[Sharma* *et al.*, 5.(6): June, 2016] ISSN: 2277-9655
IC™ Value: 3.00 Impact Factor: 4.116

IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

A GENETIC ALGORITHM FOR PERFORMANCE OPTIMIZATION OF A STOCHASTIC FLOW NETWORK WITH COST ATTRIBUTE IN TERMS OF MINIMAL CUTS

Diksha Sharma*, Pardeep Kumar
* Department Of Instrumentation,
Kurukshetra University, Kurukshetra-136119 (Haryana) INDIA

DOI: 10.5281/zenodo.54670

ABSTRACT

The paper presents a genetic algorithm to generate minimal cuts for a stochastic flow network with cost attribute to evaluate the performance index a stochastic- flow network in which each arc has several capacities i.e. reliability of the system. Calculate the system reliability such that the maximum flow is not less than a given demand. The algorithm is based on generating all the possible minimal cuts and computing reliability from those minimal cuts which satisfies the constraints and commodity conditions. The proposed algorithm can be used for a network with large number of nodes and arcs. Also, the paper investigates the problems that are found in the solutions that obtained by using other previous methods.

KEYWORDS: Genetic Algorithms, Stochastic-flow Network, Performance Index.

INTRODUCTION

In a SFN, the capacity of each arc has several possible levels. For instance, a telecommunication system can be treated as a SFN in which nodes stand for cities (or switches) and arcs for transmission lines. In fact, each transmission line consists of several physical lines, and each physical line has only successful or failure state. That implies that a transmission line has several states in which state k means k physical lines are successful. Hence, the capacity of each arc has several values and thus it is called stochastic, [11]. The system reliability of a flow network is the probability that the maximum flow of the network is not less than the given demand d. The capacity of each arc in this network, which is defined as the maximum flow passing the arc per unit time, has two levels 0 and positive integer, [10].

Lin [11],[13], constructed a stochastic-flow network to model the computer network. The studied problem is to evaluate the possibility that a given amount of multicommodity can be sent through an information network under the cost constraint. Lin [14], proposed a new performance index, the probability that the upper bound of the system capacity equals the demand vector subject to the budget constraint, to evaluate the quality level of a multicommodity limited-flow network. The researchers in this field have presented mainly two kinds of algorithms on the reliability evaluation of stochastic-flow networks: minimal paths (MPs) and Minimal Cuts (MCs). In this paper, the network reliability under a components-assignment can be computed in terms of minimal cuts. Many researchers have worked on this kind of problem using many conventional algorithms. In recent years, Genetic Algorithms (GAs) have been applied to various problems in the network design, [1-6],[8],[16]. Younes A. et al. [16], has proposed GA to evaluate the reliability of a stochastic flow network based on minimal paths (MPs) to find all lower boundary points for *d* and then calculated system reliability from those points.

In this paper genetic algorithm is propped to evaluate the performance index of a Stochastic Flow network. The proposed GA is based on minimal cuts (MCs).The paper is organized as follows: The assumptions and notation used given in **Section 2**. **Section 3** describes the problem of calculating the network reliability. **Section 4** presents the proposed algorithm for calculating the network reliability. The overall algorithm presented in **section 5**. In **Section 6** shows how to use the proposed algorithm to calculate the reliability of a stochastic-flow network for the example networks and presents the discussion.

NOTATIONS AND ASSUMPTIONS

Notations [13]:

SFN	Stochastic-Flow Network
SSFN,TSFN	Single-commodity stochastic-flow network, Two-commodity stochastic-flow network
$G(A,N,M)$	A stochastic-flow network with a unique source s and unique sink t having a set of branches $A = \{a_i \mid 1 \le a_i \le n\}$ with n number of branches
N	Set of nodes
X	Capacity vector; $X = (x_1, x_2, \ldots, x_n)$ where x_i is the current capacity for a_i, $i = 1, 2, \ldots, n$
F	Flow vector; $F = (fr_1, fr_2, \ldots, fr_{nr}, hr_1, hr_2, \ldots, hr_{nr})$
MCs	Minimal Cuts
M	Maximal capacity Vector; $M = \{M1, M2, \ldots, Mn\}$,where Mi is maximal capacity a_i for $i = 1,2,\ldots,n$
$C(X)$	Total cost under X : $\sum_{i=1}^{n} cixi$;where c_i is transmission cost per unit capacity of a_i, for $i = 1,2,., n$
R_d	System reliability to the given demand d.
B	budget
d	Demand at ***t*** in SSFN
d_i	Required quantity of commodity ***i*** at ***t*** in TSFN, $i=1, 2$
α_i	Consumed quantity of capacity on each branch per unit of commodity i, $i = 1, 2, \ldots, n$
K_r	MC #r, $r = 1, 2, \ldots, m$ where m is number of MCs
n_r	Number of branches in K_r
f_i	Flow of commodity 1 through ai, $i = 1, 2, \ldots, n$
h_i	Flow of commodity 2 through ai, $i = 1, 2, \ldots, n$
$V(X)$	System capacity under X in a TSFN
$W(X)$	System capacity under X in a SSFN
$U_{d1,\,d2,B}$	Probability that the system capacity is less than or equal to (d_1, d_2) subject to budget constraint B
B_i	$\{X \mid X \le X_i\}, i = 1, 2, \ldots, w$ where w is number of (d_1, d_2, B)- MCs
$[x]$	Largest integer such that $[x] \le x$
Ω	$\{X \mid X$ supports at most (d_1, d_2) and meeting $C(X) < B\}$: the set of candidates of (d_1, d_2, B)- MCs
popsize	Population size.
maxit	Maximum number of generations.
mutrate	GA mutation rate.
pathmem	Number of paths for each MC
iga	Generation counter

