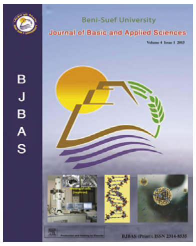
HOSTED BY
ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/bjbas



An approach for solving constrained reliability-redundancy allocation problems using cuckoo search algorithm

Harish Garg

Thapar University, School of Mathematics and Computer Applications, Patiala 147004, India

ARTICLE INFO

Article history:
Received 23 September 2014
Accepted 1 January 2015
Available online 5 March 2015

Keywords:
Redundancy allocation
Cuckoo algorithm
Reliability
Optimization

ABSTRACT

The main goal of the present paper is to present a penalty based cuckoo search (CS) algorithm to get the optimal solution of reliability – redundancy allocation problems (RRAP) with nonlinear resource constraints. The reliability – redundancy allocation problem involves the selection of components' reliability in each subsystem and the corresponding redundancy levels that produce maximum benefits subject to the system's cost, weight, volume and reliability constraints. Numerical results of five benchmark problems are reported and compared. It has been shown that the solutions by the proposed approach are all superior to the best solutions obtained by the typical approaches in the literature are shown to be statistically significant by means of unpaired pooled t-test.

1. Introduction

With the advance of technology and growing complexity of an industrial system, it has become imperative for all production systems to perform satisfactorily during their expected life span. However, failure is an unavoidable phenomenon associated with the technological advancement of the equipments used in all industries. Any unfortunate consequences of unreliable behavior of such equipments or systems have led to the desire for reliability analysis (Garg et al., 2013a, b). Therefore, in recent years system reliability becomes an important issue in evaluating the performance of an engineering system. The optimal reliability design aims to determine a system structure that achieves higher levels of reliability at the minimum cost to the manufacturer either by exchanging the existing components with more reliable components or/and using redundant components in parallel. In the former way, the system reliability can be improved to some degree, but the required reliability enhancement may never be attainable even though the highest available and reliable components are used. In the latter approach, system reliability can be enhanced by choosing the redundant components, but the cost, weight, volume etc. will be increased as well. Besides the above two ways, the system reliability can be

E-mail address: harishg58iitr@gmail.com.
Peer review under the responsibility of Beni-Suef University.
http://dx.doi.org/10.1016/j.bjbas.2015.02.003

enhanced through both reliability allocation and redundancy allocation. Such problem of allocation is known as *reliability-redundancy allocation problem* (RRAP) which aims is to maximize the system reliability under the several constraints like, cost, weight and volume (Kuo and Prasad, 2000; Hikita et al., 1992). In order to solve the reliability-redundancy allocation problem, several heuristic, global optimization as well as meta-heuristic methods exist in the literature which includes heuristic methods, Lagrangian multiplier method, branch and bound method, linear programming, and so on (Hikita et al., 1992; Kuo et al., 1978; Gopal et al., 1978; Dhingra, 1992; Hikita et al., 1978). But these approaches do not guarantee exact optimal solutions, but they achieve reasonably good solutions for hard problems in relatively short time periods.

However, the heuristic techniques require derivatives for all non-linear constraint functions, that are not derived easily because of the high computational complexity. To overcome this difficulty meta-heuristics have been selected and successfully applied to handle a number of reliability optimization problems. These heuristics include genetic algorithms (Yokota et al., 1996; Painton and Campbell, 1995), simulated annealing (Kim et al., 2006), particle swarm optimization (Yeh, 2009; Garg and Sharma, 2013; Coelho, 2009), artificial bee colony (Hsieh and Yeh, 2012; Yeh and Hsieh, 2011; Garg et al., 2013a,b), harmony search (Zou et al., 2010) etc. Yokota et al. (1996); Painton and Campbell (1995) and Hsieh et al. (1998) applied genetic algorithms (GA) to solve these mixed-integer reliability optimization problems. Coit and Smith (1996) combined GA and neural network (NN) to tackle the series-parallel redundancy problem. Chen (2006) applied the immune algorithm (IA) for solving the reliability–redundancy allocation problem. It can search over promising feasible and infeasible regions to find the feasible optimal/near optimal solution effectively and efficiently. Gen and Yun (2006) employed a soft computing approach for solving various reliability optimization problems. This method combined rough search techniques and local search techniques, which can prevent the premature convergence situation of its solution. Coelho (2009) proposed an efficient particle swarm optimization (PSO) algorithm based on Gaussian distribution and chaotic sequence (PSO − GC) to solve the reliability − redundancy optimization problems. Zou et al. (2010) proposed a novel global harmony search algorithm (NGHS) to solve reliability problems. The NGHS is an improved version of the harmony search algorithm Geem et al. (2001), and it is inspired by the swarm intelligence of the particle swarm optimization algorithm. Agarwal and Sharma (2010) presented an ant colony optimization algorithm to address the constrained redundancy allocation problem to maximize system reliability for complex binary systems. Yeh and Hsieh (2011) developed a penalty guided artificial bee colony algorithm (ABC) for solving the reliability optimization problems. In addition, they also proposed a local search to improve the solutions. Wang and Li (2012) proposed an effective coevolutionary differential evolution with harmony search algorithm for solving the reliabilityredundancy optimization problems by dividing the problem into a continuous part and an integer part. Wu et al. (2011) proposed an improved particle swarm optimization algorithm for solving the reliability problems. Hsieh and You (2011) proposed an immune based two-phase approach to solve the reliability-redundancy allocation problem. In the first phase, an immune based algorithm (IA) is developed to solve the allocation problem, and in the second phase a new procedure is presented to improve the solutions by the IA. Garg et al. (2014) solved the various multi-objective reliability optimization problem using intuitionistic fuzzy optimization technique in interval environment. The conflicting nature between the objective are handled with the help of defining their membership functions corresponding to each objective function and then corresponding problem has been solved with the particle swarm optimization. Zou et al. (2011) proposed global harmony search algorithm for solving bridge and overspeed protection system optimization problem. In it, they combine the harmony search algorithm with concepts from the particle swarm optimization to solve optimization problems. Valian et al. (2013) proposed an improved version of cuckoo search for solving the reliability optimization problems. Garg and Sharma (2012) presented a novel technique for analyzing the behavior of an industrial system after quantifying the uncertainties in the data. Various reliability parameters are computed are comparing their results with traditional and existing techniques and gave a recommendation for improving the performance of the system.

