Journal of Artificial Intelligence
Journal of Artificial Intelligence
ISSN: 2229–3965 & ISSN: 2229–3973, Volume 2, Issue 1, 2011, pp-31-41
Available online at http://www.bioinfo.in/contents.php?id=71

AN ENHANCED ANN APPROACH FOR OPTIMIZING DESIGN OF HIGHLY RELIABLE VARIABLE SIZED COMPUTER NETWORKS

BAIJNATH KAUSHIK[1*], NAVDEEP KAUR[2], AMIT KUMAR KOHLI[3]
[1]Department of Computer Science, Krishna Engineering College, Ghaziabad, India
[2]Department of Information Technology, Chandiagarh Engineering College, Chandigarh, India
3Department of Electronics & Communication, Thapar University, Patiala, India
*Corresponding Author: Email- bkaushik99@gmail.com

Received: October 01, 2011; Accepted: October 12, 2011

Abstract- The problem of optimizing reliability has been developed as potential area of research, and perceived significant contribution due to critical importance of reliability in many models and systems. The design of series-parallel system when considering cost and reliability in optimal network design, it had less practical utility because reliability of series-parallel systems can be exactly calculated very easily with closed mathematical expressions. Today, networks are complex in structure may be of variable sized growing networks, it is practically difficult to calculate exact reliability using analytical methods in minimum time. Therefore, this paper aims to provide an optimized design of highly reliable variable sized computer networks with an enhanced ANN approach for measuring reliability and cost. This paper compares Monte Carlo simulation with enhanced ANN approach to solve optimal design of variable sized networks. The results obtained shows the different designs are possible when method includes variable links measurement into optimal design problem objectives.
Key words - Monte Carlo simulation (MCS), Enhanced ANN (EANN), Optimized network design (OND), Variable links (VL), Variable sized networks (VSN), Particle Swarm Optimization (PSO), Reliability Optimization (RO), Fixed Links (FL).

I. Introduction

The study of reliability has received considerable interest of research in many areas. Examples of these areas are software reliability [1-2], reliability of computing systems [3–6], and network reliability [41]. In particular, the reliability of networks has gained great impetus from its clear applicability to complex systems, and project schedules. Each component in these networks refers to an object in a complex network, or an operation in a project schedule. Network reliability is one of the useful decision support measures in management science. It has been applied in many real-world systems such as oil/gas production systems[7], computer and communication systems [8-9], power transmission and distribution systems [10], and transportation systems [11]. Therefore, system reliability plays an important role in our modern society.

Reliability optimization has been a popular area of research, and received significant attention during the past four decades [7–11] due to the critical importance of reliability in various kinds of systems. Most of these works either assumed that the reliability functions are known in advance, or estimate the approximate symbolic network reliability function.

The network reliability optimization problem trades between cost and reliability. It can be simplified into two types of models without setting them as Multi-Objective Decision Analysis (MODA) problems [12] or as Analysis of Variance (ANOVA) [13-17]. The first type sets the system reliability as the only goal, to maximize the system reliability with components' cost limitations; the second type is the dual problem, where the total cost is minimized under the system, and components' reliability constraints.

The measurement of overall reliability in computer networks of growing size is NP—hard problem, the computational effort required is growing exponentially with growing network size [18-20] in terms of nodes and links in the network. To achieve a better solution quality (computational efficiency, and estimation accuracy), modern meta-heuristics have been presented to solve complex network reliability optimization problems such as Response Surface Methodology [21], Artificial Neural Network (ANN) [22-24], Genetic algorithms [25-27], Ant Colony Optimization [28], [29], Tabu Search [30-32], and Particle Swarm Optimization (PSO) [12], [33-36], Simulated Annealing [37]. It is impractical, and very difficult to evaluate variable sized network reliability, and reduce the cost of the system at the same time. Hence, there is a need to develop a new method to solve the variable sized network reliability optimization problem such that the cost is minimized under the reliability constraints [12-17].

To solve reliability optimization problems, the reliability function is a necessary component under the above mentioned traditional methods. Computing the exact reliability of a network is also NP-hard [18], [38]; therefore, estimation by simulation and other approaches often becomes an alternative choice. Monte-Carlo Simulation

(MCS) is a straightforward simulation method for complex systems, and is now recognized as playing an important role in networks [12], [39], [41-44]. In addition, some works put forth other alternatives to the estimation of network reliability, such as response surface methodology [21], [41], and ANN [22-24].

Monte Carlo Simulation is one of the optimal algorithms to estimate the network reliability for different kinds of network configurations. Several authors have adopted Monte Carlo Simulation to measure the system reliability for conveniences, and accuracies. Kamat & Riley [41] introduced a general approach by combining the reliability of flow graph representation, Boolean state representation, and Monte Carlo simulation to estimate the system reliability. Their proposed methods require all Minimal Cuts (MCs) for the system in advance. Kubat [42] introduced a simulation/analytical approach to analyze communication and computer networks. Lin & Donaghey [43] presented a new MCS procedure for determining the Minimal Cut/Minimal Path (MC/MP), and system reliability. They use Monte Carlo Simulation to determine the MP by tracing through the system from input components to the output components, and then use MP to simulate the system failures. MC and system reliability are determined by MCS approach. W. C. Yeh [41] proposed a new Monte Carlo Simulation without knowing MP/MC to estimate the network reliability directly in. Furthermore, MCS methods have the following advantages when we use them to evaluate system reliability.

a. Monte Carlo Simulation method can be applied to different kinds of network configuration such as series-parallel [45], and other complex network structures.
b. It can be also used to analyze systems having components with variable distributions.
c. Now, with faster systems, time required by Monte Carlo Simulation has decreased significantly [44]. As a result, the Monte Carlo Simulation algorithm has now become one of the more efficient, optimal approaches for measuring and sampling network reliability.

S. R. V. Majety, M. Dawande, and J. Rajgopal [45] states that the reliability function of a system constructed by only simple series and parallel components can be obtained easily, but it cannot apply to most real-life networks such as variable sized large networks. Most network reliability optimization problems are only focused on solving series-parallel networks (e.g., simple structured networks) of which the reliability function can be easy obtained in advance.

PSO is a new population-based optimization algorithm. To our knowledge, the first published application of PSO in reliability engineering could be found in [12], [33]. In this present paper, we have proposed to combine MCS and enhanced ANN approach to obtain high reliability under minimum cost constrained for optimal design of variable sized computer networks. We have compared the proposed enhanced ANN approach with previous work done in reliability measurement of optimal design of variable sized networks.

The reliability measurement based on Monte Carlo Simulation (MCS-PSO) method [12] does not need to know exact or approximate reliability function in advance. The enhanced ANN approach works on exact or approximate reliability values for fixed and varying links as well as also when the size of networks are independent or unknown. Therefore, enhanced ANN approach [13-17] shows significant improvement for obtaining high reliability for optimal network design for variable sized networks.