Assumptions:

i. The current capacity x_i is an integer-value random variable which takes value from $\{0, 1, 2 \ldots, M\}$ according to given probability distribution.
ii. Without loss of generality for αi, $\alpha_2 \ge \alpha_1 = 1$.
iii. Two types of commodities are transmitted from s to t.
iv. The capacities of different branches are statistically independent.
v. The flow in G must satisfy the flow-conservation law, [12].

PROBLEM DESCRIPTION

The Capacity vector X is a MC for demand and budget conditions known as (d_1, d_2, B) – MC if

i. System capacity (V) under capacity vector (X) = demands (d_1, d_2) i.e., $V(X) = (d_1, d_2)$
ii. Total cost (C) under capacity vector (X) is less than or equal to the budget (B) i.e., $C(X) \le B$
iii. System capacity (V) under capacity vector(Y) is greater than demands (d_1, d_2) and Total cost (C) under capacity vector (Y) i.e.,$V(Y) > (d_1, d_2) and\ C(Y) > B$, for any capacity vector (Y) with $Y > X$.

The set of (d_1, d_2, B) – MC is the set of maximal vectors in $\{X \mid V(X) = (d_1, d_2)\ \&\ C(X) \le B\}$.

Where $Y = (y_1, y_2, \ldots, y_n)$ and

$$(y_1, y_2, \ldots, y_n) \ge (x_1, x_2, \ldots, x_n): y_i \ge x_i \text{ for each } i = 1,2,\ldots,n$$

$$(y_1, y_2, \ldots, y_n) > (x_1, x_2, \ldots, x_n): (y_1, y_2, \ldots, y_n) \ge (x_1, x_2, \ldots, x_n) \ge and\ y_i > x_i \text{ for atleast one } i$$

$$(d_1', d_2') \ge (d_1, d_2): d_1' \ge d_1 \text{ and } d_2' \ge d_2$$

$$(d_1', d_2') > (d_1, d_2): (d_1', d_2') \geq (d_1, d_2) \text{ and } d_i' \geq d_i \text{ for atleast one } i$$

Therefore, we have following results:

***Lemma 1**:* If X is a *(d_1, d_2, B)* - MC, then $V(Y) \leq (d_1, d_2)$ *and* $C(Y) \leq B$ for each capacity vector Y with $Y < X$. Thus, the performance Index $Pr\{X \mid V(X) \leq (d_1, d_2)$ & $C(X) \leq B\}$ equals $Pr\{X|V(X) \leq X_i\}$ for a *(d_1, d_2, B)* - MC $X_i\}$. The following theorem shows a necessary condition for a capacity vector X to be *(d_1, d_2, B)* – MC.

Theorem 1: For each *(d_1, d_2, B)* – MC, there exists a $K_r = \{a_{r1}, a_{r2}, \ldots, a_{rn}\}$, and a $F = (f_{r_1}, f_{r_2}, \ldots, f_{r_{nr}}, h_{r_1}, h_{r_2}, \ldots, h_{r_{nr}})$ such that

$$\sum_{j=1}^{n_r} \mathrm{f_{rj}} = d_1 \ \& \sum_{j=1}^{nr} h_{rj} = d_2 \tag{I}$$

And

$$\begin{cases} x_{rj} = \alpha_1 f_{rj} + \alpha_2 h_{rj} & \text{for } j = 1, 2, \ldots, n \\ x_i = M_i & \text{for } \alpha_i \notin K_r \end{cases} \tag{II}$$

With respect to each K_r, we generate a X via (II) for each flow assignment $(f_{r_1}, f_{r_2}, \ldots, f_{r_{nr}}, h_{r_1}, h_{r_2}, \ldots, h_{r_{nr}})$ satisfying (I). For each $a_{rj} \in K_r$, x_{rj} supports at most *(f_{rj}, h_{rj})*. The branches in *Kr* play the roles as the parallel bridges in transportation. Hence K_r supports at most $(\sum_{j=1}^{nr} \mathrm{f_{rj}}, \sum_{j=1}^{nr} h_{rj}) = (d_1, d_2)$ That is, such an X supporting *(d_1, d_2, B)* – MC. Theorem 1 implies the following results.

***Lemma 2**:* Each (d_1, d_2, B) – MC is a candidate of *(d_1, d_2, B)* – MC.

Theorem 2 further shows that Ω is the set of *(d_1, d_2, B)* – MC.

Theorem 2: $\Omega = \{(d_1, d_2, B) - MC\}$, [13].

