ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 7, No. 4, 2012, *pp*. 284-295

Reliability-redundancy Optimization Problem with Interval Valued Reliabilities of Components via Genetic Algorithm

Sanat Kumar Mahato[1], Laxminarayan Sahoo[2+] and Asoke Kumar Bhunia[3]

[1] Department of Mathematics, Durgapur Govt. College, Durgapur-713214, West Bengal, India

[2] Department of Mathematics, Raniganj Girls' College, Raniganj-713358 ,West Bengal, India

[3] Department of Mathematics, The University of Burdwan, Burdwan-713104,West Bengal, India

(*Received May 10, 2012, accepted September 30, 2012*)

Abstract. This paper deals with the reliability-redundancy optimization problem considering the reliability of each component as an interval valued number that involves the selection of components with multiple choices and redundancy levels which maximize the overall system reliability subject to the given resource constraints arise on cost, volume and weight. Most of the classical mathematical methods have failed in solving the reliability-redundancy optimization problems because the objective functions as well as constraints are non-convex in nature. As an alternative to the classical mathematical methods, heuristic methods have been given much more attention by the researchers due to their easier applicability and ability to find the global optimal solutions. One of these heuristics is genetic algorithm (GA). GA is a computerized stochastic search method of global optimization based on evolutionary theory of Darwin: "The survival of the fittest" and natural genetics. Here we present GA based approach for solving interval valued mixed integer programming in reliability-redundancy optimization problem with advanced genetic operators. Finally, a numerical example has been solved and also to study the effects of changes of different GA parameters, sensitivity analyses have been carried out graphically.

Keywords: Reliability-redundancy optimization, Genetic algorithm, Mixed-integer programming, Interval mathematics, Interval order relations

1. Introduction

Development of the design of high-tech system depends on the selection of components and configurations to meet the functional requirements as well as performance specification. Measures of system performance are basically of four types: (i) reliability (ii) availability (iii) mean time to failure and (iv) percentile life. Reliability has been widely used and studied as the primary performance measure for non maintained systems and becomes an important concern now-a-days, because due to increase of complexity, sophistication and automation of high-tech systems, system reliability tends to decrease. In 1952, the Advisory group on the reliability of electronic equipment defined the reliability in a wider sense: reliability indicates the probability implementing specific performance or function of products and achieving successfully the objectives within a time schedule under a certain environment. For a system with known cost, reliability, weight, volume and other system parameters, the corresponding design problem becomes a combinatorial optimization problem. The most well known reliability design problems of this type are referred to as the reliability-redundancy optimization problems.

This type of problem is generally a nonlinear mixed-integer programming problem. To enhance the component reliability and providing redundancy while considering the tradeoff between the system performance and resources, optimal reliability design that aims to determine an optimal system-level configuration has long been a hot topic in reliability engineering design. Over the last four decades, a number of notable works has been done in this topic based on various system configurations, performance measures, optimization techniques and other features for the improvement of system reliability. Among these, one may refer to the recent works of Kuo and Prasad (2000), Kuo and Wan (2007), Zhao et al. (2007), Liang and

Smith (2004), Onishi and Kimura (2007), Aggarwal and Gupta (2005), Ha and Kuo (2006a, 2006b), Gen and Yun (2006), Coelho (2009), Sung and Cho (1999), Coit, Jin and Wattanapongsakorn (2004), Zafiropoulos and Dialynas (2004), Ramirez-Marquez, Coit and Kanok (2004), Ramirez-Marquez and Coit (2007), Cui et al. (2004), Painton and Campbell (1995), Utkin an Gurov (1999, 2001), Marseguerra and Podofillini (2005), Ravi and Reddy (2000), Kuo et al. (2001), Sun and Li Duan (2002) and Sun, Mckinnon and Li (2001). In their works, the reliabilities of the system components are assumed to be known and fixed positive number which lies between zero and one. However, in real life situations, the reliability of an individual component may not be fixed. It may fluctuate due to different reasons. It is not always possible for a technology to produce different components with exactly identical reliabilities. Moreover, the human factor, improper storage facilities and other environmental factors may affect the reliabilities of the individual components. So, the reliability of each component is sensible and it may be treated as a positive imprecise number. To tackle the problem with such imprecise numbers, generally stochastic, fuzzy and fuzzy- stochastic approaches are applied and the corresponding problems are converted to deterministic problems for solving them. In stochastic approach, the parameters are assumed to be random variables with known probability distributions. In fuzzy approach, the parameters, constraints and goals are considered as fuzzy sets with known membership functions or fuzzy numbers. On the other hand, in fuzzy-stochastic approach, some parameters are viewed as fuzzy sets/fuzzy numbers and others as random variables. However, it is a formidable task for a decision maker to specify the appropriate membership function for fuzzy approach and probability distribution for stochastic approach and both for fuzzy -stochastic approach. So, to avoid these difficulties for handling the imprecise numbers by different approaches, one may use interval number to represent the imprecise number, as this representation is the most significant representation among others. Due to this representation, the system reliability would be interval valued. To the best of our knowledge, only a very few researchers (Gupta (2009), Bhunia (2010) and Sahoo (2010, 2012)) have done their works considering interval valued reliabilities of components,

In this study, we have considered GA-based approach for solving mixed-integer reliability redundancy optimization problem considering the reliability of each component as interval valued. As the objective function of the redundancy allocation problem is interval valued, to solve this type of problem by GA, order relations for intervals numbers are essential. Over the last three decades, very few researchers proposed the order relations of interval numbers in different ways. Recently, Mahato and Bhunia [28] proposed the modified definitions of order relations with respect to optimistic and pessimistic decision maker's point of view for maximization and minimization problems separately. Very recently, Sahoo, Bhunia and Roy (2011) proposed the simplified definition of interval order relations ignoring optimistic and pessimistic decisions. However, it has been observed that both the definitions give the same result.

