JOURNAL OF APPLIED QUANTITATIVE METHODS

Quantitative Methods Inquires

A GENETIC ALGORITHM BASED RELIABILITY REDUNDANCY OPTIMIZATION FOR INTERVAL VALUED RELIABILITIES OF COMPONENTS[1]

L. SAHOO
Department of Mathematics, Raniganj Girls' College, Raniganj-713347, India

E-mail: lxsahoo@gmail.com

A.K. BHUNIA
Department of Mathematics, The University of Burdwan, Burdwan-713104, India

E-mail: math_akbhunia@buruniv.ac.in

D. ROY
Centre for Management Studies, The University of Burdwan, Burdwan-713104, India

E-mail: dr.diliproy@gmail.com

Abstract: *The goal of this paper is to solve the constrained redundancy allocation problem of series-parallel/parallel-series/complex system with the interval valued reliability of each component. For maximizing the overall system reliability under limited resource constraints, the problem is formulated as an unconstrained integer programming problem with interval coefficients by two different penalty techniques. To solve the transformed problem, we have developed two different real coded GAs for integer variables with tournament selection, uniform crossover, uniform mutation and different fitness functions based on penalty techniques. As a special case, considering the lower and upper bounds of the initial valued reliabilities of the component as the same, the corresponding problem has been solved. To illustrate the model, some numerical examples have been solved by our developed GAs and the results have been compared. Finally, to study the stability of our developed GAs with respect to the different GA parameters (like, population size, crossover and mutation rates), sensitivity analyses have been shown graphically.*

Key words: *Redundancy allocation; Complex system; Reliability optimization; Genetic algorithm; Interval numbers; Order relations; Penalty technique*

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 system that often require complicated operations and highly sophisticated management. From any respect, the system reliability is a crucial measure to be considered in system operation and risk management. When designing a highly reliable system, there arises an important question how to obtain a balance between the reliability and other resources e.g., cost, volume and weight. As a result, addition of redundant components or increase of component reliability leads to the increase of the system reliability. In the last few decades, optimization problems including redundancy allocation have been treated by many researchers as integer nonlinear

programming problems with one or several resource constraints [1]. During the last two decades, numerous reliability design techniques have been introduced. These techniques can be classified as implicit enumeration, dynamic programming, branch and bound technique, linear programming, Lagrangian multiplier method, heuristic methods and so on. To solve this type of problem, one may refer to the works of Ghare and Taylor[2], Tillman et al.[3], Nakagawa et al.[4], Tzafestas [5], Shantikumar and Yao[6], Misra and Sharama[7], Sundarajan[8], Chern[9],Ohtagaki et al. [10] ,Sun and Li[11], Ha and Kuo[12], Gen and Yun[13].Tillman and Kuo et al.[14] have extensively reviewed the several optimization techniques for system reliability design in their books. In those works, the reliabilities of the system components are assumed to be known at a fixed positive level which lies in the open interval $(0,1)$. However, in real life situations, the reliabilities of these individual components may vary due to different reasons. Any technology cannot 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. Therefore it is sensible to treat the component reliabilities as positive imprecise numbers in open interval $(0,1)$, instead of fixed real numbers. To tackle the problem with imprecise numbers, generally stochastic, fuzzy and fuzzy-stochastic approaches are used. In stochastic approach, the parameters are assumed as random variables with known probability distribution. 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 and other as random variables. However, 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 is a formidable task. To overcome these difficulties for representation of imprecise numbers by different approaches, one may represent the same by interval number as it is the best representation among others. Due to this new representation, the objective function of the reduced redundancy allocation problem is interval valued, which is to be maximized under given constraints.

In this study, we have considered GA-based approaches for solving reliability optimization problems with interval objective. As objective function of the redundancy allocation problem is interval valued, to solve this type of problem by GA method, order relations of interval numbers are essential for selection/ reproduction operation as well as for finding the best chromosome in each generation. Recently, Mahato and Bhunia[15] proposed the modified definitions of order relations with respect to optimistic and pessimistic decision maker's point of view for maximization and minimization problems.

In this paper, we have considered the problem of constrained redundancy allocation in series system, hierarchical series-parallel system and complex or non-parallel-series system with interval valued reliability components. The problem is formulated as a non-linear constrained integer programming problem with interval coefficients for maximizing the overall system reliability under resource constraints. During the last few years, several techniques were proposed for solving constraints optimization problem with fixed coefficient with the help of GAs[16-19]. Among them, penalty function techniques are very popular in solving the same by GAs [13, 16, 19]. This method transforms the constrained optimization problem to an unconstrained optimization problem by penalizing the objective function corresponding to the infeasible solution. In this work, to solve the constrained optimization problem we have converted it into unconstrained one by two different type 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 same GA operators like tournament selection, uniform crossover, uniform mutation 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. This type of redundancy allocation problem with fixed valued of reliabilities of

components have been solved and illustrated with some existing numerical examples. Finally, to study the stability of our developed GAs for interval valued objective with respect to different GA parameters, sensitivity analyses have been performed and shown graphically.