FLOW ASSIGNMENT IN TFSN

With respect to each MC, $K_r = \{a_{r1}, a_{r2}, \ldots, a_{rn_r}\}$, the vector $(f_{r1}, f_{r2}, \ldots, f_{rn_r}, h_{r1}, h_{r2}, \ldots, h_{rn_r})$ is feasible under the capacity vector $X = (x_1, x_2, \ldots, x_n)$ if $\alpha_1 f_{rj} + \alpha_2 h_{rj} \leq x_{rj}$ for j = 1,2, ..., Where $(\alpha_1 f_{rj} + \alpha_2 h_{rj})$ is the consumed quantity of the capacity on a_{rj} by the flows f_{rj} and h_{rj} . Such a vector is called a flow assignment (with respect to K_r). Under the capacity vector X, the MC K_r is said to support the demand *(d_1, d_2)* if there exists an $(f_{r1}, f_{r2}, \ldots, f_{rn_r}, h_{r1}, h_{r2}, \ldots, h_{rn_r})$ feasible under X such that $\sum_{j=1}^{n_r} f_{rj} = d_1$ & $\sum_{j=1}^{n_r} h_{rj} = d_2$

Under X, K_r is said to support at most *(d_1, d_2)* (i.e., K_r supports *(d_1, d_2)* but cannot support any (d_1', d_2') with $(d_1', d_2') > (d_1, d_2)$ if K_r supports *(d_1,d_2)*, and there exists no $(f_{r1}, f_{r2}, \ldots, f_{rn_r}, h_{r1}, h_{r2}, \ldots, h_{rn_r})$ feasible under X so that $\sum_{j=1}^{n_r} f_{rj} = d_1 + 1$ & $\sum_{j=1}^{n_r} h_{rj} = d_2$. Moreover, the capacity vector X is said to support *(d_1, d_2)* if, under X, all MC support *(d_1, d_2)*, and at least one MC supports at most *(d_1, d_2)*. In a TFSN, the system capacity *V(X)* is defined to be *(d_1, d_2)* if X supports at most *(d_1, d_2)*.

Without budget constraint, two performance indexes, $Pr\{X \mid V(X) \geq (d_1, d_2)\}$, and $Pr\{X \mid V(X) \leq (d_1, d_2)\}$, can be adopted to evaluate the performance for a TFSN; the former is the probability that the system capacity is larger than or equal to *(d_1, d_2)*, and the latter is the probability that the system capacity is less than or equal to *(d_1, d_2)*.

X satisfying $V(X) \leq (d_1, d_2)$, and the constraint $C(X) \leq B$ are calculated and the performance index, $U_{d_1, d_2, B} = Pr\{X|V(X) \leq (d_1, d_2)$ & $C(X) \leq B\}$ is evaluated, [13].

OBJECTIVE FUNCTION

Given the demand d_1 and d_2 and budget B, The Performance index of the system is defined by, Lin [13]:

$$U_{d_1 d_2 B} = Pr\{X|V(X) \leq (d1, d2) \ \& \ C(X) \leq B\} \tag{1}$$

Where X is the upper boundary point for demand and budget

And X can be deduced from $F = (fr_1, fr_2, \ldots, fr_{nr}, hr_1, hr_2, \ldots, hr_{nr})$ by using following equations:

$$x_{rj} = \alpha_1 f_{rj} + \alpha_2 h_{rj} \qquad \text{for } j = 1, 2, \ldots, n_r$$

$$x_i = M_i \qquad \text{for } \alpha_i \notin K_r \tag{2}$$

So, the main purpose of the proposed GA in this paper is to find the set of all feasible solutions of F that satisfies the flowing two constraints:

$$\alpha_1 f_{rj} + \alpha_2 h_{rj} \leq M_i \qquad \text{for } j = 1, 2, \ldots, n_r \tag{3}$$

$$\sum_{j=1}^{nr} \mathrm{frj} = d1 \ \text{ and } \ \sum_{j=1}^{nr} hrj = d2 \tag{4}$$

THE PROPOSED GENETIC ALGORITHM

This section describes the basic components of the proposed GA.

a) **REPRESENTATION**

If the network has m number of branches then the chromosome *CH* has *m* fields, each field represents the (current) flow on each branch i.e., $CH = \{x_1, x_2, . . . , x_m\}$ and for the example TFSN presented in this paper $m = 6$, therefore, $CH = \{x_1, x_2, x_3, x_4, x_5, x_6\}$

b) **INITIAL POPULATION**

A fixed number of chromosomes are randomly generated to form an initial population. A population size of 2000 *(popsize =2000)* is taken for the test problem in this dissertation work.

The initial population is generated according to the following steps:

Step 1: Randomly generate a chromosome CH in the initial population:
$CH = \{x_1, x_2, \dots, x_m\}$ according to the demands (d_1, d_2), where x_i is the current capacity for branches a_i, $i = 1, 2, \dots, n$ (n is number of branches)

Step 2: If the generated chromosome in step 1 doesn't satisfy eq. 4, discard it and go to step 1.