Recently, a cuckoo search (CS) has been a new meta-heuristic approach proposed by Yang and Deb (2009) in 2009. Recent studies show that CS is potentially far more efficient than PSO and GA (Rajabioun, 2011). Moreover the number of parameters in CS to be tuned is less than GA and PSO, and thus it is potentially more generic to adapt to a wider class of optimization problems. In the light of the advantages of CS technique, in the presented paper the five benchmark problems of reliability-redundancy allocation have been solved and it has been observed that the results of the new approach are all superior to the existing results in the literature. The rest of the paper is organized as follows: Section 2 describe the assumptions as well as notations that have been used in the entire paper. Section 3 deals with the benchmark problems of the reliability − redundancy allocation. In Section 4, the penalty based guided cuckoo search methodology is described. The final results by CS approach have been obtained and discussed in Section 5 while conclusions drawn are presented in Section 6.

2. Assumption and notations

Before introducing the reliability-redundancy allocation problem, we define the following assumptions and notations that have been used in the entire paper.

2.1. *Assumptions*

- If a component of any subsystem fails to function, the entire system will not be damaged or fail.
- All redundancy is active redundancy with out repair.
- Reliability, cost, weight and volume of each components in one subsystem are same.
- The state of components and system has only two states like operating state or failure state.

2.2. Notations

m	number of subsystems in the system.
M	number of constraints.
n_i	the number of components in subsystem i,$1 \leq i \leq m$.
n	$=(n_1,n_2,\ldots,n_m)$, the vector of redundancy allocation for the system.
r_i	reliability of each components in subsystem i,$1 \leq i \leq m$.
r	$=(r_1,r_2,\ldots,r_m)$, the vector of component reliabilities for the system.
g_j	the j^{th} constraint function, $j = 1,2,\ldots,M$.
w_i	the weight of each component in subsystem i,$1 \leq i \leq m$.
c_i	the cost of the each component in subsystem i,$1 \leq i \leq m$.
v_i	the volume of each component in subsystem i.
R_i	$= 1-(1-r_i)^{n_i}$ is the reliability of the ith subsystem $1 \leq i \leq m$.
Q_i	$1 - R_i$ is the unreliability of the ith subsystem.
$n_{i,max}$	maximum number of components in subsystem i,$1 \leq i \leq m$.
R_s	the system reliability.
C,W	the upper limit of the system's cost, weight respectively.
S	set of feasible solution.

3. Reliability-redundancy allocation problem

The general mathematical formulation of the reliability-redundancy allocation problem is:

$$\begin{array}{ll} \text{Maximize} & R_s(r_1,r_2,\ldots,r_m;n_1,n_2,\ldots,n_m) \\ \text{subject to} & g(r_1,r_2,\ldots,r_m;n_1,n_2,\ldots,n_m) \leq b \\ & 0 \leq r_i \leq 1\,;\, i=1,2,\cdots,m \\ & 1 \leq n_i \leq n_{i,max}; \quad n_i \in \mathbb{Z}^+, r_i \in [0,1] \subset \mathbb{R} \end{array}$$

where $g(\cdot)$ is the set of constraint functions usually associated with the system's weight, volume and cost; $R_s(\cdot)$ is the objective function for the overall system reliability; r_i and n_i are the reliability and the number of redundant components in the ith subsystem, respectively; m is the number of subsystems in the system and b is the vector of resource limitation. This problem is an NP problem and belongs to the category of constrained nonlinear mixed-integer optimization problems because the number of redundancy n_i are the positive integer values and the component reliability r_i are the real values between 0 and 1. The goal of the problem is to determine the number of components n_i and the components' reliability r_i in each subsystem so as to maximize the overall system reliability.

Five benchmark problems of the reliability – redundancy allocation have been taken. The first three examples with non-linear constraints used by authors (Hikita et al., 1992; Hsieh et al., 1998; Chen, 2006; Yeh and Hsieh, 2011; Garg et al., 2013a,b; Zou et al., 2010; Wu et al., 2011; Valian et al., 2013; Kim et al., 2006; Coelho, 2009) are a series system, series-parallel system and complex (bridge) system, respectively. The fourth example is an overspeed protection system, which was investigated by authors (Yokota et al., 1996; Chen, 2006; Coelho, 2009; Yeh and Hsieh, 2011; Dhingra, 1992) and last one is the 15 unit system reliability optimization problem (Agarwal and Sharma, 2010; Valian et al., 2013).

All the above problems are shown to maximize the system's reliability subject to multiple nonlinear constraints and can be stated as the mixed-integer nonlinear programming problems. For each problem both, the component reliabilities and redundancy allocations are to be decided simultaneously and are formulated as below.

Problem 1. Series System (Fig. 1(a)) (Hikita et al., 1992; Hsieh et al., 1998; Chen, 2006; Yeh and Hsieh, 2011; Garg et al., 2013a,b; Kim et al., 2006; Wu et al., 2011; Valian et al., 2013)

$$\text{Maximize} \quad R_s(r,n) = \prod_{i=1}^{5}\left[1-(1-r_i)^{n_i}\right]$$

$$\text{s.t.} \;\; g_1(r,n) = \sum_{i=1}^{5} v_i n_i^2 - V \leq 0 \tag{1}$$

$$g_2(r,n) = \sum_{i=1}^{5} \alpha_i(-1000/\ln r_i)^{\beta_i}[n_i + \exp(n_i/4)] - C \leq 0 \tag{2}$$

$$g_3(r,n) = \sum_{i=1}^{5} w_i n_i \exp(n_i/4) - W \leq 0 \tag{3}$$

$$0.5 \leq r_i \leq 1, \quad r_i \in [0,1] \subset \mathbb{R}^+, \quad 1 \leq n_i \leq 5, \quad n_i \in \mathbb{Z}^+ \quad ; \quad i = 1,2,\cdots,5$$

