OPTIMIZATION OF CHANCE CONSTRAINED REDUNDANCY ALLOCATION PROBLEM WITH NON-CRISP COMPONENT RELIABILITIES

Dr. Sanat Kumar Mahato

Assistant Professor, Department of Mathematics, Mejia Govt. College, West Bengal, India

ABSTRACT

In this paper, we have considered the optimal solution of the reliability-redundancy allocation problems (RAP) involving chance constraints in non-crisp environment. The reliabilities of the components are not fixed numbers rather they are non-crisp/imprecise numbers. Also the constraints in the RAP considered are chance constraints which are stochastic in nature. We have proposed a stochastic simulation based Genetic Algorithm approach for solving the reliability optimization problems of the type mentioned in this paper. The impreciseness has been considered in terms of the stochastic approach and the interval approach. In case of the stochastic approach, the reliabilities of the components are taken to be random variables which are distributed normally. After that Monte-Carlo simulation method is used to convert the chance constraints into the deterministic ones. The changed problem is then solved by the real coded genetic algorithm based on stochastic simulation and the constraint handling procedure. Few numerical examples are reported to explain the efficiency of the projected method.

Keyword: - *Optimization, Reliability-redundancy Allocation Problem, Non-crisp Number, Interval Number, Random Numbers, Genetic Algorithm, Chance Constraint, Stochastic Simulation*

---

1. INTRODUCTION

It has been assumed, in the most of the existing works available in the literature that all the probabilities are precise i.e., deterministic while solving stochastic optimization problems of reliability. The implication of deterministic probabilities is that the entire probabilistic information about the components as well as system behavior is available ready in hand. The deterministic probabilistic information relies on the two situations mentioned below:

a) All the probabilities or probability distributions are known or perfectly determinable.
b) The system components are independent i.e., all the random variables, describing the component reliability behavior are independent.

In the most of the optimization techniques in reliability optimization, the supposition on uncertainty is based on the exact probabilities and the reliabilities of the system components are to be known and fixed positive numbers which lying in [0,1] [4,8,15-17, 22- 24,31-33]. The precise system reliability can be calculated theoretically if both the aforementioned situations are fulfilled. Though, in the most of the situations of real-life phenomena where either the system is new or it exists only as a project, there do not exists plenty statistical data. Only some incomplete information about the system components is available. Thus the reliability of each component of a system becomes be an imprecise number. To deal with the problems with such imprecise numbers, generally stochastic, fuzzy and fuzzy-stochastic approaches are being adopted and the respective problems are transformed into deterministic problems before solving them. In fuzzy approach, the parameters, the constraints and objective function are considered as either fuzzy sets with known membership functions or fuzzy numbers whereas in stochastic approach, the system parameters are to be random variables with known probability distributions. On the other hand, in fuzzy-stochastic approach, some of the parameters are considered as fuzzy numbers and others as random variables. Apart from these approaches, interval method can be useful for this. In this method, an interval number is used to represent the imprecise number. For this representation, the system reliability would be interval valued. Only a few works has been reported in this area, considering the system parameters as interval valued. The works of Gupta et al. [6], Bhunia et al. [1], Sahoo et al. [26-30], Bhunia and Sahoo [5], Mahato et al. [19] must be mentioned in this context.

In this paper, we have proposed a stochastic simulation based genetic algorithm approach [11, 12] for solving chance constrained [3,10,13,21] reliability optimization problem considering the reliability of each component of a system as either precise or imprecise number lying in [0,1]. To represent this impreciseness, we have applied stochastic and interval approach. In stochastic approach, the reliability of each component is considered as a random variable with normal distribution. At first, the chance constraints have been transformed into deterministic constraints by Monte-Carlo simulation technique [25]. Then the transformed problem has been solved by real coded genetic algorithm based constrained handling technique. To conclude the methodology as well as to test the performance of the proposed technique three numerical examples have been chosen for solving them using the method mentioned.

2. NOTATIONS USED

The notations which have been used in this paper are given in the table below

Table -1: Notations Used

Notation	Description
n	number of subsystems
x_j	number of redundant components in the *j*-th subsystem
$x=(x_1,x_2,...,x_n)$	redundant vector
r_j	reliability of each component in the *j*-th subsystem which is precise
$\bar{r}_j=[r_{jL},r_{jR}]$	reliability of each component in the *j*-th subsystem which is interval
$\tilde{r}_j \sim N(m_{r_j},\sigma^2_{r_j})$	reliability of each component in the *j*-th subsystem which is stochastic in nature and follows normal distribution with parameters m_{r_j} and σ_{r_j}
$R_S(x),\bar{R}_S(x),\tilde{R}_S(x)$	precise system reliability, interval valued system reliability, stochastic system reliability
$g_i(x),\tilde{g}_i(x)$	precise and stochastic valued usability of *i*-th constraint respectively
$b_i,\tilde{b}_i$	precise and stochastic valued total availability of *i*-th resource
$l_j(\geq 1),\ u_j$	Lower and upper bounds of x_j
γ_i	level of significance of *i*-th chance constraint
P(A)	probability of the event A
$U(a,b)$	uniform distribution over [*a*, *b*]
$N(m,\sigma^2)$	normal distribution with parameters m (mean) and σ (standard deviation)
popsize	population size
maxgen	maximum number of generations
pmute	probability of mutation
pcross	probability of crossover

3. ASSUMPTIONS

The assumptions which have been taken up in constructing the reliability-redundancy optimization problem with the chance constraints are,

a) Reliability of each component may be precise, interval or, stochastic depending upon the problem.
b) The system does fail on failure of a component of any subsystem
c) All the redundancies are active and there is no provision of repairing of components.
d) The components and also the system will have only two states either operating or failure.
e) The system or any of the subsystem has only states, the operating state or the failure state.
f) The resource constraints are chance constraints with resource vector as imprecise in nature.
g) The coefficients in the left hand side of the resource constraints are precise or stochastic in nature depending upon the problem.