In this paper, we have considered the problem of reliability-redundancy optimization considering the reliability of each component as an interval valued number that maximizes the overall system reliability subject to the given resource constraints arise on cost, volume and weight. The corresponding problem has been formulated as non-linear mixed integer constrained optimization problem with interval objective and some of the variables (i.e., reliability of each component) are interval valued. For solving this problem, we have developed advanced genetic algorithm with interval valued fitness function and Big-M penalty technique. To illustrate the method, a numerical example has been solved. Finally, to test the performance of the developed method, sensitivity analyses have been performed graphically with respect to different GA parameters.

2. Nomenclature

n	number of subsystems
$x=(x_1,x_2,...,x_n)$	vector of the redundancy allocation for the system

$r_j = [r_{jL}, r_{jR}]$	interval valued reliability of j-th component
$r = (r_1, r_2, ..., r_n)$	vector of component reliability for the system
$R_S(.) = [R_{SL}(.), R_{SR}(.)]$	objective function for the overall system reliability
$[V_L, V_R]$	upper limit volume of the system which is interval valued
$[C_L, C_R]$	upper limit cost of the system which is interval valued
$[W_L, W_R]$	upper limit weight of the system which is interval valued
$g_i(.) = [g_{iL}(.), g_{iR}(.)]$	constraints functions, $i = 1, 2, ..., m$
$b_i = [b_{iL}, b_{iR}]$	availability of i-th resource ($i = 1, 2, ..., m$)
l_j, u_j	lower and upper bounds of x_j
α_j	lower bound of r_{jL}
β_j	upper bound of r_{jL} and lower bound of r_{jR}
γ_j	upper bound of r_{jR}
S	feasible region
pcross	probability of crossover/ crossover rate
pmute	probability of mutation/ mutation rate
maxgen	maximum number of generation

3. Assumptions

The following assumptions have been considered in the formulation of the problem:

(i) The reliabilities of all the components are imprecise and interval valued.
(ii) The chance of failure of any component is independent with respect to other components.
(iii) All the redundancies are active redundancy without repair.

4. Finite interval arithmetic

An interval number A is a closed interval denoted by $A = [a_L, a_R]$ and is defined by $A = [a_L, a_R] = \{x : a_L \le x \le a_R, x \in \square\}$, where a_L, a_R are the left and right bounds respectively and $\square$ is the set of all real numbers. Also, in centre and width form we have $A = [a_L, a_R] = \langle a_c, a_w \rangle$, where $a_c = (a_L + a_R)/2$ and $a_w = (a_R - a_L)/2$ are the centre and the width of the interval *A*. A real number can also be treated as an interval, such as for all $x \in \square$, x can be written as an interval $[x, x]$ which has zero width. The definitions of arithmetical operations like addition, subtraction, multiplication, division and integral power of interval numbers and also the *n-th* root as well as the rational powers of interval numbers are presented. For detailed discussion one may refer to the works of Moore (1979), Hansen and Walster (2004) and Karmakar, Mahato and Bhunia (2009).

Definition 4.1: Let $A = [a_L, a_R]$ and $B = [b_L, b_R]$ be two intervals. Then the definitions of addition, scalar multiplication, subtraction, multiplication and division of interval numbers are as follows:

Addition: $A + B = [a_L, a_R] + [b_L, b_R] = [a_L + b_L, a_R + b_R]$

Scalar multiplication: For any a real number λ, $\lambda A = \lambda[a_L, a_R] = \begin{cases} [\lambda a_L, \lambda a_R] \text{ if } \lambda \ge 0 \\ [\lambda a_R, \lambda a_L] \text{ if } \lambda < 0 \end{cases}$

Subtraction: $A - B = [a_L, a_R] - [b_L, b_R] = [a_L, a_R] + [-b_R, -b_L] = [a_L - b_R, a_R - b_L]$

Multiplication: $A \times B = [a_L, a_R] \times [b_L, b_R] = [\min(a_L b_L, a_L b_R, a_R b_L, a_R b_R), \max(a_L b_L, a_L b_R, a_R b_L, a_R b_R)]$

Division: $\dfrac{A}{B} = A \times \dfrac{1}{B} = [a_L, a_R] \times [\dfrac{1}{b_R}, \dfrac{1}{b_L}]$, provided $0 \notin [b_L, b_R]$

Definition 4.2: Let $A = [a_L, a_R]$ be an interval and n be any non-negative integer, then

$$A^n = \begin{cases} [1,\ 1] & \text{if } n = 0 \\ [a_L^n,\ a_R^n] & \text{if } a_L \geq 0 \text{ or if } n \text{ is odd} \\ [a_R^n,\ a_L^n] & \text{if } a_R \leq 0 \text{ and } n \text{ is even} \\ [0,\ \max(a_L^n,\ a_R^n)] & \text{if } a_L \leq 0 \leq a_R \text{ and } n (> 0) \text{ is even.} \end{cases}$$

4.1. Functions of intervals

The interval representations of some elementary useful functions like exponential and logarithmic for the interval $A = [a_L, a_R]$ are given below:

(i) $\exp(A) = \exp([a_L, a_R]) = [\exp(a_L), \exp(a_R)]$

(ii) $\log(A) = \log([a_L, a_R]) = [\log(a_L), \log(a_R)]$**, provided** $a_L > 0$.