Problem 2. Series-parallel system (Fig. 1(b)) (Hikita et al., 1992; Hsieh et al., 1998; Chen, 2006; Yeh and Hsieh, 2011; Garg et al., 2013a,b; Kim et al., 2006; Wu et al., 2011; Valian et al., 2013)

$$\begin{array}{lll} \text{Maximize } R_s(r,n) & = 1-(1-R_1R_2)[1-(R_3+R_4-R_3R_4)R_5) \\ \text{s.t.} & g_1(r,n), g_2(r,n), g_3(r,n) \quad (\text{as specified by}(1),(2),(3)\text{respectively}) \\ & 0.5 \leq r_i \leq 1 \quad ; \quad r_i \in [0,1] \subset \mathbb{R}^+ \quad 1 \leq n_i \leq 5 \quad ; \quad n_i \in \mathbb{Z}^+ \quad i = 1,2,\cdots,5 \\ \text{where} & R_i = 1-(1-r_i)^{n_i} \end{array}$$

Problem 3. Complex(bridge) system (Fig. 1(c)) (Hikita et al., 1992; Hsieh et al., 1998; Chen, 2006; Coelho, 2009; Garg et al., 2013a,b; Yeh and Hsieh, 2011; Kim et al., 2006; Wu et al., 2011; Valian et al., 2013)

$$\begin{array}{ll} \text{Maximize } R_s(r,n) & = R_5(1-Q_1Q_3)(1-Q_2Q_4)+Q_5[1-(1-R_1R_2)(1-R_3R_4)] \\ \text{s.t.} & g_1(r,n), g_2(r,n), g_3(r,n) \quad \text{(as specified by (1), (2), (3) respectively)} \\ & 0.5 \le r_i \le 1 \quad ; \quad r_i \in [0,1] \subset \mathbb{R}^+, \quad 1 \le n_i \le 5 \quad ; \quad n_i \in \mathbb{Z}^+, \quad i = 1,2,\cdots,5 \\ \text{where} & Q_i = 1 - R_i = (1-r_i)^{n_i} \end{array}$$

Problem 4. Overspeed protection system (Fig. 1(d)) (Yokota et al., 1996; Dhingra, 1992; Chen, 2006; Coelho, 2009; Yeh and Hsieh, 2011; Garg et al., 2013a,b; Kim et al., 2006; Wu et al., 2011; Valian et al., 2013)

The fourth problem is considered for the reliability-redundancy allocation problem of the overspeed protection system for a gas turbine. Overspeed detection is continuously provided by the electrical and mechanical systems. When an overspeed occurs, it is necessary to cut off the fuel supply. For this purpose, 4 control valves (V1–V4) must close. The control system is modeled as a 4-stage series system. The objective is to determine an optimal level of r_i and n_i at each stage i such that the system reliability is maximized. This reliability problem is formulated as follows:

$$\text{Maximize } R_s(r,n) = \prod_{i=1}^{4}\{1-(1-r_i)^{n_i}\}$$

$$\text{s.t. } g_1(r,n) = \sum_{i=1}^{4} v_i n_i^2 - V \le 0$$

$$g_2(r,n) = \sum_{i=1}^{4} \alpha_i(-1000/\ln r_i)^{\beta_i}[n_i + \exp(n_i/4)] - C \le 0$$

$$g_3(r,n) = \sum_{i=1}^{4} w_i n_i \exp(n_i/4) - W \le 0$$

$$0.5 \le r_i \le 1 \quad ; \quad r_i \in [0,1] \subset \mathbb{R}^+, \quad 1 \le n_i \le 10 \quad ; \quad n_i \in \mathbb{Z}^+, \quad i = 1,2,\cdots,4$$

where v_i is the volume of the component at stage i, w_i is the weight of each component at the stage i. The factor exp $(n_i/4)$ accounts for the interconnecting hardware. The parameters β_i and α_i are the physical feature (shaping and scaling factor) of the cost – reliability curve of each component in stage i. V is the upper limit on the volume, C is the upper limit on the cost

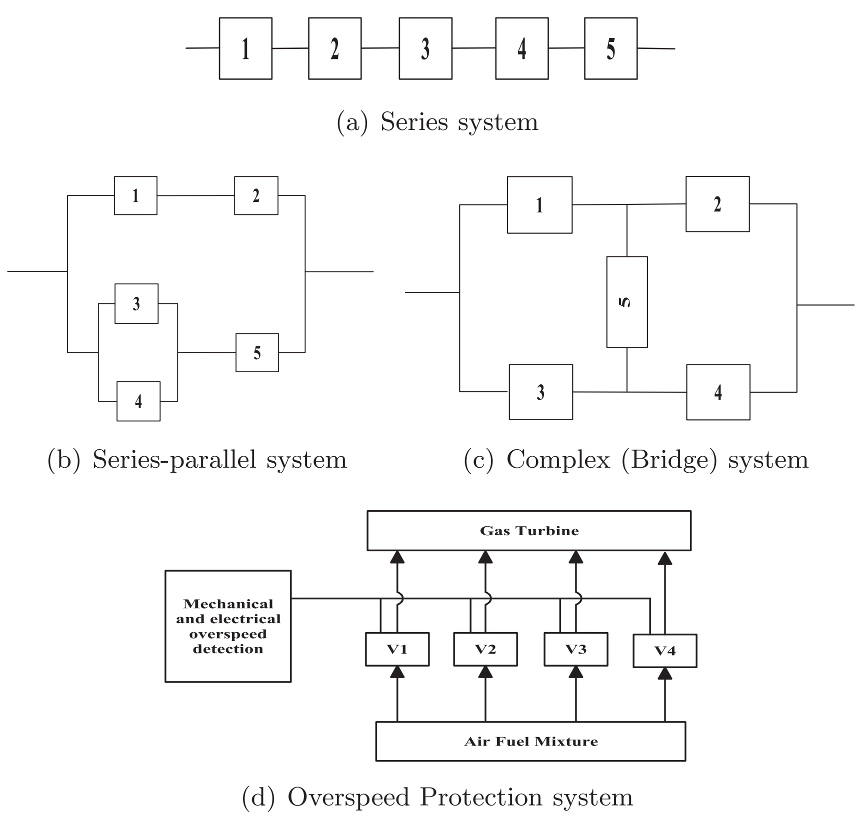


Fig. 1 – Series, series – parallel, bridge and overspeed gas turbine systems.