This paper organized as follows. The next section describe about reliability optimization problem for optimal design. Then, we move to introduce about MCS-PSO method for MP/MC when exact or approximate reliability is not required. Then, we introduce enhanced ANN approach for obtaining high reliability and optimal network design for variable sized network. And, finally we conclude with remarks.

A. Formal Reliability Problem

The formal reliability optimization problem is to determine a system structure that meets high reliability requirement with minimum cost constrained. The problem is formally defined with acronyms and nomenclature as follows:

Acronyms

ANOVA	Analysis of variance
MODA	Multi-objective analysis
MC/MP	Minimal cut/Minimal path
ANN	Artificial neural network
MCS	Monte carlo simulation

Notation

Z(X)	Objective function that minimizes cost and reliability.
G(V, E)	A graph representing a network as a set of V nodes and E links.
R(X)	Reliability of X components in network.
R_0	Expected Network reliability requirement as output in ANN.
$R(x)$	Exact reliability of x components through MCS.
C_{ij}	Cost between node i and j.
C(X)	Overall cost of a network.
X_{ij}	Decision variable which may take values between {0, 1} for static and variable links.
X	The link topology representing {X_{12},…, X_{ij},…, X_{N-1},.., X_N} links.
t	Total number of network sections consists of pairs of node or links that makes the network.
n_i	Number of node pair or link pair i, i=1,2,..,t.
$N_{nc} = \sum_{i=1}^{t}\sum_{j=1}^{n_i} X_{ij}$	Number of network node components constituting the network.
$R_k^+(G)$	Reliability of a given graph that meets lower and upper-bound on reliability.
$p(x_j)$	Reliable links, consists of jth components i.e j=1,2,….m .

$q(x_j)$ Unreliable links, consists of jth components i.e j=1,2,….m.

Some Nomenclatures

Reliability The problem of determining network reliability in a variable sized network concerned with the ability of each network node to be able to communicate with every other nodes through some non-specified path.

MC/MP A minimal cut-set is a set of minimum number of components of a system, failing of these components also cause fail of the system. A minimal path set is a set of minimum number of components of a system, whose functioning maintains the function of a system.

Problem Assumptions

a. Every network component i.e. links or node may be operational or failed.
b. The states of a network component is independent.
c. The Minimum spanning tree graph is connected i.e. it is not forming a cycle.

The network reliability optimization problem such that the cost is minimized under maximum reliability constraint[12-17].

$$Minimize C(X) \tag{1}$$

$$S.t. R(X) \geq R_O \tag{2}$$

$$X = (X_{12}, X_{ij},., X_{N-1N}) \geq R_O = (R_{O1},.., R_{ON}) \tag{3}$$

The most common objective is to design a computer network by selecting a subset of possible links so that network reliability is maximized and maximum cost constrained is met. But, in many situations, it makes more sense is to minimize cost subject to a maximum network reliability constraint which can be can measured from Eq. (5). The objective of this proposal is to find the minimum cost network architecture that meets pre-specified minimum network reliability. The equation for the objective is given as following in Eq. (4):

$$Minimize Z(X) = \sum_{i=1}^{t} \sum_{j=i+1}^{n_i} C_{ij} X_{ij} \tag{4}$$

$$R(X) = \sum_{\Omega} \left[\prod_{j \in x'} p(x_j) \right] \left[\prod_{j \in x/x'} q(x_j) \right] \tag{5}$$

The design of network is difficult when overall reliability is considered. It is defined as the probability that all nodes communicate with every other nodes. The reliability is defined as *p*, and a non-zero reliability is $q = 1 - p$, at any time, only some links in a topology *X* may be operational. A state of a topology *X* is represented by a sub-graph (N, X'), where X' represents set of operational specific links such that $X' \subseteq X$. The network reliability for the state graph $X' \subseteq X$ is given in Eq. (5). In Eq. (5), Ω = all operational states in graph.

There is a problem in optimal design of topology when network is of variable size based on space size complexity:

$$K \frac{((\bmod N) \times (\bmod N) - 1)}{2} \tag{6}$$

Where *K* is the choices for the links is to be connected in the growing networks of variable size. For fixed links, there are always two choices: 0 for no link present and 1 for link present between any pair of nodes *i* and *j*. For varying link, we can choose a single link connecting two link or two nodes or more. There are several design options. For example, a 10 node network (*N* = 10) with fixed links (*k* = 2) has $3.5*10^{13}$ possible designs. A network with (*N* = 10) and with (*K* = 5) varying links choices has 10^{35} possible designs [14-17], [22-24]. For a growing network of variable size, it is practically difficult to calculate the exact network reliability.

The objective function stated in Eq. (1) and Eq. (2) must satisfy cost and reliability constraint.

II. MCS-PSO Method for MC/MP Problem

MCS methods have been used to solve many network reliability problems. Many people have developed different kinds of Monte Carlo techniques to estimate the system reliability [41], [46]. Many researchers have only applied MCS to estimate network reliability by focusing on how to design sampling plans to reduce the variance in terms of the known MP/MC [42], [47], [48]. However, to get all MP or MC information from the network is NP-Hard [18], [49]. W.C. Yeh [41] introduced a new MCS method called MCS-PSO for a complicated system without the task of knowing MP/MC is applicable to the network called AOA (activity on arc) or AON (activity on node) have presented to estimate the reliability. W.C. Yeh [41] has proved that $R_k^+(G)$ of a given graph G(V,E) where $k = 1,2,...m$ be the expected reliability. Then,

$$E\left[R_k^+(G)\right] = 1 \times p(x_j) + 0 \times q(x_j) = R(x) \tag{7}$$

The above Eq. (7) obtains reliability of a given graph. And, the estimator $R_k^+(G)$ of the system reliability is obtained as follows:

$$E\left[\sum_{j=1}^{k} \frac{R_k^+(G)}{k}\right] = \sum_{j=1}^{k} \frac{E\left[R_k^+(G)\right]}{k} = \sum_{j=1}^{k} \frac{R(x)}{k} = R(x) \tag{8}$$

When $R_k^+(G)$ is unbiased, consistent measuring variation for R(x) can be done using

$$Var\left[R_k^+(G)\right] = \sum_{j=1}^{S} \frac{R(x)\left[1 - R(x)\right]}{S} \tag{9}$$

W.C. Yeh [41] has suggested that all the three Eq. [(7), (8), (9)] if the relative error ε, and the confidence interval $(1-\alpha)\%$ for the simulation S are required, then the total number of replications required should be $S \geq \left(K_{\alpha/2}^2 / 4\varepsilon^2\right)$ at least.

The MCS algorithm suggested in [12, 40, 41, 46] worked perfectly for the simple network problem of reliability measurement, shown in Fig. 1, under assumptions that each node is reliable.