4. FINITE INTERVAL ARITHMETIC

An interval number $A=[a_L,a_R]$ is defined to be the closed interval $A=[a_L,a_R]=\{x: a_L \le x \le a_R, x \in \mathbb{R}\}$, where a_L, a_R are the lower and upper bounds respectively and $\mathbb{R}$ is the set of all real numbers. The interval number $A=[a_L,a_R]$ can also be represented in the centre and the width form as $A=\langle a_c, a_w\rangle$, where $a_c=(a_L+a_R)/2$ and $a_w=(a_R-a_L)/2$ be the centre and radius of the interval A. It is to be noted that every real number $x\in\mathbb{R}$ can also be treated as a degenerate interval $[x,x]$ of zero width. The works of Hansen and Walster [7] and Karmakar et al. [14] may be referred for details regarding interval arithmetic, integral power of interval number and also the n-th root as well as the rational power of interval number.

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

Addition of two interval numbers A and B: $A+B=[a_L,a_R]+[b_L,b_R]=[a_L+b_L, a_R+b_R]$.

Subtraction of an interval number B from another one A:
$A-B=[a_L,a_R]-[b_L,b_R]=[a_L,a_R]+[-b_R,-b_L]=[a_L-b_R, a_R-b_L]$.

Multiplication of an interval number A by any real number k: For any real number *k*,

$$kA=k[a_L,a_R]=\begin{cases}[ka_L,ka_R] \text{ if } k\ge 0\\ [ka_R,ka_L] \text{ if } k<0.\end{cases}$$

Multiplication of two interval numbers A and B:
$A\times B=[a_L,a_R]\times[b_L,b_R]=[\min(a_Lb_L,a_Lb_R,a_Rb_L,a_Rb_R),\max(a_Lb_L,a_Lb_R,a_Rb_L,a_Rb_R)]$.

Division of an interval number A by another one B: $\frac{A}{B}=A\times\frac{1}{B}=[a_L,a_R]\times[\frac{1}{b_R},\frac{1}{b_L}]$, provided $0\notin[b_L,b_R]$.

Positive integral power of an interval number A: Let $A=[a_L,a_R]$ be an interval then for any non-negative integer *n*,

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

4.1 RANKING OF INTERVAL NUMBERS

In order to solve the chance constrained stochastic reliability optimization problem taking the component reliability as interval valued, we have proposed a simulation based genetic algorithm method. Ranking of interval numbers is of utmost need to compare the objective values at each step of the iterations while using genetic algorithm, because the objective function of the chosen optimization problem is interval valued.

The two arbitrary interval numbers $A=[a_L,a_R]$ and $B=[b_L,b_R]$ may be any of the following three types:

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

During the last few decades, a few researchers reported about the ranking of interval numbers in different ways. Recently, Mahato and Bhunia [18] presented the modified definitions of ranking with respect to optimistic and pessimistic decision makers' point of view for maximization and minimization problems separately. In 2012, Sahoo et al. [26] proposed the simplified definition of interval order relations ignoring optimistic and pessimistic decisions. It is to be mentioned that both the definitions by Mahato and Bhunia [18] and Sahoo et al. [26] report the equal outcome.

Interval ranking for maximization problem: Let $A=[a_L,a_R]=\langle a_c,a_w\rangle$ and $B=[b_L,b_R]=\langle b_c,b_w\rangle$ be two intervals. Then if

$$A>_{\max} B \Leftrightarrow \begin{cases} a_c > b_c & \text{for Type 1 and Type 2 intervals} \\ \text{either } a_c \ge b_c \wedge a_w < b_w \text{ or } a_c \ge b_c \wedge a_R > b_R & \text{for Type 3 intervals,} \end{cases}$$

the interval A is accepted for maximization problems. The order relation "$>_{\max}$" is reflexive, transitive but not symmetric.

Interval ranking for minimization problem: Let $A=[a_L,a_R]=\langle a_c,a_w\rangle$ and $B=[b_L,b_R]=\langle b_c,b_w\rangle$ be two intervals. Then if

$$A<_{\min} B \Leftrightarrow \begin{cases} a_c < b_c & \text{for Type 1 and Type 2 intervals} \\ \text{either } a_c \le b_c \wedge a_w \le b_w \text{ or } a_c \le b_c \wedge a_L < b_L & \text{for Type 3 intervals,} \end{cases}$$

the interval A is accepted for minimization problems. The order relation "$<_{\min}$" is reflexive, transitive but not symmetric.

