

Genetic Algorithm Based Approach for Reliability Redundancy Allocation Problems in Fuzzy Environment

Laxminarayan Sahoo

Department of Mathematics
Raniganj Girls' College, Raniganj-713358, West Bengal, India
E-mail: lxsahoo@gmail.com

(Received November 30, 2016; Accepted February 10, 2017)

Abstract

This paper presents the use of genetic algorithm to solve reliability redundancy allocation problem of complicated system in fuzzy environment. Generally, this problem has been formulated as single objective integer non-linear programming problem with several resource constraints. In this paper, the reliability of each component as well as other parameters related to the problem is considered to be fuzzy valued. In this work, the corresponding constrained optimization problem has been transformed to crisp constrained optimization problem using defuzzification of fuzzy number. Here, widely known Yager ranking Index has been used for defuzzification of fuzzy number. The Big-M penalty function technique has been used to transform the constrained optimization problem into an unconstrained optimization problem. The converted problem has been solved with the help of real coded genetic algorithm. To illustrate the proposed methodology, a numerical example has been considered and solved. To study the performance of the proposed genetic algorithm, sensitivity analyses have been done graphically.

Keywords- Redundancy allocation problem, Genetic algorithm, Fuzzy number, Defuzzification technique, Yager Index.

1. Introduction

Due to the development of ultra-modern highly reliable system, the reliability of a system is highly dependent on the selection of components as well as increase of component reliability. Generally, selection of components as well as increase of component reliability are evaluated by solving nonlinear integer programming problem, which is known as reliability redundancy allocation problem. For such reliability system with known cost, component reliability, weight, volume and other parameters related to that system, the corresponding design problem is described as a constrained non-linear integer programming problem. The objective of the redundancy allocation problem is to select the number of redundant components that maximize the overall system reliability and/or minimize the overall system cost subject to the given linear/nonlinear resource constraints. As redundancy allocation problem is a nonlinear integer programming problem, so it cannot be solved easily by applying gradient based or mixed search methods due to discrete search space of the feasible region of the problem. The redundancy allocation problem has been well developed by Tillman et al. (1977), Kuo and Prasad (2001). In the literature, it has been noticed that several optimization methods, viz. heuristic methods (Nakagawa and Nakashima, 1977; Kim and Yum, 1993; Aggarwal and Gupta, 2005; Ha and Kuo, 2006), branch and bound method (Kuo et al., 1987; Sun and Li, 2002; Sung and Cho, 1999), reduced gradient method (Hwang et al., 1979), integer programming method (Misra and Sharma, 1991), dynamic programming (Nakagawa and Miyazaki, 1981; Hikita et al., 1992) etc. were used to solve such types of problem. Indeed, these methods have some advantages as well as disadvantages also. Dynamic Programming (DP) technique cannot be applicable in solving such complicated problem as the problem cannot be decomposing into several smaller problems. In branch and bound method (B & B), the efficiency depends on the sharpness of the bound of the

variables/parameters and required memory increases exponentially with the problem size. Therefore, as a result, with the development of genetic algorithm (Goldberg, 1989) and other computational optimization algorithms/ evolutionary algorithms/hybrid algorithm, most of the researchers in this field are interested to use these methods for solving reliability redundancy allocation problem. These methods give more flexibility, require fewer assumptions on the objective function as well as involved constraints and easier to implement. These methods are more efficient and well user friendly to use irrespective of whether the search space of the problem is discrete and/or continuous.

In earlier studies, the parameters related to the system have usually been considered to be precise/fixed values. This means that complete probabilistic information about the system is known in advance. In this case, it is considered that the probability value is exactly determinable for every probable event. However, in reality it is not true and there are not sufficient statistical data available in most of the cases when the system is either new or it exists only as a project. In this case the exact information about the system cannot be collected precisely due to human errors, improper storage facilities and other factors relating to prescribed environment. So, the reliability of a component of a system as well as design parameters will be considered as an imprecise number. In such cases, fuzzy set theory takes an important role to tackle such imprecise number (Zadeh, 1965). Reliability optimization in imprecise environment, one can refer to the works of Sahoo et al. (2012), Bhunia et al. (2010).

In this work, we have treated imprecise parameters considering fuzzy number. Therefore, the fuzzy valued redundancy allocation problem provides an efficient framework which solves reliability redundancy optimization problem of complex system with fuzzy valued reliabilities of components as well as fuzzy cost, fuzzy weight and fuzzy volume. In this paper, we have taken redundancy allocation problem of complicated/complex system. The reliability of each component and other parameters related to this problem has been considered to be fuzzy number and the corresponding problem has been transformed to crisp problem using different defuzzification method (Yager and Filev, 1993). Here, widely known Yager ranking Index method (Yager, 1981) has been used for defuzzification of the fuzzy parameters. The Big-M penalty function technique (Gupta et al., 2009) has been used to transform the constrained optimization problem into an unconstrained optimization problem. The converted problem has been solved by real coded genetic algorithm (Bhunia et al., 2010). Finally, a numerical example has been solved and the computed results have been presented. Also, sensitivity analyses have been done graphically to study the performance of the proposed method.