and W is the upper limit on the weight of the system. The input parameters defining the specific instances of the first four problems have the same values as Hikita et al. (1992); Yokota et al. (1996); Hsieh et al. (1998); Chen (2006); Coelho (2009); Yeh and Hsieh (2011); Garg et al. (2013a, b); Dhingra (1992); Kuo et al. (1978); Kim et al. (2006); Wu et al. (2011); Zou et al. (2010); Hsieh and You (2011), and are shown in Tables 1–3.

Problem 5. 15-unit system reliability optimization problem (Agarwal and Sharma, 2010; Valian et al., 2013) Considering a 15-unit structure, shown in Fig. 2, the optimization model is defined as follows

the reproduction strategy of cuckoos. At the most basic level, cuckoos lay their eggs in the nests of other host birds, which may be of different species. The host bird may discover that the eggs are not its own so it either destroys the eggs or abandons the nest all together. This has resulted in the evolution of cuckoo eggs which mimic the eggs of local host birds. CS is based on three idealized rules:

(i) Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
(ii) The best nests with high quality of eggs (solutions) will carry over to the next generations.
(iii) The number of available host nests is fixed, and a host

$$
\begin{aligned}
\text{Maximize } R_s(n) = {} & R_1R_2R_3R_4R_5R_6 + R_9R_{10}R_{11}R_{12}R_{13}R_{14}R_{15} \times (Q_1 + R_1Q_2 + R_1R_2Q_3 + R_1R_2R_3Q_4 + R_1R_2R_3R_4Q_5 + R_1R_2R_3R_4R_5Q_6) \\
& + R_4R_5R_6R_7R_9R_{10}(Q_{11} + R_{11}Q_{12} + R_{11}R_{12}Q_{13} + R_{11}R_{12}R_{13}Q_{14} + R_{11}R_{12}R_{13}R_{14}Q_{15}) \times (Q_1 + R_1Q_2) \\
& + \{(Q_1 + R_1Q_2)(Q_3 + R_3Q_4 + R_3R_4Q_7) + R_1R_2Q_7(Q_3 + R_3Q_4)\} \times (Q_{13} + R_{13}Q_{14} + R_{13}R_{14}Q_{15})R_5R_6R_8R_9R_{10}R_{11}R_{12} \\
& + R_1R_2R_5R_6R_7R_8R_{11}R_{12} \times (R_9R_{10} + Q_9 + R_9Q_{10})(Q_3 + R_3Q_4)(Q_{13} + R_{13}Q_{14} + R_{13}R_{14}Q_{15}) + (Q_5 + R_5Q_6) \\
& \times \{(Q_7 + R_7Q_{11} + R_7R_{11}Q_{12})(Q_9 + R_9Q_{10}) + R_9R_{10}(Q_{11} + R_{11}Q12)\} \times R_1R_2R_3R_4R_8R_{13}R_{14}R_{15} \\
& + R_1R_2R_7R_{11}R_{12}R_{13}R_{14}R_{15}(Q_9 + R_9Q_{10}) \times (Q_3 + R_3Q_4 + R_3R_4Q_5 + R_3R_4R_5Q_6) + R_3R_4R_7R_8R_9R_{10}R_{13}R_{14}R_{15}(Q_1 + R_1Q_2) \\
& \times (Q_{11} + R_{11}Q_{12})(Q_5 + R_5Q_6) \\
\text{subject to } g_y(n) = {} & \sum_{i=1}^{15} c_{yi}n_i \le b_y \quad y = 1, 2, \ldots, M, \\
n_i \in \mathbb{Z}^+, \quad & i = 1, 2, \ldots, 15
\end{aligned}
$$

where $b_y = d \times \sum_{i=1}^{15} c_{yi}$ with $d = rand(1.5,3.5)$. The parameters $R_i(n_i) = 1 - (1 - r_i)^{n_i}$ and $Q_i = 1 - R_i$ be the reliability and unreliability of subsystem i respectively. The coefficients c_{yi} and r_i are generated from uniform distributions in [0,100] and [0.6, 0.85], respectively. The parameter m refers to the number of constraints. Two sets of problem are considered by taking $M = 1$ and $M = 5$. The random data used in two sets of problems is given in Table 4 (Valian et al., 2013).

4. Cuckoo search (CS)

CS is a meta-heuristic search algorithm which has been proposed recently by Yang and Deb (2009) getting inspired from

can discover an alien egg with a probability $p_a \in [0,1]$. In this case, the host bird can either throw the egg away or abandon the nest so as to build a completely new nest in a new location.

To make the things even simpler, the last assumption can be approximated by the fraction of p_a of n nests that are replaced by new nests with new random solutions. The fitness function of the solution is defined in a similar way as in other evolutionary techniques. In this technique, egg presented in the nest will represent the solution while the cuckoo's egg represents the new solution. The aim is to use the new and potentially better solutions (cuckoos) to replace worse solutions that are in the nests. Based on these three rules, the basic steps of the cuckoo search are described in Algorithm 1.

