrix

Factors affecting System reliability of Computer Network:

Computer network reliability can be classified as follows:

- (i) **Terminal reliability:** It is the probability of a transferring message successfully between the v_s to v_d (Source node to destination node) and it includes two-terminal reliability, k-terminal reliability, and all-terminal reliability.
- (ii) **Percolation reliability:** It is the reliability of estimating that failure of how many nodes leads to the failure of the system.
- (iii) **Efficiency reliability:** It is the probability of checking how many nodes and links are fully functional, how long they can, and how much the system is fault-tolerant.

Assumption Used: To do analysis simply and to concentrate on the key performance indicators, some assumptions are used which are as follows.

- (i) The nodes (source node and destination node) are constant and known
- (ii) The nodes are considered connected and unique capacity is present in each node.
- (iii) The data transmission is the same along with the neighbours of each node.
- (iv) The capacity of every link is binary, which means the link either exists or does not exist.
- (v) All the components used in the network are reliable and work properly with maximum throughput.
- (vi) Failure of the link is the only reason for network failure.
- (vii) Link failure probability is independent.
- (viii) If a link is repaired once in a lifetime it cannot be reusable.
- (ix) Links can be a move to non-operation mode any time.

Degree of each node $\deg(i, j) \geq 2$.

II. PROBLEM DEFINITION

Network designing is a big and challenging area and the main problem of network design is to achieve maximum reliability. Let N be a set Nodes and $L = \{L_i | i = 1, 2, \dots, n\}$ represent a set of n links, where L_i is the i^{th} arc. Network of computer is hence denoted by (N, L) and includes a set of k -MPs, $\{P_1, P_2, \dots, P_k\}$, where a path is a set of links connecting v_s to v_d and an MP is a path whose proper subsets are no longer ones. Only a few algorithms are available to compute all-terminal reliability in polynomial time but all these algorithms having some difficulty. To identify the best design of a reliable

network, we need to first examine the network design for the simplest undirected and unweighted graph with no redundant links between two nodes. The Optimization problem of the network is defined by equation 1

$$\text{Min } O(X) = \sum_{i=1}^{N-1} \sum_{j=i+1}^N C_{ij} X_{ij} \quad (1)$$

Where N = number of components or sites(computers)/nodes, link represent communication links, C_{ij} is cost between i and j, X_{ij} is the decision variable either 0 or 1 ($X_{ij} \in \{0,1\}$), O is the objective function and X is the link topology variable.

Assume that $Rel(x) \geq Rel_0$

Each node and each link have an associative probability of success and failure. Usually, n-type approaches are used by various authors to achieve maximum reliability goals like Monte Carlo Simulation, Particle Swarm optimization, Upper and Lower bounds of reliability, and simulation-based reliability estimation [15-18]. The reliability measurement is an important factor for optimal network design. The main objective is to design a network by selecting the best functioning paths and links for communication so that maximum reliability can be achieved with reduced cost. In the general case, reliability $\propto$ cost.

Minimum: $C(X)$

Cost Function C_i in term of the reliability R_i of each node or component is calculated as:

$$C_i = f(R_i) \quad (2)$$

This Function represents the current reliability of the component, $Rel_{(Current)}$ with maximum possible component reliability $Rel_{(Max)}$. This used for calculating reliability at a different level but having some limitations like

- Network design issues.
- State of technology.

For the exact cost estimation following condition must be followed:

- The reliability must be lie between $Rel_{(Current)} \leq R(X) \leq 1$ or at least $R(X) \geq 0$.
- The Linear line is not acceptable, but it guarantees the fact it is incrementally harder to improve reliability.
- Only asymptotic functions are acceptable for maximum achievable reliability.
- Cost Function/penalty functions are used for increasing the reliability of the components. The cost function is calculated as

$$C_i R_i = e^{1 - fes * \frac{R_i - R_{min,i}}{R_{max,i} - R_i}} \quad (3)$$

$C_i R_i$: cost function

fes : Feasibility index which improves.
component's reliability

$R_{min,i}$: current reliability.

$R_{max,i}$: maximum achievable reliability.

In this paper, the data set is divided into small multiple sets, the algorithm is applied on one set (randomly selected) but results are cross verified randomly on all sets.

III. CALCULATING AVERAGE RELIABILITY OF ALL-TERMINAL GRAPH

The all-terminal reliability of a connected graph, G on n nodes and l links (i.e., a graph of order n and size m) can be expressed.

$$Rel(G, P) = \sum_{i=0}^{m-n+1} F_i p^{m-1} (1-p)^i \quad (4)$$

Where F_i denotes the no of subsets of links whose removal from the G leaves the graph connected. The Average reliability of graph is calculated by using Bernstein formula [8][9][10][11]:

$$avgRel(G) = \frac{1}{m+1} \sum_{i=0}^{m-n+1} \frac{F_i}{\binom{m}{i}} \quad (5)$$

The reliability of connected graph is shown in table 2.

Table II: Reliability of Connected graph

Node	Link	Reliability of Graph
4	4	0.640
4	5	0.790
5	5	0.530
5	6	0.703
6	6	0.420
6	7	0.624
6	8	0.729
6	9	0.853
6	10	0.890
6	11	0.926
6	12	0.959

The main responsibility of a reliability engineers is to achieve minimum required reliability either by improving the reliability of component or components of such system. Generally, two types of approaches we are using for improving the reliability of such system:

1. Fault Avoidance: This is achieved by using standard quality components with high reliability.
2. Fault tolerance: This is achieved by using redundancy technique. Redundancy forced to increased design complexity and costs of such system.