2. Finite interval arithmetic

An interval number A is a closed interval defined by $A=[a_L,a_R]=\{x:a_L\le x\le a_R, x\in R\}$, where a_L and a_R are the left and right limits, respectively and R is the set of all real numbers. An interval A can also be expressed in terms of centre and radius as $A=\langle a_C,a_W\rangle=\{x:a_C-a_W\le x\le a_C+a_W, x\in R\}$, where a_C and a_W are, respectively, the centre and radius of the interval A i.e., $a_C=(a_L+a_R)/2$ and $a_W=(a_R-a_L)/2$. Actually, each real number can be regarded as an interval, such as for all $x\in R$, x can be written as an interval $[x,x]$ which has zero radius. The basic arithmetical operations like, addition, subtraction, multiplication, division and integral power of interval number are available in the book of interval analysis [20].

3. Order relations of interval numbers

In this paper, as we have considered some parameters as interval valued, to find the optimal solution of the optimization problem so we have to discuss the order relations of interval numbers for maximization problems. Let A and B be two closed intervals. These two intervals may be one of the following three types.

Type1. Two intervals A and B are disjoint.

Type2. Intervals are partially overlapping.

Type3. Either $A\subset B$ or $B\subset A$

In this case, we shall consider the definitions of order relations for maximization problems developed recently by Mahato and Bhunia [15] in the context of optimistic and pessimistic decision maker's point of view.

4. Assumptions and Notations

To develop the paper, the following assumptions and notations have been considered .

4.1 Assumptions

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

4.2 Notations

The following notations have been used in the entire paper.

m	Number of resource constraints
n	Number of stages of the system
x	$(x_1,x_2,...,x_n)$
j	Index for stage
x_j	Number of redundant components at stage j
l_j	Lower limit on x_j
u_j	Upper limit on x_j
r_j	Reliability of component at stage j which is interval valued

r_{jL} Lower limit of r_j

r_{jR} Upper limit of r_j

b_i Total amount of $i-th$ resource available

$R_j(x_i)$ $1-(1-r_j)^{x_i}$, the reliability of stage j

$R_{jL}(x)$ Lower bound of $R_j(x)$

$R_{jR}(x)$ Upper bound of $R_j(x)$

Q_j $1-R_j$

$R_S=[R_{SL},R_{SR}]$ System reliability which is interval valued

5. Problem formulation

Let us consider a system consisting of n components. Assume that the system and its components have two states, viz. "operating state" and "failure state". To represent the state of the component j ($j=1,2,3,...,n$), let us define a binary variable y_j as follows:

$$y_j=\begin{cases}1, & \text{if the component j is in operating state,}\\ 0, & \text{otherwise,}\end{cases}$$

In this case, we get a set of 2^n numbers of binary vectors with n components each. Similarly to represent the state of the system, we define another binary variable ϕ given by

$$\phi=\begin{cases}1, & \text{if the system is in operating state}\\ 0, & \text{otherwise}\end{cases}$$

Clearly the value of ϕ is 1 for some possibilities of y and 0 for other possibilities. Therefore, in most of the reliability systems, we can explicitly define ϕ as a function of y [say $\phi(y)$]. So that the system state ϕ corresponding to the vector y is given by $\phi=\phi(y)$. The function $\phi(y)$ is called the structure function of the system.

Now we consider a system with structure function $\phi(y)$. A component j is said to be relevant to function $\phi(y)$ either

$$\phi(y_1,...,y_{j-1},1,y_{j+1},...,y_n)=1\ \forall y_r,\ r\neq j \text{ and } r=1,2,...,n$$

or, $\phi(y_1,...,y_{j-1},0,y_{j+1},...,y_n)=0\ \forall y_r, r\neq j \text{ and } r=1,2,...,n$

Definition 5.1 A system is called coherent if and only if the structure function $\phi(y)$ satisfies the following conditions:

(i) $\phi(y)$ is a non-decreasing function for each y_j,

(ii) Each component is relevant to $\phi(y)$.

Now, we consider a coherent system consisting of n subsystems and m constraints, in which each subsystem has a number of redundancies and all components, are stochastically independent. The objective of the redundancy allocation problem is to find the number of redundancies in each subsystem in order to maximize the overall system reliability subject to the given constraints. The corresponding problem formulated as an Integer Non-Linear Programming (INLP) is as follows:

Maximize R_S

subject to the constraints

$g_i(x)\le b_i,\quad \text{for } i=1,2,...,m$

$l_j \le x_j \le u_j$, for $j = 1, 2, ..., n$

Now if we consider the reliability of each component as interval valued, the earlier mentioned problem is transformed to the non-linear constrained integer programming problem with interval objective as follows:

Maximize $[R_{SL}, R_{SR}]$

subject to the constraints

$g_i(x) \le b_i, \quad \text{for } i = 1, 2, ..., m$

$l_j \le x_j \le u_j$, for $j = 1, 2, ..., n$

Now, we shall discuss the different types of redundancy allocation problems.