5. UNIFORM DISTRIBUTION

A continuous random variable X over the interval [a,b] is said to have uniform distribution if its probability density function $f(x)$ is given by

$$f(x)=\begin{cases} \dfrac{1}{b-a} & \text{if } x\in[a,b] \\ 0 & \text{otherwise.} \end{cases}$$

It is denoted by $X \sim U(a,b)$.

6. NORMAL DISTRIBUTION

A continuous random variable X is said to follow normal distribution with parameters *m* (mean) and σ^2 (variance), denoted as $X \sim N(m,\sigma^2)$, if its probability density function $f(x)$ is given by

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2\sigma^2}(x-m)^2\right];\quad -\infty<x,m<\infty,\ \sigma>0.$$

7. GENERATION OF RANDOM NUMBERS

Random numbers are very important tool in stochastic simulation. Thus, finding of random numbers is a crucial part of simulation technique. Jana and Biswal [11,12] reported some algorithms to find random numbers based on different probability distributions.

The sub function using C library for the generation of pseudo random numbers between 0 and RAND_MAX is

#include<stdlib.h>
int rand (void).

where the value of RAND_MAX is defined in <stdlib.h>. Hence, a uniformly distributed random number can be generated from the given interval [a, b] according to the following algorithm.

Algorithm for finding random numbers in case of uniform distribution in [a,b]

Step 1: $\lambda_1 = rand()$

Step 2: $\lambda \leftarrow \lambda_1 / \text{RAND\_MAX}$

Step 3: Return $a+\lambda(b-a)$.

This random number generator is denoted as *U(a,b).*

Algorithm for finding random numbers in case of normal distribution in $N(m,\sigma^2)$.

Based on normal distribution, a random number between $[m-\sigma, m+\sigma]$ can be generated according to the following algorithm:

Step 1: Generate λ_2 and λ_3 from $U(0,1)$.

Step 2: Compute $\lambda = \left[-2\log_e(\lambda_2)\right]^{\frac{1}{2}}\sin(2\pi\lambda_3)$.

Step 3: Return $(m+\sigma\lambda)$.

This random number generator is denoted as $N(m,\sigma^2)$.

8. PROBLEM FORMULATION

Consider the n-stage parallel-series system as shown in Fig-1. This system is consisted of n subsystems connected in series, where j-th subsystem consists of x_j number of identical components connected in parallel. Assuming the reliability of each component as precise (fixed), we get the system reliability $R_S(x)$ as

$$R_S(x)=\prod_{j=1}^{n}\left[1-(1-r_j)^{x_j}\right].$$

The objective is to maximize the overall system reliability subject to several resource constraints. Sometimes, the constraints are satisfied depending on chance, which are called the chance constraints. In this case, each constraint is taken to be an event of a random experiment.

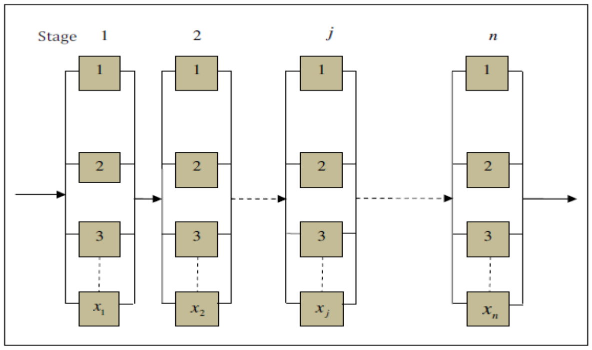

Fig-1: The n-stage parallel series system

Then, the chance constrained reliability-redundancy allocation problem for this parallel-series system with m constraints can be formulated as follows:

$$\text{Maximize } R_S(x) = \prod_{j=1}^{n}\left[1-(1-r_j)^{x_j}\right] \qquad (1)$$

subject to

$$P\left(g_i(x) \le b_i\right) \ge 1-\gamma_i, \quad i=1,2,...,m$$

$$\text{and } \; l_j \le x_j \le u_j, \quad j=1,2,...,n.$$

In problem (1), it is to be noted that all the parameters are assumed to be precise.

Now, we want to formulate the problem corresponding to the same system having the interval valued reliabilities of the components and the random variates for the parameters involved in the left side of the constraints. Also, the available resources are stochastic in nature. Then the corresponding problem can be formulated as follows:

$$\text{Maximize } \bar{R}_S(x) = \left[R_{SL}(x), R_{SR}(x)\right] \qquad (2)$$

subject to

$$P\left(\tilde{g}_i(x) \le \tilde{b}_i\right) \ge 1-\gamma_i, \quad i=1,2,...,m$$

$$\text{and } \; l_j \le x_j \le u_j, \quad j=1,2,...,n.$$

where $\bar{r}_j = \left[r_{jL}, r_{jR}\right]$, $R_{SL}(x) = 1-(1-r_{jR})^{x_j}$, $R_{SR}(x) = 1-(1-r_{jL})^{x_j}$, $j=1,2,...,n$ and