It is advisable to measure the reliability of each system before improving the reliability of any component or components of such system because it reduces the overhead.

Algorithm for Evaluating R(X) of Network

Input: (N, L), Traffic, $I_M, A_M, Comp_i$

I_M : Incident Matrix

A_M : Adjacency Matrix

N: Nodes of Network

L: Links of Network

$Comp_i$: Componets number

Output:

Call Measure ($R(X), Rel_{(Current)}$)

if

$R(Comp_i) \leq \text{Threshold}$

then

Call Improve ($R(Comp_i)$)

else

Evaluate the R(Function) Regularly of the network

If we are able to quantified the value of each node, then analysis is easy but it become clear that the reliability of system will not be satisfactory to meet the threshold value sometime. If three components are connected in a series with a reliability value 70%,80%,90% and expected reliability of system $R(X) = 87\%$ the current reliability of system is calculated as

$$R(X) = R_1 * R_2 * R_3 = 50.4\%$$

The overall reliability is failed due to our assumption $Rel_{(Current)} \leq R(X) \leq 1$, this is far lesser than system expected reliability. It is clear that, to increase the $Rel_{(Current)}$ we need to increase the individual reliability of component $R(Comp_i)$.

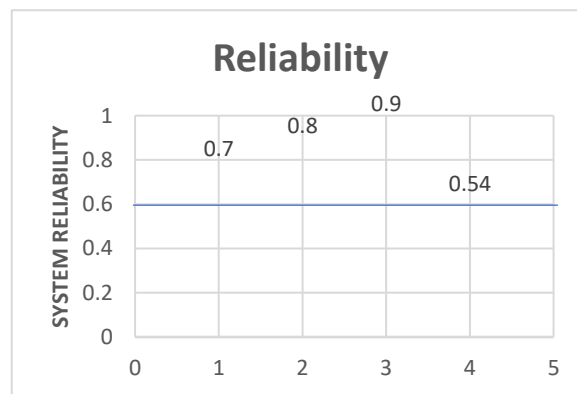


Fig V: System Reliability Vs Component Reliability

If try to increase the reliability of individual component at a single time then the goal cannot be achieved. For example, if hypothetically we increase the reliability value of one component. $R(Comp_1) = 1$ or say 100%, the $Rel_{(Current)}$ will not meet expected reliability by improving the reliability of the only $R(Comp_1)$. It means that individual components cannot force to increase system reliability. Is this possible to increase randomly reliability of any two components will put an impact on $Rel_{(Current)}$ mathematically this is not fruitful because this is a question of ambiguity to increase the reliability of which two components ($R(Comp_i)$).

The above approach is applicable if we have a limited no of nodes but for a varying size network, the problem is NP-hard. The present approach is based on a Genetic algorithm because we know that due to the flexibility and robustness of GA, we can approach so many NP-hard problems. Our approach is focused on network designing with an assumption that the network consists of N nodes or say components and with L links or arcs. Links are placed in between any two components of network; other assumption is that in between any two pairs of nodes either link is present or absent (means one or zero link connecting them). Our second assumption is that every link has same probability because as we see in previous Fig which shows that if a link has different probability or reliability it is a question of ambiguity. Reliability measurement is represented by $R(Comp_i)$ is the probability (p) that the link is working and $q = 1 - p$ is used when link is broken. Here p is called robustness parameter, p exists in two states *Open* and *Closed* states. Where *Open* means successful communication and *Closed* state mean failed or broken communication. Successful communication means there is at least one path between any two pairs of nodes and that exist a reliability value $R(X)$ which is thus a probability value (p) which indicates successful working of a system. So, we can say that network reliability means that the network is connected and a node is able to communicate with any other node in the network. The all-terminal reliability $R(X)$ is the probability that communication network remains coupled assuming that nodes are perfect and by knowing probability edges can fail independently [2, 17,19,23,24,25] or we can say that “All Terminal” reliability as it known as the network consists of one single connected component. Now our main aim is evaluated network reliability $R(X)$ for a given network topology. We use simple undirected and unweighted network graph with following assumption:

1. No articulation vertex.
2. No Redundant edge
3. No Pendant vertex
4. No null vertex.
5. Connected Graph

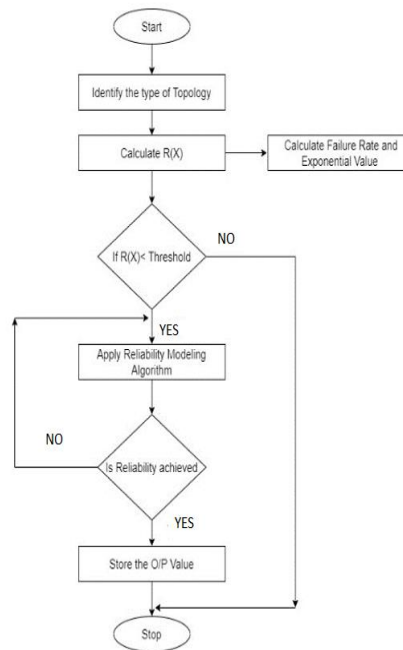


Fig VI: Flow chart to Evaluate Reliability.

we can describe their topology with a special adjacency matrix $A[i][j]$. In adjacency matrix if $A[i][j] = 1$ indicates presence of link and $A[i][j] = 0$ indicate absence of link between any two pair of vertices. As discussed in our previous paper [26] network reliability can be evaluated by adjacency matrix due to integration of a variable p .