5.2 Constrained redundancy optimization problem for a series system

It is well known that a series system (ref. Fig.1) 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 redundant components as shown in Fig. 2. 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 fixed values of reliability can be formulated as follows:

IN

OUT

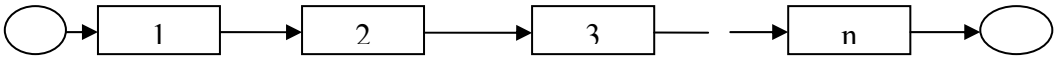

Figure 1

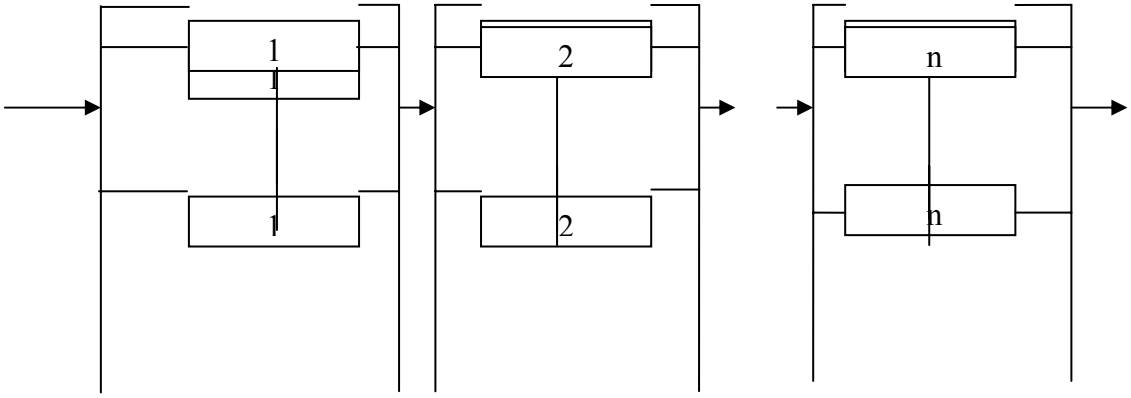

Figure 2

Maximize $R_S = \prod_{j=1}^{n} [1 - (1 - r_j)^{x_j}]$

subject to $g_i(x) \le b_i, \quad i = 1, 2, ..., m$

and $l_j \le x_j \le u_j$, for $j = 1, 2, ..., n$

where $r_j \in (0, 1)$

Now if we consider component reliabilities as interval valued, i.e., $r_j = [r_{jL}, r_{jR}]$ then the above problem reduces to

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

subject to $g_i(x) \le b_i, \quad i = 1,2,...,m$

and $l_j \le x_j \le u_j$, for $j = 1,2,...,n$

This is an INLP with interval valued objective.

5. 3. 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. 3. 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 fixed values of reliability can be formulated as follows:

Maximize $R_S = \{1-\langle 1-[1-Q_3(1-R_1R_2)]R_4\rangle(1-R_5R_6)\}(1-Q_7Q_8Q_9)R_{10}$

subject to $g_i(x) \le b_i, \quad i = 1,2,...,m$

and $l_j \le x_j \le u_j$, for $j = 1,2,...,n$

where $r_j \in (0,1)$

Now if we consider the component reliabilities as interval valued, i.e., $r_j = [r_{jL}, r_{jR}]$ then the above problem reduces to the following form:

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

subject to $g_i(x) \le b_i, \quad i = 1,2,...,m$

and $l_j \le x_j \le u_j$, for $j = 1,2,...,n$

This is an INLP with interval valued objective.

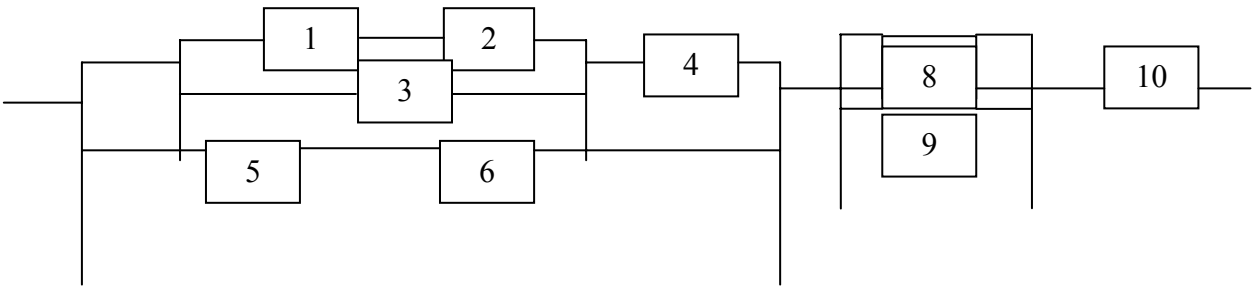

Figure 3

5.4 Complex 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 complex 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.4. 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:

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

where $R_j = R_j(x_j)$ and $Q_j = 1-R_j$ for all $j=1,...,5$.

