Genetic Algorithm Based Reliability Optimization in Interval Environment

A.K. Bhunia[1] and L. Sahoo[2]

[1] Department of Mathematics, The University of Burdwan, Burdwan-713104, West Bengal, India
math-akbhunia@buruniv.ac.in

[2] Department of Mathematics, Raniganj Girls College, Raniganj-713347, West Bengal, India
lxsahoo@gmail.com

The objective of this chapter is to develop and solve the reliability optimization problems of series-parallel, parallel-series and complicated system considering the reliability of each component as interval valued number. For optimization of system reliability and system cost separately under resource constraints, the corresponding problems have been formulated as constrained integer/mixed integer programming problems with interval objectives with the help of interval arithmetic and interval order relations. Then the problems have been converted into unconstrained optimization problems by two different penalty function techniques. To solve these problems, two different real coded genetic algorithms (GAs) for interval valued fitness function with tournament selection, whole arithmetical crossover and non-uniform mutation for floating point variables, uniform crossover and uniform mutation for integer variables and elitism with size one have been developed. To illustrate the models, some numerical examples have been solved and the results have been compared. As a special case, taking lower and upper bounds of the interval valued reliabilities of component as same the corresponding problems have been solved and the results have been compared with the results available in the existing literature. Finally, to study the stability of the proposed GAs with respect to the different GA parameters (like, population size, crossover and mutation rates), sensitivity analyses have been shown graphically.

1 Introduction

While advanced technologies have raised the world to an unprecedented level of productivity, our modern society has become more delicate and vulnerable due to the increasing dependence on modern technological systems that

N. Nedjah et al. (Eds.): Innovative Computing Methods, SCI 357, pp. 13–36.
springerlink.com

often require complicated operations and highly sophisticated management. From any respect, the system reliability is a crucial measure to be considered in systems operation and risk management. When designing a highly reliable system, there arises an important question as to how to obtain a balance between reliability and other resources e.g., cost, volume and weight. In the last few decades, several researchers considered reliability optimization problems, like redundancy allocation and cost minimization problems as integer nonlinear programming problems (INLPP) and/or mixed-integer nonlinear programming problems (MINLPP) with single or several resource constraints [1-14]. To solve those problems, different techniques have been proposed by the several researchers. In their works, the reliability of each component is 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 vary due to several reasons. There is no technology by which different components can be produced with exactly identical reliabilities. So, the reliability of each component is sensible and it may be treated as a positive imprecise number instead of a fixed real number. Studies of the system reliability where the component reliabilities are imprecise and/or interval valued have already been initiated by some authors [15-19].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 the stochastic approach, the parameters are assumed to be random variables with known probability distributions. In the fuzzy approach, the parameters, constraints and goals are considered as fuzzy sets with known membership functions or fuzzy numbers. On the other hand, in the 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 intervals number to represent an imprecise number, as this representation is the most significant representation among others. Due to this representation, the system reliability would be interval valued. Here, we have considered GA-based approaches for solving reliability optimization problems with the interval objective. As the objective function of the reliability optimization is interval valued, to solve this type of problem by the GA method, order relations of interval numbers are essential for selection operation as well as for finding the best chromosome in each generation. Here we consider the definition of order relations developed by Mahato and Bhunia [20] in the context of the optimistic and pessimistic decision maker's point of view for maximization and minimization problems.

In this chapter, we have considered the problem of constrained redundancy allocation in the series system, the hierarchical series-parallel system, the complicated or non-parallel-series system and the network reliability system with interval valued reliability components (redundancy allocation and network cost minimization). The problems are formulated as non-linear constrained integer programming problems and/or mixed integer programming problems with interval coefficients [21-22] for maximizing the overall system reliability and system cost under some resource/budget constraints. During the last few years, several techniques were proposed for solving the constrained optimization problem with fixed coefficients with the help of GAs [23-29]. Among these methods, penalty function techniques are very popular in solving the same by GAs [30-32]. This method transforms the constrained optimization problem to an unconstrained optimization problem by penalizing the objective function corresponding to the infeasible solution. Hence, to solve the constrained optimization problem the problem is converted to unconstrained one by two different types of penalty techniques and the resulting objective function would be interval valued. So, to solve this problem we have developed two different GAs for integer variables with the same GA operators like tournament selection, uniform crossover for integer variables and whole arithmetical crossover for floating point variables, uniform mutation for integer variables and boundary mutation for floating point variables and elitism of size one but different fitness function depending on different penalty approaches. These methods have been illustrated with some numerical examples and to test the performance of these methods, results have also been compared. As a special case considering the lower and upper bounds of interval valued reliabilities of components as same, the resulting problem becomes identical with the existing problem available in the literature.

2 Finite Interval Arithmetic

An interval number is a closed interval denoted by $A = [a_L, a_R]$ and is defined by $A = [a_L, a_R] = \{x : a_L \leq x \leq a_R, x \in \Re\}$ where a_L and a_R are the left and right limits respectively and $\Re$ is the set of all real numbers. A can also be expressed in terms of centre and radius as $A = \langle a_c, a_w \rangle = \{x : a_c - a_w \leq x \leq a_c + a_w, x \in \Re\}$, where a_c anda_w are the centre and radius of the interval A respectively, i.e., $a_c = (a_L + a_R)/2$, and $a_w = (a_R - a_L)/2$. Actually, every real number can be treated as an interval, such as for all $x \in \Re$, x can be written as an interval $[x, x]$ having zero width. Now we shall present the definitions of arithmetical operations like addition, subtraction, multiplication, division and integral power of interval numbers [33] and also the n-th root as well as the rational powers of interval numbers [34].

Definition 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 real number α,$\alpha A=\alpha[a_L,a_R]=\begin{cases}[\alpha a_L,\alpha a_R] \text{ if } \alpha\geq 0\\ [\alpha a_R,\alpha a_L] \text{ if } \alpha<0\end{cases}$
- **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:** $A\times B=[a_L,a_R]\times[b_L,b_R]$
$=[\min(a_Lb_L,a_Lb_R,a_Rb_L,a_Rb_R),\max(a_Lb_L,a_Lb_R,a_Rb_L,a_Rb_R)]$
- **Division**
$\frac{A}{B}=A\times\frac{1}{B}=[a_L,a_R]\times[\frac{1}{b_R},\frac{1}{b_L}]$, $provided\, 0\notin[b_L,b_R]$

Definition 2: Let $A=[a_L,a_R]$ be an interval and n be any non-negative integer, then

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

Definition 3: The $n-th$ root of an interval $A=[a_L,a_R]$, n being a positive integer, is defined as

$$(A)^{\frac{1}{n}}=[a_L,a_R]^{\frac{1}{n}}=\begin{cases}\sqrt[n]{[a_L,a_R]}=[\sqrt[n]{a_L},\sqrt[n]{a_R}] \text{ if } a_L\geq 0 \text{ or if } n \text{ is odd}\\ [0,\sqrt[n]{a_R}] \quad \text{ if } a_L\leq 0, a_R\geq 0 \text{ and } n \text{ is even}\\ \phi \text{ if } a_R<0 \text{ and } n \text{ is even}\end{cases}$$

where ϕ is the empty interval.

Again, by applying the definitions of power and different roots of an interval, we can find any rational power of an interval. For example $A^{\frac{p}{q}}$ obtained by defining $A^{\frac{p}{q}}$ as $(A^p)^{\frac{1}{q}}$.

3 Order Relation of Interval Numbers

Further, for arriving at the optimum solution involving interval algebra, we need to define the order relation of interval numbers.