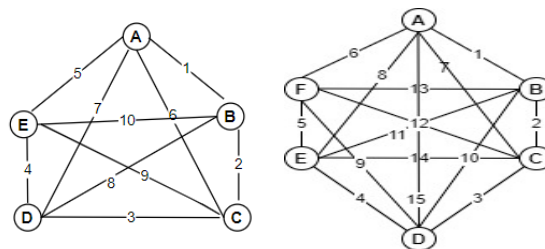
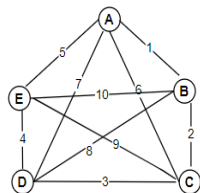


Fig VII : (a) Connected Graph with N=5 & L=10

(b) Connected Graph with N=6 & L=15

Problem can be formulated as for an undirected graph $G = (V, E)$, we consider independent failures on the links in L . Each failure is associated with its links l and occurs randomly and with probability. The network is represented by adjacency matrix $A[i][j]$. and it is useful because the chromosome is a string. In Genetic Algorithm, the search space comprises of candidate solutions to the problem which is represented by a string, known as chromosome. The chromosome holds objective function value, defined as the fitness. A collection of chromosomes together holding associated fitness is the population and at some specific iteration of the genetic algorithm, is termed as a generation [5-7][30].

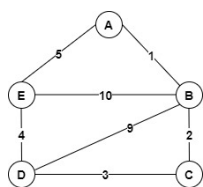


Complete Network Graph
 $G (N=5, L=10)$

Adjacency Matrix of

	A	B	C	D	E
A	0	1	1	1	1
B	1	0	1	1	1
C	1	1	0	1	1
D	1	1	1	0	1
E	1	1	1	1	0

Network Graph

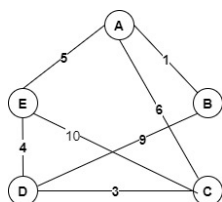


Network Graph G1 (N=5, L=7)

Adjacency Matrix of

	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	1	1
C	0	1	0	1	0
D	0	1	1	0	1
E	1	1	0	1	0

Network
Graph G1

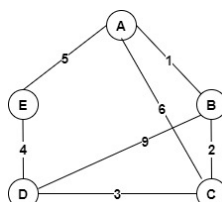


Network Graph G2 (N=5, L=7)

Adjacency Matrix for

	A	B	C	D	E
A	0	1	1	0	1
B	1	0	0	1	0
C	1	0	0	1	1
D	0	1	1	0	1
E	1	0	1	1	0

network Graph G2

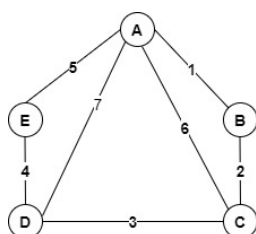


Network Graph G3 (N=5, L=7)

Adjacency Matrix for

	A	B	C	D	E
A	0	1	1	0	1
B	1	0	1	1	0
C	1	1	0	1	0
D	0	1	1	0	1
E	1	0	0	1	0

network
Graph G3

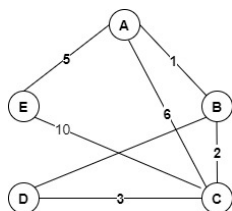


Network Graph G4 (N=5, L=7)

Adjacency Matrix for

	A	B	C	D	E
A	0	1	1	1	1
B	1	0	1	0	0
C	1	1	0	1	0
D	1	0	1	0	1
E	1	0	0	1	0

network
Graph G4



Network Graph G5 (N=5, L=8)

Adjacency Matrix for

	A	B	C	D	E
A	0	1	1	0	1
B	1	0	1	1	0
C	1	1	0	1	1
D	0	1	1	0	0
E	1	0	1	0	0

network

Graph G5

Fig VIII: Connected Graph with N=6 and Adjacency matrix

Start Algorithm

Generate the initial population K=1 to N

Compute the fitness of each network.

Upper and Lower Bound Values of each Network

While the condition is not satisfied

Select the network population from the existing population of the network.

Produce a new generation

Discard the population which is not the 2-connected network.

Evaluate the reliability of the networks

Find the most reliable network in the generation

End while

Repeat

Selection

Crossover

Mutation

Compute

fitness

Until population has converged

Output the most reliable network of the last generation

Stop Algorithm

Matrix A of the network is useful for the reliability evaluation and it cannot be useful with the implementation process of the genetic algorithm because it requires a string. In GA chromosome is a string of n length. This can be only done by making a linking or conversion of the adjacency matrix into a string. Consider an upper left triangle of Matrix the string representations are as follows:

Table III: String Representation of Graph G

Graph	STRING REPRESENTATION	REPLACED VALUE
G	1,1,1,1,1,1,1,1,1	1,2,3,4,5,6,7,8,9,10
G1	1,0,0,1,1,1,1,0,1	1,4,5,6,7,8,10
G2	1,1,0,1,0,1,0,1,1	1,2,4,6,8,9,10
G3	1,1,0,1,1,1,0,1,0	1,2,4,5,6,8,10
G4	1,1,1,1,1,0,0,1,0	1,2,3,4,5,8,10
G5	1,1,0,1,1,1,0,1,1	1,2,4,5,6,8,9