$\tilde{b}_i \sim N\left(m_{b_i}, \sigma^2{}_{b_i}\right), \quad i=1,2,...,m.$

Moreover, if the reliabilities of components in problem (1) are stochastic in nature and follow normal distribution, then the chance constrained stochastic reliability optimization problem becomes

$$\text{Maximize } \tilde{R}_S(x) = \prod_{j=1}^{n}\left[1-(1-\tilde{r}_j)^{x_j}\right] \qquad (3)$$

subject to

$$P\left(\tilde{g}_i(x) \le \tilde{b}_i\right) \ge 1-\gamma_i, \quad i=1,2,...,m$$

$$\text{and } \; l_j \le x_j \le u_j, \quad j=1,2,...,n.$$

where $\tilde{r}_j \sim N\left(m_{r_j}, \sigma^2{}_{r_j}\right), \quad j=1,2,...,n.$ and $\tilde{b}_i \sim N\left(m_{b_i}, \sigma^2{}_{b_i}\right), \quad i=1,2,...,m.$

Our objective is to solve the problems (1), (2) and (3). All these problems are nonlinear all integer programming problems with chance constraints. By transforming all these problems into deterministic problems, the reduced problems can be solved by existing methods. However, in an alternative way, we can solve the same by stochastic simulation based genetic algorithm technique.

9. STOCHASTIC SIMULATION

In stochastic simulation for chance constrained optimization problem, at first the stochastic constraints are transformed into their respective deterministic equivalent forms to the given level of confidence. Let us consider the chance constraints as follows:

$$\mathrm{P}\left(g_i\left(x,r\right)\le b_i\right)\ge 1-\gamma_i,\ 0<\gamma_i<1,\ i=1,2,...,m$$

where $r=\left(r_1,r_2,....,r_n\right)$ is a n-dimensional continuous vector and each r_i has a known distribution.

Monte-Carlo simulation technique is used for estimating these chance constraints for a given x. Let us generate N independent vectors $r^{(s)}=\left(r_1^{(s)},r_2^{(s)},....,r_n^{(s)}\right),\ s=1,2,...,N$ from their probability distributions. Let $N_i'\left(i=1,2,...,m\right)$ be the number of occurrences when all the constraints $g_i\left(x,r^{(s)}\right)\le b_i (i=1,2,...,m)$ are satisfied. Then by the definition of probability we have

$$\frac{N_i'}{N}\ge 1-\gamma_i,\ \ i=1,2,...,m. \qquad (4)$$

A solution is said to be feasible, if the condition (4) is satisfied for all *i* (*i=1,2,3,…,m*).

The algorithm for calculating the value of N_i'/N from the given chance constraints is as follows:

Step-1: Initialize $N_i'=0\left(i=1,2,...,m\right)$.

Step-2: Generate random numbers according to the known distribution of the random variables R_i.

Step-3: Find all the values of $g_i\left(x,r\right),$ for all $i,\ i=1,2,...,m.$

Step-4: If $g_i\left(x,r^{(s)}\right)\le b_i$, ***then*** $N_i'=N_i'+1, i=1,2,...,m.$

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

Step-6: Find the ratio $N_i'/N, i=1,2,...,m.$

Step-7: Stop.

10. GENETIC ALGORITHM BASED CONSTRAINTS HANDLING APPROACH

The optimization problems mentioned in the equations (1), (2) and (3) are the constrained optimization problems. Numerous techniques [20] have been reported to handle the constraints in genetic algorithms for solving the optimization problems. Gupta et al. [6] and Bhunia et al. [1] solved the optimization problems by using Big-M penalty technique. In this method, the given constrained optimization problem is transformed into an unconstrained optimization problem by penalizing by a large positive number say, M and called this penalty as Big-M penalty. In this work, we have used the Big-M penalty technique.

The respective transformed problems of (1), (2) and (3) are as follows:

$$\text{Maximize } \hat{R}_S(x) \qquad (5)$$

where $\hat{R}_S(x)=\begin{cases} R_S & \text{if } x\in S \\ -M & \text{if } x\notin S \end{cases}$

and $S=\left\{x:\mathrm{P}\left(g_i\left(x\right)\le b_i\right)\ge 1-\gamma_i,\ \ i=1,2,...,m\right\}$ be the feasible space.

Maximize $\hat{\bar{R}}_S(x)$ (6)

where $\hat{\bar{R}}_S(x) = \begin{cases} \bar{R}_S & \text{if } x \in S \\ [-M, -M] & \text{if } x \notin S \end{cases}$

and $S = \left\{ x : \mathrm{P}\left(\tilde{g}_i(x) \le \tilde{b}_i \right) \ge 1 - \gamma_i, \; i = 1, 2, ..., m \right\}$ be the feasible space.

Maximize $\hat{\tilde{R}}_S(x)$ (7)

where $\hat{\tilde{R}}_S(x) = \begin{cases} \tilde{R}_S & \text{if } x \in S \\ -M & \text{if } x \notin S \end{cases}$

and $S = \left\{ x : \mathrm{P}\left(\tilde{g}_i(x) \le \tilde{b}_i \right) \ge 1 - \gamma_i, \; i = 1, 2, ..., m \right\}$ be the feasible space.

The problems given in equations (5), (6) and (7) are integer non-linear unconstrained optimization problems. For solving these problems, we have developed stochastic simulation based genetic algorithm (GA) with advanced operators for integer variables.