4.2. Interval order relations

Let $A = [a_L, a_R]$ and $B = [b_L, b_R]$ be two intervals. Then these two intervals may be any one of the following types:

Type-1: Two intervals are disjoint.
Type-2: Two intervals are partially overlapping.
Type-3: One of the intervals contains the other one.

Recently, Sahoo, Bhunia and Kapur (2012) proposed the definitions of order relations between two interval numbers corresponding to maximization and minimization problems.

Definition-4.2.1: The order relation $>_{\max}$ between the intervals $A = [a_L, a_R] = \langle a_c, a_w \rangle$ and $B = [b_L, b_R] = \langle b_c, b_w \rangle$, then for maximization problems

(i) $A >_{\max} B \Leftrightarrow a_c > b_c$ for Type I and Type II intervals,

(ii) $A >_{\max} B \Leftrightarrow$ either $a_c \geq b_c \wedge a_w < b_w$ or $a_c \geq b_c \wedge a_R > b_R$ for Type III intervals,

According to this definition, the interval A is accepted for maximization case. Clearly, the order relation $A >_{\max} B$ is reflexive and transitive but not symmetric.

Definition-4.2.2: The order relation $<_{\min}$ between the intervals $A = [a_L, a_R] = \langle a_c, a_w \rangle$ and $B = [b_L, b_R] = \langle b_c, b_w \rangle$, then for minimization problems

(i) $A <_{\min} B \Leftrightarrow a_c < b_c$ for Type I and Type II intervals,

(ii) $A <_{\min} B \Leftrightarrow$ either $a_c \leq b_c \wedge a_w < b_w$ or $a_c \leq b_c \wedge a_L < b_L$ for Type III intervals,

According to this definition, the interval A is accepted for minimization case. Clearly, the order relation $A <_{\min} B$ is reflexive and transitive but not symmetric.

4.3. Mean, Variance and Standard deviation of Interval Numbers

According to Bhunia, Sahoo and Roy (2010), the mean, variance and standard deviation of n interval numbers are defined as follows:

Let $x_i = [x_{iL}, x_{iR}]$, $i = 1, 2, ..., n$, be the i th observation which is an interval number. Then mean $(\overline{x})$, variance $(Var(x))$, standard deviation (σ_x) and also coefficient of variation (COV) of the intervals $x_1, x_2, \cdots, x_n$ are given by

$$\bar{x} = [\bar{x}_L, \bar{x}_R] = \left[\frac{1}{n}\sum_{i=1}^{n} x_{iL}, \frac{1}{n}\sum_{i=1}^{n} x_{iR} \right],$$

$$\mathrm{Var}(x) = [\sigma_L^2, \sigma_R^2] = \frac{1}{n}\sum_{i=1}^{n}\left([x_{iL} - \frac{1}{n}\sum_{i=1}^{n} x_{iR}, x_{iR} - \frac{1}{n}\sum_{i=1}^{n} x_{iL}] \right)^2$$

$$\sigma_x = [\sigma_L, \sigma_R] = \sqrt{\mathrm{Var(x)}} = \left\{ \frac{1}{n}\sum_{i=1}^{n}\left([x_{iL} - \frac{1}{n}\sum_{i=1}^{n} x_{iR}, x_{iR} - \frac{1}{n}\sum_{i=1}^{n} x_{iL}] \right)^2 \right\}^{1/2}$$

And

$$COV = \frac{\sigma_x}{\bar{x}} \times 100 = [\frac{\sigma_L}{\bar{x}_R} \times 100, \frac{\sigma_R}{\bar{x}_L} \times 100].$$

5. Reliability redundancy optimization problem

The aim of reliability optimization is to improve reliability of a system. The reliability components are not always fixed but may be considered as intervals in reliability redundancy optimizations which are necessary for system design having large number of subsystems assembled together.

According to the assumptions, the system reliability is interval valued as all the component reliabilities are interval valued. Hence a reliability redundancy optimization problem can be constructed with system reliability as the objective function in the form of interval. The reliability redundancy problem for maximizing system reliability with interval valued reliability components is subject to nonlinear interval constraints and it can be formulated as constrained nonlinear mixed-integer programming model. The general form of the problem is as follows:

Maximize $R_S = \left[R_{SL},\, R_{SR}\right] = f\left(x, \left[r_L, r_R\right]\right)$

subject to $g_i\left(x, \left[r_L, r_R\right]\right) \le \left[b_{iL}, b_{iR}\right], i = 1, 2, ..., m$

$0 \le r_{jL} \le r_{jR} \le 1, \quad x_j \in \mathbb{Z}^+, \; 1 \le j \le n$

where R_S is the system reliability, *g* is the set of constraint functions which are generally associated with system volume, weight and cost. Also $[r_L, r_R] = \left(\left[r_{1L}, r_{1R}\right], \left[r_{2L}, r_{2R}\right], \cdots, \left[r_{nL}, r_{nR}\right]\right)$ is the reliability vector whose components are intervals and $x = \left(x_1, x_2, \cdots, x_n\right)$ is the vector of the redundancy allocation for the system, $[r_{jL}, r_{jR}]$ and x_j respectively are the reliability and the number of components in the *j*-th subsystem; $f\left(x, \left[r_L,\, r_R\right]\right)$ is the objective function of the whole system reliability; $b_i = \left[b_{iL}, b_{iR}\right]$ is the *i*-th component of the resource limitations; *n* is the number of subsystems in the system. The objective of the problem is to find the number of redundant components and the bounds of the components reliability for each subsystem in such a way that the overall system reliability is maximized.