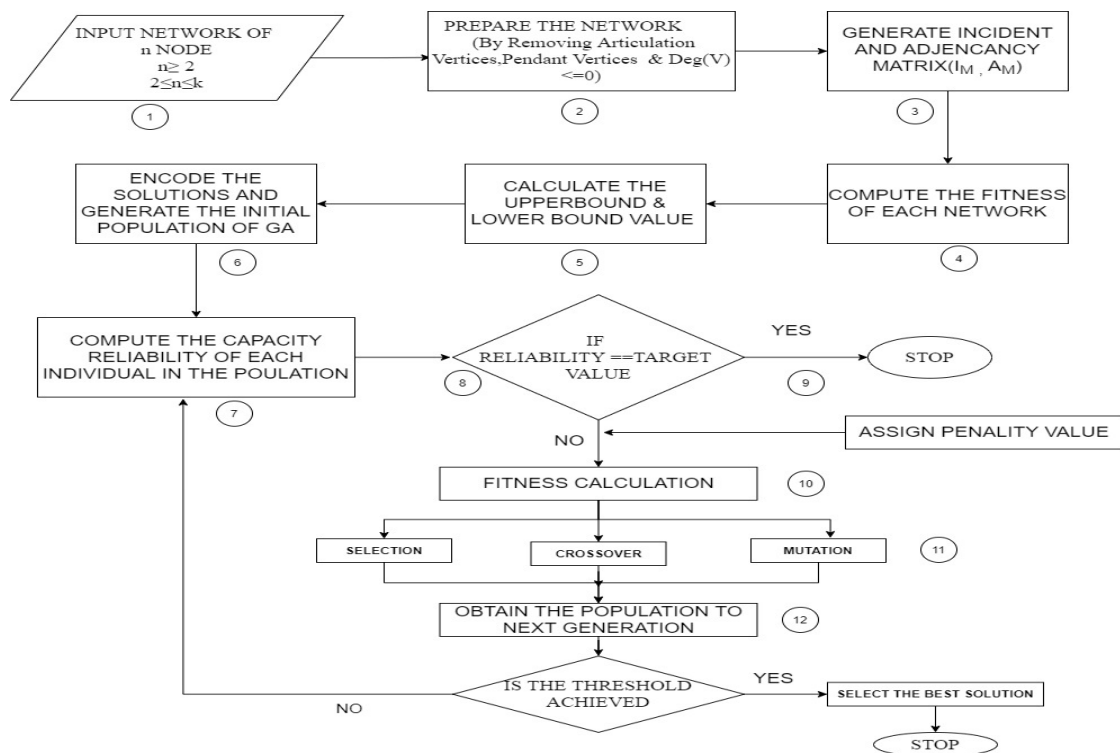


Fig IX: Flowchart of Optimal network reliability based on GA

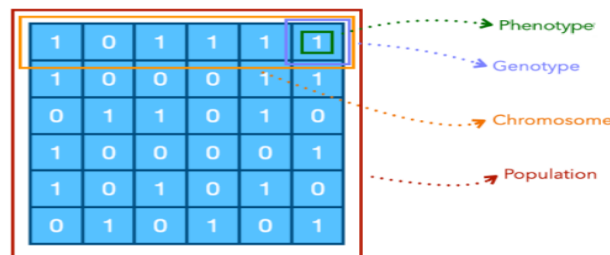


Fig X: Population Matrix of GA with respect to nodes

Genetic algorithm is relying on some biological operators such as

1. Chromosome
2. Mutation
3. Crossover
4. Selection

Main steps of genetic algorithm are as follows.

1. Selected string from existing string to be the generator of the next generation. The filtering process is used for the selection of a string from the generation. This is the selection process for reproduction.
2. Crossover is a process of generating a new string from two filtered string or from selected string.
3. Exploration of search space is called Mutation process.
4. Crossover and mutation are used for reproduction

The steps in the GA approach are given in a flowchart of the algorithm.

Maximum string size of graph given in Fig 4 that represents proposed network design and the size of the string is equal to $\frac{n*(n-1)}{2}$ if graph is fully connected and complete graph. Matrix cell value represents the following:

- 0: no edge or link between nodes v_i and v_j .
- 1: direct link between nodes v_i and v_j .

The string representation of Fig 4(b) is represented by 1001111101 which is represented by 1,4,5,6,7,8,10 means that the maximum number of digits to represent graph is 10 but in this case, it is represented by 7. A 2-connected graph is a graph if for each vertex $x \in V(G)$, $G - x$ is connected

IV. IMPLEMENTATION OF ALGORITHM

In Genetic algorithm creation process is used for the creation of a new network and is used for the initial population generation and for the evolution of low reliability modules of network whereas mutation process is used for replacing one entry from the existing string or it is used for the replacement of any one link from the graph and place it to the other position of graph. The result of replacement does not affect the degree of the graph

$$\deg(g) = \sum_{i=1}^N \deg(v_i)$$

The newly obtained network is just similar to previous one and the reliability of the network is remained same as the previous one. Ordered pair for the network are as follows:

A = (1, 2), B = (2, 3), C = (3, 4), D = (3, 4), E = (1, 6), F = (6, 7), G = (7, 4), H = (4, 5)