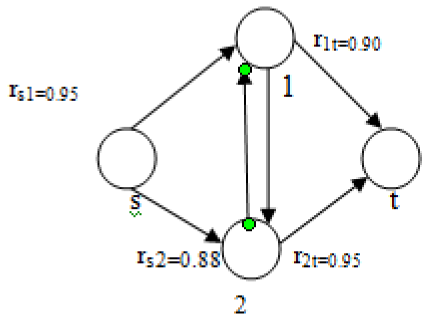

Fig. 1 A simple network

The MCS-PSO [12] can overcome the drawbacks of conventional methods which require finding all the exact or estimated reliability functions in advance to solve the complex network reliability optimization problem. But, they are also limited to small size network not for large networks.

PSO is a new population based optimization algorithm, which was first introduced by Kennedy & Eberhart[33] and by several other researchers for multi-objective reliability applications in [34-36]. The goal of this algorithm is to optimize various continuous nonlinear functions. The concept of PSO is based on the metaphor of social interaction, and communication such as fish schooling and bird flocking. In PSO, a solution is encoded as a finite-length string called "*a particle*". All of the particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles.

PSO is initialized with a population of random particles, with random positions and velocities inside the problem space, and then searches for optima by updating generations. It combines local search with global search, producing high search efficiency in problem space. The MCS-PSO method described in [12] is slightly modified and new result is produced here in fig. 3.

This method will be applied to small network problem as stated in [21], [41] is shown in fig. 3 given below:

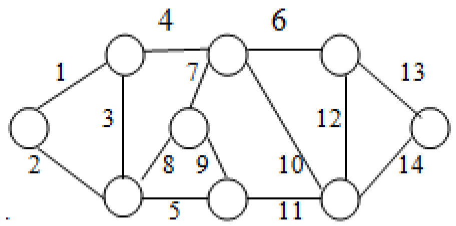

Fig. 2: A small scale network.

The main drawback of MCS-PSO method suggested by [21, 41] is that they cause algorithm to converge very slowly so that simulation results may not satisfy system reliability $R_k^+(G)$ the lower bound R(X). To overcome this, a penalty function suggested in [12] given below:

$$C(X)\times\left(\frac{R(X)}{R_k^+(G)}\right)^{\delta} \quad R_k^+(G) < R(X) \qquad (10)$$

Where δ is amplifying parameter for improving convergence rate. The exponent δ is an amplification parameter used for convergence. This penalty function in Eq. (10) encourages the particles to explore both of the feasible region, and the infeasible region.

MCS-PSO METHOD

Step 1: First particle is assigned with lower bound of reliability of the component.

Step 2: MCS is applied to find the system reliability $R_k^+(G)$.

Step 3: If $R_k^+(G) \le R(X)$, then perform step 4. Otherwise, the initial component reliability will be the solution, so, stop.

Step 4: Perform for each dimension of the particle.

Step 4.1: The ith dimension value is $0.35*(1-X_i)+X_i$.

Step 4.2: MCS is applied to obtain $R_k^+(G)$.

Step 4.3: Dimension I will be adjusted suitably.

Step 5: Now, for each dimension obtained for this particle.

Step 5.1: If the dimension of each $R_k^+(G)$ is less than length of each link of $R(X)$, then dimension of link will be find with highest value in step 5.3,

Step 5.2: The cost $C(X)$ of link length should be minimum such that $R_k^+(G) \ge R(X)$.

Step 5.3: The value of link dimension is $0.35*(1-X_i)+X_i = R_k^+(G)$, and finally go to 3.

End.

Fig. 3: A MCS-PSO Method.

This help the search not to go too far into the infeasible region. This approach assures that the feasible, and infeasible regions of the search space are explored in Eq. (6) efficiently, and effectively to identify an optimal, or nearly optimal solution. But, the drawback is limited to small scale network and the simulation is very costly as it takes large amount of time to operate.

Here, we present two tables used by MCS-PSO method to work on the problem stated in Fig. 1 and Fig. 2 given below:

TABLE I-
COST FUNCTION FOR EACH LINK COMPONENT

Component	α_i	β_i	$C(X)=\alpha_i-\beta_i\times\ln(1-X_i)$
1	140	15.5	140-15.5*ln(1-X_1)
2	140	12.5	140-12.5*ln(1-X_2)
3	95	7.6	95-7.6*ln(1-X_3)
4	100	8.6	100-8.6*ln(1-X_4)
5	68	9.4	68-9.4*ln(1-X_5)
6	110	5.5	110-5.5*ln(1-X_6)
7	92	5.75	92-5.75*ln(1-X_7)
8	165	9.95	165-9.95*ln(1-X_8)
9	170	8.85	170-8.85*ln(1-X_9)
10	175	14	175-14*ln(1-X_{10})
11	76	13	76-13*ln(1-X_{11})
12	85	12	85-12*ln(1-X_{12})
13	96	15	96-15*ln(1-X_{13})
14	180	15..8	180-15.8*ln(1-X_{14})

TABLE II- MCS-PSO PARAMETER

Total Particle	25
Dimension Length	14
Highest Velocity	1
High Position	1
Iteration	500
Cognitive factor	0.8
Social factor	0.8
Replication	50
Simulation Replication	10000
δ	1 to 10

III. Enhanced ANN Approach for Reliability Measurement

The disadvantage of MCS-PSO method has led improve the problem of reliability measurement in a variable sized networks with static and variable dimensions of links as well as convergence of optimal network design problems. This proposal focuses on the optimal design of highly reliable for variable sized computer networks using an enhanced ANN approach.

In this paper, ANN are developed, trained, based on the overall terminal's reliability of a very small set of possible network topologies and link reliabilities, for a given number of nodes. The resulting ANN is used to estimate network reliability as a function of the link reliabilities and the topology during search for the optimal design. In this way, estimates of the reliability of numerous topologies are available without costly calculation or simulation.

Artificial neural network [14-17] is used as a function approximation or a non-linear estimation technique which takes set of input values and it produces an output value. The functionality of using an ANN estimation of reliability [14-17] during optimal network design is tested by comparing it to an easily calculated upper-bound and expensive exact calculation [22-23].

A. Design Description of Enhanced ANN

Here we present an enhanced design of ANN modified from [14-17] is considered here shown below:

Enhanced ANN Algorithm

Step1: Normalize the I/P and O/P with respect to the maximum value. For each training pair, assume that in normal form

i inputs given by $\{I\}_i$,

n outputs given by $\{O\}_o$,

$n \times 1$

Step2: Assume that there are m numbers of neurons in the hidden layers where $1 < m < 25$

Step3: Let [V] represents weights of synapse that connect input and hidden neurons. Let [W] weights of synapse connect hidden and output neurons. Weights will be initialized small random value from -1 to +1.

$[V]^O$=[random weights]

$[W]^O$=[random weights]

$$[\Delta V]^O = [\Delta W]^O = [0]$$

Learning rate α may vary 10^{-3} to 10^3 and threshold is 0.