6. GA based constraint handling technique

Genetic algorithm (GA) is one of the powerful computerized heuristic methods (Goldberg (1989), Michalewicz (1996), Sakawa (2002) and Deb (2000)) which are being employed for solving the interval valued constrained optimization problems. The constrained optimization problem is transformed into unconstrained optimization problem with the help of penalty function technique given by Miettinen et al., (2003) and Aggarwal and Gupta (2005). In penalty function technique, constraints are added to or subtracted from the objective function to penalize the infeasible solutions. When the fitness function is increased or decreased with a penalty term multiplied by penalty coefficient there may appear difficulty to select the initial value of the same and also to select the upgrading strategy for the penalty coefficients (Goldberg, 1989). To overcome this difficulty we have to solve the same problem with a fitness function by penalizing with a large positive number (say, M which can be written in the interval form as$[M, M]$) following Gupta et al. (2009) and Bhunia et al. (2010). This technique is called as Big-M penalty method. The reduced objective function for this penalty method is given below.

$$\text{Maximize } \hat{R}_S(x,[r_L,r_R]) = R_S(x,[r_L,r_R]) + \theta(x,[r_L,r_R])$$

$$\text{where } \theta(x,[r_L,r_R]) = \begin{cases} [0,0] & \text{if } (x,[r_L,r_R]) \in S \\ -R_S(x,[r_L,r_R]) + [-M,-M] & \text{if } (x,[r_L,r_R]) \notin S \end{cases}$$

$$S = \left\{ (x,[r_L,r_R]) : g_i(x,[r_L,r_R]) \le [b_{iL},b_{iR}], i = 1,2,...,m \right\}$$

$$\text{and } l' \le x \le u', \alpha \le r_L \le \beta \le r_R \le \gamma$$

where $l' = (l'_1, l'_2, ..., l'_q)$, $u' = (u'_1, u'_2, ..., u'_q)$, $x = (x_1, x_2, ..., x_n)$, $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)$, $\beta = (\beta_1, \beta_2, ..., \beta_n)$ and $\gamma = (\gamma_1, \gamma_2, ..., \gamma_n)$.

7. Genetic Algorithm

This algorithm starts with an initial population of chromosomes which are the set of randomly generated solutions. These populations are improved from generation to generation by artificial evolution and genetics. During each generation, each chromosome in the population is evaluated using some measure of fitness and the population of the next generation is created through genetic and evolutionary operators. In genetic algorithm, there are two genetic operators such as crossover and mutation. Crossover operator creates new chromosome (offspring) by combining the features of two or more parent chromosomes. On the other hand, mutation operator acts on a single chromosome and produces an offspring with minor changes.

For implementing the GA in solving the optimization problems, the following basic components are considered.

- GA Parameters
- Chromosome representation and initialization of population
- Evaluation of fitness function
- Selection process
- Genetic operators (crossover, mutation and elitism)

There are basically four parameters to be decided upon in the genetic algorithm, viz. population size (*popsize*), maximum number of generations (*maxgen*), probability of crossover (*pcross*) and probability of mutation (*pmute*). Choices of *popsize* and *maxgen* are problem dependent. Again, from the principle of natural genetics, it is obvious that the crossover rate is greater than that of mutation rate. Usually, the crossover rate varies from 0.8 to 0.95 whereas the mutation rate varies from 0.05 to 0.2. Sometimes, it is considered as $1/n$ where n is the number of genes (or variables) of the individuals.

For implementation of GA, the appropriate chromosome/individual representation of solution for the given problem is reasonably important. As reported in the existing literature there are several types of representations, such as binary coded, real coded, octal coding or hexadecimal coding. Among these representations, real coding representation is very popular. In this representation, each component (gene) of the chromosome is the value of the decision variable.

After appropriate representation of chromosome, the next step is to initialize the chromosomes that will take part in the artificial genetic operations. This process creates *popsize* number of chromosomes in which every component for each chromosome is randomly generated within the bounds of the corresponding decision variable.

After obtaining a population of potential solutions, we need to check how good they are. So we have to measure the fitness value for each chromosome by using fitness function. In this work, the value of the transformed unconstrained objective function (due to different penalty technique) corresponding to the chromosome is taken as the fitness value of that chromosome.

In the application of GA, the selection operator plays an important role because it is the first operator applied to the population. The goal of this operator is to select the above average solutions and eliminate below average solutions from the population for the next generation under the principle “survival of the relatively fit”. In this work, we have used tournament selection process of size two with replacement as the selection operator with the following assumptions:

- When both the chromosomes/individuals are feasible then one with better fitness value is selected.

DOI https://doi.org/2024-JICS-22635 | Generated on 2025-04-12 03:46:35

- When one chromosome/individual is feasible and another is infeasible then the feasible one is selected.
- When both the chromosomes/individuals are infeasible with unequal constraints violation, then the chromosome with less constraints violation is selected.
- When both the chromosomes/individuals are infeasible with equal constraints violation, then any one chromosome/individual is selected.

After the selection process, crossover operator is applied to the resulting chromosomes which have survived. This is an operation that really empowers the GA. It operates on two or more parent chromosomes/solutions at a time and creates the offspring by recombining the feature of the parent solutions. In this work, we have used uniform crossover for integer variables and whole arithmetical crossover for floating point variables.

The aim of mutation operation is to introduce the random variations into the population. Sometimes, it helps to regain the information lost in earlier generations and responsible for fine tuning of the system. This operator is applied to a single chromosome only. Usually, its rate is very low; otherwise it would defeat the order building being generated through selection and crossover operations. This operator attempts to bump the population gently into a slightly better course. This means that mutation changes either single gene or all the genes of a randomly selected chromosome slightly. In this work we have used one-neighborhood mutation for integer variables and boundary mutation for floating point variables.