Step 3: Repeat steps 1 to 3 to generate *popsize* chromosomes.

c) **THE OBJECTIVE FUNCTION**

The problem can be formulated as:

Find the set of all feasible solutions F Such that equations (3) and (4) have been satisfied.

d) **EVALUATING FITNESS OF CHROMOSOMES**

The fitness of the chromosomes is calculated by using the objective function (discussed in section 5). All the chromosomes satisfying the flow demand and budget constraints i.e., the feasible ones are arranged in descending order of their fitness value and ascending order of the cost of transportation and are associated as rank 1 capacity vectors. A copy of these chromosomes is maintained as a pool of $(d_1, d_2, B) -$ MC solutions. All infeasible solutions are ranked in terms of their ascending order of violation of flow conditions and cost starting from 2. The population thus obtained is considered as the first generation population, [9].

e) **CROSSOVER OPERATOR**

In the proposed GA, similar to as proposed by A. Younes et al., [16], one-cut point crossover (i.e. an integer value is randomly generated in the range (0, m-1) where m is the length of the chromosome) is used to breed two offsprings (two new chromosomes) from two parents selected randomly according to selection value, as shown in the following example in Figure 1 (The network has 5 MCs).

Rank based weighted random pairing technique is used to perform the crossover operation and one-point crossover method is used to mate the parents and cut positions are generated randomly from the interval [1, *n*],similar to what proposed by Kumar Pardeep, [9].

f) **MUTATION OPERATOR**

An offspring undergoes mutation according to the mutation probability "*mutrate*".

Step 1: Generate a number $ii = 1$ to *nmut* (total number of mutations)

Step 2: If $ii < nmut$, the chromosome is chosen to mutate and go to step 3; otherwise skip this chromosome.

Step 3: For each offspring, randomly one component (randomly chosen row and column) of the chromosome is chosen and mutate/ changed to generate new offspring. This is done for each and every offspring. Figure 2 shows an example of performing the mutation operation on a given chromosome.

g) **TERMINATION CONDITION**

The execution of the GA is terminated when the number of generations exceeds the specified number of maximum generations or the set of all feasible solutions of F have been generated which does not cross the lower limit of minimum cost and also maximum number of iterations has been reached.

AN ALGORITHM FOR EVALUATING PERFORMANCE INDEX OF THE SYSTEM:

If $\Omega = \{X_1, X_2, . . . , X_w\}$, the performance index $U_{d1, d2,B}$ can be computed by :

$$Pr\{X \mid V(X) \le (d_1, d_2) \ \&\ C(X) \le B\}$$
$$= Pr\{X \mid V(X) \le X_i \text{ for a } (d_1, d_2,B)\text{- MC } X_i\}$$
$$= Pr\{B1 \cup B_2 \cup B_3 \cup B_4 \cup B_w\}$$

The following algorithm is presented in this paper to calculate $U_{d1, d2,B}$ according to the above rules:-

Step 1: Generate all possible intersections for all upper points of *X*.

Step 2: Calculate the probability (accumulative probability) for each *X* and also for each intersection.

Step 3: Calculate $U_{d1, d2,B}$ as follows:

Set $B_1 = \{X \mid X \leq X_1\}$, $B_2 = \{X \mid X \leq X_2\}, \ldots,\ B_q = \{X \mid X \leq X_q\}$

Apply inclusion-exclusion rule to calculate $U_{d1,\, d2,B}$ by using the relation**:**

$U_{d1,\, d2,B} = Pr\{B1 \cup B_2 \cup B_3 \cup B_4 \cup B_w\} = \sum_i Pr\{B_i\} - (-1)^2 \sum_{i<j} Pr\{B_i \cap B_j\} - (-1)^3 \sum_{i<j<k} Pr\{B_i \cap B_j \cap B_k\} - \ldots - (-1)^w Pr\{B_1 \cap B_2 \cap \ldots \cap B_w\}$

THE OVERALL ALGORITHM

This section presents the proposed GA for computing the performance index of a stochastic-flow network. The steps of this algorithm are as follows:

Step 1: Set the parameters: *popsize*, *mutrate*, *selection*, *pathmem*, *constraints*, *B*, *cost*, number of commodities and set *iga = 0.*

Step 2: Generate the initial population, as described in section 6.1.

Step 3: To obtain chromosomes for the new population; select two chromosomes from the parent population. Apply crossover, then mutate the new offsprings according to *mutrate*.

Step 4: If the new offspring satisfies the two constraints (equations 3 and 4) and it doesn't equal to any pre generated offspring, then keep it and increase ***ic***. If it fails to satisfy them, discard this offspring and reapply the mutation operator to the original parent.

Step 5: Set *iga =iga + 1*;

If *iga* > *maxit*, then go to step 3, otherwise goto step 3 to get a new generation.

Step 6: Report the set of all feasible solutions, FS and generate *X* from *F* using eq. 2.

Step 7: Suppose the result of step 6 is: $X_1, X_2, \ldots, X_w$. Then, obtain all upper boundary points for (d_1, d_2, B) by removing non-minimal ones in $X_1, X_2, \ldots, X_w$.

Step 8: Calculate $U_{d1,\, d2,B}$ according to the algorithm given in section 7 and note down the results.