Step4: For training data, we need to present one set of input and outputs. Present the pattern as input to input layer $\{I\}_I$., then by using liner activation function, the output of the input layer may be obtained as follows:

$\{O\}_I = \{I\}_I$

$i \times 1 \quad i \times 1$

Step5: Compute the inputs to the hidden layer by multiplying weights of synapse as:

$$\{I\}_H = [V]^T \{O\}_I$$

$m \times 1 \quad n \times i \quad i \times 1$

Step6: The output at hidden layer obtained by using sigmoidal activation function

$$\{O\}_H = \left\{ \frac{1}{(1+e^{-(I_{HI})})} \right\}$$

$m \times 1$

Step7: The input of the output layer is obtained by multiplying it by weights of synapse:

$$\{I\}_O = [W]^T \{O\}_H$$

$n \times 1 \quad n \times m \quad m \times 1$

Step8: The output layer units, evaluate output using sigmoidal activation function as given below:

$$\{O\}_O = \left\{ \frac{1}{\left(1+e^{-(I_{OJ})}\right)} \right\}$$

Important: This output is the network output.

Step9: The error at the output layer is calculated using the difference between the network output from step 9 and the desired output as for the jth training set is given below:

$$E^P = \frac{\sqrt{\sum (Tj - O_{Oj})^2}}{n}$$

Step10: Find a difference term $\{d\}$ as given below:

$$\{d\} = \{(T_K - O_{OK})O_{OK}(1 - O_{OK})\}$$

Step11: Find [Y] matrix as:

$$[Y] = \{O\}_H \langle d \rangle$$

$m \times n \quad m \times 1 \quad 1 \times n$

Step12:

Find $[\Delta W]^{t+1} = \alpha[\Delta W]^t + \eta[Y]$

$m \times n \quad m \times n \quad m \times n$

Step13:

Find $\{e\} = [W]\{d\}$

$m \times 1 \quad m \times n \quad n \times 1$

$$\{d^*\} = \{ei(O_{Hi})(1 - O_{Hi})\}$$

$m \times 1 \quad m \times 1$

Find [X] matrix as

$$[X] = \{O\}_I \langle d^* \rangle = \{I\}_I \langle d^* \rangle$$

$1 \times m \quad l \times 1 \quad 1 \times m \quad l \times 1 \quad 1 \times m$

Step14: Find

$$[\Delta V]^{t+1} = \alpha[\Delta V]^t + \eta[X]$$

$1 \times m \quad 1 \times m \quad 1 \times m$

Step15: Find

$$[V]^{t+1} = [V]^t + [\Delta V]^{t+1}$$

$[W]^{t+1} = [W]^t + [\Delta W]^{t+1}$

Step16: The error rate can be find as

error rate = $\dfrac{\sum E^P}{nset}$

Step17: Repeat from step 4 to 16 until the convergence in the error rate is less than the tolerance value is achieved.

Step18: End of Algorithm.

Fig. 4: Enhanced ANN Algorithm.

In enhanced ANN, training of neural networks is iterated until a given condition is satisfied.

B. Training and Validation of ANN

In this proposal a back-propagation learning algorithm [51-53] is selected for training and validation of neural networks. BPN is systematic method for training a multilayer ANN. The central idea behind this solution is that errors for the units of the hidden layer are determined by back propagating the errors of the output layer. Back-propagation can also be considered as a generalization of the delta rule for non-linear activation functions and multi layer networks.

The back-propagation algorithm minimizes the squared error between the ANN output and the target. A hyperbolic activation function was used in all neurons to set the learning rate of hidden neurons and a learning rate for output neurons.

A standard ANN software package, neuralworks explorer , was used to perform training and validation of neural networks for: Networks with fixed and varying link reliabilities. After preliminary experiments, the architecture of ANN consists of 107, 70, and 1 neurons in input, hidden and output layers, respectively. The ANN models were trained for 500000 epochs, that is 500000 passes through the training set, with the normalized cumulative delta rule (learning rule) with 10^{-3} and 10^{3} learning coefficients for the hidden layer and output layer, respectively, using Neuralworks Predict software package [54].

IV. Computational Results

This proposal compares Monte Carlo simulation and Particle Swarm Optimization (MCS-PSO) mentioned in [12], [40]-[46] with enhanced ANN approach to solve optimal design of variable sized networks. The results obtained shows that the ANN method provides optimal designs with high reliability are possible when method includes static and variable links measurement into optimal design problem objectives.

A. Optimal Network Design with Fixed Link Values

The problem of measuring reliability can be simplified by limiting the links chosen in a network topology with same reliability i.e., with $K = 2$ because the number of possible topologies grows exponentially with increase in links from Eq. (6). In this case, if $X_{ij} = 1$, the link is chosen for the network topology and if $X_{ij} = 0$, no link is present. To make the ANN more applicable to a variety of design problems, five different values of link reliability were chosen to be included in a single ANN.

The inputs to the ANN were:

a. The architecture of the network as indicated by a series of binary variables (*Xij*).
b. The length of the string of 0'*s* and 1'*s* is equal to (*N* (*N* – 1))/2.
c. The link reliability is chosen in between 0 and 1 may be (0.88, 0.90, 0.92, 0.95, 0.99).
d. The calculated upper-bound of network reliability using the method of [22]-[24] and Kaushik, Navdeep, and Kohli [14]-[17].

The ANN is used to measure the upper-bound of network reliability which is significantly improved for fixed link using Eq. (11).

$$R_K^{+}(G) \le 1 - \left[\sum_{i=1}^{N_{nc}} q_i^d \prod_{k=1}^{n_i} (1 - q_{k-1}^d) \prod_{k=n_i-1}^{i=1} (1 - q_k^d) \right] \quad (11)$$

B. Optimal Network Design with Variable Link Values

Network design problems with variable link values greatly expand the number of possible topologies of a network, also complicates the network design problem and computation of overall reliability of network. For real world example, consider a network with links value is $K = 10$, that is, network can take any nine reliability value or 0, which indicates that link is not present. For further clarification, for any network design problems, we can use any of five link reliabilities in any combination.

The inputs to ANN are:

a. The architecture of network is given by a series of real-value variables (X_{ij})
b. The length of string is given by $\dfrac{(N(N-1))}{2}$
c. The Konak, Smith [22]-[24] and Kaushik, Navdeep and Kohli [14]-[17] method is used to calculate the upper-bound reliability.

The upper-bound for variable link values in reliability calculation is improved using Eq. (12).

$$R(X) \le 1 - \left[\sum_{i=1}^{N_{nc}} \left(\prod_{(k,i)\varepsilon E_i} (1 - p_{ki} \right) \prod_{j=1}^{N_{nc}} \left(1 - \frac{\prod_{(k,j)\varepsilon E_j} (1 - p_{kj})}{(1 - p_{ij})} \right) \right] (12)$$

Where *p* is the reliability of a link and *E* is the set of links connected to a given node. The output of the enhanced ANN will be the measurement of variable sized network reliability.

First, we use MCS-PSO method to determine the targeted value of network reliability of each network in variable sized network is determined using the Monte Carlo simulation method explained in Section II. In this section, we compare the MCS-PSO method with enhanced ANN approach for cost optimization problems under reliability constraints. The MCS-PSO method evaluates a small-scale modified network problem taken form [12], [21], [41]. Fig. 3 is modified example of ARPANET, contains 9 nodes and 14 unreliable links.