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

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

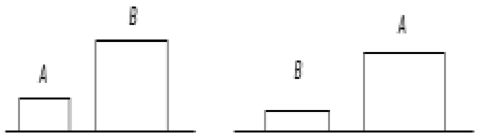

Fig. 1 Type-1 interval

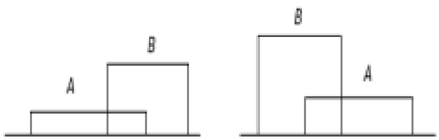

Fig. 2 Type-2 intervals

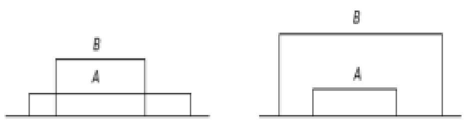

Fig. 3 Type-3 intervals

Here we consider the definitions of order relations developed by Mahato and Bhunia [20] in the context of optimistic and pessimistic decision makers' point of view.

3.1 Optimistic Decision-Making

In optimistic decision-making, decision maker prefers the lowest value for minimization problems and highest value for maximization problems ignoring the uncertainty.

Definition 4: Let us define the order relation $\geq_{\text{omax}}$ between the intervals $A = [a_L, a_R]$ and $B = [b_L, b_R]$ then for maximization problems $A \geq_{\text{omax}} B$ $\Leftrightarrow a_R > b_R$,$A >_{\text{omax}} B \Leftrightarrow A \geq_{\text{omax}} B \wedge A \neq B$.

According to this definition, the optimistic decision maker accepts A . The order relation $\geq_{\text{omax}}$ is reflexive and transitive but not symmetric.

Definition 5: The order relation $\leq_{\text{omin}}$ between the intervals $A = [a_L, a_R]$ and $B = [b_L, b_R]$ then for minimization problems $A \leq_{\text{omin}} B \Leftrightarrow a_L \leq b_L$, $A <_{\text{omin}} B \Leftrightarrow A \leq_{\text{omin}} B \wedge A \neq B$. The order relation $\leq_{\text{omin}}$ is not symmetric.

3.2 Pessimistic Decision-Making

In pessimistic decision making, the decision maker prefers the highest/lowest value under the principle ”Less uncertainty is better than more uncertainty” for maximization/minimization problems.

Definition 6: The order relation $>_{\text{pmax}}$ 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)$A >_{\text{pmax}} B \Leftrightarrow a_c > b_c$ for type − 1 and type − 2 intervals,
(ii)$A >_{\text{pmax}} B \Leftrightarrow a_c \geq b_c \wedge a_w < b_w$ for type − 3 intervals

However, for Type-3 intervals, pessimistic decision cannot be taken when $a_c > b_c \wedge a_w > b_w$. In this case, we consider the optimistic decision.

Definition 7: The order relation $<_{\text{pmin}}$ 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)$A <_{\text{pmin}} B \Leftrightarrow a_c < b_c$ for type − 1and type−2 intervals,
(ii)$A <_{\text{pmin}} B \Leftrightarrow a_c \leq b_c \wedge a_w < b_w$ for type − 3 intervals

However, for Type-3 intervals, pessimistic decision cannot be taken when $a_c < b_c \wedge a_w > b_w$. In this case, we consider the optimistic decision.

4 Assumptions and Notations

Witout loss of generality, tet us assume the following:

- The component reliabilities are imprecise and interval valued.
- The failure of any component is independent of that of the other components.
- All redundancy is active redundancy without repair.

The following notations have been used in the entire paper.

- x_j: the number of redundant components in j-th subsystem
- r_j: reliability of j-th component
- $R_j(x)$: $1-(1-r_j)^{x_j}, j = 1, 2, ..., q$, the reliability of j-th parallel subsystem
- x: $(x_1, x_2, ..., x_n)$
- r_{jL},r_{jR}: lower and upper limits of r_j
- m: number of resource constraints
- n: number of stages of the system
- $R_{jL}(x)$: lower bound of $R_j(x)$
- $R_{jR}(x)$: upper bound of $R_j(x)$
- Q_j: $1 - R_j$
- R_j: the reliability of j-th subsystem, $j = q + 1, q + 2, \cdots, n$

- (x, R): $(x_1, x_2, ..., x_q, R_{q+1}, ..., R_n)$
- $R_S(x, R)$: system reliability
- $R_{SL}(x, R)$: lower bound of $R_S(x, R)$
- $R_{SR}(x, R)$: upper bound of $R_S(x, R)$
- $C_i(x, R)$: consumption of i-th resource $(i = 1, 2, ..., m)$
- $C_w(x, R)$: weighted cost
- c_i: availability of i-th resource $(i = 1, 2, ..., m)$
- l_j,u_j: lower and upper bounds of x_j
- α_j,β_j: lower and upper bounds of R_j, $j = q+1, q+2, \cdots, n$
- R^*: minimum prescribed reliability in case of cost minimization problem
- *p_size*: population size
- *p_cross*: probability of crossover or crossover rate
- *p_mute*: probability of mutation or mutation rate

5 Constrained Redundancy Optimization Problem for different Systems

5.1 Series System

It is well known that a series system (ref. Fig. 4) with n independent components must be operating only if all the components are functioning. In order to improve the overall reliability of the system; one can use more reliable components. However, the expenditure and more often the technological limits may prohibit an adoption of this strategy. An alternative technique is to add some redundant components as shown in Fig. 5. The goal of the problem is to determine an optimal redundancy allocation so as to maximize the overall system reliability under limited resource constraints. These constraints may arise out of the size, cost and quantities of the resources. Mathematically, the constrained redundancy optimization problem for such a system for interval valued of reliability can be formulated as follows:

Problem-1: Maximize $[R_{SL}, R_{SR}] = \prod_{j=1}^{q} [\{1-(1-r_{jL})^{x_j}\}, \{1-(1-r_{jR})^{x_j}\}]$
subject to $g_i(x) \leq c_i$, $i = 1, 2, ..., m$ and $l_j \leq x_j \leq u_j$, for $j = 1, 2, ..., q$, where $r_j = [r_{jL}, r_{jR}]$

This is a constrained nonlinear integer programming problem with interval valued objective.

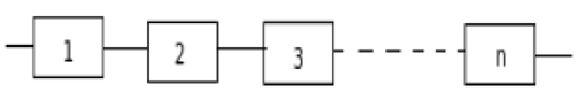

Fig. 4 Series System

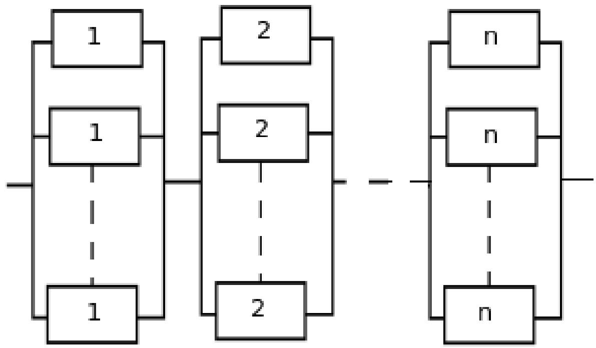

Fig. 5 Parallel series system

5.2 Hierarchical Series-Parallel System

A reliability system is called a hierarchical series parallel system (HSP) if the system can be viewed as a set of subsystems arranged in a series parallel; each subsystem has a similar configuration; subsystems of each subsystem have a similar configuration and so on. For example, let us consider a HSP system ($n = 10, m = 2$) shown in the Fig.6. This system has a nonlinear and non separable structure and consists of nested parallel and series system. The system reliability of HSP is given by $R_S = \{1 - \langle 1 - [1 - Q_3(1 - R_1R_2)]R_4\rangle(1 - R_5R_6)\}(1 - Q_7Q_8Q_9)R_{10}$. Mathematically; the constrained redundancy optimization problem for this system for interval valued reliability can be formulated as follows:

Problem-2: Maximize $[R_{SL}, R_{SR}] = \{1 - \langle 1 - (1 - [Q_{3L}, Q_{3R}](1 - [R_{1L}, R_{1R}]$ $[R_{2L}, R_{2R}]))\ [R_{4L}, R_{4R}]\rangle(1 - [R_{5L}, R_{5R}][R_{6L}, R_{6R}])\}(1 - [Q_{7L}, Q_{7R}][Q_{8L}, Q_{8R}]$

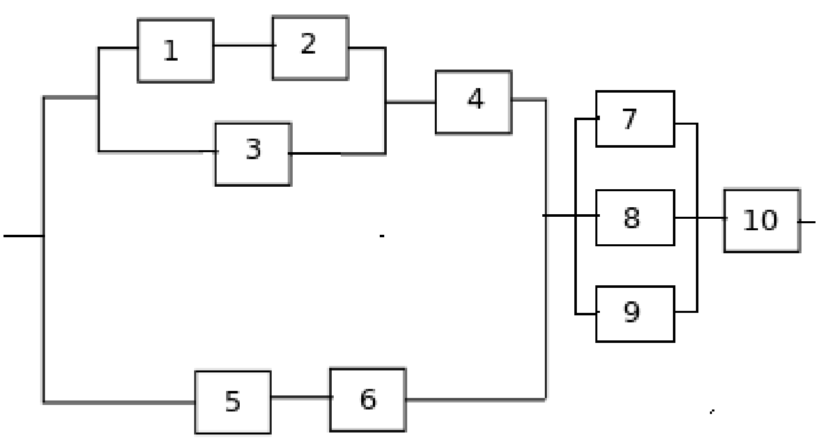

Fig. 6 Hierarchical series-parallel system

$[Q_{9L}, Q_{9R}]$) $[R_{10L}, R_{10R}]$ subject to $g_i(x) \leq c_i, \quad i = 1, 2, ..., m$ and $l_j \leq x_j \leq u_j$ for $j = 1, 2, ..., q$. This is an INLP with interval valued objective.

5.3 Complicated System

When a reliability system can be reduced to series and parallel configurations, there exist combinations of components which are connected neither in a series nor in parallel. Such systems are called complicated or non parallel series systems. This system is also called the bridge system. For example, let us consider a bridge system ($n = 5, m = 3$) shown in Fig.7. This system consists of five subsystems and three nonlinear and non-separable constraints. The overall system reliability R_S is given by the expression as follows:

$$R_S = R_5(1 - Q_1Q_3)(1 - Q_2Q_4) + Q_5[1 - (1 - R_1R_2)(1 - R_3R_4)]$$

$$R_i = R_i(x_i) \text{ and } Q_i = 1 - R_i \text{ for all } i = 1, 2, 3, 4, 5 .$$

Mathematically, the constrained redundancy optimization problem for such complicated system for interval valued reliability can be formulated as follows:

Problem-3: Maximize $[R_{SL}, R_{SR}] = [R_{5L}, R_{5R}](1 - [Q_{1L}, Q_{1R}][Q_{3L}, Q_{3R}])$ $(1 - [Q_{2L}, Q_{2R}][Q_{4L}, Q_{4R}]) + [Q_{5L}, Q_{5R}]\{1 - (1 - [R_{1L}, R_{1R}][R_{2L}, R_{2R}])(1 - [R_{3L}, R_{3R}][R_{4L}, R_{4R}])\}$ subject to $g_i(x) \leq c_i, \quad i = 1, 2, ..., m$ and $l_j \leq x_j \leq u_j$, for $j = 1, 2, ..., q$

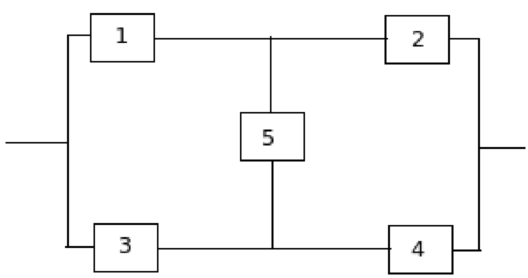

Fig. 7 Complicated system

5.4 k-out-of-n System

A $k - out - of - n$ system is a n-component system which functions when at least k of its components function. This redundant system is sometimes used in the place of pure parallel system. It is also referred to as $k - out - of - n : G$ system. An n-component series system is a $n - out - of - n : G$ system whereas a parallel system with n-components is a $1 - out - of - n : G$ system. When all of the components are independent and identical, the reliability of

$k-out-of-n$ system can be written as $R_S = \sum_{j=k}^{n} \binom{n}{j} r^j (1-r)^{n-j}$, where r is the component reliability.

5.5 Reliability Network System

Let us consider a network with n subsystems. The goal of the redundancy allocation problem is to determine the number of redundant components in each of q parallel subsystems and reliability levels of $(n-q)$ general subsystems so as to maximize the overall system reliability subject to the given resource constraints and also to minimize the overall system cost subject to the given constraint on system reliability. The corresponding problems are mixed-integer nonlinear programming problems as follows:

Problem-4: Maximize $R_S(x, R) = f(R_1(x_1), R_2(x_2), ..., R_q(x_q), R_{q+1}, ..., R_n)$ subject to $C_i(x, R) \leq c_i,\ i = 1, ..., m$, and $1 \leq l_j \leq x_j \leq u_j$, x_j integer, $j = 1, ..., q$, $0 < \alpha_j \leq R_{j+1} \leq \beta_j < 1,\ \ j = 1, ..., n-q$,

Problem-5: Minimize$C_w(x, R)$ subject to $R_S(x, R) \geq R^*$, where $R_S(x, R) = f(R_1(x_1), R_2(x_2), ..., R_q(x_q), R_{q+1}, ..., R_n)$

6 GA Based Constrained Handling Technique

In the application of GA for solving reliability optimization problem with interval objective, there arises an important question for handling the constraints relating to the problem. During the past, several methods have been proposed to handle the constraints in evolutionary algorithms [30], [32] for solving the same problem with fixed objective. These methods can be classified into several types, viz. penalty function techniques, methods that preserve the feasibility of solutions, methods that clearly distinguish between feasible and infeasible solutions and hybrid methods. Among these methods, penalty function technique is very well known and widely applicable. In this technique, the amount of constraint violations is added /subtracted to the objective function in different ways. When the objective function is increased/ decreased with a penalty term multiplied by so called penalty coefficient there arises a difficulty to select the initial value and upgrading strategy for the penalty coefficient. To overcome this difficulty, Deb [30] proposed a GA based Parameter Free Penalty (PFP) technique. In this technique, the worst fitness value of GA for feasible solutions is considered as the fitness value of infeasible solution without multiplying the penalty coefficient i.e., the fitness function values of infeasible solutions are independent of the objective function value for the same solution. Therefore, according to the PFP technique, the converted problem of problem (1-3) is as follows:

$$\text{Maximize } [\hat{R}_{SL}(x), \hat{R}_{SR}(x)] = [R_{SL}(x), R_{SR}(x)] - [\sum_{i=1}^{m} \max(0, g_i(x) - c_i), \sum_{i=1}^{m} \max(0, g_i(x) - c_i)] + \theta(x) \quad (1)$$

where $\theta(x) = \begin{cases} [0,0] & \text{if } x \in S \\ -[R_{SL}(x), R_{SR}(x)] + \min[R_{SL}, R_{SR}] & \text{if } x \notin S \end{cases}$
and $S = \{x : g_i(x) \leq c_i, i = 1, 2, ..., m \text{ and } l \leq x \leq u\}$