Algorithm 1 Pseudo code of Cuckoo Search (CS)

```
Objective function: f(x),   x = (x_1, x_2, ..., x_d);
Generate an initial population of n host nests x_i ;   i = 1, 2, ··· , n;
While (t < MaxGeneration) or (stop criterion)
    Get a cuckoo randomly (say, i)
    Generate a new solution by performing Lévy flights;
    Evaluate its fitness f_i
    Choose a nest among n (say, j) randomly;
    if (f_i > f_j)
        Replace j by new solution
    end if
    A fraction(p_a) of the worse nests are abandoned and new ones are built;
    Keep the best solutions/nests;
    Rank the solutions/nests and find the current best;
    Pass the current best solutions to the next generation;
end while
```

Table 1 – Parameter used for Problem 1 and 3.

i	$10^5\alpha_i$	β_i	v_i	w_i	C	V	W
1	2.330	1.5	1	7	175	110	200
2	1.450	1.5	2	8			
3	0.541	1.5	3	8			
4	8.050	1.5	4	6			
5	1.950	1.5	2	9			

Table 2 – Parameter used for Problem 2.

i	$10^5\alpha_i$	β_i	v_i	w_i	C	V	W
1	2.500	1.5	2	3.5	175	180	100
2	1.450	1.5	4	4.0			
3	0.541	1.5	5	4.0			
4	0.541	1.5	8	3.5			
5	2.100	1.5	4	3.5			

Table 3 – Parameter used for Problem 4.

i	$10^5\alpha_i$	β_i	v_i	w_i	C	V	W
1	1.0	1.5	1	6	400	250	500
2	2.3	1.5	2	6			
3	0.3	1.5	3	8			
4	2.3	1.5	2	7			

The new solution $x_i^{(t+1)}$ of the cuckoo search is generated, from its current location x_i^t and probability of transition, with the following equation

$$x_i^{(t+1)} = x_i^{(t)} + \alpha \oplus \text{Lévy}(\lambda) \tag{4}$$

where α,($\alpha > 0$) represents a step size. This step size should be related to the problem specification and t is the current iteration number. The product $\oplus$ represents entry-wise multiplications as similar to other evolutionary algorithms like PSO but random walk via Lévy flight is much more efficient in exploring the search space as its step length is much longer in the long run.

In Mantegna's algorithm, the step length can be calculated by

$$\text{Lévy}(\alpha) \sim \frac{u}{|v|^{1/\alpha}} \tag{5}$$

where u and v are drawn from normal distribution, i.e.

Table 4 – Data used in the 15-unit system reliability problem.

i	r	c_1	c_2	c_3	c_4	c_5
1	0.6796	33.2468	35.6054	13.7848	44.1345	10.9891
2	0.7329	27.5668	44.9520	96.7365	25.9855	68.0713
3	0.6688	13.3800	28.6889	85.8783	19.2621	1.0164
4	0.6102	0.4710	0.4922	63.0815	12.1687	29.4809
5	0.7911	51.2555	39.6833	78.5364	23.9668	59.5441
6	0.8140	82.9415	59.2294	11.8123	28.9889	46.5904
7	0.8088	51.8804	78.4996	97.1872	47.8387	49.6226
8	0.7142	77.9446	86.6633	45.0850	25.0545	59.2594
9	0.8487	26.8835	7.8195	3.6722	76.9923	87.4070
10	0.7901	85.8722	27.7460	55.3950	53.3007	55.3175
11	0.6972	41.8733	90.4377	75.7999	95.0057	54.1269
12	0.6262	61.6181	58.0131	98.5166	97.9127	59.1341
13	0.6314	90.0418	77.8206	60.6308	37.2226	40.9427
14	0.6941	75.5947	36.4524	70.4654	96.9179	40.2141
15	0.6010	88.5974	61.0591	18.8802	42.1222	80.0045
d	–	3.2150	3.4710	3.3247	2.6236	3.4288

$$u \sim N(0, \sigma_u^2), \quad v \sim N(0, \sigma_v^2) \quad \sigma_u = \left\{ \frac{\Gamma(1+\alpha)\sin\left(\frac{\pi\alpha}{2}\right)}{\Gamma\left[\frac{1+\alpha}{2}\right]\alpha 2^{\frac{\alpha-1}{2}}} \right\}^{1/\alpha}, \quad \sigma_v = 1$$

where the distribution parameter $\alpha \in [0.3,1.99]$, Γ denotes the gamma function.

4.1. *Constraint handling technique*

Due to presence of constraints in the optimization problems, it is not an easy to find the feasible solution of the problem which optimize the performance of the system. For this, penalty function method has been used for handling the constraints. The main function of the penalty functions is to penalize the unfeasible solution. Despite the popularity of penalty functions, they have several drawbacks out of which the main one is that of having too many parameters to be adjusted and finding the right combination of the same may not be easy. Moreover, there is no guarantee that the optima will be attained. For overcoming this drawback, Deb (2000) modified these algorithms using concept of parameter-free penalty functions by defining the modified objective function F in the search space S

$$F(x) = \begin{cases} f(x) & \text{if } x \in S \\ f_w + \sum_{j=1}^{M} g_j(x) & \text{if } x \notin S \end{cases} \tag{6}$$

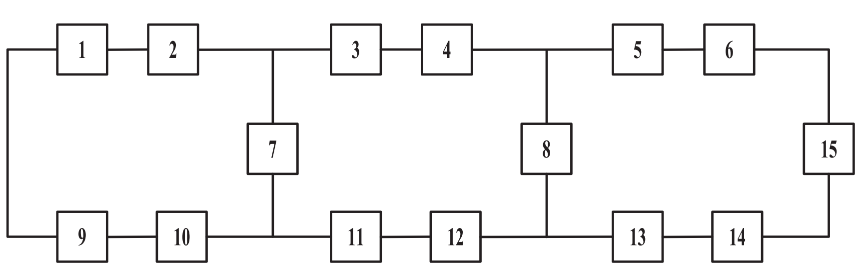


Fig. 2 – Structure of the 15-unit system.

Table 5 – Optimal solutions of the Problem 1.