In order to preserve and use the best solution obtained in the previous generations, a process is considered called elitism process. Sometimes there is a chance that the best chromosome may be lost when a new population is generated by crossover and mutation operations. To remove this situation the worst individual/chromosome is replaced by the best individual/chromosome in the current generation. Instead of single chromosome, one or more chromosomes may take part in this operation.

The termination condition is a condition for which the algorithm/process is going to stop. For this purpose any one of the following three conditions is considered as termination condition.

- the best individual does not improve over specified generations,
- the total improvement of the last certain number of best solutions is less than a pre-assigned small positive number
- the number of generations reaches a prescribed finite number of generation (called maximum number of generation).

Basically genetic algorithm is a computerized stochastic search optimization. In this algorithm results are obtained for different sets of random numbers. For this purpose a number of independent runs have been performed to calculate the system reliability. Among all the results in different runs the best found value of system reliability has been considered. For this purpose we have used the nomenclature “Best found system Reliability”. On the other hand, “Mean system Reliability” is the mean value of all the values of system reliability in different runs.

8. Numerical Examples

To test the performance of the proposed GA in solving constrained nonlinear mixed integer program problem with interval valued objective and interval constraint functions, we have solved the following example:

$$\text{Maximize} \quad f\left(x,\left[r_L,r_R\right]\right)=\prod_{i=1}^{n}\left[1-\left(1-\left[r_{iL},r_{iR}\right]\right)^{x_i}\right]$$

subject to

$$g_1\left(x,\left[r_L,r_R\right]\right)=\sum_{i=1}^{n} v_i\, x_i^2 \le \left[V_L,V_R\right]$$

$$g_2\left(x,\left[r_L,r_R\right]\right)=\sum_{i=1}^{n} C\left(r_i\right)\left[x_i+e^{0.25x_i}\right]\le\left[C_L,C_R\right]$$

DOI https://doi.org/2024-JICS-22635 | Generated on 2025-04-12 03:46:35

$$g_3\left(x,[r_L,r_R]\right)=\sum_{i=1}^{n} w_i\, x_i e^{0.25x_i} \le [W_L,W_R]$$

where $1 \le x_i \le 10,\quad x_i \in \square^{+}$,

$0.5 \le r_{iL} \le r_{iR} \le 1-10^{-6},\quad r_i=[r_{iL},r_{iR}]$

where $c(r_i)=\alpha_i\left[-T/\log\left([r_{iL},r_{iR}]\right)\right]$ is the cost component with reliability r_i of the i-th subsystem; T is the operating time during which the component must not fail.

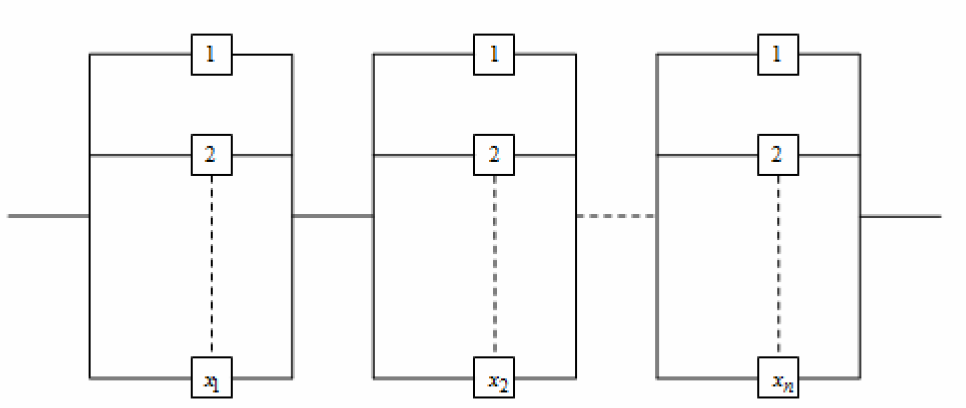

Fig. 1: Reliability-redundant system of *n* components

The proposed method has been coded in C programming language. The computational work has been done on a PC with Intel Core-2-due 2.5 Ghz processor in LINUX environment. For each case 20 independent runs have been performed to calculate the best found system reliability which is nothing but the optimal value of the system reliability. In this computation, the values of genetic parameters like, population size, crossover rate, mutation rate and maximum number of generations have been taken as *popsize*=20, *pcross*=0.85, *pmute*=0.15 and *maxgen*=30 respectively.

Table 1: values of different parameters used in the example

Stage	$10^5\alpha_i$	β_i	v_i	w_i	v	c	w	T
1	1.0	1.5	1	6	[240, 260]	[390, 410]	[490, 510]	1000
2	2.3	1.5	2	6				
3	0.3	1.5	3	8				
4	2.3	1.5	4	7				

The best found result obtained from 20 independent runs is given below:

$r_1=[0.862454,\ 0.908425], r_2=[0.880435,\ 0.953215], r_3=[0.886079,\ 0.961301], r_4=[0.865099,\ 0.970941]$, $x_1=7$, $x_2=5$, $x_3=5$, $x_4=6$.

The value of system reliability is [0.999949, 1.000000] having elapsed time 1.0×10^{-7} sec. for $\alpha=(0.85,0.85,0.85,0.85)$, $\beta=(0.89,0.89,0.89,0.89)$ and $\gamma=(0.99,0.99,0.99,0.99)$.

The mean, variance and standard deviation of the system reliability obtained from 20 independent runs are computed in the form of interval and are [0.999922, 1.000000], [0.0, 0.122×10^{-7}], [0.0, 0.111×10^{-3}] and [0.0, 0.011058] respectively.