2. Mathematical Concept

2.1. Definitions About Fuzzy Number

Let X be a non-empty set. A fuzzy set \tilde{A} is defined as $\tilde{A} = \{(x, \mu_{\tilde{A}}(x)) : x \in X\}$, where $\mu_{\tilde{A}}$ is a mapping from $X \rightarrow [0,1]$. The real valued function $\mu_{\tilde{A}}(x)$ is called the membership function of \tilde{A} or degree of membership of x in \tilde{A} .

Definition 1: (α -cut). The α -cut of a fuzzy set \tilde{A} is a crisp subset of X and is denoted by $\tilde{A}_{\alpha} = \{x \in X : \mu_{\tilde{A}}(x) \geq \alpha\}$, where $\mu_{\tilde{A}}(x)$ is the membership function of \tilde{A} and $\alpha \in [0,1]$.

Definition 2: (Strong α -cut). If $\tilde{A}_\alpha = \{x \in X : \mu_{\tilde{A}}(x) > \alpha\}$, then it is called strong α -cut.

Definition 3: (Normal fuzzy set). A fuzzy set \tilde{A} is called a normal fuzzy set if there exists at least one $x \in X$ such that $\sup_x \mu_{\tilde{A}}(x) = 1$.

Definition 4: (Convex fuzzy set). A fuzzy set \tilde{A} is called convex iff for every pair of points $x_1, x_2 \in X$, the membership function of \tilde{A} satisfies the inequality $\mu_{\tilde{A}}(\lambda x_1 + (1-\lambda)x_2) \geq \min\{\mu_{\tilde{A}}(x_1), \mu_{\tilde{A}}(x_2)\}$, where $\lambda \in [0, 1]$.

Definition 5: (Fuzzy number). A fuzzy number \tilde{A} is a fuzzy set, must satisfy the following conditions.

- (i) There exists at least one $x_0 \in R$ for which $\mu_{\tilde{A}}(x_0) = 1$.
- (ii) $\mu_{\tilde{A}}(x)$ is pair wise continuous.
- (iii) \tilde{A} must be convex and normal.

Definition 6: (Triangular fuzzy number). The triangular fuzzy number (TFN) is a normal fuzzy number denoted as $\tilde{A} = (a_1, a_2, a_3)$ where $a_1 \leq a_2 \leq a_3$ and its membership function $\mu_{\tilde{A}}(x) : X \rightarrow [0, 1]$ is defined by

$$\mu_{\tilde{A}}(x) = \begin{cases} \frac{x - a_1}{a_2 - a_1} & \text{if } a_1 \leq x \leq a_2 \\ 1 & \text{if } x = a_2 \\ \frac{a_3 - x}{a_3 - a_2} & \text{if } a_2 \leq x \leq a_3 \\ 0 & \text{otherwise} \end{cases}$$

2.2. Yager's Ranking Method

The Yager's ranking method (Yager, 1981) is one of the best fuzzy ranking methods, which satisfies both linearity and additive property. Yager's ranking index for a fuzzy number \tilde{A} is denoted by $Y(\tilde{A})$ and is computed by the formula $Y(\tilde{A}) = 0.5 \int_0^1 (A_L(\alpha) + A_U(\alpha)) d\alpha$ where

$[A_L(\alpha), A_U(\alpha)]$ is the α -level interval of the fuzzy number \tilde{A} . The Yager's ranking index $Y(\tilde{A})$ gives the best representative value of the fuzzy number \tilde{A} .

Definition 7 (The Yager ranking index of Triangular Fuzzy Number). Let $\tilde{A} = (a_1, a_2, a_3)$ is a triangular fuzzy number. The α -level interval of \tilde{A} is $[a_1 + (a_2 - a_1)\alpha, a_3 - (a_3 - a_2)\alpha]$. Then the Yager's ranking index of $\tilde{A} = (a_1, a_2, a_3)$ is $Y(\tilde{A}) = \frac{1}{4}(a_1 + 2a_2 + a_3)$.

2.3 Ranking/ Order Relation of Fuzzy Numbers

Let $Y(\tilde{A})$ and $Y(\tilde{B})$ are the Yager ranking indices of two fuzzy numbers \tilde{A} and \tilde{B} respectively. Now, we have defined the ranking or order relations of two fuzzy numbers as follows: (i) $\tilde{A} > \tilde{B} \Leftrightarrow Y(\tilde{A}) > Y(\tilde{B})$, (ii) $\tilde{A} < \tilde{B} \Leftrightarrow Y(\tilde{A}) < Y(\tilde{B})$ and (iii) $\tilde{A} = \tilde{B} \Leftrightarrow Y(\tilde{A}) = Y(\tilde{B})$.

3. Problem Description of Reliability Redundancy Allocation Problem

The general form of the redundancy allocation problem in crisp form is as follows:

$$\begin{aligned} &\text{Maximize } R_S(x) && (1) \\ &\text{subject to} \\ &f_j(x) \leq b_j, \quad j = 1, 2, \dots, m \\ &\text{where } x = (x_1, x_2, \dots, x_n), \quad 1 \leq l_i \leq x_i \leq u_i, \quad x_i \text{ is integer, } i = 1, \dots, n, \quad b_j \text{ is the } j\text{-th available} \\ &\text{resource, } j = 1, 2, \dots, m. \end{aligned}$$

When all the parameters are fuzzy valued, then the problem (1) reduces to

$$\begin{aligned} &\text{Maximize } \tilde{R}_S(x) && (2) \\ &\text{subject to} \\ &\tilde{f}_j(x) \leq \tilde{b}_j, \quad j = 1, 2, \dots, m \\ &\text{where } x = (x_1, x_2, \dots, x_n), \quad 1 \leq l_i \leq x_i \leq u_i, \quad x_i \text{ is integer, } i = 1, \dots, n \text{ and } \tilde{b}_j \text{ is the } j\text{-th available} \\ &\text{resource, } j = 1, 2, \dots, m. \end{aligned}$$

Here $\tilde{R}_S(x)$, $\tilde{f}_j(x)$ and \tilde{b}_j are all fuzzy valued system reliability, j -th constraint and j -th resource respectively.

