ISSN 1746-7659, England, UK
Journal of Information and Computing Science
Vol. 4, No. 3, 2009, *pp*. 163-172

Reliability Optimization of Complex Systems through C-SOMGA

Kusum Deep [1, +] and Dipti [2]

[1, 2] Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee – 247667 Uttarakhand, India

(*Received June 12, 2008, accepted September 29, 2008*)

Abstract. In this paper, three problems from the field of reliability engineering are considered. The first problem is a nonlinear constraint optimization problem. The problem is to determine the minimum cost of a life support system in a space capsule subject to the constraints on reliability of the system. The objective is to find the minimum cost of the system as well as maximum reliability. The second problem is to determine the minimum cost of a complex bridge network system with constraints on system reliability. The objective is to minimize the cost and maximize reliability at the same time. The third problem is a discrete optimization problem. This problem is to determine the optimal number of redundancies in a multistage mixed system so that the reliability of the system can be maximized. Three cases of this problem are considered. All the problems are solved using Self Organizing Migrating Genetic Algorithm (C-SOMGA) which is a recently published algorithm for obtaining the global optimal solution of constrained optimization problems. C-SOMGA is a hybridized genetic algorithm inspired by the features of Self Organizing Migrating Algorithm (SOMA) as well as Simple binary GA. The results obtained by C-SOMGA are compared with the existing published results in order to exhibit the roboutness of C-SOMGA for solving reliability engineering problems.

Keywords: Genetic algorithms, Self Organizing Migrating Algorithm, Self Organizing Migrating Genetic algorithm, Reliability Optimization.

1. Introduction

System reliability plays a very important role in real world applications. Various kinds of complex and multistaged systems are studied in literature. The reliability of a system can be improved in two ways (1) by optimizing reliability of the system components and (2) by choosing optimal redundant components in various subsystems. In the first method the chances of improving the reliability are less inspite of using the currently available reliable components. In the second method chances are more but as we optimize the redundant components, the cost, weight and volume will increase as well. These are very common issues while solving reliability design problems.

These kinds of reliability optimization problems have been attempted in literature. For example Misra [1] solved reliability optimization problem through sequential simplex search. Beraha and Misra [2] used random search algorithms to solve this kind of problems. Mohan and Shanker[3] solved a complex bridge network problem of reliability optimization using their Random Search Technique. Rao and Dhingra [4], use fuzzy multiobjective optimization approach to determine the reliability and redundancy appointment. Ravi et al [5] applied Nonequilibrium Simulated Annealing-Algorithm. Again Ravi et al [6] solved this problem in fuzzy environment. Chen [7] used a penalty guided artificial immune algorithm to solve mixed integer reliability design problem. A number of research papers are available for solving the reliability optimization problem using evolutionary algorithms for instance see Salazar et al [8], Salazar and Rocco [9] and Moghaddam et al. [10] etc. In some papers these problems are solved as a multicriterion problem, for instance Salazar et al [8], Salazar and Rocco [9]. Zhao et al [11], solved this problem as a multiobjective reliability optimization problem by using Ant Colony approach.

The objective of this paper is to solve three common types of reliability optimization problems. The first problem is a nonlinear constraint optimization problem. The problem is to determine the minimum cost of a life support system in a space capsule subject to the constraints on reliability of the system. The objective is

+ Corresponding author. *Email address:* kusumfma@iitr.ernet.in, diptipma@rediffmail.com

to find the minimum cost of the system as well as maximum reliability. The second problem is to determine the minimum cost of a complex bridge network system with constraints on system reliability. The objective is to minimize the cost and maximize reliability at the same time. The third problem is a discrete optimization problem. This problem is to determine the optimal number of redundancies in a multistage mixed system so that the reliability of the system can be maximized. Three cases of this problem are considered. The methodology to solve these problems is Self Organizing Migrating Genetic Algorithm (C-SOMGA) which is a recently published algorithm for obtaining the global optimal solution of constrained optimization problems. C-SOMGA is a hybridized genetic algorithm inspired by the features of Self Organizing Migrating Algorithm (SOMA) as well as Simple binary GA.