8.1 Sensitivity Analysis

To investigate the overall performance of the proposed GA based Big-M penalty technique for solving Reliability Redundancy Optimization problem, sensitivity analyses have been carried out graphically on the lower bound (LR), upper bound (UR) and centre value (MR) of the system reliability with respect to different GA parameters (*popsize*, *maxgen*, *pcross* and *pmute*) separately by changing one parameter at a time and keeping the other at their original value. These have been shown in Fig. 2 – Fig. 5. From Fig. 2, it is observed that both the bounds and the centre value of the system reliability be the same for all the values of population size from 20 to 100. On the other hand, Fig. 3 shows that both the bounds along with the centre value of the system reliability remain same for all the values of maximum number of generations from 30 to 100. In all cases, the upper and lower bounds are fixed and they are 1.0 and 0.999950 respectively.

In Fig.4, system reliability is shown with the change of probability of crossover (*pcross*) (0.7~0.95). From Fig.4, it is seen that the upper bound of the system reliability remains unaltered, whereas the lower bound be the same for $pcross = 0.75$ and more.

From Fig.5, it is observed that the centre value and the lower bound of the system reliability remain unchanged for $pmute = 0.15$ and more, whereas the upper bound be the same for all the values of *pmute*.

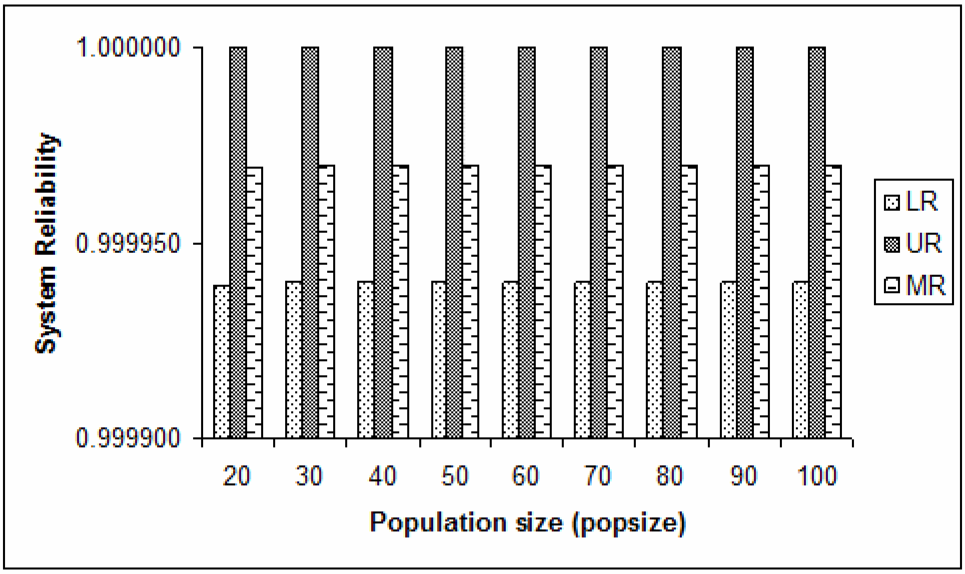

Fig. 2: Population size (*popsize*) vs. System Reliability

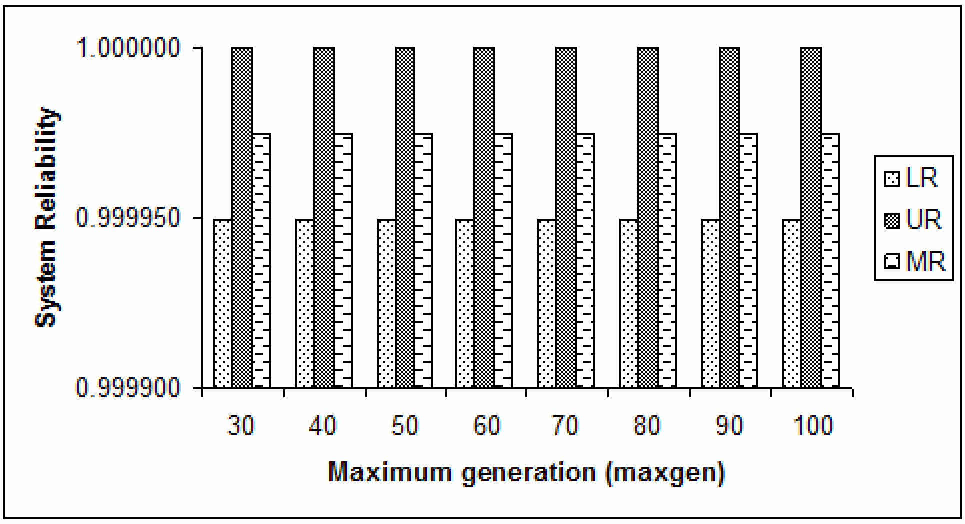

Fig. 3: Maximum number of generations (*maxgen*) vs. System Reliability

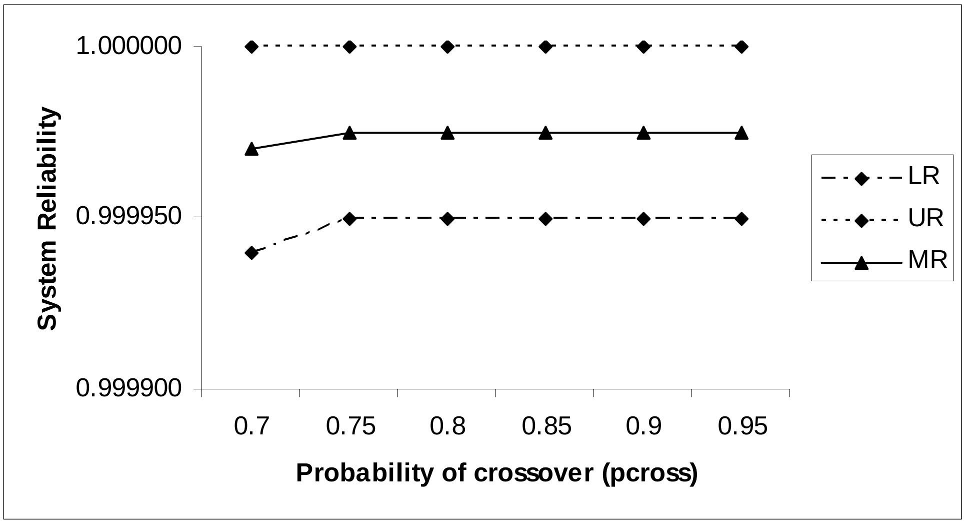

Fig. 4: Probability of crossover (*pcross*) vs. System Reliability

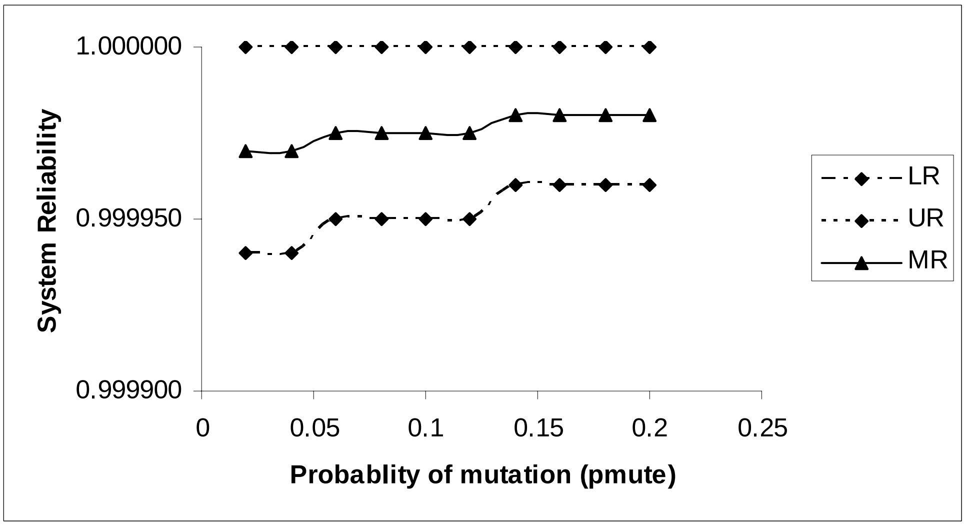

Fig. 5: Probability of mutation (*pmute*) vs. System Reliability

9. Concluding Remarks

Generally, the goal of the reliability-redundancy allocation problem is to find the number of redundant components so as to maximize the system reliability. However, there are some redundancy problems where the numbers of redundant components as well as the component reliabilities are to be found. In this paper, we have solved such type of problem with the help of GA based constraint handling technique and interval order relations. This type of problems occurs in complex bridge system and over speed protection system for a gas turbine.