10.1 THE GENETIC ALGORITHM

The different steps of genetic algorithm [5,9] are given in the following algorithm:

Algorithm

Step-1: Initialize GA parameters (popsize, maxgen, pmute, pcross) and the bounds of each decision variables.
Step-2: Set iteration $= 0$.
Step-3: Initialize the population i.e. popsize number of chromosomes.
Step-4: Set iteration $= iteration + 1$.
Step-5: Check the constraints using stochastic simulation.
Step-6: Evaluate fitness function for each chromosome.
Step-7: Use tournament selection to select chromosomes having better fitness values.
Step-8: Apply crossover, mutation and elitist operators to update the chromosomes.
Step-9: If iteration<maxgen, go to Step-4; otherwise go to Step-10.
Step-10: Print the best chromosome along with the fitness value.
Step-11: Stop.

There are several GA parameters, viz. population size (*popsize*), maximum number of generation (*maxgen*), crossover rate i.e., the probability of crossover (*pcross*) and mutation rate i.e., the probability of mutation (*pmute*). There is no such hard and fast rule for selecting the population size for GA, how large it should be. The population size is problem dependent and will need to increase with the dimensions of the problem. Regarding the maximum number of generations, there is no clear indication for considering this value. It varies from problem to problem and depends upon the number of genes (variables) of a chromosome and prescribed as stopping/termination criteria to make sure that the solution has converged. From natural genetics, it is obvious that the rate of crossover is always greater than that of the rate of mutation. Generally, the crossover rate varies from 0.60 to 0.95 whereas the mutation rate varies from 0.05 to 0.20. Sometimes the mutation rate is considered as $1/n$ where n is the number of genes (variables) of the chromosome. At the beginning, GA needs the initialization of the population of solutions. If $x_1, x_2, ..., x_n$ $\left(x_j \text{ is integer, } j = 1, 2, ..., n \right)$ be the decision variables of the optimization problem to be solved, then each chromosome can be represented as $X_p = (x_1, x_2, ..., x_n)_p, \; p = 1, 2, ..., popsize$. Here the integer values of $x_j \left(j = 1, 2, ..., n \right)$ are initialized uniformly between l_j and $u_j \left(j = 1, 2, ..., n \right)$. There are several procedures for selecting a random number of integer types. In this work, we have used the following algorithm for selecting an integer number randomly. A random integer random number between a and b can be generated as either $x = a + g$ or, $x = b - g$ where g is a random integer between 1 and $|a - b|$.

Since the constraints of the problem are chance constraints with some known degree of significance, stochastic simulation technique has been applied for checking the constraints.

Fitness function plays an important role in GA. This role is same for natural evolution process in the biological and physical environments. In our work, the value of objective function of the optimization problems corresponding to the chromosome is considered as the fitness value of that chromosome.

The selection operator which is the first operator in artificial genetics plays an interesting role in GA. This selection process is based on the well known Darwin's principle on natural evolution "survival of the fittest". The primary objective of this process is to select the above average individuals/chromosomes from the population according to the fitness value of each chromosome and eliminate the rest of the individuals/chromosomes. There are several methods for implementing the selection process. In this work, we have used the well known tournament selection with size two.

The exploration and exploitation of the solution space can be made possible by exchanging genetic information of the current chromosomes. After the selection process, other genetic operators, like crossover and mutation are applied to the resulting chromosomes those which have survived. Crossover is an operator that creates new individuals/chromosomes (offspring) by combining the features of both parent solutions. It operates on two or more parent solutions at a time and produces offspring for next generation. In this work, we have used intermediate crossover for integer variables.

The aim of mutation operator is to introduce the random variations into the population and is used to prevent the search process from converging to the local optima. This operator helps to regain the information lost in earlier generations and is responsible for fine tuning capabilities of the system and is applied to a single individual only. Usually, its rate is very low; because otherwise it would defeat the order building being generated through the selection and crossover operations. In this work we have used one-neighborhood mutation for integer variables.

11. NUMERICAL EXAMPLES

To illustrate the methodology and also to test the performance of the proposed algorithm, we have solved three different examples. In the first example, the values of component reliabilities are assumed to be precise whereas in second example, the component reliabilities are interval valued and the coefficients of the chance constraints and the available resources are normally distributed. In the third example, component reliabilities, coefficients of the chance constraints and available resources are normally distributed.

Example-1: A four stage system with simple chance constraints is considered as a pure stochastic integer programming problem using the data from Table- 2. The problem in this case is

$$\text{Maximize } R_S(x) = \prod_{j=1}^{4}\left[1-(1-r_j)^{x_j}\right]$$

subject to

$$P\left(\sum_{j=1}^{4} a_{ij}x_j \le b_i\right) \ge 1-\gamma_i, \quad i = 1, 2$$

$$\text{and } 1 \le x_j \le 10, \quad j = 1, 2, 3, 4.$$

The problem has been solved using the proposed algorithm and the solution is given by the redundancy vector $x = (5, 4, 5, 4)$ and the corresponding best found system reliability is $R_S(x) = 0.995946.$

Table- 2: Input data for Example 1

j	1	2	3	4	Available resource		γ_i
r_j	0.75	0.80	0.75	0.85			
a_{1j}	1.5	3.3	3.2	4.4	b_1	55	0.10
a_{2j}	4.0	5.0	7.0	9.0	b_2	125	0.15