This MCS-PSO method is implemented in NeuralWorks Predict [54]. The numerical parameters and data are presented in Table II for evaluating performance, and confirming the validity of MCS-PSO. We have used 25 particles, and dimension is 14. The reliability is in interval [0, 1], and maximum velocity and maximum position are set to 1. The cognitive factor, and social factor will be 0.8 because reliability is [0, 1]. The testing experiment is repeated 35 times. We have applied the MCS method for each particle 100000 replications to get $R_k^+(G)$. The penalty functions will be applied only when simulation result $R_k^+(G)$ does not satisfy lower bound of the constraint $R(X)$. The penalty function from Eq. (10) is used to measure specific cost of component's reliability $C(X)$ for the amplification parameter δ is set to 10 after number of trails. From Table III, we find that the solutions using the heuristic method to find the initial solution of the first particle to perform better than the method that initializes all particles randomly. In addition, our proposed method can have lower variance, lower average error, and the best solution closest to the optimal solution.

In the next section, we perform computation of reliability in variable sized networks for fixed and varying link reliabilities using enhanced ANN approach as discussed in section A and B of IV. A standard ANN software package, neural works explorer [54], was used to perform training and validation of neural networks for: Networks with fixed and varying link reliabilities. After preliminary experiments, the architecture of ANN consists of 107, 70, and 1 neurons input, hidden and output layers, respectively. The ANN models were trained for 300000 epochs, that is 300000 passes through the training set, with the normalized cumulative delta rule (learning rule) with 0.30 and 0.15 learning coefficients for the hidden layer and output layer, respectively, using Neuralworks Explorer software package [54]. All data sets are divided into five subsets to use the five-fold cross validation technique. The five-fold validation ANN used 4/5 of the data set for training and the remaining 1/5 data set for testing, where the testing set changed with each validation of ANN.

Table IV gives five-fold cross validation results in root mean squared error (RMSE) for the ANN models built with the data sets with homogenous link reliabilities. The error, which is used to calculate 0.0000* difference between Monte Carlo and ANN estimations of the network reliability When the RMSE columns of training and testing sets are examined, it can be seen that the ANN models built with *D*4 generate minimum average RMSE values of 0.02809 on the training, and 0.03639 on the testing sets. Ordering all data sets from the best to the worst according to their average RMSE values of testing sets, the sequence of *D*4, *D*3, *D*1, *and D2* is obtained. Upper-bound RMSE columns represent the RMSE of the upper-bound only (no ANN estimation) on the testing sets. It can also be seen that the ANN always improve upon the upper-bound estimates. Based on the test there are significant differences between ANN models with a p value of < 0.000 at α = 0.05. Table V shows pairs, mean differences and p values. As shown in this table that there are no statistically significant differences between the optimized ANN models built with *D*4 and *D*3, while other pairs are statistically significantly different [13-17].

Similar comparisons and tests were carried out for networks with varying links to determine the effects of data sets for the ANN performance. Table IV shows that the ANN models built with the D4 data set generate minimum average RMSE values of 0.03608 and 0.04510 on the training and testing sets, respectively. When data sets are ordered from the best to the worst according to their average RMSE values of testing sets, the sequence of *D*4, *D*3, *D*1 and *D*2 is obtained. It is also observed that each ANN model estimation always improve the upper-bound. Table V shows pairs, mean differences and p-values at α=0.01. While there are no statistically significant difference between ANN models built on *D*1 and *D*2, other pairs are statistically significantly different.

V. Conclusions and Future Research

In this paper, we have compared MCS-PSO [12] and an enhanced ANN approach for optimal design of highly reliable variable sized networks constructed under minimum cost and maximum reliability constrained.

The MCS-PSO was proposed for solving for complex network reliability by minimizing the cost of components that constituted the network under reliability constraints. Compared with previous methods [21], [41], MCS-PSO does not require knowing the approximate reliability functions to solve this network problem. From experiment results, MCS-PSO has proved to have better efficiency in solving to the extent of limited sized complex network reliability optimization problem as it can provide a solution which is closer to the exact solution.

However, there are limitations with MCS-PSO method in variance reduction techniques and sampling plan method for various MCS method that they still need to be standardized.

It can be seen from the result that the ANN models give unbiased results significantly better than the Monte Carlo results. The ANN estimations are statistically closer but significantly better than the Monte Carlo estimations than the upper-bound for variable sized networks. This model is developed and tested for 50 nodes with fixed and varying link reliabilities. The results show that optimized ANN models built with the data generated by experimental design considering connectivity and link produce more accurate results than those developed by random/experimental design considering system reliability.

The recommended approach is to use the ANN models to measure network reliability of all candidate designs during the topological optimization (network design) phase. Then, the network reliability for only the best design or for a few good designs can be exactly calculated. In this way, the computational efforts of exact reliability calculation using Monte Carlo estimation can be reduced. The neural network approach gives superior designs at manageable computational cost.

The proposed method of ANN can be used for modeling the reliability in large multi-model or commercial network design problems.