The reliability estimation of above network is calculated as follows:

$$\begin{aligned} Rel(G) = & -(R_a * R_b * R_c * R_e * R_f * R_g * R_h) + (R_a * R_b * R_c * R_d * R_e * R_f * R_g * R_h) + (R_a * R_b * R_d * R_h) + (R_e \\ & * R_f * R_g * R_h) - (R_a * R_b * R_d * R_e * R_f * R_g * R_h) + (R_a * R_b * R_c * R_h) - (R_a * R_b * R_c * R_d \\ & * R_h) \end{aligned}$$

Let us assume that

$R_a=0.92, R_b=0.85, R_c=0.85, R_d=0.85, R_e=0.95, R_f=0.75, R_g=0.8, R_h=0.92,$

$Rel(G) = 0.82679$

A = (1, 2), B = (2, 3), C = (3, 4), D = (7, 4), E = (1, 6), F = (6, 7), G = (7, 4), H = (4, 5)

$$\begin{aligned} Rel(G) = & -(R_a * R_b * R_c * R_e * R_f * R_d * R_h) + (R_a * R_b * R_c * R_d * R_e * R_f * R_g * R_h) + (R_e * R_f * R_g * R_h) + (R_d \\ & * R_e * R_f * R_h) - (R_d * R_e * R_f * R_g * R_h) + (R_a * R_b * R_c * R_h) - (R_a * R_b * R_c * R_f * R_g * R_h) \end{aligned}$$

Let us assume that $R_a=0.92, R_b=0.85, R_c=0.85, R_d=0.85, R_e=0.95, R_f=0.75, R_g=0.8, R_h=0.92,$
 $Rel(G) = 0.82679.$

A new network is created by Creation. It helps to generate the population at initial level and remove the networks having low reliability in the succeeding evolution. A single entry can be change of the chromosome by mutation. In terms of network, it can change position of links corresponding to rewiring of the network. The new network obtained after changing position (mutation) of links has equivalent reliability as of the previous network.

G1	1,0,0,1,1,1,1,0,1	1,4,5,6,7,8,10
G2	1,1,0,1,0,1,0,1,1,1	1,2,4,6,8,9,10
G3	1,1,0,1,1,0,1,0,1	1,2,4,5,6,8,10
G4	1,1,1,1,1,0,0,1,0,1	1,2,3,4,5,8,10
G5	1,1,0,1,1,1,0,1,1,0	1,2,4,5,6,8,9

	A	B	C	D	E
A	0	1	0	0	1
B	1	0	1	1	1
C	0	1	0	1	0
D	0	1	1	0	1
E	1	1	0	1	0

A=G1 & B=G2 crossover points at 2 and 5 in first pass and 7 and 9 in second pass if required t retain the original one.

Table IV: Crossover operation of two chromosome

A	1	0	0	1	1	1	1	1	0	1
B	1	1	0	1	0	1	0	1	1	1
C	1	1	0	1	0	1	1	1	0	1
D	1	0	0	1	1	1	0	1	1	1
E	1	1	0	1	0	1	0	1	1	1
G	1	0	0	1	1	1	1	1	0	1

The second stage crossover process generates the same string even after crossover will perform on some other disjoint points.

Penalty Function: It is defined as quadratic penalty value, used with objective function i.e., sum of the total cost for all links in a computer communication network. The penalty function assists the network to meet minimum reliability requirements. Its main objective is to help in finding a minimal optimal solution or feasible solutions. The penalty function is act as a dotted line between the feasible and non-feasible solution of network reliability.

Fitness Function is a kind of Evaluation function that evaluates a given solution concerning the optimum solution of the desired problem. It determines how to fit a solution is.

```
import numpy
import ga
y = w1x1+w2x2+w3x3+w4x4+w5x5+w6x6
qu_in= [1,4,5,6,7,8,10]
num_wg = len(equ_i_p)
sol_per_pop = 8
num_Par_mating = 4
pop_size = (sol_per_pop,num_wg)
#Creating the initial population
Pop_new = numpy.random.uniform(low=-4.0, high=4.0, size=pop_size)
print(Pop_new)
Pop_new[0, :] = [1,4,5,6,7,8,10]
Pop_new[1, :] = [1,2,4,6,8,9,10]
Pop_new[2, :] = [1,2,4,5,6,8,10]
Pop_new[3, :] = [1,2,3,4,5,8,10]
Pop_new[4, :] = [1,2,4,5,6,8,9]
best_outputs = []
num_gens = 50
for generation in range(num_generations):
    print("Generation : ", generation)
    fit1 = ga.cal_pop_fit1(equ_i_p, Pop_new)
    print(fit1)
    best_outputs.append(numpy.max(numpy.sum(Pop_new*equ_i_p, axis=1)))
    print("Best result : ", numpy.max(numpy.sum(Pop_new*equ_i_p, axis=1)))
    Par = ga.select_mating_pool(Pop_new, fit1,num_Par_mating)
    print("Par")
    Off_Spring_cross = ga.crossover(Par,offspring_size=(pop_size[0]-Par.shape[0], num_wg))
    print("Crossover")
    print(Off_Spring_cross)
    offspring_mutation = ga.mutation(Off_Spring_cross, num_mutations=2)
    print("Mutation")
    print(offspring_mutation)
    # Creating the new population based on the Par and offspring.
    Pop_new[0:Par.shape[0], :] = Par
    Pop_new[Par.shape[0], :] = offspring_mutation
    fit1 = ga.cal_pop_fit1(equ_i_p, Pop_new)
    best_match_idx = numpy.where(fit1 == numpy.max(fit1))
    print("Best solution : ", Pop_new[best_match_idx, :])
    print("Best solution fit1 : ", fit1[best_match_idx])
    import matplotlib.pyplot
    matplotlib.pyplot.plot(best_outputs)
    matplotlib.pyplot.xlabel("Iteration")
    matplotlib.pyplot.ylabel("Fit1")
    matplotlib.pyplot.show()
```