Example-2: A four stage system with stochastic chance constraints is considered as a pure stochastic integer programming problem with interval valued component reliability using the data from Table-3. The problem in this case is

$$\text{Maximize } \bar{R}_S(x) = [R_{SL}(x), R_{SR}(x)]$$

subject to

$$P\left(\sum_{j=1}^{4} \tilde{a}_{ij} x_j \le \tilde{b}_i\right) \ge 1-\gamma_i, \quad i=1,2$$

where $\bar{r}_j = [r_{jL}, r_{jR}]$, $R_{SL}(x) = 1-(1-r_{jR})^{x_j}$, $R_{SR}(x) = 1-(1-r_{jL})^{x_j}$ and $1 \le x_j \le 10, \quad j=1,2,3,4.$

Table-3: Input data for Example 2

j	1	2	3	4	Available resource		γ_i
$\bar{r}_j$	[0.50, 0.99]	[0.50, 0.99]	[0.50, 0.99]	[0.50, 0.99]			
a_{1j}	N(1.5, 0.012)	N(3.3, 0.052)	N(3.2, 0.022)	N(4.4, 0.012)	b_1	N(55, 22)	0.10
a_{2j}	N(4.0, 0.032)	N(5.0, 0.042)	N(7.0, 0.032)	N(9.0, 0.022)	b_2	N(125, 32)	0.15

For different sets of values of γ and also for different sets of values of r, the problem of Example- 2 has been solved. The computational results have been shown in Tables-4 and 5.

Table-4: Results for different sets of values of $\gamma = (\gamma_1, \gamma_2)$ in Example-2

$\gamma = (\gamma_1, \gamma_2)$	$x = (x_1, x_2, x_3, x_4)$	obj=[objL, objR]	Centre value	Computational time (in sec.)
(0.10,0.15)	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416	0.52
(0.15,0.15)	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416	0.52
(0.20,0.15)	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416	0.52
(0.25,0.15)	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416	0.52
(0.25,0.20)	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416	0.52
(0.25,0.25)	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416	0.52

Table-5: Results for different sets of values of $r = (r_1, r_2, r_3, r_4)$ in Example-2

r_1	r_2	r_3	r_4	$x = (x_1, x_2, x_3, x_4)$	obj=[objL, objR]	Centre value
[0.75, 0.75]	[0.80, 0.80]	[0.75, 0.75]	[0.85, 0.85]	(5,5,5,3)	[0.994361, 0.994361]	0.994361
[0.74, 0.76]	[0.78, 0.81]	[0.73, 0.78]	[0.83, 0.86]	(5,4,5,4)	[0.994211, 0.997004]	0.995608
[0.70, 0.80]	[0.75, 0.85]	[0.70, 0.80]	[0.80, 0.90]	(5,4,5,4)	[0.989673, 0.998754]	0.994213
[0.70, 0.90]	[0.70, 0.90]	[0.70, 0.90]	[0.70, 0.90]	(5,4,5,4)	[0.979090, 0.999780]	0.989435
[0.65, 0.99]	[0.65, 0.99]	[0.65, 0.99]	[0.65, 0.99]	(5,4,5,4); (5,5,4,4)	[0.960048, 0.999999]	0.980024
[0.70, 0.99]	[0.70, 0.99]	[0.70, 0.99]	[0.70, 0.99]	(5,4,5,4)	[0.987387, 0.999999]	0.993693
[0.50, 0.99]	[0.50, 0.99]	[0.50, 0.99]	[0.50, 0.99]	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416

Table-6: Sensitivity results w.r.t. popsize in Example-2

popsize	$x = (x_1, x_2, x_3, x_4)$	obj=[objL, objR]	Centre value
10	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
20	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
30	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
40	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
50	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
60	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
70	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
80	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
90	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416
100	(5,4,5,4); (5,5,4,4)	[0.824833, 0.999999]	0.912416

Example-3: A four stage system with stochastic chance constraints is considered as a pure stochastic integer programming problem with stochastic reliability components using the data from Table 7. The problem in this case becomes

$$\text{Maximize } \tilde{R}_S(x) = \prod_{j=1}^{4}\left[1-(1-\tilde{r}_j)^{x_j}\right]$$

subject to

$$\mathrm{P}\left(\sum_{j=1}^{4}\tilde{a}_{ij}x_j \le \tilde{b}_i\right) \ge 1-\gamma_i,\quad i=1,2.$$

where $\tilde{r}_j \sim N\left(m_{r_j}, \sigma^2{}_{r_j}\right)$, $1 \le x_j \le 10,\ j = 1,2,...,4$ and $\tilde{b}_i \sim N\left(m_{b_i}, \sigma^2{}_{b_i}\right),\ \ i = 1,2.$

The computational results of this example are presented in the Table-8. From Table-8, we can see that the results are given for twenty independent runs for different random reliabilities components. It can be observed that the best found result corresponds to the 9th run. The best found system reliability is **0.999925** which corresponds for the reliability vector **(0.85,0.80,0.68,0.85)** and the redundancy vector **(6,4,5,3).**

Table-7: Input data for Example-3

j	1	2	3	4	Available resource		γ_i
$\tilde{r}_j$	N(0.75, 0.012)	N(0.80, 0.022)	N(0.75, 0.012)	N(0.85, 0.022)			
a_{1j}	N(1.5, 0.012)	N(3.3, 0.052)	N(3.2, 0.022)	N(4.4, 0.012)	b_1	N(55, 22)	0.10
a_{2j}	N(4.0, 0.032)	N(5.0, 0.042)	N(7.0, 0.032)	N(9.0, 0.022)	b_2	N(125, 32)	0.15