In section 2, the problem formulation for two types of complex systems and a multi-stage mixed system, is presented. In section 3, the C-SOMGA algorithm is discussed. In section 4, C-SOMGA is used to solve these three reliability optimization problems. Its results are discussed and compared with the previously quoted results. Finally in section 5, conclusions based on this study are drawn.

2. Problem Description

A complex system can consist of any logical configurations of stages, including combinations in series, parallel and bridge arrangements. In this paper three reliability optimization problems are considered. The description of these three problems is given below:

Problem 1: Life support system in a space capsule

This problem is a continuous nonlinear optimization problem. The source of this problem is Ravi et al [6]. The problem is to determine the minimum cost of a life support system in a space capsule subject to the constraints on reliability of the system. The objective is to find the minimum cost of the system as well as maximum reliability.

The schematic diagram is shown in Fig 1. The system has 4 components, each having component reliability R_i, i =1, 2, 3,…4. The problem formulation is as follows:

$$R_S = 1 - R_3\left(\overline{R_1}\cdot\overline{R_4}\right)^2 - \overline{R_3}.\left[1 - R_2\cdot\left(1 - \overline{R_1}\cdot\overline{R_4}\right)\right]^2 \tag{1}$$

Minimize C_S subject to:

$$R_{i,\min} \le R_i \le 1,\ i = 1,2,3,4; \tag{2}$$

$$R_{S,\min} \le R_S \le 1. \tag{3}$$

Where

C_S is the system cost

R_i is the reliability of the *i*th component

R_S is the reliability of the system

$R_{i,\min}$ (=0.5) is the lower bound on the reliability of *i*th component and

$R_{S,\min}$ is the lower bound on system respectively

The system cost C_S is:

$$C_S = 2K_1R_1^{\alpha_1} + 2K_2R_2^{\alpha_2} + K_3R_3^{\alpha_3} + 2K_4R_4^{\alpha_4}, \tag{4}$$

$$K_1 = 100,\ K_2 = 100,\ K_3 = 200,\ K_4 = 150,\ R_{S,\min} = 0.9,\ \alpha_i = 0.6,\ \textit{for all i.}$$

Problem 2: Complex Bridge-Network

Here the bridge network system is considered. The source of this problem is Mohan and Shanker (now Deep) [3]. This problem is to determine the minimum cost of a complex bridge network system with constraints on system reliability. The objective is to minimize the cost and maximize reliability at the same time. This problem is earlier solved by Ravi et. al [5] and Ravi et. al [6]. The schematic diagram is shown in Fig 2. The system has 5 components, each having component reliability R_i, i =1, 2, 3,…5. The reliability R_S of the system, which is probability of success of the system, is given by

$$R_S = R_1 \cdot R_4 + R_2 \cdot R_5 + R_2 \cdot R_3 \cdot R_4 + R_1 \cdot R_3 \cdot R_5 + 2R_1 \cdot R_2 \cdot R_3 \cdot R_4 \cdot R_5 - R_2 \cdot R_3 \cdot R_4 \cdot R_5 \\ - R_1 \cdot R_3 \cdot R_4 \cdot R_5 - R_1 \cdot R_2 \cdot R_4 \cdot R_5 - R_1 \cdot R_2 \cdot R_3 \cdot R_5 - R_1 \cdot R_2 \cdot R_3 \cdot R_4. \tag{5}$$

Thus in general R_S is a nonlinear function f (R_1, R_2, R_3, R_4, R_5) of the variables R_i, i=1, 2,…5. Following Misra [1], the cost C of the jth component is taken as

$$C_j = a_j \exp\left\{\frac{b_j}{1-R_j}\right\} \tag{6}$$

Thus the total cost of the system, which is to be minimized is

$$C_S = \sum_{j=1}^{5} C_j \tag{7}$$

The overall reliability R_S of the system is given by (7) and the overall cost C_S of the system is given by (9) have to be respectively maximized and minimized, keeping in view the following restrictions on the reliability component.

$$0 \le R_j \le 1, \; j=1, 2, 3,\dots, 5. \tag{8}$$

Theoretically speaking the maximum reliability of the system is 1, which is obtained for each $R_j = 1$. However, from (8) it is clear that the value of $C_j \to \infty$ as $R_j \to 1$. It means that ordinarily in achieving reliability close to unity the cost of the system will increase considerably. Although it is a multiobjective optimization problem, it is used here as a single objective problem as follows:

$$\text{Minimize } \; C_S = \sum_{i=1}^{5} a_i \cdot \exp\left(\frac{b_i}{R_i}\right), \tag{9}$$