Mathematically, the constrained redundancy optimization problem for such complex system for fixed values of reliability can be formulated as follows:

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

subject to $g_i(x) \le b_i, \quad i=1,2,...,m$

and $l_j \le x_j \le u_j$, for $j=1,2,...,n$

where $r_j \in (0,1)$

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

Now if we consider the component reliabilities as interval valued, i.e., $r_j = [r_{jL}, r_{jR}]$ then the above problem reduces to the following form as follows:

$$\text{Maximize}[R_{SL}, R_{SR}] = [R_{5L}, R_{5R}](1-[Q_{1L}, Q_{1R}][Q_{3L}, Q_{3R}])(1-[Q_{2L}, Q_{2R}][Q_{4L}, Q_{4R}])$$
$$+[Q_{5L}, Q_{5R}]\{1-(1-[R_{1L}, R_{1R}][R_{2L}, R_{2R}])(1-[R_{3L}, R_{3R}][R_{4L}, R_{4R}])\}$$

subject to $g_i(x) \le b_i, \quad i=1,2,...,m$

and $l_j \le x_j \le u_j$, for $j=1,2,...,n$

This is an INLP with interval valued objective.

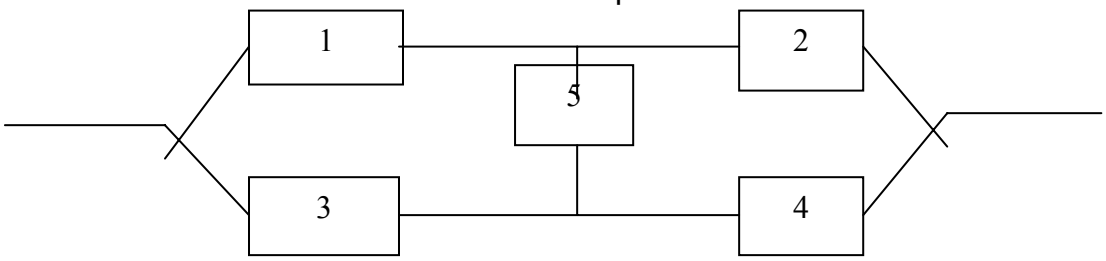

Figure 4

5.5 k-out-of-n **System**

A $k-out-of-n$ system is an n-component system which functions when at least k of its n 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.

6. GA based constraint handling technique

In the application of Genetic Algorithm for solving the reliability optimization problems with interval objective[21-23], there arises an important question for handling the constraints relating to the problem. During the past, several approaches viz. penalty-based method, methods that preserve the feasibility of solutions, methods that clearly distinguish between feasible and unfeasible solutions and hybrid methods have been proposed to handle the constraints in evolutionary algorithms [16-19] for solving the problems with fixed objective [24]. Among these approaches, penalty-based technique is very well known and widely applicable. In this approach, constraints are added /subtracted to the objective function in different ways. To avoid the infeasible solution, the fitness function is increased /decreased with a penalty term multiplied by a so called penalty coefficient. However, there arises a difficulty to select the initial value and updating strategy for the penalty coefficient. To avoid this situation, Deb [24] proposed a GA based penalty technique called Parameter Free Penalty (PFP) technique replacing the objective function value of GA by worst objective value of feasible solution of previous generation and without multiplying the penalty coefficient. This means that the fitness function values of infeasible solution do not depend on the objective function value. As the mentioned constrained optimization problem is of interval valued, therefore the problem to be solved with a new fitness function is of the following interval form:

Maximize

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

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 $S = \{x : g_i(x) \le b_i, i = 1, 2, ..., m \text{ and } l \le x \le u\}$

Here the parameter $\min[R_{SL}, R_{SR}]$ 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 interval form as $[M, M]$) [25]. We denote this penalty as Big-M penalty and its form is as follows:

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

(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 $S = \{x : g_i(x) \le b_i, i = 1, 2, ..., m \text{ and } l \le x \le u\}$

The above problems (1) and (2) are non-linear unconstrained integer programming problem with interval coefficient.

7. Solution procedure

Now we have to solve the above nonlinear maximization problems (1) and (2) with the help of GA. For this purpose, we have developed two types of GA with different fitness function based on PFP and Big-M penalty techniques. We call these GAs as PFP-GA and Big-M-GA respectively.

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 the objective function of either (1) or (2) 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.