Table-8: Results for stochastic parametric values of $r = (r_1, r_2, r_3, r_4)$ in Example-3

Run	$r = (r_1, r_2, r_3, r_4)$	$x = (x_1, x_2, x_3, x_4)$	R
1	(0.70,0.72,0.78,0.77)	(5,4,5,3)	0.999385
2	(0.75,0.80,0.66,0.86)	(5,4,5,3)	0.998852
3	(0.75,0.80,0.75,0.85)	(6,4,5,3)	0.999332
4	(0.80,0.71,0.70,0.85)	(6,4,5,3)	0.999714
5	(0.83,0.90,0.75,0.82)	(6,4,5,3)	0.999543
6	(0.74,0.87,0.80,0.74)	(6,4,5,3)	0.999598
7	(0.75,0.76,0.75,1.00)	(6,4,5,3)	0.999318
8	(0.75,0.80,0.75,0.85)	(6,4,5,3)	0.999344
9	**(0.85,0.80,0.68,0.85)**	**(6,4,5,3)**	**0.999925**
10	(0.79,0.80,0.75,0.85)	(6,4,5,3)	0.999817
11	(0.68,0.82,0.75,0.80)	(6,4,5,3)	0.999738
12	(0.72,0.80,0.62,0.89)	(6,4,4,3)	0.999430
13	(0.75,0.80,0.68,0.83)	(6,4,5,3)	0.999541
14	(0.70,0.80,0.75,0.72)	(6,4,5,3)	0.999633
15	(0.71,0.77,0.74,0.85)	(6,4,4,4)	0.999384
16	(0.74,0.80,0.75,0.98)	(6,4,5,3)	0.999422
17	(0.71,0.94,0.75,0.80)	(6,4,5,3)	0.999561
18	(0.75,0.80,0.79,0.85)	(6,4,5,3)	0.999461
19	(0.73,0.91,0.79,0.85)	(6,4,5,3)	0.999399
20	(0.75,0.86,0.77,0.75)	(6,4,5,3)	0.999555

We have used real coded genetic algorithm to solve the problems under consideration. In this algorithm, we have used tournament selection, intermediate crossover and one neighborhood mutation as genetic operators. For this purpose, we have prepared the code for this algorithm in C Programming language. The corresponding computational work has been done on a PC with Intel Core-2 duo processor in LINUX environment. For each problem, twenty independent runs have been performed to determine the best found system reliability which is very near the optimal value of system reliability. In this computation, the values of genetic parameters are taken as *popsize*=100, *maxgen*=50, *pmute*=0.85 and *pcross*=0.15 respectively.

12. CONCLUSIONS

For the first time, we have proposed simulation based genetic algorithm for solving Chance Constrained redundancy allocation problem considering imprecise component reliabilities. To represent this impreciseness, we have used the stochastic and the interval approaches. For converting the chance constraints to its deterministic equivalent form, Monte-Carlo simulation technique is applied. Then the transformed problem has been converted into unconstrained optimization problem with the help of Big-M penalty technique. To solve the transformed problem we have developed real coded genetic algorithm with tournament selection, intermediate crossover and one-neighbourhood mutation. In tournament selection process we have used the definitions of interval ranking. For solving the optimization problem, we have used the GA based Big-M penalty approach. In this approach, the value of fitness function is not computed for infeasible solution. For infeasible solution, the value of M may be taken depending on the fitness function value. A small value (in case of maximization problem) or a large value (in case of minimization problem) may be considered for M to solve the constrained optimization problem. For further research one may use the proposed methodology and simulation based genetic algorithm for solving optimization problems which are mostly arising in the areas of engineering disciplines and management sciences.

ACKNOWLEDGEMENT

The author would like to acknowledge the University Grants Commission for the financial support through the minor research project with the sanction order number **F -PSW-006/12-13(ERO) DATED 05/02/2013** to carry out this work. The author also likes to acknowledge the help and cooperation of Dr. A. K. Bhunia, Professor, Department of Mathematics, Burdwan University and Dr. L.N. Sahoo, Assistant Professor, Department of Mathematics, Raniganj Girls' College.