Now, using ranking of fuzzy number the problem (2) reduces to

$$\begin{aligned} &\text{Maximize } Y(\tilde{R}_S(x)) && (3) \\ &\text{subject to} \\ &Y(\tilde{f}_j(x)) \leq Y(\tilde{b}_j), \quad i = 1, 2, \dots, m \\ &\text{where } x = (x_1, x_2, \dots, x_n), \quad 1 \leq l_i \leq x_i \leq u_i, \quad x_i \text{ is integer, } i = 1, \dots, n \text{ and } Y(\tilde{b}_j) \text{ is the } j\text{-th} \\ &\text{available resource, } j = 1, 2, \dots, m. \end{aligned}$$

Here $Y(\tilde{R}_S)$, $Y(\tilde{f}_j)$ and $Y(\tilde{b}_j)$ are all crisp valued system reliability, j -th constraint and j -th resource respectively.

4. Complicated/Complex System

Let us consider a complicated system with ($n = 5, m = 3$) shown in Fig. 1. This system consists of five subsystems and three nonlinear and non-separable constraints. The expression of overall system reliability $R_S(x)$ is below:

$R_S(x) = R_5(x_5)(1 - Q_1(x_1)Q_3(x_3))(1 - Q_2(x_2)Q_4(x_4)) + Q_5(x_5)[1 - (1 - R_1(x_1)R_2(x_2))(1 - R_3(x_3)R_4(x_4))]$
 , where $Q_j(x_j) = 1 - R_j(x_j)$ for all $j = 1, 2, 3, 4, 5$ and $x = (x_1, x_2, x_3, x_4, x_5)$.

Since the overall system has a complex structure, the objective function is also nonlinear and non-separable.

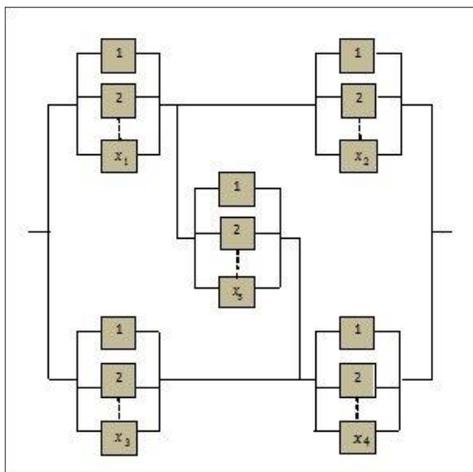


Fig. 1. A complex bridge system

5. Constraints Handling Technique and Solution Procedure

The problem (3) is a constrained optimization problem. To solve the problem (3) by using genetic algorithm we have transformed the considered optimization problem into an unconstrained optimization problem. In this work, we have used well known Big-M penalty function method. Using Big-M penalty function method, the corresponding unconstrained optimization problem is as follows:

$$\text{Maximize } Y(\tilde{R}_S) = \begin{cases} Y(\tilde{R}_S) & \text{if } x \in \Omega \\ -M & \text{if } x \notin \Omega \end{cases} \quad (4)$$

where $\Omega = \{x : Y(\tilde{f}_j(x_1, x_2, \dots, x_n)) \leq Y(\tilde{b}_j), j = 1, 2, \dots, m\}$ be the feasible space. Here, $Y(A)$

denotes the defuzzified value of the fuzzy number A using Yager Index. The problem (4) is a nonlinear unconstrained optimization problem with precise/fixed valued objective of n integer variables x_1, x_2, \dots, x_n .

Since the problem in (4) is nonlinear optimization problem with integer variables, so to find out analytic solution (if exist) to the problem is very difficult (Gen and Cheng, 1997). Also, efficient care of integer nonlinear optimization problem is one of the most difficulties in practical optimization problem (El-Sharkawi, 2008). Hence, to solve it, we need to use a heuristic/meta-heuristic algorithm.

A group of researcher has successfully used heuristic algorithms to solve the very complicated optimization problems. Some of these heuristic algorithms are tabu search, genetic algorithm, particle swarm optimization, ant colony optimization, differential evolution and bee colony optimization. Among these genetic algorithm is most popular.

5.1. Genetic Algorithm (GA)

The basic idea of Genetic Algorithm (Goldberg, 1989) is to replicate the natural evolution process artificially in which populations undergo continuous changes through genetic operators. Generally, genetic algorithm is very useful for solving large scale/complicated optimization problems which cannot be solved easily by direct or gradient based mathematical methods. It is very well to handle large-scale, real-life, discrete and continuous optimization problems avoiding unrealistic assumptions. To solve the problem (4) we have implemented Genetic Algorithm (GA) with advance genetic operators.