For implementing the above GA in solving the problems (1) and (2) 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

7.2 GA Parameters

There are different parameters used in the genetic algorithm, viz. population size (*p_size*), maximum number of generations (*m_gen*), crossover rate/probability of crossover (*p_cross*) and mutation rate/probability of mutation (*p_mute*). There is no hard and fast rule for choosing the population size for GA. However, if the population size is considered to be large, storing of the data in the intermediate steps of GA may create some difficulties at the time of computation with the help of computer. On the other hand, for very small population size, some genetic operations can not be implemented. Particularly, mutation operator does not work properly as the mutation rate is very low. Regarding the maximum number of generations, there is no indication for considering this value. Generally, it is problem

dependent on problem to problem. Particularly, it depends upon the number of genes (variables) of a chromosome (solution) in artificial genetics. Again, from the natural genetics, it is obvious that the crossover rate is always 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 be the number of genes (variables) of the chromosomes (solutions).

7.3 Chromosome representation and initialization

In the applications of GA, the appropriate chromosome (individual) representation of solution for the given problems is an important task to the users. There are different types of representations, viz. binary, real, octal, hexadecimal coding, available in the existing literature. Among these representations, real coding representation is very popular as this type of chromosome representation looks like a vector. In this representation, each component (gene) of the chromosome is the values of decision variables of the optimization problems which are to be solved by GA.

7.4 Initialization of population

After the selection of chromosome representation, the next step is to initialize the chromosomes that will take part in the artificial genetic operations like natural genetics. This procedure produces population size number of chromosomes in which every component for each chromosome is randomly generated within the bounds of the corresponding decision variable. There are different procedures for selecting a random number for each component of the chromosomes. Here for each component of the chromosome, a random value is selected from the discrete set of values within its bounds.

7.5 Evaluation of fitness function

After getting a population of potential solutions, we need to check how good they are. So we have to calculate the fitness value for each chromosome. In this evaluation, the value of objective function corresponding to the chromosome is taken as the fitness value of that chromosome. Here, the transformed unconstrained objective function due to different penalty technique is considered as the fitness function.

7.6 Selection

In artificial genetics, the selection operator plays an important role. Usually, it is the first operator applied to the population. The primary objective of this operator is to emphasize on the above average solutions and eliminate below average solutions from the population for the next generation under the well-known evolutionary principle "survival of the fittest". In this work, we have used tournament selection scheme of size two with replacement as the selection operator. This operator selects the better chromosome/individual from randomly selected two chromosomes/individuals. This selection procedure is based on the following assumptions:

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

(ii) When one chromosome/individual is feasible and another is infeasible then the feasible one is selected.

(iii) When both the chromosomes/individuals are infeasible with unequal constraint violations, then the chromosome with less constraint violation is selected.

(iv) When both the chromosomes/individuals are infeasible with equal constraint violations, then any one chromosome/individual is selected.

7.7 Crossover

After the selection process, other genetic operators like crossover and mutation are applied to the resulting chromosomes (those which have survived). Crossover is an operation that really empowers the GA. It operates on two or more parent solutions at a time and

generates offspring by recombining the feature of the parent solutions. In this operation, expected $[\text{p\_cross*p\_size}]$ (* denotes the product and $[\ ]$ denotes the integral value) number of chromosomes will take part. Hence in order to perform the crossover operation, select $[\text{p\_cross*p\_size}]$ numbers of chromosomes are selected. In our work, the operation is done in the following manner.

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

Step-2: Select the chromosomes v_k and v_i randomly from the population for crossover.

Step-3: The components v'_{kj} and v'_{ij} $(j=1,2,...,n)$ of two offspring will be created by either $v'_{kj}=v_{kj}-g$ and $v'_{ij}=v_{ij}+g$ if $v_{kj}>v_{ij}$

Or, $v'_{kj}=v_{kj}+g$ and $v'_{ij}=v_{ij}-g$, where g is a random integer number between 0 and $|v_{kj}-v_{ij}|$.

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

7.8 Mutation

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. Mainly, this operator is responsible for fine tuning capabilities 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 generated through selection and crossover operations. Mutation attempts to bump the population gently into a slightly better way, i.e., the mutation changes single or all the genes of a randomly selected chromosome slightly. In this work, we have used uniform mutation. If the gene v_{ik} of chromosome v_i is selected for mutation and domain of v_{ik} is $[l_{ik},u_{ik}]$, then the reduced value of v_{ik} is given by

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

where $k\in\{1,2,...,n\}$ and $\Delta(y)$ returns a value in the range $[0,y]$.

In our work, we have taken

$\Delta(y)$ = A random integer between $[0,y]$.

7.9 Elitism

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 that best individual/chromosome in the current generation. Instead of single chromosome one or more chromosomes may take part in this operation. This process is called as elitism.

7.10 Termination criteria

The termination condition is to stop the algorithm when either of the following three conditions is satisfied:

(i) the best individual does not improve over specified generations.

(ii) the total improvement of the last certain number of best solutions is less than a pre-assigned small positive number or

(iii) The number of generations reaches maximum number of generation i.e., *max_gen*.

In this work we have used first condition and we take 10 generations as a specified generation.