EXPERIMENTAL RESULTS & DISCUSSION

This section shows how to use the proposed GA to calculate the Performance Index of a Two-commodity stochastic-flow network for an example network with two different budgets and presents the discussion of the obtained solutions

To illustrate the proposed algorithm for computing the system reliability, consider the following example network shown in Fig. 3 taken from [13]. This network has 4 nodes and 6 arcs. The arcs are numbered from a_1 to a_6. "*s*" is the source node and "*t*" is the sink node (destination). The capacity distribution of each component is given in Table 1 for the purpose. An International trade network is taken as shown in Figure 3. According to the law of Taiwan, no direct route from ***s*** (Shanghai, China) to ***t*** (Taipei, Taiwan) is permitted. Hence, each route from ***s*** should pass Hong Kong or Tokyo before arriving ***t***. The supplier would like to transmit 15-inch LCD monitors (commodity-1), and 17-inch LCD monitors (commodity-2) from Shanghai to Taipei. All commodities are loaded into the same type of container. The capacity of each arc is stochastic due to the fact that either container or traffic tools through each arc may be in maintenance, reserved by other suppliers or in other conditions. One unit of commodity means 120 homogeneous commodities. The supplier wants to have its 2 units of 15-inch LCD monitors, and 2 units of 17-inch LCD monitors (i.e. $d_1 = 2$, and $d_2 = 2$) to be transported from ***s*** to ***t***. The sizes of each container, 15-inch LCD monitor, and 17-inch LCD monitor, are $591 \times 230 \times 220\ (length \times bredth \times height)\ cm^3$, $92 \times 52 \times 38\ cm^3$, and $125 \times 70 \times 43\ cm^3$, respectively. Hence, one unit of commodity 1 (resp. 2) consumes 1 (resp. 2) container, i.e. $\alpha_1 = 1 (resp. \alpha_2 = 2)$. The cost c_i through each branch a_i is counted by the number of containers. There are 6 branches $(a_1, a_2, a_3, a_4, a_5, a_6)$ and 4 nodes (***s***: Shanghai, Tokyo, Hong Kong, and ***t***: Taipei).

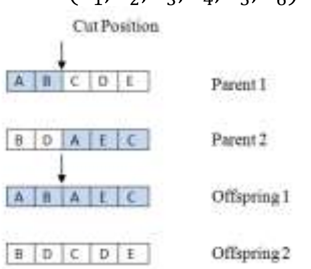

FIGURE 1. Single point crossover

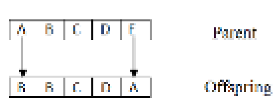

FIGURE 2. The mutation operator

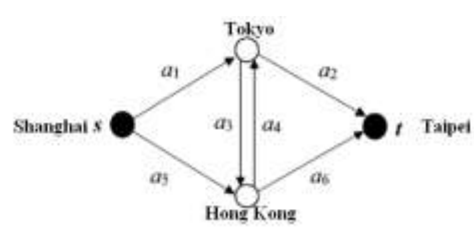

FIGURE 3. *International Trade network*

The data for the branches, capacities, cumulative probabilities, cost and consumed quantity in each commodity etc. are shown in Table 1.

Table 1. Data of Branches for Fig. 3, [13]

Branch	Capacity	Cumulative Probability	Cost **	α_1	α_2	Branch	Capacity	Cumulative Probability	Cost	α_1	α_2
a_1	4	1.00	250			a_4	4	1.00	150		
	3*	0.50					3	0.50			
	2	0.30					2	0.30			
	1	0.20					1	0.20			
	0	0.10					0	0.10			
a_2	3	1.00	250			a_5	3	1.00	250		
	2	0.40		1	2		2	0.40		1	2
	1	0.20					1	0.20			
	0	0.10					0	0.10			
a_3	3	1.00	300			a_6	3	1.00	150		
	2	0.40					2	0.40			
	1	0.20					1	0.20			
	0	0.10					0	0.10			

*the cumulative probability: $Pr\{x_1 \leq 3\} = 0.5$ ** US dollar

There are 4 minimal cuts of this above network:

$K_1 = \{a_1, a_3\}$ $K_2 = \{a_2, a_4\}$

$K_3 = \{a_1, a_4, a_5\}$ $K_4 = \{a_2, a_3, a_6\}$

The proposed algorithm will generate all (d_1, d_2, B)- MCs i.e. (2, 2, 4400) - MCs as follows:

I. For K_1:

(3 3 3 4 3 3) | (4 3 2 4 3 3)

For K_2:

(4 2 3 4 3 3) | (4 3 3 3 3 3)

For K_3:

(0 3 3 3 3 3) (1 3 3 4 1 3) (2 3 3 4 0 3) (3 3 3 3 0 3)
(0 3 3 4 2 3) (2 3 3 1 3 3) (3 3 3 0 3 3) (4 3 3 0 2 3)
(1 3 3 2 3 3) (2 3 3 2 2 3) (3 3 3 1 2 3) (4 3 3 1 1 3)
(1 3 3 3 2 3) (2 3 3 3 1 3) (3 3 3 2 1 3) (4 3 3 2 0 3)

For K_4:

(4 0 3 4 3 3) (4 2 1 4 3 3) (4 3 0 4 3 3) (4 3 2 4 3 1)
(4 1 2 4 3 3) (4 2 2 4 3 2) (4 3 1 4 3 2) (4 3 3 4 3 0)
(4 1 3 4 3 2) (4 2 3 4 3 1)

Similarly, all (2, 2, 4200)-MCs obtained are:

(3 3 3 4 3 3) (1 3 3 4 1 3) (3 3 3 3 0 3) (4 2 1 4 3 3)
(4 3 2 4 3 3) (2 3 3 1 3 3) (4 3 3 0 2 3) (4 2 2 4 3 2)
(4 2 3 4 3 3) (2 3 3 2 2 3) (4 3 3 1 1 3) (4 2 3 4 3 1)
(4 3 3 3 3 3) (2 3 3 3 1 3) (4 3 3 2 0 3) (4 3 0 4 3 3)
(0 3 3 3 3 3) (2 3 3 4 0 3) (4 0 3 4 3 3) (4 3 1 4 3 2)
(0 3 3 4 2 3) (3 3 3 0 3 3) (4 1 2 4 3 3) (4 3 2 4 3 1)
(1 3 3 2 3 3) (3 3 3 1 2 3) (4 1 3 4 3 2) (4 3 3 4 3 0)
(1 3 3 3 2 3) (3 3 3 2 1 3)

II. Obtain all upper boundary points for *(2,2,4400)* by removing non-minimal ones in $\{X_1, X_2, \ldots, X_{30}\}$, using the same algorithm in [15]. The only potential candidates for *(2,2,4400)-MC* are:

$X_1 = (3, 3, 3, 4, 3, 3)$, $X_2 = (4, 3, 2, 4, 3, 3)$, $X_3 = (4, 3, 3, 3, 3, 3)$, $X_4 = (4, 2, 3, 4, 3, 3)$, and $X_{22} = (4, 3, 3, 4, 3, 0)$

III. Calculate the performance index of the system for the given demand and budget constraint according to the algorithm given in section 4.8:

Set $B_1 = \{X \mid X \le X_1\}$, $B_2 = \{X \mid X \le X_2\}$, $B_3 = \{X \mid X \le X_3\}$, $B_4 = \{X \mid X \le X_4\}$ *and* $B_5 = \{X \mid X \le X_{22}\}$

Apply inclusion-exclusion rule to calculate $\mathbf{U}_{d1,\, d2,B}$ by using the relation**:**

$$U_{2,2,4400} = Pr\{B_1\} + Pr\{B_2\} + Pr\{B_3\} + Pr\{B_4\} + Pr\{B_5\} - Pr\,\{B_1 \cap B_2\} - Pr\,\{B_1 \cap B_3\} - \ldots - Pr\,\{B_1 \cap B_2 \cap B_3 \cap B_4\} + Pr\{B_1 \cap B_2 \cap B_3 \cap B_4 \cap B_5\}$$
$$= \Pr\{X \mid X \le (3,3,3,4,3,3)\} + Pr\{X|X \le (4,3,2,4,3,3)\} - \ldots + Pr\{X|X \le (3,3,2,3,3,0)\}$$
$$= 0.919$$

Similarly, the Performance index of the system for *(2, 2, 4200)* is: $U_{2,2,4200} = 0.8493$ and, the results of *F*, *X*, and the set of upper boundary points for that demand and budget are obtained.

The set of all feasible solutions of *F* is:

(1, 1, 1, 1), (2, 0 ,1 ,1), (0, 2, 2, 0), (2, 0 ,0 ,2), (0, 2, 1, 1), (0, 2, 0, 1, 1, 0), (0, 2, 0, 1, 0, 1), (0, 2, 0, 0, 1, 1), (0, 0, 2, 1, 1, 0), (0, 0, 2, 2, 0, 0), (0, 0, 2, 0, 2, 0),(0, 1, 1, 1, 1, 0), (0, 1, 1, 1, 0, 1), (0, 1, 1, 0, 1, 1), (0, 1, 1, 1, 0, 1), (0, 1, 1, 0, 1, 1), (0, 1, 1, 2, 0, 0), (1, 1, 0, 1, 1, 0), (1, 1, 0, 1, 0, 1), (1, 1, 0, 0, 1, 1), (1, 0, 1, 1, 1, 0), (1, 0, 1, 1, 0, 1), (1, 0, 1, 0, 1, 1), (1, 0, 1, 0, 2, 0), (2, 0, 0, 1, 1, 0), (2, 0, 0, 1, 0, 1), (2, 0, 0, 0, 1, 1) and (2, 0, 0, 0, 2, 0)

The set of *X* that can be obtained from *F* is as shown in Table 2:

TABLE 2: Set of Ω for (2, 2, 4200) - MCs

$X_1 = (3, 3, 3, 4, 3, 3)$	$X_{11} = (3, 3, 3, 1, 2, 3)$	$X_{21} = (4, 3, 3, 4, 3, 0)$
$X_2 = (4, 3, 2, 4, 3, 3)$	$X_{12} = (1, 3, 3, 3, 2, 3)$	$X_{22} = (4, 3, 1, 4, 3, 2)$
$X_3 = (4, 3, 3, 3, 3, 3)$	$X_{13} = (1, 3, 3, 4, 1, 3)$	$X_{23} = (4, 1, 3, 4, 3, 2)$
$X_4 = (4, 2, 3, 4, 3, 3)$	$X_{14} = (3, 3, 3, 2, 1, 3)$	$X_{24} = (4, 3, 2, 4, 3, 1)$
$X_5 = (2, 3, 3, 4, 0, 3)$	$X_{15} = (3, 3, 3, 0, 3, 3)$	$X_{25} = (4, 3, 0, 4, 3, 3)$
$X_6 = (2, 3, 3, 2, 2, 3)$	$X_{16} = (0, 3, 3, 3, 3, 3)$	$X_{26} = (4, 0, 3, 4, 3, 3)$
$X_7 = (4, 3, 3, 2, 0, 3)$	$X_{17} = (1, 3, 3, 2, 2, 3)$	$X_{27} = (4, 1, 2, 4, 3, 3)$
$X_8 = (4, 3, 3, 0, 2, 3)$	$X_{18} = (2, 3, 3, 1, 3, 3)$	$X_{28} = (4, 2, 1, 4, 3, 3)$
$X_9 = (0, 3, 3, 4, 2, 3)$	$X_{19} = (2, 3, 3, 3, 1, 3)$	$X_{29} = (4, 2, 3, 4, 3, 1)$
$X_{10} = (3, 3, 3, 3, 0, 3)$	$X_{20} = (4, 2, 2, 4, 3, 2)$	$X_{30} = (4, 3, 3, 1, 1, 3)$

The only potential candidates for ***(2, 2, 4200)*-MC** are:

$X_1 = (3, 3, 3, 4, 3, 3)$, $X_2 = (4, 3, 2, 4, 3, 3)$, $X_4 = (4, 2, 3, 4, 3, 3)$, $X_7 = (4, 3, 3, 2, 0, 3)$, $X_8 = (4, 3, 3, 0, 2, 3)$, $X_{21} = (4, 3, 3, 4, 3, 0)$ and $X_{30} = (4, 3, 3, 1, 1, 3)$

Note: For the studied network example, the parameters setting in the proposed algorithm are:

The population size (popsize) = 2000

The selection (fraction of population kept) = 0.5

The GA mutation rate (mutrate) = 0.2

The maximum number of generations (maxit) =100

Minimmum Cost (maxreli) = 0.9999000.

CONCLUSION & FUTURE WORK

This section investigates the problem of the obtained solution to the above examples given in Lin, [13]. For the given demand $\boldsymbol{d_1 = 2}$ and $\boldsymbol{d_2 = 2}$ and according to [13], the set of all feasible solutions of *F* is:

(1, 1, 1, 1), (2, 0 ,1 ,1), (0, 2, 2, 0), (2, 0 ,0 ,2), (0, 2, 1, 1), (0, 1, 1, 1, 1, 0), (0, 1, 1, 1, 0, 1), (0, 1, 1, 0, 1, 1), (0, 1, 1, 1, 0, 1), (0, 1, 1, 0, 1, 1), , (1, 1, 0, 1, 1, 0), (1, 1, 0, 1, 0, 1), (1, 1, 0, 0, 1, 1), (1, 0, 1, 1, 1, 0), (1, 0, 1, 1, 0, 1), (1, 0, 1, 0, 1, 1), (1, 0, 1, 0, 2, 0), (2, 0, 0, 1, 1, 0), (2, 0, 0, 1, 0, 1), (2, 0, 0, 0, 2, 0), (0, 1, 1, 2, 0, 0), (0, 2, 0, 1, 1, 0), (0, 2, 0, 1, 0, 1), (0, 2, 0, 0, 1, 1), (0, 0, 2, 1, 1, 0), (0, 0, 2, 2, 0, 0), (0, 0, 2, 0, 2, 0), (2, 0, 0, 0, 1, 1)

But, the last eight solutions (0, 1, 1, 2, 0, 0), (0, 2, 0, 1, 1, 0), (0, 2, 0, 1, 0, 1), (0, 2, 0, 0, 1, 1), (0, 0, 2, 1, 1, 0), (0, 0, 2, 2, 0, 0), (0, 0, 2, 0, 2, 0) and (2, 0, 0, 0, 1, 1) don't satisfy the constraint:

So, these eight solutions (0, 1, 1, 2, 0, 0), (0, 2, 0, 1, 1, 0), (0, 2, 0, 1, 0, 1), (0, 2, 0, 0, 1, 1), (0, 0, 2, 1, 1, 0), (0, 0, 2, 2, 0, 0), (0, 0, 2, 0, 2, 0) and (2, 0, 0, 0, 1, 1) must be eliminated and the set of solutions become: (1, 1, 1, 1), (2, 0 ,1 ,1), (0, 2, 2, 0), (2, 0 ,0 ,2), (0, 2, 1, 1), (0, 1, 1, 1, 1, 0), (0, 1, 1, 1, 0, 1), (0, 1, 1, 0, 1, 1), (0, 1, 1, 1, 0, 1), (0, 1, 1, 0, 1, 1), , (1, 1, 0, 1, 1, 0), (1, 1, 0, 1, 0, 1), (1, 1, 0, 0, 1, 1), (1, 0, 1, 1, 1, 0), (1, 0, 1, 1, 0, 1), (1, 0, 1, 0, 1, 1), (1, 0, 1, 0, 2, 0), (2, 0, 0, 1, 1, 0), (2, 0, 0, 1, 0, 1), (2, 0, 0, 0, 2, 0), which satisfy the constraints (3) and (4) and compatible with the solution obtained by the proposed GA by comparing the set of upper boundary points and the value of Performance Index for most of the cases. Even the proposed GA has more efficiency for the case *(2, 2, 4400)* – MCs as 30 MCs are obtained using GA while Lin [13] has obtained 29 MCs using conventional method. The reliability of the system is found out to be the same by both the methods.