Here $\min[R_{SL}, R_{SR}]$ is the value of interval valued objective function of the worst feasible solution in the population. Alternatively, the problem may be solved with another fitness function by penalizing a large positive number (say M which can be written in the interval form as$[M, M]$) [18]. This penalty function method is known as Big-M penalty and its form is as follows:

$$\text{Maximize } [\hat{R}_{SL}(x), \hat{R}_{SR}(x)] = [R_{SL}(x), R_{SR}(x)] + \theta(x) \quad (2)$$

where $\theta(x) = \begin{cases} [0,0] \text{ if } x \in S \\ -[R_{SL}(x), R_{SR}(x)] + [-M, -M] \text{ if} x \notin S \end{cases}$
and $S = \{x : g_i(x) \leq c_i, i = 1, 2, ..., m \text{ and } l \leq x \leq u\}$

The above problems (1) and (2) are nonlinear unconstrained integer programming problem with interval coefficients. Also, according to the PFP technique, the converted problem of problem-4 is as follows:

$$\text{Maximize } \hat{R}_S(x, R) = R_S(x, R) - \sum_{i=1}^{m} [\max(0, C_i(x, R) - c_i), \max(0, C_i(x, R) - c_i)] + \theta(x, R) \quad (3)$$

where $\theta(x, R) = \begin{cases} [0,0] & , & if\ (x, R) \in X \\ -R_S(x, R) + \min[R_{SL}(x, R), R_{SR}(x, R)] & & if\ (x, R) \notin X \end{cases}$
and $X = \{(x, R) : C_i(x, R) \leq c_i,\ i = 1, ..., m \text{ and } l \leq x \leq u, \alpha \leq R \leq \beta\}$

Here $\min[R_{SL}(x, R), R_{SR}(x, R)]$ is the value of the interval valued objective function of the worst feasible solution in the population.

Alternatively, the problem may also be solved with another fitness function by penalizing a large positive number. The converted form is as follows:

$$\text{Maximize } \hat{R}_S(x, R) = R_S(x, R) + \theta(x, R) \quad (4)$$

where $\theta(x, R) = \begin{cases} [0,0] & if\ (x, R) \in X \\ -R_S(x, R) + [-M, -M] & if\ (x, R) \notin X \end{cases}$
and $X = \{(x, R) : C_i(x, R) \leq c_i,\ i = 1, ..., m \text{ and } l \leq x \leq u, \alpha \leq R \leq \beta\}$

Similarly, for Problem-5, the converted problem is as follows:

$$\text{Minimize } \hat{C}_w(x,R) = C_w(x,R) + \sum_{j=1}^{m} [\max(0, -R_{SL}(x,R) + R^*)] + \theta(x,R) \quad (5)$$

where $\theta(x,R) = \begin{cases} [0,0] & if\ (x,R) \in X \\ -C_w(x,R) + \max\{C_w(x,R)\} & if\ (x,R) \notin X \end{cases}$
and $X = \{(x,R) : -R_{SL}(x,R) + R^* \leq 0,\ i = 1,2,...,m \text{ and } l \leq x \leq u, \alpha \leq R \leq \beta\}$

Here $\max\{C_w(x,R)\}$ is the value of the interval valued objective function of the worst feasible solution in the population. Alternatively, the problem may also be solved with another fitness function by penalizing a large positive number. The converted problem is of the form

$$\text{Minimize } \hat{C}_w(x,R) = C_w(x,R) + \theta(x,R) \quad (6)$$

where $\theta(x,R) = \begin{cases} [0,0] & if (x,R) \in X \\ -C_w(x,R) + M & if (x,R) \notin X \end{cases}$
and
$X = \{(x,R) : -R_{SL}(x,R) + R^* \leq 0,\ i = 1,...,m \text{ and } l \leq x \leq u, \alpha \leq R \leq \beta\}$

The above problems (1-2) are non-linear unconstrained integer programming problem with interval coefficients whereas problems (3-6) are non-linear unconstrained mixed integer programming problem with interval coefficients.

7 Genetic Algorithm

Genetic Algorithm is a well-known stochastic method of global optimization based on the evolutionary theory of Darwin: ' The survival of the fittest' and natural genetics (Goldberg [23]). It has successfully been applied in different real world application problems. This algorithm starts with an initial population of chromosomes. These populations will be improved from generation to generation by an artificial evolution process. During each generation, each chromosome in the entire population is evaluated using the measure of fitness and the population of the next generation is created through different genetic operators. This algorithm can be implemented easily with the help of computer programming. In particular, it is very useful for solving complicated optimization problems which cannot be solved easily by analytical /direct/gradient based mathematical techniques.

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

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

Initially, the chromosomes/individuals are generated randomly. In this work, each chromosome/individual has n components/genes of which first q genes are relating to integer variables whereas the last $(n-q)$ are relating to floating point variable. These chromosomes/individuals compete with each other with their fitness values. Here, the transformed unconstrained objective function due to Big-M and PFP penalty are considered as the fitness function. In the proposed GA, the well-known tournament selection process is employed as the selection operator. The primary objective of this process is to emphasize the above average solutions and eliminate the below average solutions from the population for the next generation under the well-known evolutionary principle "Survival of the fittest". This selection procedure is based on the following assumptions:

1. When both the chromosomes / individuals are feasible then the one with better fitness value is selected.
2. When one chromosome/individual is feasible and another is infeasible then the feasible one is selected.
3. When both the chromosomes/individuals are infeasible with unequal constraint violations, then the chromosome with less constraint violation is selected.
4. When both the chromosomes/individuals are infeasible with equal constraint violations, then any one chromosome/individual is selected.

After the selection process, new offspring will be created through crossover and mutation processes. In this work, we have used uniform crossover and uniform mutation in the genes corresponding to the integer variables, whole arithmetical crossover and boundary mutation for the last gene of the chromosome.

The computational steps of crossover are as follows:

Step-1: Find the integral value of the product of p_cross and p_size and store it in N.
Step-2: Select two chromosomes v_k and v_i randomly from the population.

Step-3: For first q genes, compute the components $\bar{x}_{kj}$ and $\bar{x}_{ij}$ $(j = 1, 2, ..., q)$ of two offspring by either $\bar{x}_{kj} = x_{kj} - g$ and $\bar{x}_{ij} = x_{ij} + g$ if $x_{kj} > x_{ij}$ or, $\bar{x}_{kj} = x_{kj} + g$ and $\bar{x}_{ij} = x_{ij} - g$, where g is a random integer number between 0 and $|x_{kj} - x_{ij}|$, $j = 1, 2, ..., q$ and for the last gene, compute the last components x'_{kj} and x'_{ij} of two offspring will be created by $x'_{kj} = cx_{kj} + (1 - c)x_{ij}$ and $x'_{ij} = (1 - c)x_{kj} + cx_{ij}$ where c is a random number between 0 and 1.

Step-4: Repeat step-2 and step-3 for $\frac{N}{2}$ times.

The computational steps of mutation are as follows:

Step-1: Find the integral value of the product of *p_mute* and *p_size* and store it in N.

Step-2: Select a chromosome v_i randomly from the population.

Step-3: Select a particular gene v_{ik} $(k = 1, 2, ..., q)$ of chromosome v_i for mutation and domain of v_{ik} is $[l_{ik}, u_{ik}]$.

Step-4: Create new gene v'_{ik} corresponding to the selected gene v_{ik} by mutation process as follows:

For $k = 1, 2, ..., q$

$$v'_{ik} = \begin{cases} v_{ik} + \Delta(u_{ik} - v_{ik}), \text{ if random digit is } 0 \\ v_{ik} - \Delta(v_{ik} - l_{ik}), \text{ if random digit is } 1 \end{cases}$$

$\Delta(y)$ returns a value in the range $[0, y]$, is a random integer between $[0, y]$.

$$\text{Otherwise } v'_{ik} = \begin{cases} l_{ik} & \text{if a random digit is } 0. \\ u_{ik} & \text{if a random digit is } 1. \end{cases}$$

Step-5: Repeat Step-2 to Step-4 for N times.