8. Numerical Examples

To illustrate the proposed GAs (viz. PFP-GA and Big-M-GA) for solving constrained redundancy allocation problems with interval valued reliabilities of components, we have solved four numerical examples. It is to be noted that for solving the said problem with fixed valued reliabilities of components, the reliability of each component are taken as interval with the same lower and upper bounds of interval. In first three examples, the reliabilities of the components are interval valued whereas in the last example (taken from Ha and Kuo [12]), it is fixed. For each example, 20 independent runs have been performed by both the GAs, of which the following measurements have been collected to compare the performance of PFP-GA and Big-M-GA.

(i) Best found system reliability
(ii) Average generations
(iii) Average CPU times

The proposed Genetic Algorithms are coded in C programming language and run in Linux environment. The computation 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-8.1 (Ref. section 5.2)

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

subject to

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

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

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

Table 1: Parameter used in Example -8.1

j	r_j	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	

Example-8.2 (Ref. section 5.3)

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

subject to

$$c_1 \exp(\frac{x_1}{2})x_2 + c_2 \exp(\frac{x_3}{2}) + c_3 x_4 + c_4[x_5 + \exp(\frac{x_5}{4})] + c_5 {x_6}^2 x_7 + c_6 x_8 + c_7 {x_9}^3 \exp(\frac{x_{10}}{2}) - 120 \le 0,$$

$$w_1 {x_1}^2 x_2 + w_2 \exp(\frac{x_3 x_4}{2}) + w_3 x_5 \exp(\frac{x_6}{4}) + w_4 x_7 {x_8}^3 + w_5[x_9 + \exp(\frac{x_9}{2})] + w_6 x_2 \exp(\frac{x_{10}}{4}) - 130 \le 0,$$

Table 2: Parameter used in Example-8.2

j	1	2	3	4	5	6	7	8	9	10
r_j	[.80,.84]	[.87,.90]	[.89,.93]	[.84,.86]	[.88,.90]	[.9,.95]	[.8,.85]	[.91,.95]	[.8,.83]	[.88,.92]
c_j	8	4	2	2	1	6	2	8	-	-
w_j	16	6	7	12	7	1	9	-	-	-
l_j	1	1	1	1	1	1	1	1	1	1
u_j	4	5	6	7	5	5	3	3	4	6

Example-8.3 (Ref. section 5.4)

$$\text{Maximize}[R_{SL}, R_{SR}] = [R_{5L}, R_{5R}](1 - [Q_{1L}, Q_{1R}][Q_{3L}, Q_{3R}])(1 - [Q_{2L}, Q_{2R}][Q_{4L}, Q_{4R}])$$
$$+[Q_{5L}, Q_{5R}]\{1 - (1 - [R_{1L}, R_{1R}][R_{2L}, R_{2R}])(1 - [R_{3L}, R_{3R}][R_{4L}, R_{4R}])\}$$

subject to

$$10 \exp(\frac{x_1}{2})x_2 + 20x_3 + 3{x_4}^2 + 8x_5 - 200 \le 0,$$

$$10 \exp(\frac{x_1}{2}) + 4 \exp(x_2) + 2{x_3}^3 + 6[{x_4}^2 + \exp(\frac{x_4}{4})] + 7 \exp(\frac{x_5}{4}) - 310 \le 0,$$

$$12[{x_2}^2 + \exp(x_2)] + 5x_3 \exp(\frac{x_3}{4}) + 3x_1 {x_4}^2 + 2{x_5}^3 - 520 \le 0,$$

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

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-8.4 (Ref. section 5.4)

$$\text{Maximize}[R_{SL}, R_{SR}] = [R_{5L}, R_{5R}](1-[Q_{1L}, Q_{1R}][Q_{3L}, Q_{3R}])(1-[Q_{2L}, Q_{2R}][Q_{4L}, Q_{4R}])$$
$$+[Q_{5L}, Q_{5R}]\{1-(1-[R_{1L}, R_{1R}][R_{2L}, R_{2R}])(1-[R_{3L}, R_{3R}][R_{4L}, R_{4R}])\}$$

subject to

$$10\exp(\frac{x_1}{2})x_2 + 20x_3 + 3x_4^{\,2} + 8x_5 - 200 \le 0,$$

$$10\exp(\frac{x_1}{2}) + 4\exp(x_2) + 2x_3^{\,3} + 6[x_4^{\,2} + \exp(\frac{x_4}{4})] + 7\exp(\frac{x_5}{4}) - 310 \le 0,$$

$$12[x_2^{\,2} + \exp(x_2)] + 5x_3\exp(\frac{x_3}{4}) + 3x_1x_4^{\,2} + 2x_5^{\,3} - 520 \le 0,$$

$$(1,1,1,1,1) \le (x_1, x_2, x_3, x_4, x_5) \le (6,3,5,6,6),\ \mathrm{x} \in \square_n^+$$