Subject to

$$0 \le R_i \le 1, \; i = 1,2.....5, \tag{10}$$

$$0.99 \le R_S \le 1, \tag{11}$$

$$a_i = 1, b_i = 0.0003, \; for\, all\, i.$$

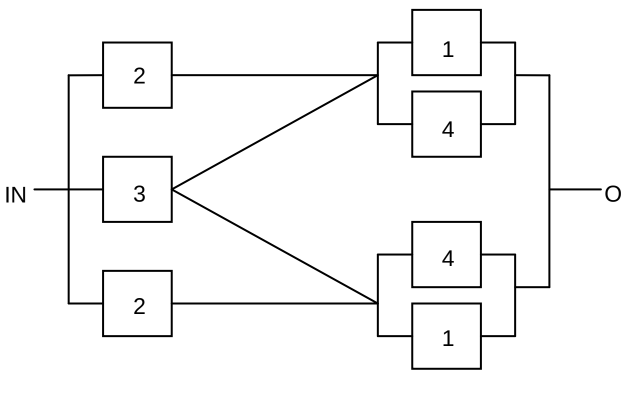

Fig. 1: A Complex system.

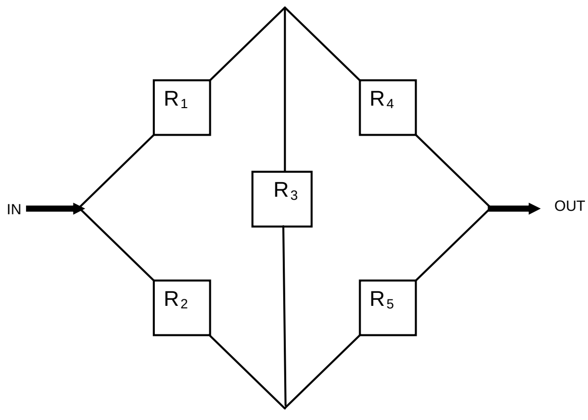

Fig. 2: Complex Bridge Network system.

Problem 3: n-Stage Mixed System

The source of this n-stage mixed system problem is Ravi et al [6]. This problem is to determine the optimal number of redundancies in a multistage mixed system so that the reliability of the system can be maximized. It is an integer nonlinear programming problem. The schematic diagram is shown in Fig 3. Three cases of this problem are considered here.

Case (i): when n=4.

Find the optimal x_j, $j = 1,2,3,4,$ which maximize:

$$R_S = \prod_{j=1}^{4} \left(1 - \overline{R_j}^{\,x_j}\right), \tag{12}$$

Subject to:

$$g_1 = \sum_{j=1}^{4} C_j \cdot x_j \le 56, \qquad g_2 = \sum_{j=1}^{4} W_j \cdot x_j \le 120; \tag{13}$$

The values of constants C_j and W_j, $j = 1,2,3,4.$ are given in Table 1.

Case (ii): when n=5.

Find the optimal x_j, $j = 1,2,3,4,5,$ to maximize:

$$R_S = \prod_{j=1}^{5} \left(1 - \overline{R_j}^{\,x_j}\right), \tag{14}$$

Subject to:

$$g_1 = \sum_{j=1}^{5} P_j \cdot x_j^2 \le 110, \quad g_2 = \sum_{j=1}^{5} C_j \cdot \left[x_j + \exp\left(\frac{x_j}{4}\right)\right] \le 175, \quad g_3 = \sum_{j=1}^{5} W_j \cdot \left[x_j \cdot \exp\left(\frac{x_j}{4}\right)\right] \le 200; \tag{15}$$

The values of constants P_j, C_j and W_j, $j = 1,2,3,4,5.$ are given in Table 2.