In this approach, there is also a drawback. As the objective function (system reliability) is interval valued so that the best found solution obtained from proposed method is also interval valued. In that case, there is an uncertainty which is nothing but the width of the interval. In other approaches the objective function is fixed.

For solving the optimization problem, the GA based Big-M penalty approach has been used. In his approach, the value of M is considered for the value of the fitness function corresponding to infeasible solution. Gupta, Bhunia and Roy (2009) applied this for solving constrained reliability optimization problems. In their works there is no indication regarding the vale of M. However, for infeasible solution the value of M may be taken depending on the fitness function value. In case of maximization problem, a small

DOI https://doi.org/2024-JICS-22635 | Generated on 2025-04-12 03:46:35

value and in case of minimization problems a value may be considered for M for solving constrained optimization problems.

Acknowledgement

For this research, the last author would like to acknowledge the financial support provided by the Council of Scientific and Industrial Research (CSIR), New Delhi, India.

10. References

[1] K.K. Aggarwal, and J.S. Gupta, *Penalty function approach in heuristic algorithms for constrained*, IEEE Transactions on Reliability, 54 (2005), pp.549-558.

[2] A.K.Bhunia, L. Sahoo, and D. Roy, *Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm*, Applied Mathematics and Computation, 216(2010), pp. 929-939.

[3] L. d. S. Coelho, *An efficient particle swarm approach for mixed-integer programming in reliability-redundancy optimization applications*, Reliability Engineering and System Safety, 94(2009), pp. 830-837.

[4] D.W. Coit, T.D. Jin, and N. Wattanapongsakorn, *System optimization with component reliability estimation uncertainty: A multi-criteria approach*, IEEE Transactions on Reliability, 53(2004), pp. 369-380.

[5] L. Cui, , W. Kuo, H.T. Loh,, and M. Xie, *Optimal allocation of minimal & perfect repairs under resource constraints*, IEEE Transactions on Reliability, 53(2004), pp. 193-199.

[6] K. Deb, *An efficient constraint handling method for genetic algorithms*, Computer Methods in Applied Mechanics and Engineering, 186(2000), pp. 311-338.

[7] M. Gen, and Y. Yun, *Soft computing approach for reliability optimization*, Reliability Engineering & System Safety, 91(2006), pp. 1008-1026.