where

$$R_1(x_1) = \{[.8,.8],[.85,.85],[.9,.9],[.925,.925],[.95,.95],[.975,.975]\};$$
$$R_2(x_2) = 1-(1-[0.75,0.75])^{x_2};$$
$$R_3(x_3) = \sum_{k=2}^{x_3+1}\binom{x_3+1}{k}([0.88,0.88])^k([0.12,0.12])^{x_3+1-k};$$

$$R_4(x_4) = 1-(1-[0.7,0.7])^{x_4};$$
$$R_5(x_5) = 1-(1-[0.85,0.85])^{x_5};$$

Table 3: Numerical results for Example 8.1-8.4

Method	Example	x	Best found system reliability R_s	Average CPU time (sec.)	Average Generation	Population Size
PFP-GA	8.1	(3,2,2,3,3)	[0.860808,0.930985]	0.0001	12.10	50
	8.2	(1,2,2,5,4,4,2,2,1,5)	[0.999909,0.999987]	0.0105	17.55	100
	8.3	(5,1,2,4,4)	[0.991225,0.999872]	0.0200	11.20	200
	8.4	(3,2,4,4,2)	[0.999382,0.999382]	0.0100	12.40	100
Big-M-GA	8.1	(3,2,2,3,3)	[0.860808,0.930985]	0.0001	12.80	50
	8.2	(1,2,2,5,4,4,2,2,1,5)	[0.999909,0.999987]	0.0110	17.75	100
	8.3	(5,1,2,4,4)	[0.991225,0.999872]	0.0200	10.90	200
	8.4	(3,2,4,4,2)	[0.999382,0.999382]	0.0100	12.55	100

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 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 rate separately keeping the other parameters at their original values. These are shown in Fig.5-Fig.7.

From Fig.5, 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.6, 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.7, sensitivity analyses have been done with respect to mutation rate. In both GAs, the value of system reliability be the same.