The procedure of Genetic Algorithm is described as follows:

```
begin  
  Initialize different parameters, bounds of variables of the problem;  
  set  $t \leftarrow 0$ ; //  $t$  represents the generation number  
  Initialize  $P(t)$ ; //  $P(t)$  represents the population at  $t$  – th generation;  
  evaluate the fitness function of each chromosome of  $P(t)$  with the help of interval arithmetic;  
  find the best chromosome from  $P(t)$  according to the fitness value using interval ranking;  
  while ( not termination condition) do  
    begin  
      set  $t \leftarrow t + 1$  ;  
      select  $P(t)$  from  $P(t-1)$  by selection process;  
      alter  $P(t)$  by crossover, mutation and elitism process;  
      evaluate  $P(t)$  ;  
      find the best chromosome from  $P(t)$  ;  
      compare the best results of  $P(t)$  and  $P(t-1)$  and store better one;  
    end  
  print the components of best chromosome with fitness value;  
end
```

The following basic operators/processes are to be considered for implementing the above GA in solving the problems.

- Selection operator
- Crossover operator
- Mutation operator

Now, we shall discuss the three operators of GA.

The selection operator is very popular and it is the first operator applied to the entire population and this operator plays an important role for selection. This operator selects the above average solutions and eliminate below average solutions from the entire population for the next

generation/iteration. Here, we have used tournament selection process of size two with replacement as the selection operator (Bhunia et al., 2010).

After selection, a crossover operator is applied to the selecting chromosomes which have survived. It operates on two or more parent solutions/chromosomes at a time and produces the offspring by recombining the feature of the parent solutions/chromosomes. In this work, we have used intermediate crossover.

The steps of intermediate crossover operation are as follows:

Step-1: Calculate $N_c = \text{Int}(P_c * P_s)$.

Step-2: Select two chromosomes $s_k^{(t)}$ and $s_i^{(t)}$ randomly from the entire population.

Step-3: (a) Generate $g = U(0, |s_{kj}^{(t)} - s_{ij}^{(t)}|)$, $j = 1, 2, \dots, n$

$$(b) \text{ Compute the components } \bar{s}_{kj}^{(t)} = \begin{cases} s_{kj}^{(t)} - g & \text{if } s_{kj}^{(t)} > s_{ij}^{(t)} \\ s_{kj}^{(t)} + g & \text{otherwise} \end{cases}$$

$$\text{and } \bar{s}_{ij}^{(t)} = \begin{cases} s_{ij}^{(t)} + g & \text{if } s_{kj}^{(t)} > s_{ij}^{(t)} \\ s_{ij}^{(t)} - g & \text{otherwise} \end{cases} .$$

Step-4: Compute $s_k^{(t+1)} = \text{argument of best of } \{ \text{fit}(s_k^{(t)}), \text{fit}(s_i^{(t)}), \text{fit}(\bar{s}_k^{(t)}), \text{fit}(\bar{s}_i^{(t)}) \}$

and $s_i^{(t+1)} = \text{argument of best of } \{ \text{fit}(s_k^{(t)}), \text{fit}(s_i^{(t)}), \text{fit}(\bar{s}_k^{(t)}), \text{fit}(\bar{s}_i^{(t)}) \}$

where $\text{fit}(s_k^{(t)})$ is the fitness value of the corresponding chromosome $s_k^{(t)}$.

Step-5: Repeat Step-2 and Step-4 for $\frac{N_c}{2}$ times.

Mutation operator take places the random variations into the population used to preserve the search process from converging to the local optimal solution. This operator helps to inculcate the information lost in earlier generations and responsible for fine tuning of the whole. This operator is applied to a single solution/chromosome only. In this work, we have used well known one-neighbourhood mutation (Bhunia et al., 2010).

The different steps of mutation operation are as follows:

Step-1: Calculate $N_m = [P_m * P_s]$.

Step-2: Select one parent chromosome $s_i^{(t)}$ randomly from the population.

Step-3: Select a gene $s_{ik}^{(t)}$ ($k = 1, 2, \dots, n$) on chromosome $s_i^{(t)}$ for mutation and domain of $s_{ik}^{(t)}$ is $[l_{ik}, u_{ik}]$.

Step-4: Generate $\rho = U(0,1)$.

Step-5: Create new gene $s_{ik}^{(t+1)}$ ($k = 1, 2, \dots, n$) corresponding to the selected gene $s_{ik}^{(t)}$ by mutation process as follows:

$$s_{ik}^{(t)} = \begin{cases} s_{ik}^{(t)} + 1 & \text{if } s_{ik}^{(t)} = l_{ik} \\ s_{ik}^{(t)} - 1 & \text{if } s_{ik}^{(t)} = u_{ik} \\ s_{ik}^{(t)} + 1 & \text{if } \rho < 0.5 \\ s_{ik}^{(t)} - 1 & \text{if } \rho \geq 0.5 \end{cases} .$$

Step-6: Compute $s_i^{(t+1)} = \text{argument of better of } \{ \text{fit}(s_i^{(t)}), \text{fit}(s_i^{(t+1)}) \}$.

Step-7: Repeat Step-2 to Step-6 for N_m times.