Table III: Solutions for MCS-PSO Method with Different R(X) Values

	R(X)=0.92		R(X)=0.96		R(X)=0.99		R(X)=0.935		R(X)=0.985		R(X)=0.995	
Repli-cation	Random Value	Heuristic Value	Random Value	Heuristic Value	Random Value	Heuristic Value	Random	Heuristic	Random	Heuristic	Modified Random	Modified Heuristic
1	1200.65	1160.05	1200.45	1184.15	1255.45	1236.96	1199.68	1170.29	1211.1	1209.85	1216.08	1208.02
2	1198.36	1160.05	1199.85	1183.59	1152.42	1234.56	1199.86	1171.19	1212.29	1208.45	1215.89	1208.00
3	1197.86	1160.05	1199.75	1185.69	1253.45	1235.9	1206.66	1171.96	1213.39	1210.95	1217.39	1207.44
4	1206.5	1164.65	1199.65	1183.39	1254.79	1233.65	1203.56	1171.95	1215.56	1206.65	1216.59	1207.67
5	1194.4	1160.05	1199.55	1184.15	1242.86	1238.78	1214.78	1171.68	1216.64	1208.56	1219.98	1207.09
6	1195.65	1160.05	1199.45	1186.78	1242.96	1237.65	1213.24	1171.45	1217.78	1209.45	1219.38	1207.61
7	1205.65	1160.05	1199.35	1187.78	1251.46	1232.76	1210.49	1171.54	1219.92	1206.86	1218.40	1208.47
8	1199.76	1160.05	1200.95	1188.9	1236.35	1234.54	1209.19	1171.68	1222.68	1210.85	1218.48	1208.10
9	1188.95	1160.05	1200.86	1186.2	1238.45	1239.68	1208.48	1171.96	1221.65	1205.29	1218.25	1206.89
10	1189.65	1160.05	1200.76	1187.1	1239.56	1236.9	1205.46	1171.29	1220.24	1204.86	1217.42	1207.83
11	1198.55	1165.65	1200.48	1185.09	1240.49	1234.95	1201.89	1171.47	1223.96	1209.69	1216.52	1207.92
12	1185.76	1160.05	1208.45	1184.08	1242.89	1233.43	1194.69	1170.29	1211.95	1208.39	1218.04	1208.61
13	1198.05	1160.05	1209.65	1183.07	1241.78	1231.67	1190.56	1171.19	1212.45	1210.45	1217.96	1208.50
14	1205.63	1160.05	1209.75	1182.06	1245.96	1232.98	1191.96	1171.19	1213.59	1205.88	1218.14	1208.19
15	1200.65	1161.45	1209.85	1188.98	1246.56	1234.34	189.12	1171.19	1214.56	1206.96	1213.27	1208.38
16	1195.08	1162.34	1209.95	1183.76	1247.68	1238.61	1188.86	1171.96	1215.18	1207.19	1217.89	1206.83
17	1204.56	1163.68	1204.45	1185.6	1248.85	1237.23	1187.89	1171.96	1216.62	1203.49	1215.91	1207.21
18	1203.05	1164.48	1204.65	1184.56	1249.65	1238.54	1193.44	1171.96	1217.75	1205.95	1216.62	1206.91
19	1202.95	1165.65	1204.75	1187.78	1250.55	1235.98	1194.86	1171.45	1218.89	1207.78	1216.84	1207.94
20	1203.36	1160.05	1204.86	1186.45	1251.62	1234.85	1196.46	1171.54	1219.92	1209.63	1217.10	1208.02
21	1202.91	1160.05	1204.96	1185.58	1253.45	1233.89	1197.62	1171.34	1221.45	1204.86	1217.31	1208.22
22	1203.48	1161.45	1198.45	1184.29	1254.87	1236.75	1198.59	1171.96	1220.34	1205.95	1215.13	1207.17
23	1198.88	1162.34	1198.67	1182.69	1244.39	1234.52	1199.65	1171.96	1211.34	1206.66	1215.41	1207.43
24	1190.86	1161.45	1198.87	1183.89	1243.45	1231.43	1201.96	1171.19	1212.68	1207.97	1215.95	1208.61
25	1202.68	1162.34	1198.97	1184.15	1241.86	1232.68	1205.86	1171.29	1213.95	1208.2	1216.92	1208.35
26	1201.65	1163.68	1197.67	1188.65	1240.68	1239.65	1204.88	1171.47	1214.85	1205.67	1216.20	1207.39
27	1206.97	1164.48	1196.56	1187.96	1239.65	1240.97	1203.67	1171.86	1218.86	1203.96	1215.49	1206.95
28	1185.56	1163.68	1196.65	1186.78	1238.78	1236.14	1206.54	1171.96	1219.98	1209.49	1216.29	1207.48
29	1184.98	1165.65	1196.85	1185.65	1236.34	1235.24	1208.55	1170.29	1222.56	1208.86	1216.97	1208.43
30	1197.76	1160.85	1196.97	1184.56	1237.87	1233.39	1212.96	1171.45	1223.54	1205.37	1218.59	1208.17
31	1206.34	1160.78	1200.45	1183.59	1248.34	1234.87	1214.56	1171.65	1211.54	1206.76	1220.10	1207.71
32	1205.68	1160.87	1200.45	1182.87	1246.65	1235.88	1215.67	1171.68	1216.87	1205.97	1220.51	1207.45
33	1204.45	116095	1200.45	1185.45	1244.45	1240.96	1190.65	1171.76	1217.57	1206.69	1214.71	1206.87
34	1203.49	1160.05	1200.45	1187.96	1253.78	1239.29	1189.56	1171.86	1223.37	1210.89	1214.58	1207.17
35	1202.21	1160.05	1200.45	1184.86	1254.65	1235.88	1188.25	1171.88	1219.96	1207.76	1214.44	1207.47
36	1201.48	1160.05	1200.45	1184.15	1255.05	1236.45	1187.34	1171.89	1218.86	1207.86	1214.34	1207.30
37	1185.62	1160.05	1200.45	1182.98	1243.68	1231.96	1195.46	1171.95	1220.85	1207.96	1215.37	1207.98
38	1186.67	1160.05	1200.45	1183.86	1242.65	1232.34	1194.68	1171.96	1214.45	1208.96	1215.25	1207.95
39	1188.69	1163.68	1203.45	1185.65	1249.64	1237.76	1193.34	1171.19	1212.71	1208.76	1216.18	1207.64
40	1189.59	116096	1204.86	1184.15	1253.65	1238.66	1192.45	1171.29	1213.39	1207.56	1216.57	1207.32

Random	R	Net-R	Avg. Abs.	Max. Abs.	RMS	Accuracy (20%)	Conf. Interval (95%)	Records
All	0.03222802	-0.01899077	32.18199	1212.125	189.3302	0.7682927	374.6904	40
Train	0.03266806	-0.02159939	23.82949	1212.125	160.581	0.7894737	320.3186	27
Test	0.03963978	-0.02218639	51.22569	1212.125	242.4499	0.72	500.1256	25
Heuristic	R	Net-R	Avg. Abs.	Max. Abs.	RMS	Accuracy (20%)	Conf. Interval (95%)	Records
All	0.279230	0.276474	1.48002	3.724365	1.825757	0.55	3.682468	40
Train	0.261526	0.260266	1.48004	3.724365	1.827568	0.5555556	3.753112	27
Test	0.279230	0.276474	1.48002	3.724365	1.825757	0.55	3.682468	40

Table IV: Five-fold Cross Validation Results for Fixed and Varying Link Reliabilities

Experiment Different Data sets	Five-fold Cross Validation Results for Fixed and Varying Link Reliabilities					
	Fixed Link Reliability			Varying Link Reliability		
	G = (50, 1225)			G = (50, 1225)		
	RMSE			RMSE		
	Training	Testing	Upper-bound	Training	Testing	Upper-bound
Results for D1						
1	0.0326	0.04201	0.07272	0.04774	0.06066	0.09277
2	0.03337	0.03937	0.07034	0.04944	0.05041	0.08848
3	0.03279	0.03995	0.06863	0.04966	0.05425	0.08439
4	0.03465	0.03935	0.07301	0.05056	0.05248	0.08362
5	0.03377	0.03603	0.06307	0.04857	0.05725	0.08916
Average	**0.033436**	**0.039342**	**0.069554**	**0.049194**	**0.05501**	**0.087684**
Results for D2						
1	0.04505	0.05279	0.07661	0.05152	0.06059	0.09947
2	0.03722	0.04815	0.08575	0.05247	0.05788	0.09095
3	0.03876	0.04036	0.07011	0.05132	0.05777	0.09979
4	0.03858	0.04001	0.07169	0.05123	0.05858	0.09379
5	0.03852	0.04776	0.07796	0.05144	0.05959	0.09813
Average	**0.039626**	**0.045814**	**0.076424**	**0.051596**	**0.058882**	**0.096426**
Results for D3						
1	0.0291	0.03853	0.06251	0.03967	0.05061	0.08505
2	0.02826	0.04213	0.06936	0.03897	0.04703	0.08184
3	0.0299	0.03279	0.0728	0.0405	0.05202	0.0803
4	0.02994	0.0334	0.07345	0.04111	0.04626	0.08473
5	0.02856	0.0394	0.07443	0.03988	0.04561	0.07621
Average	**0.029152**	**0.03725**	**0.07051**	**0.040026**	**0.048306**	**0.081626**
Results for D4						
1	0.02743	0.03595	0.05888	0.03679	0.04468	0.07657
2	0.02732	0.04129	0.07156	0.03588	0.04582	0.07606
3	0.02958	0.02966	0.0629	0.03525	0.04405	0.07132
4	0.02797	0.03713	0.06635	0.0375	0.04652	0.07388
5	0.02794	0.0379	0.06793	0.03499	0.04443	0.07613
Average	**0.028048**	**0.036386**	**0.065524**	**0.036082**	**0.0451**	**0.074792**