Sometimes, in any generation, there is a chance that the best chromosome may be lost when a new population is created by crossover and mutation operations. To remove this situation the worst individual/chromosome is replaced by the best individual/chromosome in the current generation. This process is called elitism. The different steps of this algorithm are described as follows:

7.1 Algorithm

Step 1: Initialize the parameters of genetic algorithm, bounds of variables and different parameters of the problem.

Step 2: $t = 0$ [t represents the number of current generation].

Step 3: Initialize the chromosome of the population$P(t)$[$P(t)$ represents the population at $t - th$ generation].

Step 4: Evaluate the fitness function of each chromosome of $P(t)$ considering any one of the objective function from (1-6) as fitness function.

Step 5: Find the best chromosome from the population $P(t)$.

Step 6: t is increased by unity.

Step 7: If the termination criterion is satisfied go to step-14, otherwise, go to next step.
Step 8: Select the population $P(t)$ from the population$P(t-1)$ of earlier generation by tournament selection process.
Step 9: Alter the population $P(t)$ by crossover, mutation and elitism process.
Step 10: Evaluate the fitness function value of each chromosome of $P(t)$.
Step 11: Find the best chromosome from $P(t)$.
Step 12: Compare the best chromosome of $P(t)$ and $P(t-1)$ and store better one.
Step 13: Go to step-6.
Step 14: Print the last found best chromosome (which is the solution of the optimization problem).
Step 15: End.

8 Numerical Example

To illustrate the proposed GAs (viz. PFP-GA and Big-M-GA) for solving earlier mentioned optimization problems with interval valued reliabilities of components, we have solved nine numerical examples. It is to be noted that for solving the said problem with fixed valued reliabilities of components, the reliability of each component is taken as interval with the same lower and upper bounds of interval. For each example, 20 independent runs have been performed by both the GAs, of which the following measurements have been collected to compare the performances of PFP-GA and Big-M-GA.

1. Best found system reliability
2. Average generations
3. Average CPU times

The proposed Genetic Algorithms are coded in C programming language and run in Linux environment. The computational work has been done on the PC which has Intel core-2 duo processor with 2 GHz. In this computation, different population size has been taken for different problems. However, the crossover and mutation rates are taken as 0.95 and 0.15 respectively.

Example 1: (related to Problem-1):

Maximize $[R_{SL}, R_{SR}] = \prod_{j=1}^{5} [\{1-(1-r_{jL})^{x_j}\}, \{1-(1-r_{jR})^{x_j}\}]$ subject to:

$\sum_{j=1}^{5} p_j x_j^2 - P \leq 0,$

$\sum_{j=1}^{5} c_j[x_j + \exp(\frac{x_j}{4})] - C \leq 0,$

$\sum_{j=1}^{5} w_j x_j \exp(\frac{x_j}{4}) - W \leq 0,$

The values of different parameters along with the interval valued reliabilities of Example-1 are given in Table 1.

Example 2: (related to Problem-2)
Maximize $[R_{SL}, R_{SR}] = \{1 - \langle 1 - (1 - [Q_{3L}, Q_{3R}](1 - [R_{1L}, R_{1R}][R_{2L}, R_{2R}]))[R_{4L}, R_{4R}]\rangle \ (1 - [R_{5L}, R_{5R}][R_{6L}, R_{6R}])\}(1 - [Q_{7L}, Q_{7R}][Q_{8L}, Q_{8R}][Q_{9L}, Q_{9R}])[R_{10L}, R_{10R}]$
subject to
$c_1 \exp(\frac{x_1}{2})x_2 + c_2 \exp(\frac{x_3}{2}) + c_3x_4 + c_4[x_5 + \exp(\frac{x_5}{4})] + c_5x_6^2x_7 + c_6x_8 + c_7x_9^3 \exp(\frac{x_{10}}{2}) - 120 \leq 0$,
$w_1x_1^2x_2 + w_2 \exp(\frac{x_3x_4}{2}) + w_3x_5 \exp(\frac{x_6}{4}) + w_4x_7x_8^3 + w_5[x_9 + \exp(\frac{x_9}{2})] + w_6x_2 \exp(\frac{x_{10}}{4}) - 130 \leq 0$,

Table 1 Parameters in Example 1

j	$[r_{jL}, r_{jR}]$	p_j	P	c_j	C	w_j	W
1	[0.76, 0.83]	1		7		7	
2	[0.82, 0.87]	2	110	7	175	8	200
3	[0.88, 0.93]	3		5		8	
4	[0.61, 0.67]	4		9		6	
5	[0.70, 0.80]	2		4		9	

The values of different parameters along with the interval valued reliabilities of Example-2 are given in Table 2.

Table 2 Parameters in Example 2

j	$[r_{jL}, r_{jR}]$	c_j	w_j	l_j	u_j
1	[0.80, 0.84]	8	16	1	4
2	[0.87, 0.90]	4	6	1	5
3	[0.89, 0.93]	2	7	1	6
4	[0.84, 0.86]	2	12	1	7
5	[0.88, 0.90]	1	7	1	5
6	[0.90, 0.95]	6	1	1	5
7	[0.80, 0.85]	2	9	1	3
8	[0.90, 0.95]	8	−	1	3
9	[0.80, 0.83]	−	−	1	4
10	[0.88, 0.92]	−	−	1	6

Example 3: (related to Problem 2)
Maximize
$[R_{SL}, R_{SR}] = \{1 - \langle 1 - (1 - [Q_{3L}, Q_{3R}](1 - [R_{1L}, R_{1R}][R_{2L}, R_{2R}]))[R_{4L}, R_{4R}]\rangle(1 - [R_{5L}, R_{5R}][R_{6L}, R_{6R}])\}(1 - [Q_{7L}, Q_{7R}][Q_{8L}, Q_{8R}][Q_{9L}, Q_{9R}])[R_{10L}, R_{10R}]$

subject to
$c_1 \exp(\frac{x_1}{2})x_2 + c_2 \exp(\frac{x_3}{2}) + c_3x_4 + c_4[x_5 + \exp(\frac{x_5}{4})] + c_5x_6^2x_7 + c_6x_8 + c_7x_9^3 \exp(\frac{x_{10}}{2}) - 120 \leq 0,$
$w_1x_1^2x_2 + w_2 \exp(\frac{x_3x_4}{2}) + w_3x_5 \exp(\frac{x_6}{4}) + w_4x_7x_8^3 + w_5[x_9 + \exp(\frac{x_9}{2})] + w_6x_2 \exp(\frac{x_{10}}{4}) - 130 \leq 0,$

The values of different parameters along with the interval valued reliabilities of Example-3 are given in Table 3.