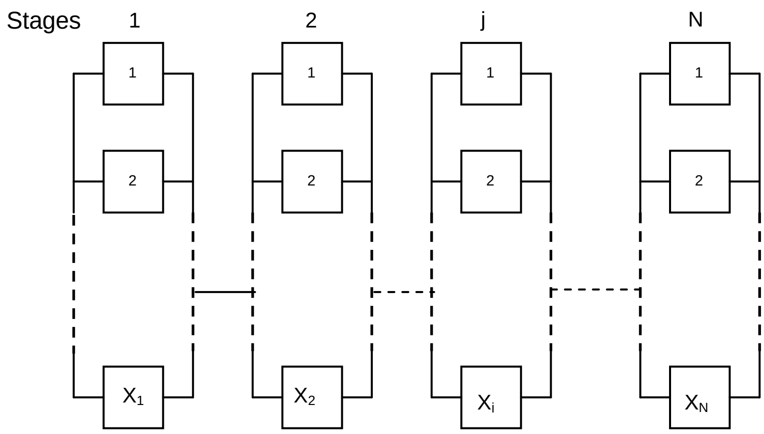

Fig. 3: Mixed series parallel system.

Table 1: Constants for problem 3 Case (i)

j	1	2	3	4
R_j	0.80	0.70	0.75	0.85
C_j	1.2	2.3	3.4	4.5
W_j	5	4	8	7

Table 2: Constants for problem 3 Case (ii)

j	1	2	3	4	5
Rj	0.80	0.85	0.90	0.65	0.75
Pj	1	2	3	4	2
Cj	7	7	5	9	4
Wj	7	8	8	6	9

Case (iii): when n=15.

Two models are considered for this case. Model 1 is taken from Ravi et al [5] and model 2 is taken from Ravi et al [6]. These are described below:

Model 1:

Find the optimal x_j, $j = 1,2,\ldots,15,$ which maximize:

$$R_S = \prod_{j=1}^{15}\left(1 - \overline{R_j}^{x_j}\right), \tag{16}$$

Subject to:

$$\begin{aligned} g_1 &= \sum_{j=1}^{15} C_j \cdot x_j \le 400, \\ g_2 &= \sum_{j=1}^{15} W_j \cdot x_j \le 414; \end{aligned} \tag{17}$$

The values of constant C_j and W_j, $j = 1,2,\ldots 15.$ are given in Table 3.

Model 2:

Find the optimal x_j, $j = 1,2,\ldots,15,$ which maximize:

$$R_S = \prod_{j=1}^{15}\left(1 - \overline{R_j}^{x_j}\right), \tag{18}$$

Subject to:

$$g_1 = \sum_{j=1}^{15} C_j \cdot x_j \le 400,$$

$$g_2 = \sum_{j=1}^{15} W_j \cdot x_j \ge 414; \tag{19}$$

The values of constants C_j and W_j, $j = 1,2,\ldots 15.$ are given in Table 3.

Table 3: Constants for problem 3 Case (iii)

j	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R_J	0.90	0.75	0.65	0.80	0.85	0.93	0.78	0.66	0.78	0.91	0.79	0.77	0.67	0.79	0.67
C_J	5	4	9	7	7	5	6	9	4	5	6	7	9	8	6
W_J	8	9	6	7	8	8	9	6	7	8	9	7	6	5	7

3. C-SOMGA for optimization

Deep and Dipti [12] present a hybridized genetic algorithm, named SOMGA, for nonlinear unconstrained optimization problems of the type:

$$\text{Min } f(X) \qquad \text{where } X = (x_1, x_2, \ldots\ldots x_n) \tag{20}$$

SOMGA is a hybridization of Self Organizing Migration Algorithm (SOMA) of Zelinka and Lampinen [13] and the simple Binary Genetic Algorithm (GA). SOMA is a not-so-well-known algorithm which surfaced in 2000. The main features that motivate us to incorporate SOMA into GA are that SOMA works with very low population size and has more exploration capabilities than other low population based approaches. Hybridizing SOMA with simple GA is expected to increase the reliability, efficiency and robustness of the search algorithm.

In order to handle constraints of the type:

$$\text{subject to} \quad g_j(X) \geq 0 \text{ for } j=1, 2, \ldots m \tag{21}$$

the following constraint handling mechanism is incorporated into SOMGA:

The well known tournament selection operator is employed between two chromosomes in the following way, for the entire population in a particular generation:

- If both chromosomes are not in the feasible region, then the one which is closer to the feasible region is carried over to the next generation. The values of the objective function are not calculated for either chromosomes.
- If one chromosome is in the feasible region and the other one is out of the feasible region, then the one which is in the feasible region is carried over to the next generation. The values of the objective function are not calculated for either chromosome.
- If both chromosomes are in the feasible region, then the values of the objective function are calculated for both chromosomes and the one which has a better value of the objective function (higher fitness value) is carried over to the next generation.