Method	Kuo et al. (1978)	Gopal et al. (1978)	Hikita et al. (1978)	Hikita et al. (1992)	Hsieh et al. (1998)	Gen and Yun (2006)	Chen (2006)	Yeh and Hsieh (2011)	Wu et al. (2011)	Proposed approach
n	(3,3,2,3,2)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)	(3,2,2,3,3)
r	0.77960	0.8	0.774887	0.777143	0.779427	0.780874	0.779266	0.779399	0.78037307	0.779439734086
	0.80065	0.8625	0.870065	0.867514	0.869482	0.871292	0.872513	0.871837	0.87178343	0.871995212292
	0.90227	0.90156	0.898549	0.896696	0.902674	0.902316	0.902634	0.902885	0.90240890	0.902873050171
	0.71044	0.7	0.716524	0.717739	0.714038	0.711945	0.710648	0.711403	0.71147356	0.711127088245
	0.85947	0.8	0.791368	0.793889	0.786896	0.786995	0.788406	0.787800	0.78738760	0.787986374473
R_s	0.92975	0.930289	0.931451	0.931363	0.931578	0.931676	0.931678	0.931682	0.931680	0.931682106582
MPI	2.75032%	1.99840%	0.33714%	0.464912%	0.15215%	0.00893%	0.00601%	[a]	0.0030834%	
Slacks of	27	27	27	27	27	27	27	27	27	27
$g_1 \sim g_3$	0.00001	0.0265	0.108244	0	0.121454	0.003352	0.001559	−0.0002184[b]	0.121454	$4.42986674 \times 10^{-7}$
	10.57248	7.518918	7.518918	7.518918	7.518918	7.518918	7.518918	7.518918	7.518918	7.518918241
Mean	–	–	–	–	–	–	–	0.930580	–	0.931671924231087
Std	–	–	–	–	–	–	–	8.14×10^{-4}	5.2382×10^{-3}	1.49487×10^{-5}

[a] Infeasible.
[b] Violate constraint.

where x are solutions obtained by approaches and f_w is the worst feasible solution in the population and set it to be zero if there is no feasible solution.

5. Numerical results and discussions

The nests for each example use the variable vectors n and r. During the evolution process, the integer variables n_i are treated as real variables, and in evaluating the objective functions, the real values are transformed to the nearest integer values. The presented algorithm is implemented in Matlab (MathWorks) and the program has been run on a T6400 @ 2 GHz Intel Core(TM) 2 Duo processor with 2 GB of Random Access Memory (RAM). In order to eliminate stochastic discrepancy, in each case study, 25 independent runs are made which involves 25 different initial trial solutions for each optimization method. To evaluate the performance of proposed approach, the following maximum possible improvement (MPI) index has been used to compute the relative improvement

$$\mathrm{MPI} = \frac{R_s(\mathrm{CS}) - R_s(\mathrm{other})}{1 - R_s(\mathrm{other})} \tag{7}$$

where R_s(CS) is the best-known solution obtained from CS approach and R_s(other) is the best solution by other typical approaches. Numerical results are reported in Tables 5–9, in which the best solutions for each problem are reported and compared with solutions reported previously in the literature, which show that proposed approach leads to improvement in reliability. Clearly, greater MPI implies greater improvement.

For the series system (i.e. Problem 1), Table 5 shows that the best solution by our approach is 0.931682106582 which is superior to all those of the other typical approaches in the literature (Kuo et al., 1978; Gopal et al., 1978; Hikita et al., 1978, 1992; Hsieh et al., 1998; Gen and Yun, 2006; Chen, 2006; Yeh and Hsieh, 2011; Wu et al., 2011) with their improvement indices are 2.75032%, 1.99840%, 0.33714%, 0.00747%, 0.464912%, 0.15215%,0.00893%, 0.00601% and 0.0030834% respectively. It is worth notifying here that solution by ABC algorithm, as given by Yeh and Hsieh (2011), is infeasible solution as it violates the cost constraint. The results of the experiment for the Problem 2, shown in Table 6, indicate that the best solution of the CS algorithm ($R_s = 0.999976648818$) is much better than the solution given by (Hikita et al., 1992; Hsieh et al., 1998; Chen, 2006; Kim et al., 2006; Wu et al., 2011; Valian et al., 2013; Zou et al., 2010) by an improvement factor 25.27621%, 9.56164%, 0.29384%, 1.43021%, 0.037748%, 0% and 0.037748% respectively. It should be noticed that even very small improvements in reliability are critical and beneficial to system security and system efficiency. It is worth mentioning that the solution obtained by Yeh and Hsieh (2011) by using the ABC algorithm is not a feasible solution as it violates the cost constraint function. From Table 7 one can observe that the solution to the Problem 3 as obtained by us is relatively with most significant improvement over the solutions presented by Hikita et al. (1992); Hsieh et al. (1998); Chen (2006); Kim et al. (2006); Coelho (2009). It may again be pointed out that the solution by ABC algorithm, obtained by Yeh and

Table 6 – Optimal solutions of the Problem 2.

Method	Hikita et al. (1992)	Hsieh et al. (1998)	Chen (2006)	Kim et al. (2006)	Yeh and Hsieh (2011)	Wu et al. (2011)	Valian et al. (2013)	Proposed approach
n	(3,3,1,2,3)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)	(2,2,2,2,4)
r	0.83819295	0.785452	0.812485	0.812161	0.8197457	0.81918526	0.819927087	0.819483232488
	0.85506525	0.842998	0.843155	0.853346	0.8450080	0.84366421	0.845267657	0.844783084455
	0.87885933	0.885333	0.897385	0.897597	0.8954581	0.89472992	0.895491554	0.895810553887
	0.91140223	0.917958	0.894516	0.900710	0.9009032	0.89537628	0.895440692	0.895220216915
	0.85035522	0.870318	0.870590	0.866316	0.8684069	0.86912724	0.868318775	0.868542486973
R_s	0.99996875	0.99997418	0.99997658	0.99997631	0.99997731	0.99997664	0.9999766488	0.999976648818
MPI	25.27621%	9.56164%	0.29384%	1.43021%	[a]	0.037748%	0%	–
Slacks of	53	40	40	40	40	40	40	40
$g_1 \sim g_3$	0.000011	1.194440	0.002627	0.007300	−1.469522[b]	0.000561	0.0000161	$2.7216628 \times 10^{-10}$
	7.110849	1.609289	1.609289	1.609289	1.609289	1.609289	1.6092890	1.609288966
Mean	–	–	–	–	0.99997517	–	–	0.999976290153280
Std	–	–	–	–	2.89×10^{-6}	1.3362×10^{-5}	4.45034×10^{-6}	1.14187×10^{-6}

[a] Infeasible.
[b] Violate constraint.