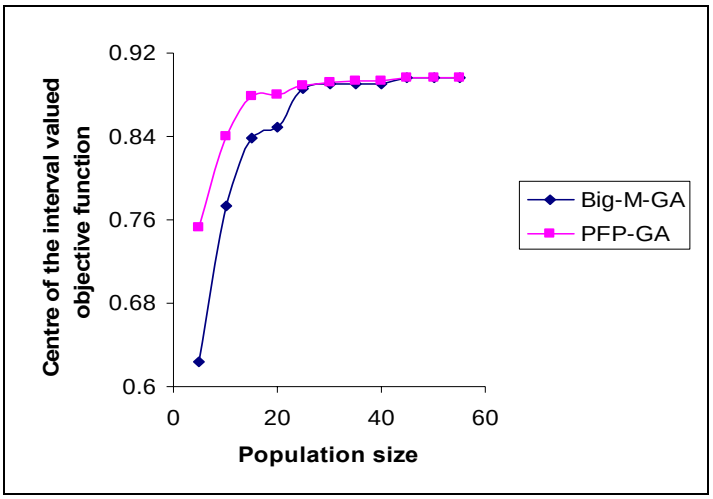

Figure 5. Population size vs. centre of the objective function value

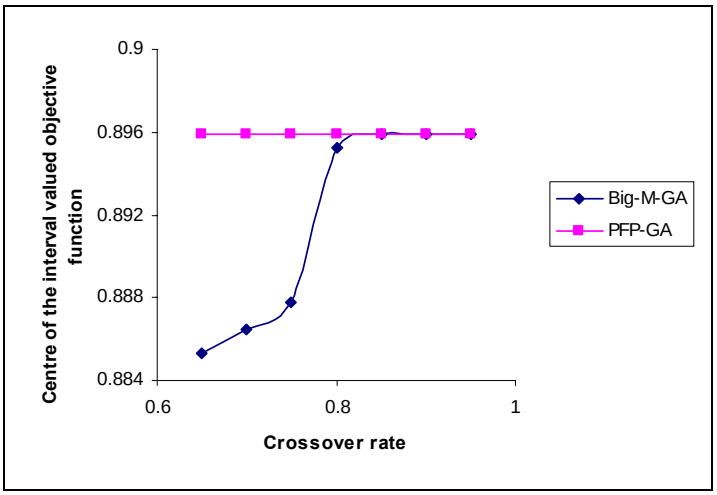

Figure 6. Crossover rate vs. centre of the objective function value

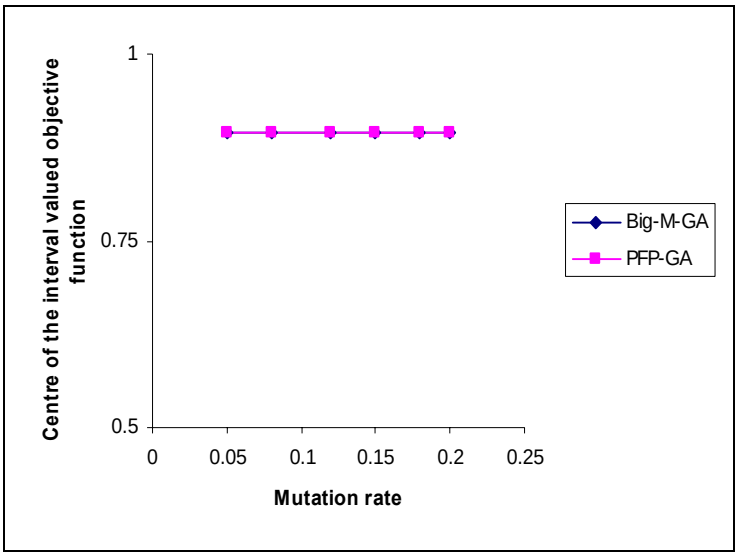

Figure 7. Mutation rate vs. centre of the objective function value

10. Conclusions

In this paper, reliability redundancy allocation problems of series-parallel/parallel-series/complex 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 of resource constraints, the corresponding problem has been converted into 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 variables. To solve the transformed problem, we have developed real coded GA for integer variables with interval valued fitness function, tournament selection, uniform crossover, uniform mutation and elitism of size one. In tournament selection and elitism operation, recently developed definitions of interval ranking have been used. In the existing penalty function technique, tuning of penalty parameter is a formidable task. From the performance of GAs, it is observed that both the GAs with both fitness function due to different penalty techniques take very lesser CPU time 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. However, the interval approach presented in this paper has a wider applicability. For future research, one may use the proposed GAs in solving other reliability optimization problems like Chance- constrained reliability optimization problems, Network reliability optimization problems, etc.

References

1. Kuo, W., Prasad, V. R., Tillman, F. A. and Hwuang, C. L. **Optimal Reliability Design Fundamentals and application**, Cambridge University Press, 2001
2. Ghare, P. M. and Taylor, R. E. **Optimal redundancy for reliability in series system**, Operations Research 17, 1969, pp. 838-847

3. Tillman, F. A., Hwuang, C. L. and Kuo, W. **Optimization technique for system reliability with redundancy: A Review**. IEEE Trans. Reliability 26, 1977, pp.148-155

4. Nakagawa, Y., Nakashima, K. and Hattori, Y. **Optimal reliability allocation by branch-and -bounded technique**. IEEE Trans. Reliability, 27, 1978, pp.31-38

5. Tzafestas, S. G. **Optimization of system reliability: A survey of problems and techniques**, International Journal of System Science, 11, 1980, pp.455-486

6. Shantikumar, J. G. and Yao, D. D. **On server allocation in multiple centre manufacturing systems**, Operations Research 36, 1988, pp. 333-341

7. Misra, K.B. and Sharama, U. **An efficient algorithm to solve integer-programming problems arising in system reliability design**. IEEE Trans. Reliability 40, 1991, pp. 81-91

8. Sundarajan C**. Guide to Reliability Engineering: Data, Analysis, Applications, Implementation, and Management**, New York: Van Nostrand Reinhold, 1991

9. Chern, M.S. **On the computational complexity of reliability redundancy allocation in a series system**, Operations Research Letter 11, 1992, pp. 309-315

10. Ohtagaki, H., Nakagawa, Y., Iwasaki, A. and Narihisa, H. **Smart greedy procedure for solving a nonlinear knapsacclass of reliability optimization problems**, Mathl. Comput. Modeling 22, 1995, pp.261-272

11. Sun Xiaoling and Li Duan. **Optimal Condition and Branch and Bound Algorithm for Constrained Redundancy Optimization in Series System**, Optimization and Engineering, 3, 2002, pp. 53-65

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

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

14. Kuo, W., Prasad, V. R., Tillman, F. A. and Hwuang, C. L. **Optimal Reliability Design Fundamentals and application**, Cambridge University Press, 2001

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

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

17. Michalawich, Z. **Genetic Algorithms + Data structure = Evaluation Programs**, Berlin: Springer Verlag, 1996

18. Gen, M. and Cheng, R. **Genetic algorithms and engineering optimization**, John Wiley & Sons inc., 2000

19. Sakawa, M. **Genetic Algorithms and fuzzy multiobjective optimization**, Kiuwer Academic Publishers, 2002

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

21. Ishibuchi, H. and Tanaka, H. **Multiobjective programming in optimization of the interval objective function**, European Journal of Operational Research 48, 1990, pp.219-225

22. Chanas, S. and Kuchta, D. **Multiobjective programming in the optimization of interval objective functions-A generalized approach**, European journal of Operational Research, 94, 1996, pp.594-598

23. Sengupta, A. and Pal, T. K. **Theory and methodology on comparing interval numbers**, European Journal of Operational Research,127, 2007, pp.28-43

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

25. Gupta, R. K., Bhunia, A. K. and 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 Mathematics232, 2009, pp. 275-284

[1] **Acknowledgements**
The second author would like to acknowledge the support of Defence Research Development Organization (DRDO), India, for conducting this work.