The new method such formed for solving constrained nonlinear optimization problems is named as C-SOMGA. Its comparison with GA, using the above mentioned constraint handling mechanism is performed and well established in Deep and Dipti [14].

The C-SOMGA is described as follows: First the chromosomes/individuals are generated randomly. These individuals compete with each other through the above constraint handling mechanism New individuals are created using single point crossover and bitwise mutation. The best individual among them is considered as leader and all others are considered as active members. For each active individual a new population of size N is created, where N=path length/step size. This population is nothing but the new positions of the active individual, proceeds in the direction of the leader in n steps of the defined length. The movement of an individual is given as follows:

$$\begin{aligned} &x_{i,j}^{MLnew} = x_{i,j,start}^{ML} + \left(x_{L,j}^{ML} - x_{i,j,start}^{ML}\right) tPRTVector_j \\ &where \quad t \in < 0, by\ Step\ to, PathLength >, \\ &ML\ is\ actual\ migration\ loop. \end{aligned} \tag{22}$$

Then sort this population according to the fitness value in decreasing order. Starting from the best one of the new population evaluates the constraint violation function described below:

$$\psi(x) = \sum_{m=1}^{M} \left[h_m(x)\right]^2 + \sum_{k=1}^{K} G_k \left[g_k(x)\right]^2 \tag{23}$$

If $\psi(x)=0$, replace the active individual with the current position and move to the next active individual. And If $\psi(x) > 0$ then move to the next best position of the sorted new population. In this way, all the active individuals are replaced by the new updated feasible position. If no feasible solution is available then active individual remains the same. At last, the best individuals (number equal to population size) from the previous and current generations are selected for the next generation. The computational steps of this approach are given below:

Step 1: Generate the initial population.

Step 2: Evaluate all individuals.

Step 3: Apply tournament selection for constrained optimization on all individuals to select the better individuals for the next generation.

Step 4: Apply crossover operator on all individuals with crossover probability P_c to produce new individuals.

Step 5: Evaluate the new individuals.

Step 6: Apply mutation operator on every bit of every individual of the population with mutation probability P_m.

Step 7: Evaluate the mutated individuals.

Step 8: Find leader (best fitted individual) of the population and consider all others as active individuals of the population.

Step 9: For each active individual a new population of size N is created. This population is nothing but the new positions of the active individual towards the leader in n steps of the defined length. The movement of this individual is given in equation (1).

Step 9.1: Sort new population with respect to fitness in decreasing order.

Step 9.2: For each individual in the sorted population, check feasibility criterion.

Step 9.3: If feasibility criterion is satisfied replace the active individual with the new position, else move to next position in sort order and go to step 9.2.

Step 10: Select the best individuals (in fitness) of previous and current generation for the next generation via tournament selection.

Step 11: If termination criterion is satisfied go to step 12 else go to step 3.

Step 12: Report the best chromosome as the final optimal solution.

4. Results and Discussion

The three problems described in section 2 are solved using C-SOMGA described in section 3. The experimental setup used to solve these problems is presented in Table 4. In all the problems, the population size is kept fixed to 20, other parameters are also same. 10 runs are performed for each problem and best one is reported. The stopping criterion used, is fixed number of function evaluations (100000).

Table 4: Experimental setup

Population size	20
P_c	0.65
P_m	0.002
Step size	0.31
Path length	3
String length	30

The results obtained for this problem using C-SOMGA are reported in the Table 5. These results are compared with the results given by Ravi et. al [5] using NESA and by Ravi et al [6] using fuzzy concepts. It can be seen in the Table 5 that the cost of the system obtained by C-SOMGA that is 641.824, is less than the cost obtained by NESA (Ravi et. al [5]) that is 641.824 and is similar to the cost obtained by fuzzy (Ravi et al [6]). In other words we can say that C-SOMGA is not expensive. On the other side the reliability of the system given by C-SOMGA i.e. 0.900001 is slightly superior to Ravi et. al [6] that is 0.9. Hence on the basis of these results we can say that C-SOMGA is giving superior results than that of other existing techniques.