7. Numerical Example and Computational Results

In this section, we have considered a redundancy allocation problem in fuzzy environment. Here, in this example, the system reliability, reliability of each component and available resources are fuzzy valued and we represent the fuzzy number as a Triangular Fuzzy Number (TFN).

Example 1:

Maximize $\tilde{R}_S(x)$

subject to

$$(8,10,11) \exp\left(\frac{x_1}{2}\right) x_2 + (19, 20, 21) x_3 + (2.5, 3, 3.2) x_4^2 + (7.5, 8, 9) x_5 \leq (195, 200, 202) ,$$

$$(8,10,11) \exp\left(\frac{x_1}{2}\right) + (3.5, 4, 4.2) \exp(x_2) + (1.5, 2, 3) x_3^3 + (5, 6, 7) [x_4^2 + \exp\left(\frac{x_4}{4}\right)] + (6.5, 7, 7.3) \exp\left(\frac{x_5}{4}\right) \leq (300, 310, 315),$$

$$(11.5, 12, 13) [x_2^2 + \exp(x_2)] + (4, 5, 5.5) x_3 \exp\left(\frac{x_3}{4}\right) + (2.5, 3, 3.2) x_1 x_4^2 + (1.5, 2, 3) x_5^3 \leq (518, 520, 525),$$

$$(1, 1, 1, 1, 1) \leq (x_1, x_2, x_3, x_4, x_5) \leq (6, 3, 5, 6, 6),$$

where

$$\tilde{R}_S(x) = \tilde{R}_5(x_5) (1 - \tilde{Q}_1(x_1) \tilde{Q}_3(x_3)) (1 - \tilde{Q}_2(x_2) \tilde{Q}_4(x_4)) + \tilde{Q}_5(x_5) [1 - (1 - \tilde{R}_1(x_1) \tilde{R}_2(x_2)) (1 - \tilde{R}_3(x_3) \tilde{R}_4(x_4))]$$

$$R_1(x_1) = \{(.7, .8, .83), (.84, .85, .86), (.86, .90, .92), (.900, .925, .950), (.93, .95, .97), (.972, .975, .978)\};$$

$$R_2(x_2) = 1 - (1 - (0.72, 0.75, 0.76))^{x_2};$$

$$R_3(x_3) = \sum_{k=2}^{x_3+1} \binom{x_3+1}{k} ((0.86, 0.88, 0.89))^k ([0.11, 0.12, 0.14])^{x_3+1-k};$$

$$R_4(x_4) = 1 - (1 - (0.69, 0.70, 0.72))^{x_4};$$

$$R_5(x_5) = 1 - (1 - (0.84, 0.85, 0.86))^{x_5};$$

The Yager's ranking indices of all the fuzzy parameters related to the example-1 are presented as follows:

$$Y(8, 10, 11) = 9.75, Y(19, 20, 21) = 20.0, Y(2.5, 3.0, 3.2) = 2.925, Y(7.5, 8.0, 9.0) = 8.125,$$

$$Y(3.5, 4.0, 4.2) = 3.925, Y(1.5, 2.0, 3.0) = 2.125, Y(5, 6, 7) = 6.0, Y(6.5, 7.0, 7.3) = 6.95,$$

$$Y(11.5, 12.0, 13.0) = 12.125, Y(4.0, 5.0, 5.5) = 4.875, Y(2.5, 3.0, 3.2) = 2.925,$$

$$Y(195, 200, 202) = 199.25, Y(300, 310, 315) = 308.75, Y(518, 520, 525) = 520.75,$$

$$\begin{aligned}
 Y(0.70,0.80,0.83) &= 0.783, Y(0.84,0.85,0.86) = 0.85, Y(0.86,0.90,0.92) = 0.895, \\
 Y(0.900,0.925,0.950) &= 0.925, Y(0.93,0.95,0.97) = 0.95, Y(0.972,0.975,0.978) = 0.975, \\
 Y(0.72,0.75,0.76) &= 0.745, Y(0.86,0.88,0.89) = 0.878, Y(0.11,0.12,0.14) = 0.122, \\
 Y(0.69,0.70,0.72) &= 0.703
 \end{aligned}$$

Now, using defuzzified values of all the parameters, example 1 reduces to

Maximize $R_S(x)$

subject to

$$9.75 \exp\left(\frac{x_1}{2}\right)x_2 + 20x_3 + 2.925x_4^2 + 8.125x_5 \leq 199.25,$$

$$9.75 \exp\left(\frac{x_1}{2}\right) + 3.925 \exp(x_2) + 2.125x_3^3 + 6.0[x_4^2 + \exp\left(\frac{x_4}{4}\right)] + 6.95 \exp\left(\frac{x_5}{4}\right) \leq 308.75,$$

$$12.125[x_2^2 + \exp(x_2)] + 4.875x_3 \exp\left(\frac{x_3}{4}\right) + 2.925x_1x_4^2 + 2.125x_5^3 \leq 520.75,$$

$$(1,1,1,1,1) \leq (x_1, x_2, x_3, x_4, x_5) \leq (6,3,5,6,6),$$

where

$$R_S(x) = R_5(x_5)(1 - Q_1(x_1)Q_3(x_3))(1 - Q_2(x_2)Q_4(x_4)) + Q_5(x_5)[1 - (1 - R_1(x_1)R_2(x_2))(1 - R_3(x_3)R_4(x_4))]$$