Also, for the case of *(2, 2, 4200)* – MCs**,** the MCs obtained by Lin [13] are 29 whereas by the proposed GA, the MCs are 30 and the reliability for this case is also more than that, which had been calculated by Lin [13]. By proposed GA, $U_{2,2,4200} = 0.8493$ while by Lin [13], it had been calculated to be $U_{2,2,4200} = 0.84772$.

The time consumed for generating all MCs and calculating the System Reliability or Performance Index of the System is much less than the conventional method which is very time consuming and cumbersome method. Therefore, the proposed GA is optimal method to solve such problems of Stochastic- Flow networks under flow conditions and constraints.This paper presented a genetic algorithm to calculate the performance index of a stochastic-flow network with cost attribute to given demands d_1 and d_2 and budget B. The algorithm is based on determining the set of all feasible solutions of the vector using minimal cuts (MCs) and generates the set of all upper boundary points for the given demands d_1 and d_2 and budget B and then calculates the performance index in terms of capacity as system reliability. Finally we illustrate the use of the proposed algorithm by calculating the reliability of a flow network to given sample network taken from literature. Also, the algorithm is efficient for two commodity case and may be extended to compute the reliability of a flow network in multicommodity cases.

REFERENCES

[1] Berna Dengiz, Fulya Altiparmark & Smith Alice E., 1995, *A genetic algorithm approach to optimal topological design of all terminal networks*, Intelligent Engineering Systems through Artificial Neural Network, 5: 405-410.

[2] Berna Dengiz, Fulya Altiparmark & Smith Alice E., May 17-18, 1997, *Genetic algorithms design of networks considering all-terminal reliability*, The Sixth Industerial Engineering Research Conference Proceedings IERC'97, Miami Beach, Florida, USA: 30-35.

[3] Berna Dengiz, Fulya Altiparmak & Smith Alice E., 1997, *Local search genetic algorithm for optimization of highly reliable communications networks*, IEEE Transactions on Evolutionary Computation, 1: 179-188.

[4] Berna Dengiz & Aabbas C., 2001, *A simulated annealing algorithm for design of computer communication networks*, In Proceedings of world Multiconference on Cybernetics & Informatics-SCI 2001, Volume 5.

[5] Coit David W. & Smith Alice E., 1994, *Use of a genetic algorithm to optimize a combinatorial reliability design problem*, Proceeding of the Third IIE Research Conference: 467-472.

[6] Coit David W. & Smith Alice E., 1996, *Penalty guided genetic search for reliability design optimization*, Accepted to Computers & Industrial Engineering, Special Issue on Genetic Algorithms Vol. 30(4).

[7] Ford L. R. & Fulkerson D. R., 1962, *Flows in networks*, Princeton, New Jersey: Princeton University Press.

[8] Fulya Altiparmark, Berna Dengiz & Smith Alice E., 1998, *Reliability optimization of computer communication networks using genetic algorithms*, Proceedings of the 1998 IEEE International Conference on Systems, Man, & Cybernetics-Intelligent Systems For Humans In A Cyberworld, SMC'98, Hyatt Regency La Jolla, San Diego, California, USA, October 11-14: 4676-4681.

[9] Kumar Pardeep, june 2013, *A Genetic Algorithm for Reliability Evaluation Of A Flow Network Subject To Budget Constraint*, International Journal of Advanced Research in Electrical, Electronics & Instrumentation Engineering, volume 2, issue 6 : 2627-2635.

[10] Lin Yi-Kuei, 2001, *A simple algorithm for reliability evaluation of a stochastic-flow network with node failure*, Computers & Operations research, 28: 1277-1285.

[11] Lin Yi-Kuei, 2006, *Evaluate the performance of a stochastic-flow network with cost attribute in terms of minimal cuts*, Reliability Engineering & System Safety 91: 539–545.

[12] Lin Yi-Kuei, 2006, *Reliability of a computer network in case capacity weight varying with arcs, nodes & types of commodity*, Reliability Engineering & System Safety.

[13] Lin Yi-Kuei, March 2007, *Reliability of a Flow Network Subject to Budget Constraints*, IEEE Transactions on Reliability, vol 56, No. 1,10-16.

[14] Lin Yi-Kuei, 2007, *System reliability of a limited-flow network in multicommodity case*, IEEE Transactions on Reliability, 56(1): 17-25.

[Sharma* *et al.*, 5.(6): June, 2016] **ISSN: 2277-9655**
IC™ Value: 3.00 **Impact Factor: 4.116**
[15] Peixin Zhao & Xin Zhang, December 19-20, 2009, *A Survey on Reliability Evaluation of Stochastic-Flow Networks in terms of Minimal Paths*, Information Engineering & Computer Science, ICIECS, International Conference : 1- 4.

[16] Younes A.& Hassan M. R., 2011, *A Genetic Algorithm for Reliability Evaluation of a Stochastic-Flow Network with Node Failure*, International Journal of Computer Science & Security (IJCSS), Volume (4): Issue (6) : 528- 536.