[8] D.E. Goldberg, *Genetic Algorithms: Search, Optimization and Machine Learning*, Reading, Addison Wesley, MA, 1989.

[9] R.K. Gupta, A.K. Bhunia, and D. Roy, *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(2009), pp. 275-284.

[10] C. Ha, and W. Kuo, *Multi-Path heuristic for redundancy allocation: the tree heuristic*, IEEE Transactions on Reliability, 55(2006a), pp. 37-43.

[11] C. Ha, and W. Kuo, *Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems*, European Journal of Operational Research, 171(2006b), pp. 124-138.

[12] E. Hansen, and G.W. Walster, *Global optimization using interval analysis*, Marcel Dekker Inc., New York, 2004.

[13] S. Karmakar, S. Mahato, and A. K. Bhunia, *Interval oriented multi-section techniques for global optimization*, Journal of Computation and Applied Mathematics, 224(2009), pp. 476-491.

[14] W. Kuo, and V.R. Prasad, *An Annotated Overview of System Reliability Optimization*, IEEE Transactions on Reliability, 49 (2000), pp. 487-493.

[15] W. Kuo, V.R. Prasad, F.A. Tillman, and C.L. Hwang, *Optimal Reliability Design Fundamentals and application*, Cambridge University Press, 2001.

[16] W. Kuo, and R. Wan, *Recent Advances in Optimal Reliability Allocation*, IEEE Transactions on System, Man, and Cybernetics Part A, 37(2007), pp. 143-156.

[17] Y.C. Liang, and A.E. Smith, *An ant colony optimization algorithm for the redundancy allocation problem (RAP)*, IEEE Transactions on Reliability, 53 (2004), pp. 417-423.

[18] S.K. Mahato, and A.K. Bhunia, *Interval-Arithmetic-Oriented Interval Computing Technique for Global Optimization*, Applied Mathematics Research eXpress, 2006, pp. 1-19.

[19] M. Marseguerra, E. Zio, L. Podofillini, D.W. Coit, *Optimal design of reliable network systems in presence of uncertainty*, IEEE Transactions on Reliability, 44(2005), pp. 243-253.

[20] Z. Michalewicz, *Genetic Algorithms + Data structure = Evaluation Programs*, Springer Verlag, Berlin, 1996.

[21] K. Miettinen, M.M. Makela, and J. Toivanen, *Numerical comparison of some Penalty-Based Constraint Handling Techniques in Genetic Algorithms*, Journal of Global Optimization, 27(2003), pp. 427-446.

[22] R. E. Moore, *Methods and applications of interval Analysis*, SIAM, 1979.

[23] J. Onishi, and S. Kimura, *Solving the redundancy allocation problem with a mix of components using the improved surrogate constraint method*, IEEE Transactions on Reliability, 56(2007), pp. 94-101.

[24] L. Painton, J. Campbell, *Genetic Algorithms in Optimization of System Reliability*, *I*EEE Transactions on Reliability, 44(1995), pp. 172-178.

[25] J.E. Ramirez-Marquez, and D.W. Coit, *Optimization of system reliability in the presence of Common cause failures*, Reliability Engineering and System Safety, 92(2007), pp. 1421-1434.

[26] J.E. Ramirez-Marquez, D.W. Coit, and A. Konak, *Redundancy allocation for series-parallel systems using a max-min approach*, IIE Trans, 36(2004), pp. 891-898.

[27] V. Ravi, and J.P. Reddy, *Fuzzy global optimization of complex system reliability*, IEEE Transactions on Fuzzy Systems, 8(2000), pp. 241-248.

[28] L. Sahoo, A.K. Bhunia,and P.K. Kapur, *Genetic algorithm based multi-objective reliability optimization in interval environment*, Computer & Industrial Engineering, 62 (2012), pp. 152-160.

[29] L. Sahoo, A.K. Bhunia, and D. Roy, *A Genetic Algorithm Based Reliability Redundancy Optimization for Interval valued Reliabilities of components*, Journal of Applied Quantitative Methods, 5(2010), pp. 270-287.

[30] M. Sakawa, *Genetic Algorithms and fuzzy multiobjective optimization*, Kluwer Academic Publishers, 2002

[31] X. Sun, and D. Li. Duan, *Optimal Condition and Branch and Bound Algorithm for Constrained Redundancy Optimization in Series System*, Optimization and Engineering, 3 (2002), 53-65.

[32] X.L. Sun, , K.I.M. Mckinnon, and D. Li, *A Convexification method for a Class of global optimization problems with application to reliability optimization*, Journal of Global Optimization, 21(2001), pp. 185-199.

[33] C.S. Sung, and Y.K. Cho, *Branch and Bound redundancy optimization for a series system with multiple-choice constraints*, IEEE Transactions on Reliability, 48(1999), pp. 108-117.

[34] L.V. Utkin, and S. V. Gurov, *Imprecise reliability of general structures*, Knowledge and Information Systems, 1(1999), pp. 459-480.

[35] L.V. Utkin, and S. V.Gurov, *New reliability models based on imprecise probabilities*, World Scientific: In C. Hsu, Advanced Signal Processing Technology, pp.110-139, Chapter 6, 2001.

[36] E.P. Zafiropoulos, and E.N. Dialynas, *Reliability and cost optimization of electronic devices considering the component failure rate uncertainty*, Reliability Engineering & System Safety, 84 (2004), pp. 271-284.

[37] X. Zhao, L. Cui, and W. Kuo, *Reliability for Sparsely Connected Consecutive-k Systems*, IEEE Transactions on Reliability, 56(2007), pp. 516-524.