Table 5: Optimal Solution of Problem **1**

	SOMGA	Ravi et.al.(1997) Using NESA	Ravi et.al.(2000) Ising FUZZY
R_1	0.50002256	0.50006	0.50000
R_2	0.83889956	0.83887	0.83892
R_3	0.55600000	0.50001	0.50000
R_4	0.50000000	0.50002	0.50000
R_S	0.90000100	0.90001	0.90000
C_S	641.82400000	641.83320	641.82400

The results of the Problem 2 obtained by C-SOMGA along with previously quoted results are presented in Table 6. It can be seen in Table 6 that the cost of the system obtained by C-SOMGA i.e. 5.01992 is lower than the cost obtained by Ravi et. al [5] i.e. 5.01993 and the cost obtained by Ravi et. al [6] i.e. 5.02042. But the reliability obtained by C-SOMGA i.e. 0.99 is lesser than the reliability obtained by Ravi et. al [6] i.e. 0.9905. Hence for this system C-SOMGA is less reliable than Fuzzy (Ravi et. al [6]).

Table 6: Optimal Solution of Problem 2

	SOMGA	Ravi et.al.(1997) Using NESA	Ravi et.al.(2000) Using FUZZY
R_1	0.935359	0.93747	0.93635
R_2	0.934304	0.93291	0.93869
R_3	0.790332	0.78485	0.80615
R_4	0.935504	0.93641	0.93512
R_5	0.934575	0.93342	0.93476
R_S	0.99	0.99000	0.9905
C_S	5.01992	5.01993	5.02042

To solve Problem 3 one modification is made in C-SOMGA, since this is a integer optimization problem After each step, the real variables are first converted to integers. Three cases of this problem are considered. Case (i), when the number of stages in multistage mixed system is 4. In case (ii), it is 5 and in the case (iii), it is increased to 15. As the stages are increased the complexity of the system will increase as well.

Table 7: Optimal solution of Problem 3 Case (i)

	SOMGA	Raviet.al.(1997) Using NESA
x_1	5	5
x_2	6	6
x_3	5	5
x_4	4	4
R_S	0.99750	0.99750
g_1	54.80000	54.80000
g_2	117.00000	117.00000
Q_1	0.00032	0.00032
Q_2	0.00073	0.00073
Q_3	0.00098	0.00098
Q_4	0.00051	0.00051

Table 8: The optimal solution for problem 3 Case (ii)

	SOMGA	Ravi et.al.(1997) Using NESA	Mohan & Shanker(1987)
x_1	3	3	3
x_2	2	2	2
x_3	2	2	2
x_4	3	3	3
x_5	3	3	3
R_S	0.90450	0.90450	0.90450
g_1	83	83	83
g_2	146.12500	146.12500	146.21000
g_3	192.48000	192.48000	192.47000
Q_1	0.00800	0.00800	0.00800
Q_2	0.02250	0.02250	0.02250
Q_3	0.01000	0.01000	0.01000
Q_4	0.04287	0.04287	0.04287
Q_5	0.01562	0.01562	0.01562

The results for case (i) and case (ii) obtained by C-SOMGA are presented in Table 7 and 8 respectively and compared with previously quoted results. Refer [5] and [3]. In both the cases the results obtained by C-SOMGA are similar to the previously quoted results. Thus the performance of C-SOMGA is similar to the performance of existing techniques for this system (case (i) and case (ii)).

In case (iii), two models are available in literature. One model is taken Ravi et. al [5] and the second model is taken from Ravi et. al [6]. The results obtained for both models of case (iii) using C-SOMGA along with previously quoted results are given in Table 9. In model 1 results obtained by C-SOMGA are same as results given by Ravi et. al [5]. In model 2, C-SOMGA achieves the maximum reliability of the system that is 0.9563. Also the solution quoted by Ravi et. al [6] is infeasible whereas C-SOMGA produced feasible solution with better reliability. Hence in this case C-SOMGA is better than Fuzzy.