$$R_1(x_1) = \{0.783, 0.850, 0.895, 0.925, 0.950, 0.975\};$$

$$R_2(x_2) = 1 - (1 - 0.745)^{x_2};$$

$$R_3(x_3) = \sum_{k=2}^{x_3+1} \binom{x_3+1}{k} (0.875)^k (0.123)^{x_3+1-k};$$

$$R_4(x_4) = 1 - (1 - 0.703)^{x_4};$$

$$R_5(x_5) = 1 - (1 - 0.85)^{x_5};$$

Now, we have solved the example 1 by proposed genetic algorithm. For this purpose, we have implemented the same in C programming code and corresponding computational work has been done on a Personal Computer (PC) with Intel Core-2 duo processor in LINUX operating system. For each problem, 50 independent trials/runs have been taken to obtain the best-found system reliability which is nothing but the optimal value of system reliability. In this computation, the values of genetic parameters, like Population Size (Ps), Maximum no. of Generation (Mg), Probability of Crossover (Pc) and Probability of Mutation have been taken as 200, 200, 0.85 and 0.15 respectively. The optimal computational results have been shown in Table 1.

$(x_1, x_2, x_3, x_4, x_5)$	System Reliability (R^*)	CPU time (in sec.)
(1,3,4,3,3)	0.999330	0.080

Table 1. Computational result

To study the efficiency of the proposed genetic algorithm for solving redundancy allocation problem, sensitivity analyses have been done graphically. The graphical representation of the analysis has been shown in Fig. 2-8. From Fig. 2 and 3, it is obvious that the system reliability is stable and converges to optimal solution for all the values of population size on and after 75.

From Fig. 4 and 5, it is observed that the system reliability (R_s) is stable and converges to optimal solution for all the values of Maximum no. of generation on and after 75. Fig. 6 depict the uniformity of the system reliability vs. the Probability of Crossover (varied from 0.60 to 0.90). Also from Fig. 7 and 8 it is seen that the system reliability is stable and converges to optimal solution for all the values of probability of Mutation on and after 0.12 (varied from 0.11 to 0.17).