Fig XI: Screen shoot of code Implementation in Python

V. RESULTS AND DISCUSSION

The effective results of the above algorithm can be obtained for the two-terminal and all-terminal reliability problem based on close analysis of the complexity of the network tree-width. Our assumption for reliability evaluation of node 1 to node N, with link reliability of 50%. The network within the RED line in the link is already taken into consideration. To illustrate the proposed algorithm for computing the system reliability, consider the example shown in Fig. III This network has 4 nodes and 6 arcs. The arcs are numbered from a_1 to a_6 and the nodes from a_7 to a_{10} . The same capacity distribution of each component given in [5] will be used to calculate R_5 (i.e., the demand $d = 5$).

- To solve the optimization problem of reliability, GA provides a framework that does not depend on any variable. GA is an iterative algorithm, this ideal cannot be attained as many candidate networks must be evaluated during the search. The result can be obtained in the following steps Use a decision variable and set the various constraints.
- Construct an optimization function.

- c) Generate a Chromosome table. Gene expression. Paper is focus on binary coding method, to determine the expression of N gene with computer network node, as shown in table 3.
- d) Fitness function is arranged according to the numerical size and cost is calculated as:

$$f(x) = \frac{(x - 1)}{p - s - 1}$$

Where p : Population size and $1 \leq x \leq p - s$.

- e) Record the representation of the chromosome coding technique, that is to determine the genotype of X individuals and genetic algorithm search space.

Determine the decoding method and identified by the individual genotype correspondence or convert X to individual phenotype x method.

Table V: Shows the Component, capacity, and probability of each component.

Component	K1				
Capacity	C ₀	C ₁	C ₂	C ₃	
Probability	0.01	0.04	0.05	0.90	
Component	K2				
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄
Probability	0.01	0.01	0.02	0.04	0.92
Component	K3				
Capacity	C ₀	C ₁	C ₂	C ₃	
Probability	0.01	0.04	0.05	0.9	
Component	K4				
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄
Probability	0.01	0.01	0.01	0.05	0.92
Component	K5				
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄
Probability	0.01	0.01	0.02	0.04	0.92
Component	K6				
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄
Probability	0.01	0.01	0.01	0.05	0.92

Component	V1							
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇
Probability	0.005	0.005	0.005	0.005	0.01	0.02	0.04	0.91
Component	V2							
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄			
Probability	0.01	0.01	0.02	0.04	0.92			
Component	V3							
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄			
Probability	0.01	0.01	0.01	0.05	0.92			
Component	V4							
Capacity	C ₀	C ₁	C ₂	C ₃	C ₄	C ₅	C ₆	C ₇
Probability	0.005	0.005	0.005	0.005	0.01	0.02	0.04	0.91

Parameters	Name	Sample1	Sample 2	Sample 3
G A Parameter	No of Population	27	37	50

	Probability of cross over	0.8	0.8	0.8
	Probability of Mutation	0.01	0.01	0.01

Table VI: Comparison of Results of GA over n-Seeds (best, mean and coefficient of variation) with Optimal Design.

Set No	NODE	LINK	PROBABILITY	RELIABILITY	Best	Mean	CV
Set1	$n_1 - n_5$	$l_1 - l_{10}$	0.8	0.90	255	255	0.0012
Set2	$n_1 - n_7$	$l_1 - l_{21}$	0.9	0.90	720	720	0.0011
Set3	$n_1 - n_8$	$l_1 - l_{28}$	0.9	0.90	203	205.4	0.0198
Set4	$n_1 - n_9$	$l_1 - l_{36}$	0.9	0.90	239	245	0.0495
Set5	$n_1 - n_{10}$	$l_1 - l_{36}$	1.0	0.95	209	227	0.0837
Set6	$n_1 - n_{10}$	$l_1 - l_{45}$	0.9	0.90	156	170	0.0615
Set7	$n_1 - n_{10}$	$l_1 - l_{45}$	0.9	0.95	205	207	0.0095
Set8	$n_1 - n_{15}$	$l_1 - l_{105}$	0.9	0.95	317	344	0.7028
Set9	$n_1 - n_{20}$	$l_1 - l_{190}$	1.0	0.95	926	956	0.0301
Set10	$n_1 - n_{25}$	$l_1 - l_{300}$	1.0	0.90	1606	1651	0.0240