Table 9: The optimal solution for problem 3 Case (iii)

x_i	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	R_S
C-SOMGA (Model 1)	3	4	5	3	3	2	4	5	4	3	3	4	5	5	5	0.9450
C-SOMGA (Model 2)	3	4	5	4	3	2	4	5	4	3	4	4	5	4	5	0.9563
Ravi et.al. (1997) NESA	3	4	5	3	3	2	4	5	4	3	3	4	5	5	5	0.9450
Ravi et.al.(2000) FUZZY	3	4	5	4	3	3	4	5	4	3	3	4	5	5	5	0.9552

On the basis of these results it is clear that C-SOMGA is able to solve reliability optimization problems and produce results better or comparable to the previously quoted results with less population size.

5. Conclusions

In this paper C-SOMGA has been used to optimize the reliability as well as cost of the three complex systems. These problems have been taken from the field of reliability engineering. In two problems, C-SOMGA obtains better solutions than the previously quoted results and in one problem results are comparable. The population size required to solve each problem is only 20. On the basis of results presented in this paper it is concluded that C-SOMGA has a physical utility in solving reliability optimization problem.

6. Acknowledgements

The second author would like to thank University Grant Commission< new Delhi, India for the fund under grant no.10-2(5)2003(I)-EU.II

7. References

[1] K.B. Misra. Reliability Optimization Through Sequential Simple Search. *Int. J. Control*.. 1973, **18**: 173-183.

[2] D. Beraha, and K.B. Misra. Reliability Optimization through Random Search Algorithms. *Microelectron Reliability*. 1974, **13**: 295-297.

[3] C. Mohan and Kusum Shanker (now Kusum Deep). Reliability Optimization of Complex Systems using Random Search Technique. *Microelectronics and Reliability.* 1988, **28**(4): 513-518.

[4] S.S. Rao, and A.K. Dhingra. Reliability and Redundancy Apportionment Using Crisp and Fuzzy Multi –Objective Optimization Approaches. Reliability. *Eng. Syst. Safety.* 1992, 37: 253-261.

[5] V. Ravi, B.S.N. Murty, and P.J. Reddy. Non Equilibrium Simulated Annealing-Algorithm Applied to Reliability Optimization of Complex Systems. *IEEE Trans. Rel.*. 1997, **46**: 233-239.

[6] V. Ravi, , P.J. Reddy, and H.J. Zimmermann. Fuzzy Global Optimization of Complex System Reliability. *IEEE Transactions on Fyzzy systems*. 2000, **8**(3): 241-248.

[7] T.C Chen. IAs Based Approach For Reliability Redundancy Allocation Problems. *Applied Mathematics and Computation.* 2006, **182**: 1556-1567

[8] D. Salazar, C.M Rocco, and B.J Galvin. Optimization of Constrained Multiple-Objective Reliability Problems Using Evolutionary Algorithms. *Reliability Engineering and System Safety*. 2006, **91**: 1057-1070.

[9] D. Salazar, and C.M Rocco. Solving Advanced Multi-Objective Robust Designs By Means Of Multiple Objective

Evolutionary Algorithms (MOEA): A Reliability Application. *Reliability Engineering and System Safety*. 2007, **92**: 697-706.

[10] R.T Moghaddam, J. Safari, and F. Sassani. Reliability Optimization Of Series-Parallel Systems With A Choice Of Redundancy Strategies Using A Genetic Algorithm. *Reliability Engineering and System Safety*. 2007, Article in Press.

[11] H.J. Zhao, L. Zhaoheng, and M.T Dao. Reliability Optimization Using Multi-Objective Ant Colony System Approaches. *Reliability Engineering and System Safety*. 2007, **92**: 109-120.

[12] Kusum Deep and Dipti. A New Hybrid Self Organizing Migrating Genetic Algorithm for Function Optimization. *Proc. of IEEE Congress on Evolutionary Computation, Singapore.* 2007, pp. 2796 - 2803.

[13] I. Zelinka and J. Lampinen. SOMA-Self Organizing Migrating Algorithm, Mendal. *6th international conference on soft computing.* Brno, Czech Republic. 2000, **80** (2): pp. 214.

[14] Kusum Deep and Dipti. A Self Organizing Migrating Genetic Algorithm for Constrained Optimization. *Applied Mathematics and Computation.* 2008, **198**: 237-250.