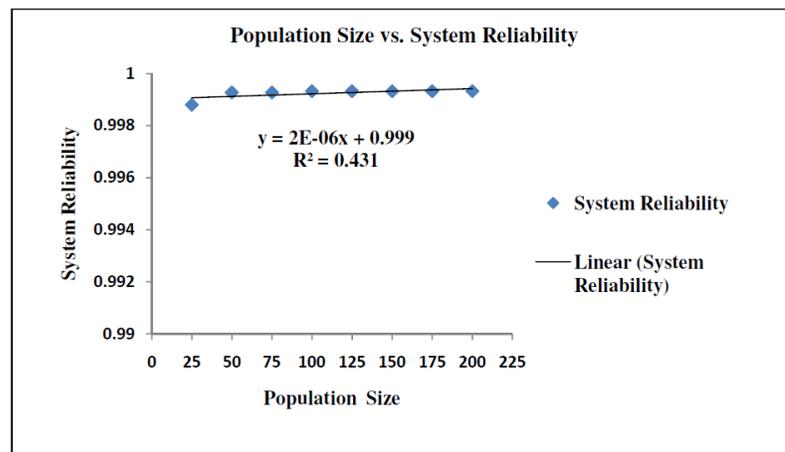


Fig. 2. Pc vs. Rs

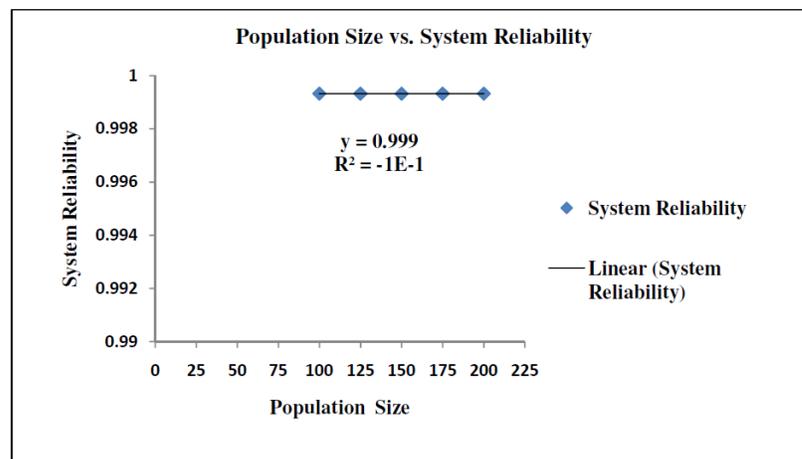


Fig. 3. Pc vs. Rs

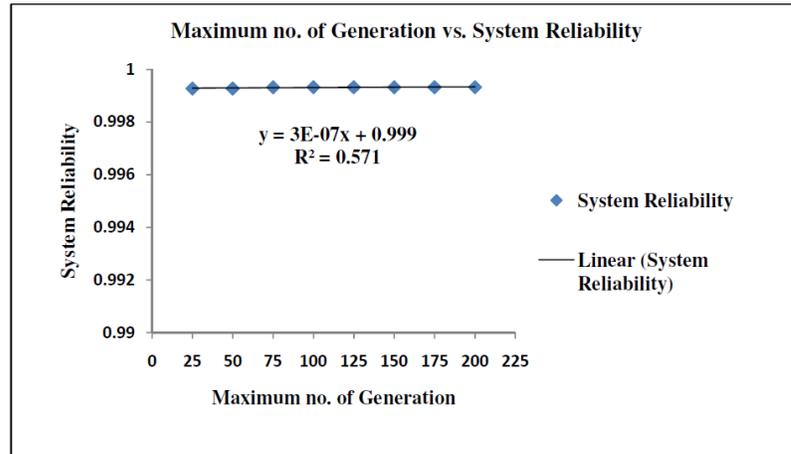


Fig. 4. Mg vs. Rs

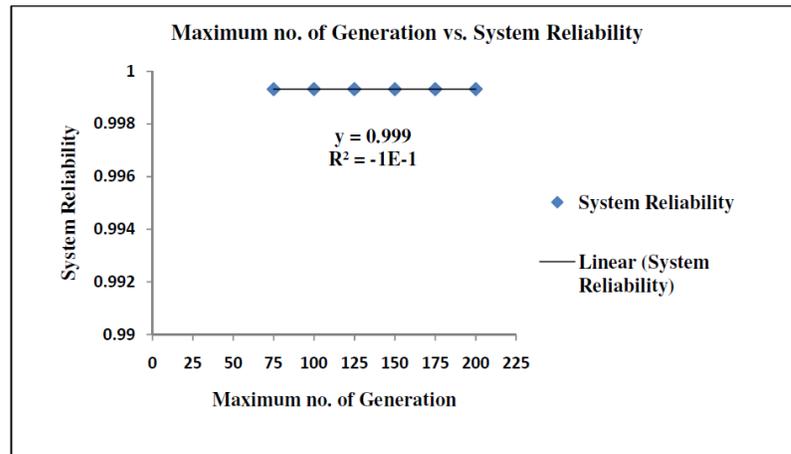


Fig. 5. Mg vs. Rs

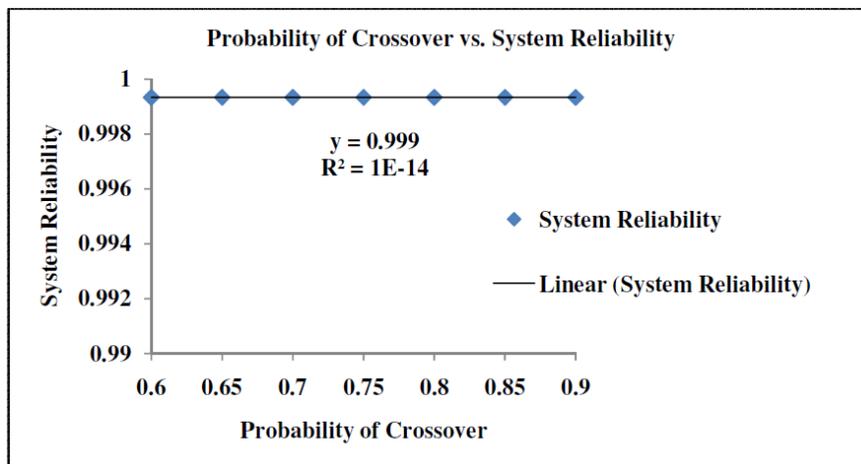


Fig. 6. Pc vs. Rs

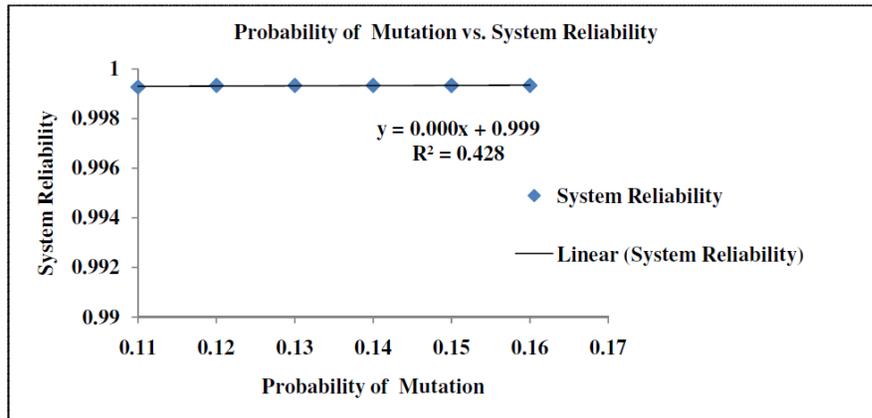


Fig. 7. Pm vs. Rs

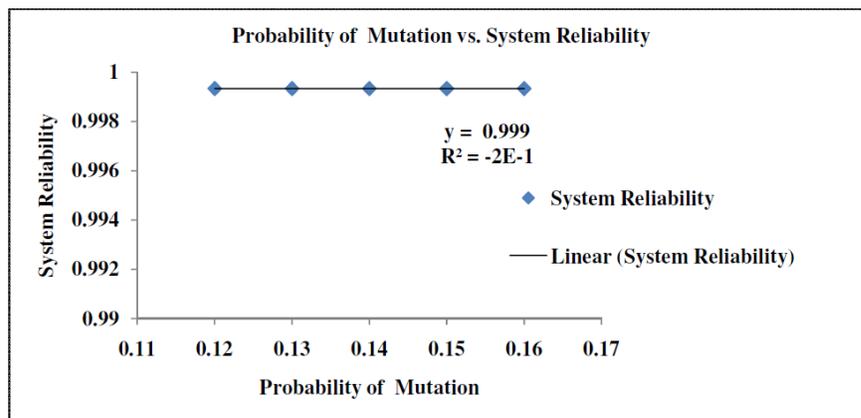


Fig. 8. Pm vs. Rs

From Figs 2-8, it is clear that the developed GA is stable and converges to its optimal solution. So, it can be claimed that the proposed GA described in the entire work is more appropriate to use.