Hsieh (2011) is also infeasible, since it again violates the cost constraint function. Table 8 depicts that the solution of Problem 4 as obtained by the proposed approach is better than the previously known solutions by Yokota et al. (1996); Chen (2006); Dhingra (1992); Kim et al. (2006); Zou et al. (2010); Wu et al. (2011); Valian et al. (2013). The optimal component redundancy by the proposed approach is (5,5,4,6) which is completely different from those by the other approaches. Here again we have observed through calculations that the solutions given by Yeh and Hsieh (2011) and Yokota et al. (1996) are not feasible solutions as both of these violate the cost constraint function. The improvement indices are 88.37809%, 21.85273%, 17.59015%, 3.55294%, 0.010114% and 0.001291% respectively from Dhingra (1992); Chen (2006); Kim et al. (2006); Coelho (2009); Zou et al. (2010); Wu et al. (2011); Valian et al. (2013) respectively. Moreover, the solutions found by the proposed approach can dominate any other methods for the four example problems discussed in literature. In other words, we may say that CS algorithm is able to find solutions of quality comparable to those published earlier in the literature. Moreover, the standard deviations of system reliabilities by proposed approach are pretty low, and it further implies that the approach seems reliable to solve the reliability-redundancy allocation problems. For example, the standard deviations of system reliabilities for Problems 1–4 are 1.49487×10^{-5}, 1.14187×10^{-6}, 7.03799×10^{-7} and 6.97619×10^{-9} respectively. For the two sets of Problem 5, the 15-unit system reliability optimization, the results computed by the proposed approach along with the results given by other algorithms are listed in Table 9. From the table it has been concluded that the results provided by proposed approach is far better than the other algorithms. The simulation results of 25 independent runs in terms of mean, median, worst and standard deviations provided that the proposed approach performs better than the others.

In order to study the performance of the proposed algorithm statistically with other meta-heuristic algorithm namely PSO and ABC, the simulation experiments were repeated for 25 observations. All 25 observations are generated with 25 different initial solutions. For each experiment reliability of the system is calculated. An unpaired pooled t−test assuming equal variances has been applied with significance level of 5 percent. The pooled t-test is applied for the comparisons of CS results with PSO and ABC results for each problem. The results of the t-test for the maximizing the reliability of the system are shown in Tables 10–14 for the problems P1–P5 respectively. It is indicated from the tables that the values of t-stat are greater than the t-critical values. Also the p-value obtained for both one-tail and two-tail test is less than the significant level. Thus the means of system reliability for CS is higher than the mean from PSO as well as ABC and this difference is statistically significant.

6. Conclusion

This paper presents penalty guided cuckoo search (CS) for solving various reliability design problem, which include series systems, series-parallel system, complex (bridge) system and overspeed protection system. The objective of the problem is to maximize the system reliability subject to three nonlinear resource constraints. In these optimization problems, both the redundancy and the corresponding reliability of each component in each subsystem are decided simultaneously under cost, weight and volume constraints. To evaluate the performance of CS algorithm, numerical experiments are conducted and compared with the previous studies for mixed-integer reliability problems. As shown, the best solutions found by penalty guided CS are all better than the well-know best solutions by other heuristic methods for mixed-integer reliability problems. Also by using the means of unpaired pooled t-test, the proposed cuckoo search based penalty guided reliability redundancy allocation problem is shown to be statistically significant as compared to other methods. Thus, the CS was demonstrated to be a promising and viable tool to solve reliability – redundancy optimization problems.

Table 7 – Optimal solutions of the Problem 3.

Method	Hikita et al. (1992)	Hsieh et al. (1998)	Chen (2006)	Kim et al. (2006)	Coelho (2009)	Yeh and Hsieh (2011)	Wu et al. (2011)	Zou et al. (2011)	Zou et al. (2010)	Valian et al. (2013)	Proposed approach
n	(3,3,2,3,2)	(3,3,3,3,1)	(3,3,3,3,1)	(3,3,3,3,1)	(3,3,2,4,1)	(3,3,2,4,1)	(3,3,2,4,1)	(3,3,2,4,1)	(3,3,2,4,1)	(3,3,2,4,1)	(3,3,2,4,1)
r	0.814483	0.814090	0.812485	0.807263	0.826678	0.828087	0.82868361	0.82983999	0.82883148	0.828094038	0.827855652338
	0.821383	0.864614	0.867661	0.868116	0.857172	0.857805	0.85802567	0.85798911	0.85836789	0.858004485	0.857626105413
	0.896151	0.890291	0.861221	0.872862	0.914629	0.704163	0.91364616	0.91333926	0.91334996	0.914162924	0.914752916604
	0.713091	0.701190	0.713852	0.712673	0.648918	0.648146	0.64803407	0.64674479	0.64779451	0.647907792	0.648217208595
	0.814091	0.734731	0.756699	0.751034	0.715290	0.914240	0.70227595	0.70310972	0.70178737	0.704565982	0.702670374782
R_s	0.99978937	0.99987916	0.99988921	0.99988764	0.99988957	0.99948407[a]	0.99988963	0.99988960	0.99988962	0.99988963	0.999889631978
MPI	47.60099%	8.66598%	0.38088%	1.77285%	0.05612%	[b]	0.00179%	0.02896%	0.01085%	0.00726%	–
Slacks of	27	18	18	18	5	5	5	5	5	5	5
$g_1 \sim g_3$	0.000000	0.376347	0.001494	0.007300	0.000339	−25.433926[c]	0.00000359	0.00000594	0.00004063	0.00007929	$1.06723518 \times 10^{-10}$
	10.572475	4.264770	4.264770	1.609289	1.560466	1.560466288	1.56046629	1.56046629	1.56046629	1.560466288	1.560466288
Mean	–	–	–	–	0.99988594	0.99988362	–	0.99988263	0.99988656	–	0.999889270567758
Std	–	–	–	–	0.00000069	1.026×10^{-5}	4.0163×10^{-5}	1.6×10^{-5}	1.0699×10^{-5}	1.40652×10^{-5}	7.037994×10^{-7}

[a] In Yeh and Hsieh (2011), it was reported 0.99988962.
[b] Infeasible.
[c] Violate constraint.