Table 3 Parameters in Example -3

j	$[r_{jL}, r_{jR}]$	c_j	w_j	l_j	u_j
1	$[0.83, 0.83]$	8	16	1	4
2	$[0.89, 0.89]$	4	6	1	5
3	$[0.92, 0.92]$	2	7	1	6
4	$[0.85, 0.85]$	2	12	1	7
5	$[0.89, 0.89]$	1	7	1	5
6	$[0.93, 0.93]$	6	1	1	5
7	$[0.83, 0.83]$	2	9	1	3
8	$[0.94, 0.94]$	8	−	1	3
9	$[0.82, 0.82]$	−	−	1	4
10	$[0.91, 0.91]$	−	−	1	6

Example 4: (related to Problem-3)
Maximize $[R_{SL}, R_{SR}] = [R_{5L}, R_{5R}](1-[Q_{1L}, Q_{1R}][Q_{3L}, Q_{3R}])(1-[Q_{2L}, Q_{2R}] [Q_{4L}, Q_{4R}]) + [Q_{5L}, Q_{5R}]\{1 - (1 - [R_{1L}, R_{1R}][R_{2L}, R_{2R}])(1 - [R_{3L}, R_{3R}] [R_{4L}, R_{4R}])\}$ subject to:

$10 \exp(\frac{x_1}{2})x_2 + 20x_3 + 3x_4^2 + 8x_5 - 200 \leq 0,$
$10 \exp(\frac{x_1}{2}) + 4 \exp(x_2) + 2x_3^3 + 6[x_4^2 + \exp(\frac{x_4}{4})] + 7 \exp(\frac{x_5}{4}) - 310 \leq 0,$
$12[x_2^2 + \exp(x_2)] + 5x_3 \exp(\frac{x_3}{4}) + 3x_1x_4^2 + 2x_5^3 - 520 \leq 0,$
$(1, 1, 1, 1, 1) \leq (x_1, x_2, x_3, x_4, x_5) \leq (6, 3, 5, 6, 6),$
where
$R_1(x_1)$
$= \{[0.78, 0.82], [0.83, 0.88], [0.89, 0.91], [0.915, 0.935], [0.94, 0.96], [0.965, 0.985]\};$
$R_2(x_2) = 1 - (1 - [0.73, 0.77])^{x_2};$
$R_3(x_3) = \sum_{k=2}^{x_3+1} \binom{x_3+1}{k} ([0.87, 0.89])^k ([0.11, 0.13])^{x_3+1-k};$
$R_4(x_4) = 1 - (1 - [0.68, 0.72])^{x_4};$
$R_5(x_5) = 1 - (1 - [0.83, 0.86])^{x_5};$

Example 5: (related to Problem-3)
Maximize $[R_{SL}, R_{SR}] = [R_{5L}, R_{5R}](1-[Q_{1L}, Q_{1R}][Q_{3L}, Q_{3R}])(1-[Q_{2L}, Q_{2R}] [Q_{4L}, Q_{4R}]) + [Q_{5L}, Q_{5R}]\{1 - (1 - [R_{1L}, R_{1R}][R_{2L}, R_{2R}])(1 - [R_{3L}, R_{3R}] [R_{4L}, R_{4R}])\}$ subject to

$10\exp(\frac{x_1}{2})x_2 + 20x_3 + 3x_4^2 + 8x_5 - 200 \leq 0,$
$10\exp(\frac{x_1}{2}) + 4\exp(x_2) + 2x_3^3 + 6[x_4^2 + \exp(\frac{x_4}{4})] + 7\exp(\frac{x_5}{4}) - 310 \leq 0,$
$12[x_2^2 + \exp(x_2)] + 5x_3\exp(\frac{x_3}{4}) + 3x_1x_4^2 + 2x_5^3 - 520 \leq 0,$
$(1,1,1,1,1) \leq (x_1,x_2,x_3,x_4,x_5) \leq (6,3,5,6,6),$
where
$R_1(x_1)$
$= \{[0.8,0.8],[0.85,0.85],[0.9,0.9],[0.925,0.925],[0.95,0.95],[0.975,0.975]\};$
$R_2(x_2) = 1-(1-[0.75,0.75])^{x_2};$
$R_3(x_3) = \sum_{k=2}^{x_3+1}\binom{x_3+1}{k}([0.88,0.88])^k([0.12,0.12])^{x_3+1-k};$
$R_4(x_4) = 1-(1-[0.7,0.7])^{x_4};$
$R_5(x_5) = 1-(1-[0.85,0.85])^{x_5};$

The examples 1, 2, 3, 4 and 5 have been solved by two different methods PFP-GA and Big-M-GA and the results have been shown in Table 4.

Table 4 Numerical results for Example 1-5

Method	Example	Population size	x	Best found system reliability R_S	Average CPU seconds	Average Generation
PFP-GA	1	50	$(3,2,2,3,3)$	[0.860808, 0.930985]	0.0001	12.10
	2	100	$(1,2,2,5,4,4,2,2,1,5)$	[0.999909, 0.999987]	0.0105	17.55
	3	100	$(1,2,2,5,4,4,2,2,1,5)$	[0.999975, 0.999975]	0.0100	17.55
	4	200	$(5,1,2,4,4)$	[0.991225, 0.999872]	0.0200	11.20
	5	100	$(3,2,4,4,2)$	[0.999382, 0.999382]	0.0100	12.40
Big-M-GA	1	50	$(3,2,2,3,3)$	[0.860808, 0.930985]	0.0001	12.80
	2	100	$(1,2,2,5,4,4,2,2,1,5)$	[0.999909, 0.999987]	0.0110	17.75
	3	100	$(1,2,2,5,4,4,2,2,1,5)$	[0.999975, 0.999975]	0.0100	17.75
	4	200	$(5,1,2,4,4)$	[0.991225, 0.999872]	0.0200	10.90
	5	100	$(3,2,4,4,2)$	[0.999382, 0.999382]	0.0100	12.55

Example 6: (related to the Problem-4)
Maximize $R_S(x,R) = R_1R_2 + Q_2R_3R_4 + Q_1R_2R_3R_4 + R_1Q_2Q_3R_4R_5$ $+Q_1R_2R_3Q_4R_5$ subject to:

$$C_1(x) = x_1x_2 + 2.2x_2x_3 + 1.5x_2x_4 + 2\exp\left(\frac{0.01}{1-R_5}\right) \leq 28,$$
$$C_2(x) = x_1 + 0.1x_2 + 2x_3 + x_4 + 5\exp\left(\frac{0.01}{1-R_5}\right) \leq 25,$$
$$C_3(x) = x_1^2 + (x_2-2)^3 + 1.5x_3 + x_4 + 0.6\exp\left(\frac{0.01}{1-R_5}\right) < 21,$$

where $1 \leq x_i \leq 6,$ and are integers, $i = 1, 2, 3, 4,\ 0.50 \leq R_5 \leq 0.99,$
and $R_i = R_i(x_i) = 1 - (1 - r_i)^{x_i}, i = 1, 2, 3, 4,\ Q_i = 1 - R_i, i = 1, ..., 5$
$r_1 = [0.69, 0.72]$,$r_2 = [0.83, 0.86]$,$r_3 = [0.73, 0.76]$,$r_4 = [0.79, 0.81]$

Example 7: (related to the Problem-4)
Maximize $R_S(x, R) = R_1R_2 + Q_2R_3R_4 + Q_1R_2R_3R_4 + R_1Q_2Q_3R_4R_5$ $+Q_1R_2R_3Q_4R_5$ subject to:

$$C_1(x) = x_1x_2 + 2.2x_2x_3 + 1.5x_2x_4 + 2\exp\left(\frac{0.01}{1-R_5}\right) \leq 28,$$

$$C_2(x) = x_1 + 0.1x_2 + 2x_3 + x_4 + 5\exp\left(\frac{0.01}{1-R_5}\right) \leq 25,$$

$$C_3(x) = x_1^2 + (x_2 - 2)^3 + 1.5x_3 + x_4 + 0.6\exp\left(\frac{0.01}{1-R_5}\right) < 21,$$

where $1 \leq x_i \leq 6,$ and are integers, $i = 1, 2, 3, 4,\ 0.50 \leq R_5 \leq 0.99,$
and $R_i = R_i(x_i) = 1 - (1 - r_i)^{x_i}, i = 1, 2, 3, 4,\ Q_i = 1 - R_i, i = 1, ..., 5$
$r_1 = [0.70, 0.70]$,$r_2 = [0.85, 0.85]$,$r_3 = [0.75, 0.75]$,$r_4 = [0.80, 0.80]$

The examples 6 and 7 have been solved by two different methods PFP-GA and Big-M-GA and the results have been shown in Table 5.