8. Concluding Remarks

The component reliability in a complicated system might be imprecise rather than precise due to some reasons, like improper storage facilities, environmental conditions etc. This impreciseness may be represented by many ways. In this paper, we have represented this impreciseness by Triangular Fuzzy Number (TFN). Then the problem has been converted into crisp nonlinear programming problem after defuzzification of fuzzy number using Yager ranking index method in which the objective functions and all the constraints are converted into crisp valued. For further research, one may apply other heuristic methods, like, DE, SA, PSO, and other evolutionary algorithms for solving the problem presented in this paper. Finally, the computational results been presented and it can be concluded that the entire method attempted in this work well to solve large scale day to day life decision making problems in the near future having fuzzy parameters.

References

- Aggarwal, K. K., & Gupta, J. S. (2005). Penalty function approach in heuristic algorithms for constrained. *IEEE Transactions on Reliability*, 54(3), 549-558.
- Bhunia, A. K., Sahoo, L., & Roy, D. (2010). Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm. *Applied Mathematics and Computations*, 216(3), 929-939.
- El-Sharkawi, L. (2008). *Modern heuristic optimization techniques* (1st ed.). Wiley Inter Science, New Jersey, USA.
- Gen, M., & Cheng, R. (1997). *Genetic algorithm and engineering design* (1st ed.). USA, John Wiley & Sons, New York, NY.
- Goldberg, D. E. (1989). *Genetic algorithms in search, optimization and machine learning reading*. MA: Addison-Wesley, USA.
- Gupta, R. K, Bhunia, A. K., & Roy, D. (2009). A GA Based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliabilities of components. *Journal of Computational and Applied Mathematics*, 232(2), 275-284.
- Ha, C., & Kuo, W. (2006). Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems. *European Journal of Operational Research*, 171(1), 124-138.
- Hikita, M., Nakagawa, K., Nakashima, K., & Narihisa, H. (1992). Reliability optimization of systems by a surrogate-constraints algorithm. *IEEE Transactions on Reliability*, 41(3), 473-480.
- Hwang, C. L., Tillman, F. A., & Kuo, W. (1979). Reliability optimization by generalized lagrangian-function based and reduced-gradient methods. *IEEE Transactions on Reliability*, 28(4), 316-319.
- Kim, J. H., & Yum, B. J. (1993). A heuristic method for solving redundancy optimization problems in complex systems. *IEEE Transactions on Reliability*, 42(4), 572-578.
- Kuo, W., & Prasad, V. R. (2001). An annotated overview of system-reliability optimization. *IEEE Transactions on Reliability*, 49,176-187.
- Kuo, W., Lin, H., Xu, Z. & Zhang, W. (1987), Reliability optimization with the Lagrange-multiplier and branch-and-bound technique, *IEEE Transactions on Reliability*, 36(5), 624-630.
- Misra, K. B., & Sharma, U. (1991). An efficient algorithm to solve integer- programming problems arising in system-reliability design. *IEEE Transactions on Reliability*, 40(1), 81-91.
- Nakagawa, Y., & Miyazaki, S. (1981). Surrogate constraints algorithm for reliability optimization problems with two constraints. *IEEE Transactions on Reliability*, 30(2), 175-180.
- Nakagawa, Y., & Nakashima, K. (1977). A heuristic method for determining optimal reliability allocation. *IEEE Transactions on Reliability*, 26(3), 156-161.
- Sahoo, L., Bhunia, A. K., & Kapur, P. K. (2012). Genetic algorithm based multi-objective reliability optimization in interval environment. *Computer Industrial Engineering*, 62(1), 152-160.
- Sun, X. L., & Li, D. (2002). Optimization condition and branch and bound algorithm for constrained redundancy optimization in series system. *Optimization and Engineering*, 3(1), 53-65.
- Sung, C. S. & Cho, Y. K. (1999). Branch and bound redundancy optimization for a series system with multiple-choice constraints. *IEEE Transactions on Reliability*, 48(2), 108-117.
- Tillman, F. A., Hwang, C. L., & Kuo, W. (1977). Optimization technique for system reliability with redundancy: A Review. *IEEE Transactions on Reliability*, 26(3), 148-155.

Yager, R. R. (1981). A procedure for ordering fuzzy subsets of the unit interval. *Information Sciences*, 24(2), 143-161.

Yager, R. R., & Filev, D. P. (1993). On the issue of defuzzification and selection based on a fuzzy set. *Fuzzy Sets and Systems*, 55(3), 255-271.

Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8(3), 338-352.