REFERENCES

[1]. Bhunia AK, Sahoo L, Roy D (2010) Reliability stochastic optimization for a series system with interval component reliability via genetic algorithm, Applied Mathematics and Computations, 216: 929-939.
[2]. Bhunia AK, Sahoo L (2011) Genetic algorithm based reliability optimization in interval environment, N. Nedjah (Eds.), Innovative Computing Methods and Their Applications to Engineering Problems, SCI 357, pp. 13-36, Springer-Verlog Berlin Heidelberg.
[3]. Charnes A, Cooper WW, Symonds GH (1958) Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. Management Sciences, 4: 235-263.
[4]. Chern MS (1992) On the computational complexity of reliability redundancy allocation in a series system, Operations Research Letters, 11(5): 309-315.
[5]. Goldberg DE (1989) Genetic Algorithms in Search, Optimization, and Machine Learning, Addison- Wesley, Longman Publishing Co., Boston
[6]. Gupta RK, Bhunia AK, Roy D (2009) A GA based penalty function technique for solving constrained redundancy allocation problem of series system with interval valued reliabilities of components, Journal of Computational and Applied Mathematics, 232(2): 275-284
[7]. Hansen E, Walster GW (2004) Global optimization using interval analysis, Marcel Dekker Inc., New York.
[8]. Hikita M Y, Nakagawa K, Nakashima, Narihisa H (1992) Reliability optimization of systems by a surrogate-constraints algorithm, IEEE Transactions on Reliability, 41(3): 473-480
[9]. Holland JH (1975) Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI
[10]. Iwamura K, Liu B (1996) A genetic algorithm for chance constraint programming. Journal of Information and Optimization Sciences, 17(2): 409-422
[11]. Jana RK, Biswal MP (2004a) Stochastic simulation based genetic algorithm for chance constraint programming problems with continuous random variables. International Journal of Computer Mathematics, 81(9): 1069-1076
[12]. Jana RK, Biswal MP (2004b) Stochastic simulation based genetic algorithm for chance constraint programming problems with continuous random variables. International Journal of Computer Mathematics, 81(12): 1455-1463
[13]. Kall P, Wallace SW (1994) Stochastic programming. John Wiley & Sons, Chichester
[14]. Karmakar S, Mahato S, Bhunia, AK (2009) Interval oriented multi-section techniques for global optimization", Journal of computational and Applied Mathematics, 224: 476-491
[15]. Kuo W, Lin H, Xu Z, Zhang W(1987) Reliability optimization with the Lagrange-multiplier and Branch-and-Bound technique, IEEE Transactions on Reliability, 36(5): 624-630
[16]. Kuo W, Prasad VR (2000) An annotated overview of system reliability optimization, IEEE Transactions on Reliability, 49(2): 487-493

[17]. Kuo, W., V.R. Prasad, F.A. Tillman and C.L. Hwang, Optimal Reliability Design: Fundamentals and Applications, Cambridge University Press, 2001.
[18]. Mahato SK, Bhunia AK (2006) Interval-arithmetic-oriented interval computing technique for global optimization, Applied Mathematics Research Express, 01-18
[19]. Mahato SK, Sahoo L, Bhunia AK (2012) Reliability-redundancy optimization problem with interval valued reliabilities of components via genetic algorithm, Journal of Information and Computing Science (UK), 7(4): 284-295
[20]. Miettinen K, Mäkelä MM, Toivanen J (2003) Numerical comparison of some penalty-based constraint handling techniques in genetic algorithms, Journal of Global Optimization, 27: 427-446
[21]. Miller LB, Wagner H (1965) Chance-constrained programming with joint constraints. Operations Research, 13: 930-945
[22]. Misra B, Sharma U (1991) An efficient algorithm to solve integer-programming problems arising in system-reliability design, IEEE Transactions on Reliability, 40(1): 81-91
[23]. Nakagawa Y, Miyazaki S (1981) Surrogate constraints algorithm for reliability optimization problems with two constraints, IEEE Transactions on Reliability, 30(2): 181-184
[24]. Ravi V, Murty BSN, Reddy PJ (1997) Nonequilibrium simulated-annealing algorithm applied to reliability optimization of complex systems, IEEE Transactions on Reliability, 46(2): 233-239
[25]. Rubinstein RY (19981) Simulation and Monte Carlo method. John Wiley and Sons, New York
[26]. Sahoo L, Bhunia AK, Kapur PK (2012a) Genetic algorithm based multi-objective reliability optimization in interval environment, Computer and Industrial Engineering, 62(1): 152-160
[27]. Sahoo L, Bhunia AK, Roy D (2010) A genetic algorithm based reliability redundancy optimization for interval valued reliabilities of components, Journal of Applied Quantitative Methods, 5(2): 270-287
[28]. Sahoo L, Bhunia AK, Roy D (2012b) An application of genetic algorithm in solving reliability optimization problem under interval component Weibull parameters, Mexican Journal of Operations Research, 1(1): 2-19
[29]. Sahoo L, Bhunia AK, Roy D (2013b) Reliability optimization in Stochastic Domain via Genetic Algorithm , International Journal of Quality & Reliability Management (Accepted)
[30]. Sahoo L, Mahato SK, Bhunia AK (2013a) Optimization of System Reliability for Series System with Fuzzy Component Reliabilities by Genetic Algorithm, Journal of Uncertain System 8(2), 136-148, 2014
[31]. Sheikhalishahi MV, Ebrahimipour H, Zaman, Jeihoonian M (2013) A hybrid GA-PSO approach for reliability optimization in redundancy allocation problem, International Journal of Advanced Manufacturing Technology, 68, 317-338.
[32]. Sun X, Li D (2002) Optimal condition and Branch and Bound algorithm for constrained redundancy optimization in series system, Optimization and Engineering, 3: 53-65
[33]. Sung CS, Cho YK (1999) Branch and Bound redundancy optimization for a series system with multiple-choice constraints, IEEE Transactions on Reliability, 48 (2): 108-117

BIOGRAPHY

Dr. Sanat Kumar Mahato is an Assistant Professor of Mathematics, Mejia Govt. College, West Bengal, India. He has awarded the Ph. D. degree from The University of Burdwan in August, 2014. Dr. Mahato has published seven research papers in different peer reviewed international journals. His area of research work includes Application of Genetic Algorithm in inventory, in reliability optimization, interval analysis and its application, particle swarm optimization etc.