Table 5 Numerical results for Examples 6-7

Method	Example	Population size	(x, R)	Best found system reliability R_S	Average CPU seconds
PFP-GA	6	150	$(2, 3, 1, 2, 0.9900)$	$[0.958412, 0.997223]$	0.2705
	7	150	$(2, 1, 6, 5, 0.9396)$	$[0.999927, 0.999927]$	0.2655
Big-M-GA	6	150	$(2, 3, 1, 2, 0.9900)$	$[0.958412, 0.997223]$	0.2700
	7	150	$(2, 1, 6, 5, 0.9396)$	$[0.999927, 0.999927]$	0.2590

Example 8: (related to the Problem-5)
Minimize $C_w(x, R) = 0.3C_1(x_1) + 0.5C_2(x_2) + 0.2C_3(x_3)$ subject to:
$R_S(x, R) \geq [0.999, 0.999]$, where $1 \leq x_i \leq 6,$ and are integers, $i = 1, 2, 3, 4,\ 0.50 \leq R_5 \leq 0.99$, and $R_S(x, R)$, $C_i(i = 1, 2, 3)$ are defined in Example 6.

Example 9: (related to the Problem-5)
Minimize $C_w(x, R) = 0.3C_1(x_1) + 0.5C_2(x_2) + 0.2C_3(x_3)$ subject to:
$R_S(x, R) \geq [0.999, 0.999]$, where $1 \leq x_i \leq 6,$ and are integers, $i = 1, 2, 3, 4,\ 0.50 \leq R_5 \leq 0.99$, and $R_S(x, R)$, $C_i(i = 1, 2, 3)$ are defined in Example 7.

The examples 8 and 9 have been solved by two different methods PFP-GA and Big-M-GA and the results have been shown in Table 6.

Table 6 Numerical results for Example 8-9

Method	Example	Population size	(x, R)	Best found system cost C_w	Best found system reliability R_S	Average CPU seconds
PFP	8	150	$(6, 4, 2, 1, 0.8601)$	33.03866	$[0.997290, 0.999885]$	0.3675
-GA	9	150	$(2, 1, 4, 4, 0.5)$	17.97505	$[0.999081, 0.999081]$	0.3525
Big-M	8	150	$(6, 4, 2, 1, 0.8601)$	33.03866	$[0.997290, 0.999885]$	0.3010
-GA	9	150	$(2, 1, 4, 4, 0.5)$	17.97505	$[0.999081, 0.999081]$	0.2815

Table 7 Comparison of results of Ha and Kuo [1] and the proposed methods.

	Example	x	System Reliability R_s	Average CPU seconds
Ha and Kuo [1]	4(E2)	$(1, 1, 3, 4, 2, 1, 1, 3, 1, 4)$	0.999876	–
PFP-GA(this work)	3	$(1, 2, 2, 5, 4, 4, 2, 2, 1, 5)$	**0.999975**	**0.0100**
Big-M-GA(this work)	3	$(1, 2, 2, 5, 4, 4, 2, 2, 1, 5)$	**0.999975**	**0.0100**
Ha and Kuo [1]	4(E1)	$(1, 3, 4, 3, 3)$	0.999373	–
PFP-GA(this work)	5	$(3, 2, 4, 4, 2)$	**0.999382**	**0.0100**
Big-M-GA(this work)	5	$(3, 2, 4, 4, 2)$	**0.999382**	**0.0100**

Table 8 Comparison of results of Sun et al. [9] and the proposed methods.

	Example	(x, R)	System cost C_w	System Reliability R_s	Average CPU seconds
Sun et al.[9]	2	$(2, 1, 6, 5, 0.9396)$		0.99992653	9.84
PFP-GA(this work)	7	$(2, 1, 6, 5, 0.9396)$		**0.999927**	**0.2655**
Big-M-GA(this work)	7	$(2, 1, 6, 5, 0.9396)$		**0.999927**	**0.2590**
Sun et al.[9]	4	$(1, 1, 5, 4, .05)$	18.53505		15.97
PFP-GA(this work)	9	$(2, 1, 4, 4, 0.5)$	**17.97505**		**0.3525**
Big-M-GA(this work)	9	$(2, 1, 4, 4, 0.5)$	**17.97505**		**0.2815**

9 Sensitivity Analysis

To study the performance of our proposed GAs like PFP-GA and Big-M-GA based on two different types of penalty techniques, sensitivity analyses (for Example-1) have been carried out graphically on the centre of the interval valued system reliability with respect to GA parameters like, population size, crossover and mutation rates separately keeping the other parameters at their

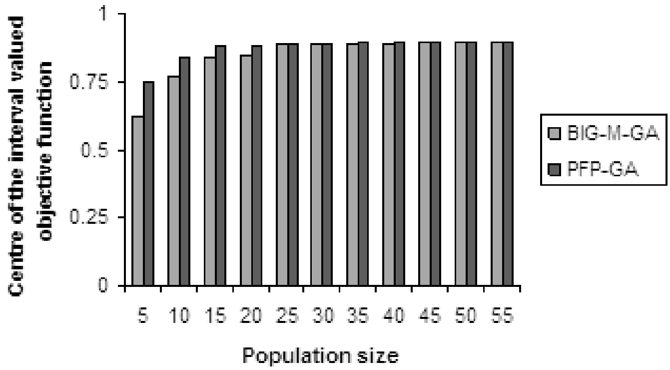

Fig. 8 Population size vs. centre of the objective function value

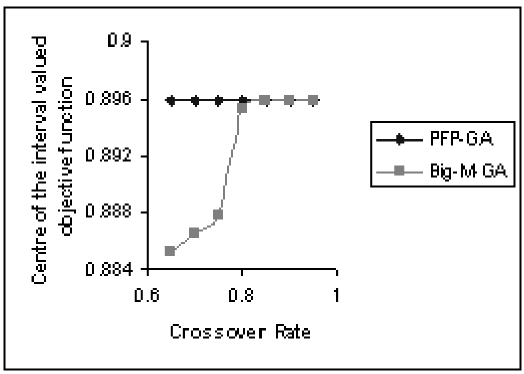

Fig. 9 Crossover rate vs.centre of the objective function value

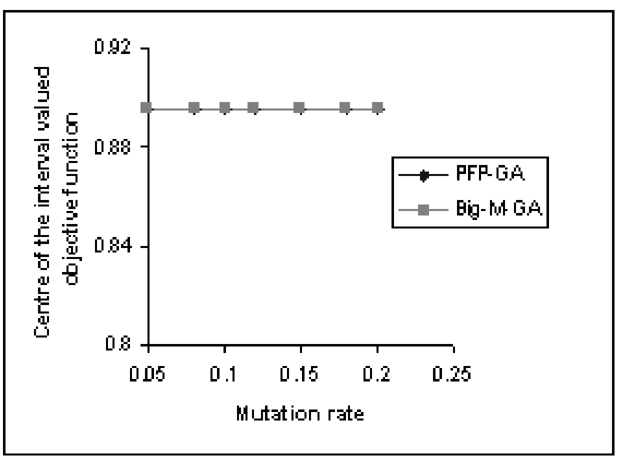

Fig. 10 Mutation rate vs.centre of the objective function value

original values. These are shown in Fig.8–Fig.10. From Fig.8, it is evident that in case of PFP-GA, smaller population size gives the better system reliability. However, both the GAs are stable when population size exceeds the number 30. From Fig.9, it is observed that the system reliability is stable if we consider the crossover rate between the interval (0.65, 0.95) in case of PFP-GA. In both GAs, it is stable when crossover rate is greater than 0.8. In Fig.10, sensitivity analyses have been done with respect to mutation rate. In both GAs, the value of system reliability be the same.