Table VII: GA and Reliability Estimation Performance

Set No	NODE	CV	LINK	RELIABILITY	ACTUAL RELIABILITY	ESTIMATED RELIABILITY	DIFFERENCE
Set1	$n_1 - n_5$	0.0012	$l_1 - l_{10}$	0.90	0.9169	0.9170	-0.0001
Set2	$n_1 - n_7$	0.0011	$l_1 - l_{21}$	0.90	0.9029	0.9029	0.0000
Set3	$n_1 - n_8$	0.0198	$l_1 - l_{28}$	0.90	0.9078	0.9078	0.0000
Set4	$n_1 - n_9$	0.0495	$l_1 - l_{36}$	0.90	0.9066	0.9069	-0.0003
Set5	$n_1 - n_{10}$	0.0837	$l_1 - l_{36}$	0.95	0.9669	0.9668	0.0001
Set6	$n_1 - n_{10}$	0.0615	$l_1 - l_{45}$	0.90	0.9050	OV	DNC
Set7	$n_1 - n_{10}$	0.0095	$l_1 - l_{45}$	0.95	0.9516	OV	DNC
Set8	$n_1 - n_{15}$	0.7028	$l_1 - l_{105}$	0.95	UC	0.9509	DNC
Set9	$n_1 - n_{20}$	0.0301	$l_1 - l_{190}$	0.95	UC	0.9925	DNC
Set10	$n_1 - n_{25}$	0.0240	$l_1 - l_{300}$	0.90	UC	0.9618	DNC

UC: Unable to Calculate

OV: Optimal Value

DNC: Do Not Calculate

DNC: Do Not Calculate

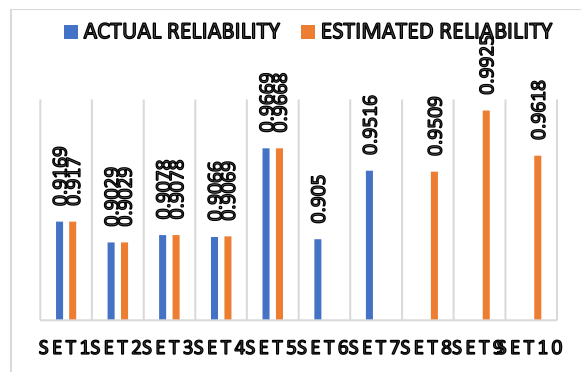


Fig 12: Actual Vs Estimated Reliability

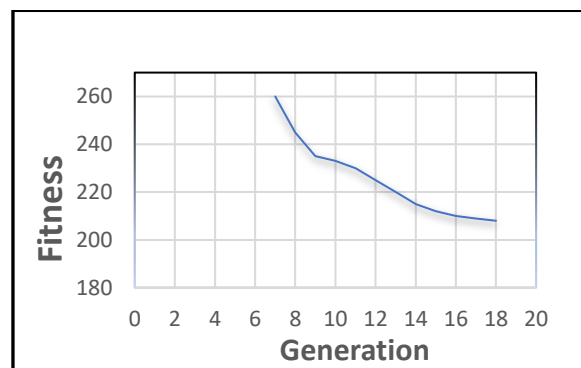


Fig 13: Fitness Vs Generation Graph

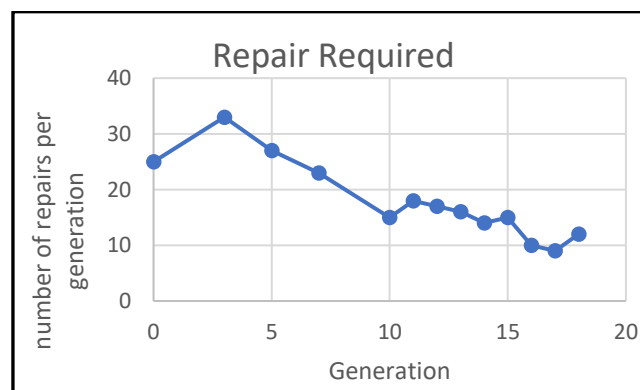


Fig 14: Repair required to achieve maximum reliability.

VI. CONCLUSION

In conclusion, we have presented a special method to find the most reliable network. For achieving this we proposed a Genetic algorithm-based approach to identify useful patterns and found that the efficiency of the GA is much better than the other algorithm proposed for network reliability analysis. The statistics were obtained from the prototype with the Python language. Our algorithm results provide an efficient way to find a reliable network path from a set of N nodes and L links. The main limitation is that our algorithm is an approximated due to the error in the reliability estimation. The algorithm is testing on the network size of 2,4,6,8, -----256 nodes with link $N(N-1)/2$.

VII. REFERENCES

- [1] Berna, D., Fulya. A. and Smith A., (1999). A genetic algorithm approach to optimal topological design of all terminal networks, *Intelligent Engineering Systems Through Artificial Neural Network*, 5: 405-410.
- [2] Srivaree-ratana, C., Konak, A., Smith, A., (2002). Estimation of all-terminal network reliability using an artificial neural network. *Computers & Operations Research* 29 849,868.