Table VI: Comparisons Result between Pairs of Data Set for Fixed and Varying Link Reliabilities

	Fixed Link Reliability		Varying Link Reliability	
Pairs	Mean Difference	p-value	Mean Difference	p-value
D4-D3	-0.00105	1.22E-01	-0.00347	0.0018*
D4-D1	-0.00567	5.97E-10*	-0.01261	0.0000*
D4-D2	-0.00883	0.0000*	-0.01591	0.0000*
D3-D1	-0.00463	5.85E-07*	-0.00915	2.47E-12*
D3-D2	-0.00778	1.68E-14*	-0.01244	0.0000*
D1-D2	-0.00316	1.40E-03*	-0.00329	0.0102

*: Represents Significant Difference

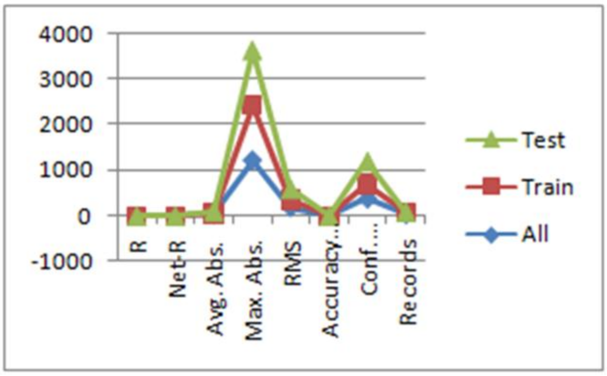

Fig. 5: Modified Random Plot for ANN Test

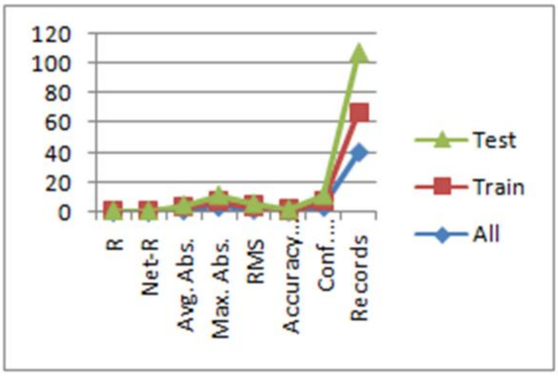

Fig. 6: Modified Heuristic Plot for ANN Test

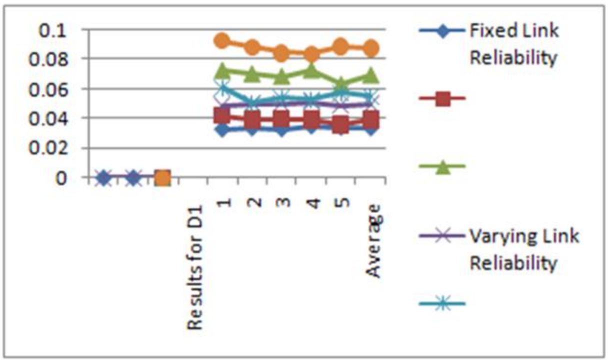

Fig. 7: Fixed Link Vs Varying Link for Data Set D_1

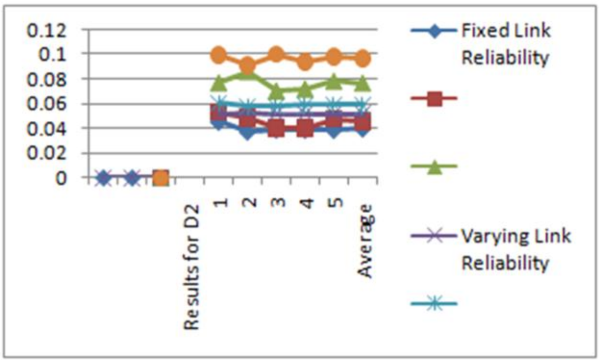

Fig. 8: Fixed Link Vs Varying Link for Data Set D_2

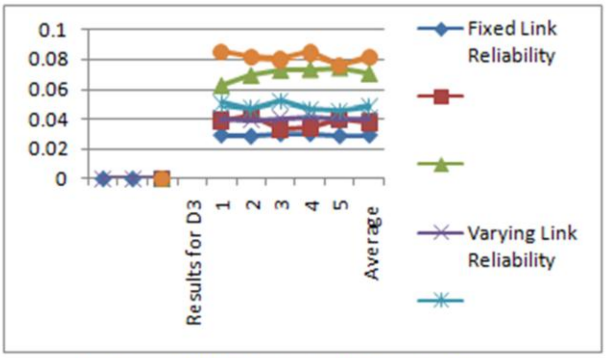

Fig. 9: Fixed Link Vs Varying Link for Data Set D_3

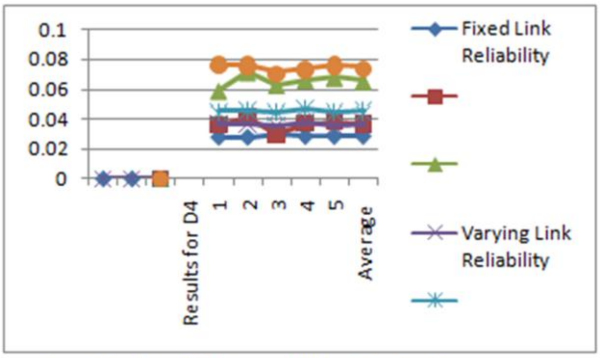

Fig. 9: Fixed Link Vs Varying Link for Data Set D_4

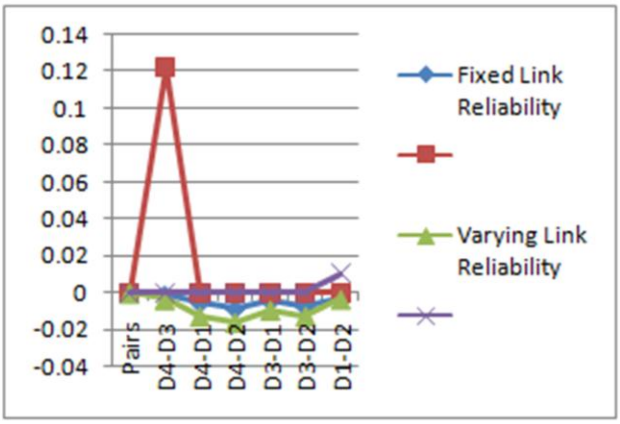

Fig. 10: Fixed Link Vs Varying Link for Data Set Pairs

Acknowledgment

First of all authors are thankful to almighty god. The authors are grateful to Krishna Engineering College, Ghaziabad for providing financial assistance. One of the author Baijnath Kaushik thankful to Dr. Navdeep Kaur and Dr. Amit Kumar Kohli for providing valuable suggestions and guidance helping him in completing this paper.