10 Conclusions

In this chapter, the problems of redundancy allocation problems of series system, hierarchical series-parallel system, complicated system and reliability network system with some resource constraints have been solved. In those systems, reliability of each component has been considered as imprecise number and this imprecise number has been represented by an interval number which is more appropriate representation among other representations like, random variable representation with known probability distribution, fuzzy set with known fuzzy membership function or fuzzy number. For handling the resource constraints, the corresponding problem has been converted to unconstrained optimization problem with the help of two different parameter free penalty techniques. Therefore, the transformed problem is of unconstrained interval valued optimization problem with integer and/or mixed integer variables. To solve the transformed problems, two different real coded GA based on different fitness functions have been developed for integer and mixed integer variables with interval valued fitness function, tournament selection, crossover (uniform crossover for integer variables and whole arithmetical crossover for floating point variables), mutation (uniform mutation for integer variables and boundary mutation for floating point variables) and elitism of size one. In the existing penalty function technique, tuning of penalty parameter is a formidable task. However, here tuning of parameters is not required as these are penalty parameter free techniques. From the performance of GAs, it is observed that the GAs with both fitness functions due to different penalty techniques take very lesser CPU times with very small generations to solve the problems. It is clear from the expression of the system reliability that the system reliability is a monotonically increasing function with respect to the individual reliabilities of the components. Therefore, there is one optimum setup irrespective of the choice of the upper bound or lower bound of the component reliabilities. As a result, the optimum setup obtained from the upper bound/lower bound will provide both the upper bound and the lower bound of the optimum system reliability. These approaches have wider applicabilities in solving the constrained optimization problems arisen in every sector of real life situation. However, as the proposed techniques are parameter free, these do not require the tuning of penalty parameter.

References

1. Ha, C., Kuo, W.: Reliability redundancy allocation: An improved realization for nonconvex nonlinear programming problems. European Journal of Operational Research 171, 124–138 (2006)
2. Coelho, L.S.: An efficient particle swarm approach for mixed-integer programming in reliability redundancy optimization applications. Reliability Engineering and System Safety 94, 830–837 (2009)
3. Tillman, F.A., Hwuang, C.L., Kuo, W.: Optimization technique for system reliability with redundancy: A Review. IEEE Trans. Reliability 26, 148–155 (1977)
4. Kuo, W., Prasad, V.R., Tillman, F.A., Hwuang, C.L.: Optimal Reliability Design Fundamentals and application. Cambridge University Press, Cambridge (2001)
5. Misra, K.B., Sharama, U.: An efficient algorithm to solve integer-programming problems arising in system reliability design. IEEE Trans. Reliability 40, 81–91 (1991)
6. Nakagawa, Y., Nakashima, K., Hattori, Y.: Optimal reliability allocation by branch-and-bounded technique. IEEE Trans. Reliability 27, 31–38 (1978)
7. Ohtagaki, H., Nakagawa, Y., Iwasaki, A., Narihisa, H.: Smart greedy procedure for solving a nonlinear knapsacclass of reliability optimization problems. Mathl. Comput. Modeling 22, 261–272 (1995)
8. Sun, X., Duan, L.: Optimal Condition and Branch and Bound Algorithm for Constrained Redundancy Optimization in Series System. Optimization and Engineering 3, 53–65 (2002)
9. Sun, X.L., Mckinnon, K.I.M., Li, D.: A convexification method for a class of global optimization problems with applications to reliability optimization. Journal of Global Optimization 21, 185–199 (2001)
10. Gen, M., Yun, Y.: Soft computing approach for reliability optimization. Reliability Engineering and System Safety 91, 1008–1026 (2006)
11. Chern, M.S.: On the computational complexity of reliability redundancy allocation in a series system. Operations Research Letter 11, 309–315 (1992)
12. Martorell, S., Sanchez, A., Carlos, S., Serradell, V.: Alternatives and challenges in optimizing industrial safety using genetic algorithms. Reliability Engineering and System Safety 86, 25–38 (2004)
13. Zhao, J., Liu, Z., Dao, M.: Reliability optimization using multiobjective ant colony system approaches. Reliability Engineering and system safety 92, 109–120 (2007)
14. Zio, E.: Reliability engineering: Old problems and new challanges. Reliability Engineering and System Safety 94, 125–141 (2009)
15. Coolen, F.P.A., Newby, M.J.: Bayesian reliability analysis with imprecise prior probabilities. Reliability Engineering and System Safety 43, 75–85 (1994)
16. Utkin, L.V., Gurov, S.V.: Imprecise reliability of general structures. Knowledge and Information Systems 1(4), 459–480 (1999)
17. Utkin, L.V., Gurov, S.V.: New reliability models based on imprecise probabilities. In: Hsu, C. (ed.) Advanced Signal Processing Technology, pp. 110–139. World Scientific, Singapore (2001)
18. Gupta, R.K., Bhunia, A.K., Roy, D.: A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliability of components. Journal of Computational and Applied Mathematics 232, 275–284 (2009)

19. Bhunia, A.K., Sahoo, L., Roy, D.: Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm. Applied Mathematics and Computation 216, 929–939 (2010)
20. Mahato, S.K., Bhunia, A.K.: Interval-Arithmetic-Oriented Interval Computing Technique for Global Optimization. In: Applied Mathematics Research eXpress 2006, pp. 1–19 (2006)
21. Ishibuchi, H., Tanaka, H.: Multiobjective programming in optimization of the interval objective function. European Journal of Operational Research 48, 219–225 (1990)
22. Chanas, S., Kuchta, D.: Multiobjective programming in the optimization of interval objective functions-A generalized approach. European journal of Operational Research 94, 594–598 (1996)
23. Goldberg, D.E.: Genetic Algorithms: Search, Optimization and Machine Learning. Addison-Wesley, Reading (1989)
24. Gen, M., Cheng, R.: Genetic algorithms and engineering optimization. John Wiley and Sons Inc., Chichester (2000)
25. Michalawich, Z.: Genetic Algorithms + Data structure = Evaluation Programs. Springer, Berlin (1996)
26. Sakawa, M.: Genetic Algorithms and fuzzy multiobjective optimization. Kluwer Academic Publishers, Dordrecht (2002)
27. Levitin, G.: Genetic algorithms in reliability engineering. Reliability Engineering and System Safety 91, 9751–9976 (2006)
28. Villanueva, J.F., Sanchez, A.I., Carlos, S., Martorell, S.: Genetic algorithm-based optimization of testing and maintenance under uncertain unavailability and cost estimation: A survey of strategies for harmonizing evoluation and accuracy. Reliability Engineering and System Safety 93, 1830–1841 (2008)
29. Ye, Z., Li, Z., Xie, M.: Some improvements on adaptive genetic algorithms for reliability-related applications. Reliability Engineering and System Safety 95, 120–126 (2010)
30. Deb, K.: An efficient constraint handling method for genetic algorithms. Computer Methods in Applied Mechanics and Engineering 186, 311–338 (2000)
31. Aggarwal, K.K., Gupta, J.S.: Penalty function approach in heuristic algorithms for constrained. IEEE Transactions on Reliability 54(3), 549–558 (2005)
32. Miettinen, K., Makela, M.M., Toivanen, J.: Numerical comparison of some Penalty-Based Constraint Handling Techniques in Genetic Algorithms. Journal of Global Optimization 2, 427–446 (2003)
33. Hansen, E., Walster, G.W.: Global optimization using interval analysis. Marcel Dekker Inc., New York (2004)
34. Karmakar, S., Mahato, S., Bhunia, A.K.: Interval oriented multi-section techniques for global optimization. Journal of Computation and Applied Mathematics 224, 476–491 (2009)