- [3] Altıparmak, F., Dengiz, B., and Smith, A., (2003), "Reliability estimation of computer communication networks: ANN models," in IEEE ISCC'03 Conference Proceedings, Antalya-Kemer, Turkey, 2003, pp.1353–1358.
- [4] Altıparmak, F., Dengiz, B., (2003). Optimal Design of Reliable Computer Networks: A Comparison of Metaheuristics. *Journal of Heuristics*, 9: 471–487.
- [5] Yeh, W., (2004). An algorithm for evaluating the k-out-of-n network reliability. *Reliability Engineering and System Safety*. Vol. 83, pp.93-101.
- [6] Yeh, W., (2005). A new algorithm for generating minimal cut sets in k-out-of-n network. *Reliability Engineering and System Safety*. Vol. 91, pp.36-43.
- [7] Konak, A., and Alice, E., (2005). Designing Resilient Networks Using a Hybrid Genetic Algorithm Approach. GECCO'05, Washington, DC, USA. ACM 1-59593-010-8/05/0006
- [8] Konak, A., & Smith, A. E.(2005). Designing resilient networks using a hybrid genetic algorithm approach. *Proceedings of the Conference on Genetic and Evolutionary Computation - GECCO '05*.doi:10.1145/1068009.1068217.
- [9] Rajpal, P. S., Shishodia, K. S., and Sekhon, S. (2006) "An artificial neural network for modeling reliability, availability and maintainability of a repairable system". *Reliability Engineering & System Safety*, 91(7), 809–819.
- [10] Lee, S., Lin, H., (2006) . Artificial Neural Network Analysis for Reliability Prediction of Regional Runoff Utilization. CIB W062 Symposium
- [11] He, F., and Qi, H., (2008). A method of estimating network reliability using an artificial neural network. Published in *IEEE Pacific-Asia* DOI:10.1109/PACIIA.2008.130.
- [12] Altıparmak, F., Dengiz, B., (2009) A general neural network model for estimating telecommunications network reliability. *IEEE Transactions on Reliability* Vol.58,No.1.
- [13] Mutawa, A. M., Alazemi, H. M. K., and Rayes, A. (2009). A novel steady-state genetic algorithm approach to the reliability optimization design problem of computer networks. *International Journal of Network Management*, 19(1), 39–55. doi:10.1002/nem.687.
- [14] Hamed A., (2010) A genetic algorithm for finding the k shortest paths in a network. *Egyptian Informatics Journal* 11, 75–79.
- [15] Yeh, W., (2010). A Particle swarm optimization approach based on Monte Carlo Simulation for Solving the complex network reliability problem. *IEEE Transactions on Reliability* Vol.59, No.1.
- [16] Hassan, M., (2011). A Genetic Algorithm for Reliability Evaluation of a Stochastic- Flow Network with Node Failure. *International Journal of Computer Science and Security (IJCSS)*, Volume (4): Issue (6) 528.
- [17] Yano, A., Wadayama, T., (2011). Probabilistic Analysis of the Network Reliability Problem on a Random Graph Ensemble. *IEICE Transactions on Fundamentals of Electronics Communications and Computer Sciences* ·arXiv:1105.5903v3
- [18] Rebaiaia, M., (2013). A New Technique for Generating Minimal Cut Sets in Nontrivial Network", *AASRI Procedia* 5. 67 – 76.
- [19] Chandrashekar, T., and Goyal, N.,(2013). An approach to evaluate multiple node pair reliability for simultaneous capacity requirements. *International Journal of performability Engineering*, Vol 9,No 4 July 2013 PP 357-366.
- [20] Zhang, H., (2014). Network Performance Reliability Evaluation Based on Network Reduction Reliability and Maintainability Symposium 978-1-4799-2848-4/14 IEEE.
- [21] Chakraborty, S., and Goyal, N., (2015). Irredundant subset cut enumeration for reliability evaluation of flow networks, *IEEE Trans. Rel.*, vol. 64, no. 4, pp. 1194–1202,
- [22] Anuradha., Kumar, A., (2016). Network Reliability measurement using Artificial Neural Network. *International conference BEST-2016 IEEE, ISTE UP Section*.
- [23] Chakraborty, S., (2016). An Efficient Reliability Evaluation Approach for Networks with Simultaneous Multiple-Node- Pair Flow Requirements (wileyonlinelibrary.com) DOI: 10.1002/qre.2097.
- [24] Chatterjee, S., (2017). An improved algorithm for k-terminal probabilistic network reliability analysis. *Journal of reliability and statistical studies*; ISSN (print): 0974-8024, (online): 2229-5666 vol. 10, issue 1 (2017): 15-26
- [25] Pérez-Rosés, H., (2018). Sixty Years of Network Reliability. *Mathematics in Computer Science*.https://doi.org/10.1007/s11786-018-0345-5.
- [26] Anuradha. and Kumar, A., (2020). Calculation and Evaluation of Network Reliability using ANN Approach. *Procedia Computer Science* 167 2153–2163, 10.1016/j.procs.2020.03.265.

- [27] Khan SM, Nilavalan R, Sallama AF (2015) A novel approach for reliable route discovery in mobile ad-hoc network. *Wirel Pers Commun* 83:1519–1529. <https://doi.org/10.1007/s11277-015-2461-8>
- [28] Khanna G, Chaturvedi SK (2018) A comprehensive survey on multihop wireless networks: milestones, changing trends and concomitant challenges. *Wirel Pers Commun*. <https://doi.org/10.1007/s11277-018-5711-8>
- [29] Khanna G, Chaturvedi SK, Soh S (2019) On computing the reliability of opportunistic multihop networks with mobile relays. *Qual Reliab Eng Int*. <https://doi.org/10.1002/qre.2433>
- [30] Suraj R, Tapaswi S, Yousef S et al (2016) Mobility prediction in mobile ad hoc networks using a lightweight genetic algorithm. *Wirel Netw* 22:1797–1806. <https://doi.org/10.1007/s11276-015-1059-0>
- [31] Malay K. Pakhira and Rajat K. De(2007),"Generational Pipe Lined Genetic Algorithm (PLGA) using Stochastic Selection" *IJCSSE VOLUME 1 NUMBER 1 2007* ISSN 1307-430X