References

[1] Dai Y.S., Xie M. and Poh K.L. (2005) *IEEE Trans. Reliability*, 54(1), 100–106.

[2] Dai Y.S., Xie M., Long Q. and Ng S.H. (2007) *IEEE Trans. Software Engineering*, 33(11), 781–795.

[13] Inohira E. and Yokoi H. (2007) *Journal of Advanced Computational Intelligence and Intelligent Informatics,* 11(6), 593-594.

[14] Kaushik B., Kaur N. and Kohli A.K. (2011) *International Journal of Computer Science and Communication,* 2(1), 79-87.

[15] Kaushik B., Kaur N. and Kohli A.K. (2011) *International Journal of Computational Engineering & Management*, 11, 12-21.

[16] Kaushik B., Kaur N. and Kohli A.K. (2010) *IEEE xplore USA.*

[17] Kaushik B., Kaur N. and Kohli A.K. (2011) *International Journal Of Computer Technology & Applications,* 2(1), 1-14.

[18] Garey M.R. and Johnson D.S. (1979) *Computers and Intractability: A Guide to the*

Theory of NP Completeness, San Francisco: E.H. Freeman and Company.
[19] Venetsanopoulos A.N. and I. Singh (1986) *Problem of Control and Information Theory*, 15, 63-78.
[20] Coit D.W. and Smith A.E. (1996) *Computers and Operations Research*, 23, 515-526.
[21] Yeh W.C. and Lin C.H. (Accepted) A squeeze response surface methodology for finding symbolic network reliability functions, *IEEE Trans. Reliability*.
[22] Ratana C. S. and Smith A. E. (1999) Estimation of All-Terminal Network Reliability Using an Artificial Neural Network, *Computers & Operations Research*.
[23] Ratana C.S., Konak A. and Smith A.E. (2002) *Computers and Operations Research*, 29(7), 849–868.
[24] Altiparmak F., Dengiz B. and Smith A.E. (2008) *IEEE Transactions on Reliability*.
[25] Coit D.W. and Smith A.E. (1996) *IEEE Trans. Reliability*, 45(2), 254–260.
[26] Dengiz B., Altiparmak F. and Smith A.E. (1997) *IEEE Trans. Evolutionary Computation*, 1(3), 179–188.
[27] Dengiz B., Altiparmak F. and Smith A.E. (1997) *IEEE Transactions on Reliability*, 46, 18-26.
[28] Liang Y.C. and A. E. Smith (2004) *IEEE Trans. Reliability*, 53(3), 417–423.
[29] Meziane R., Massim Y., Zeblah A., Ghoraf A. and Rahli R. (2005) *Electric Power Systems Research*, 76(1), 1–8.
[30] Konak A., Smith A.E. and Coit D.W. (2003) *IIE Trans.*, 35(6), 515–526.
[31] Dengiz B., Alabas C. and Dengiz O. (2007) *Journal of Operation Research.*
[32] Ouzineba M. Nourelfatha M. and Gendreau M. (2008) *Reliability Engineering and System Safety*, 93(8), 1257–1272.
[33] Kennedy J. and Eberhart R.C. (1995) *Proceedings of IEEE International Conference on Neural Networks* NJ USA, 1942-1948.
[34] Levitin G., Hu X.H. and Dai Y.S. (2006) *Computational Intelligence in Reliability Engineering* New York USA, 83–112.
[35] Pandey M.K., Tiwari M.K. and Zuo M.J. (2007) *Proceedings of the Institution of Mechanical Engineers, Published in Journal of Risk and Reliability,* 177-221.
[36] Yeh W.C. (2009) *Expert Systems with Applications.*
[37] Dengiz O., Konak A. and Smith A.E. (2011) *Connectivity Management in Mobile Ad hoc Networks Using Particle Swarm Optimization*, *Ad Hoc Networks Journal* (accepted).
[38] Pierre S., Hyppolite M.A., Bourjolly J.M. and Dioume O. (1995) *Engineering Applications of Artificial Intelligence*, 8, 61-69.
[39] Shier D.R. (1991) *Network Reliability and Algebraic Structure, Oxford University New York..*
[40] Wang H. and Pham H. (1997) *Microelectronics and Reliability*, 37(2), 187–209.
[41] Yeh W.C. (2003) *International Journal of Advanced Manufacturing Technology*, 22, 681–688.
[42] Kamat S.J. and Riley M.W. (1975) *IEEE Trans. Reliability*, 24(1), 73–75.
[43] Kubat P. (1989) *IEEE Trans. Communications*, 37(9), 927–933.
[44] Lin J.Y. and Donaghey C.E. (1993) *Proceedings of the Annual Reliability and Maintainability Symposium* Atlanta GA USA.
[45] Wang H. and Pham H. (1997) *Microelectronics Reliability*, 37(2), 187–209.
[46] Majety S.R.V., Dawande M. and Rajgopal J. (1999) *Operations Research*, 47(6), 899–906.
[47] Kamat S.J. and Franzmeier W.E. (1976) *IEEE Trans. Reliability*, 25(3), 254–255.
[48] Fishman G.S. (1986) *Operations Research*, 34, 581–594.
[49] Landers T.L., Taha H.A. and King C.L. (1991) *IEEE Trans. Reliability*, 40(2), 177–181.
[50] Colbourn C.J. (1987) *The Combinatorics of Network Reliability, Oxford University Press New York.*
[51] Mehrotra K., Mohan C.K. and Sanjay Ranka (1996) *Elements of Artificial Neural Networks, MIT Press, Chapter 3,* 65-106.
[52] Karray F. and Silva C.D. (2000) *Soft Computing and Intelligent Systems Design-Theory, Tools and Applications, Addison Wesley, Chapter 5,* 249-293.
[53] Rajasekran S. and Vijaylakshmi P. (2005) *Neaural Networks, Fuzzy Logic, and Genetic Algorithms-Synthesis and Applications, Chapter 3, PHI,* 34-86.
[54] *Neuralworks: Reference Guide and Software from http:/ /www.neuralware.com (Various Years) NeuralWare Pittsburgh PA.*