Table 8 – Optimal solutions of the Problem 4.

Method	Dhingra (1992)	Yokota et al. (1996)	Chen (2006)	Kim et al. (2006)	Coelho (2009)	Zou et al. (2010)	Yeh and Hsieh (2011)	Wu et al. (2011)	Valian et al. (2013)	Proposed approach
n	(6,6,3,5)	(3,6,3,5)	(5,5,5,5)	(5,5,5,5)	(5,6,4,5)	(5,6,4,5)	(5,6,4,5)	(5,6,4,5)	(5,5,4,6)	(5,5,4,6)
r	0.81604	0.965593	0.903800	0.895644	0.902231	0.90186194	0.901614	0.90163164	0.901614595	0.901598077027
	0.80309	0.760592	0.874992	0.885878	0.856325	0.84968407	0.849920	0.84997020	0.888223369	0.888226184172
	0.98364	0.972646	0.919898	0.912184	0.948145	0.94842696	0.948143	0.94821828	0.948141029	0.948101861662
	0.80373	0.804660	0.890609	0.887785	0.883156	0.88800590	0.888223	0.88812885	0.849920899	0.849980778637
R_s	0.99961	0.999468	0.999942	0.999945	0.999953	0.99995467	0.999955	0.99995467	0.999954674	0.999954674585
MPI	88.37809%	[a]	21.85273%	17.59015%	3.56294%	0.010114%	[a]	0.010114%	0.001291%	–
Slacks of	65	92	50	50	55	55	55	55	55	55
$g_1 \sim g_3$	0.064	−70.73357[b]	0.002152	0.9380	0.975465	0.00120356	−0.0003364[b]	0.000009	0.0000000096	$8.82494077 \times 10^{-10}$
	4.348	127.583189	28.803701	28.8037	24.801882	24.8018827	24.80188272	24.081883	15.36346309	15.3634630874
Mean	–	–	–	–	0.999907	0.99992624	0.9999487	–	–	0.999954670626769
Std	–	–	–	–	0.000011	2.8874×10^{-5}	9.244×10^{-6}	1.3895×10^{-5}	4.45034×10^{-6}	6.97619×10^{-9}

[a] Infeasible solution.
[b] Violate constraint.

Table 9 – Comparison results for the 15-unit system reliability problem.

Problem 5	Algorithm	Worst	Best	Mean	Median	SD
15 × 1	GA	0.90702516	0.96357169	0.92726226	0.92688183	0.014598
	PSO	0.96223410	0.97967292	0.97092969	0.97087057	0.004091
	ABC	0.93679247	0.96805041	0.95263439	0.95228937	0.009003
	CS	0.97108268	0.98421284	0.97440402	0.97311046	0.003477
15 × 5	GA	0.90005601	0.95716040	0.92911247	0.93188471	0.015449
	PSO	0.95542482	0.97390351	0.96667014	0.96881115	0.005614
	ABC	0.94427204	0.97011592	0.95496871	0.95394238	0.007653
	CS	0.96649615	0.97503662	0.97073179	0.96857881	0.003785

Table 10 – T – test: Two samples assuming equal variances for system reliability of P1.

	System reliability of P1		
	PSO	ABC	CS
Mean	0.92364034	0.92988735	0.93167192
Variance	3.556160×10^{-5}	6.467485×10^{-6}	2.234643×10^{-10}
Std	5.963355×10^{-3}	2.543125×10^{-3}	1.494872×10^{-5}
Observation	25	25	25
Pooled variance	1.778091×10^{-5}	3.233854×10^{-6}	
Hypothesized mean difference	0	0	
Degree of freedom	48	48	
T stat	6.73408984	3.50854474	
P ($T \le t$) one tail	9.403838×10^{-9}	4.948373×10^{-4}	
T critical one-tail	1.677224	1.677224	
P ($T \le t$) two tail	1.880767×10^{-8}	9.896747×10^{-4}	
T critical two-tail	2.010634	2.01063475	

Table 11 – T – test: Two samples assuming equal variances for system reliability of P2.

	System reliability of P2		
	PSO	ABC	CS
Mean	0.99996191	0.99996472	0.99997629
Variance	1.235391×10^{-10}	2.127280×10^{-10}	1.303870×10^{-12}
Std	1.111481×10^{-5}	1.458519×10^{-5}	1.141871×10^{-6}
Observation	25	25	25
Pooled variance	6.242152×10^{-11}	1.070159×10^{-10}	
Hypothesized mean difference	0	0	
Degree of freedom	48	48	
T stat	6.43417609	3.95406113	
P ($T \le t$) two tail	2.713218×10^{-8}	1.260151×10^{-4}	
T critical one-tail	1.677224	1.677224	
P ($T \le t$) two tail	5.426437×10^{-5}	2.520303×10^{-4}	
T critical two-tail	2.010634	2.010634	

Table 12 – T – test: Two samples assuming equal variances for system reliability of P3.

	System reliability of P3		
	PSO	ABC	CS
Mean	0.99987054	0.99988152	0.99988927
Variance	9.255383×10^{-10}	1.408923×10^{-10}	4.953337×10^{-13}
Std	3.042266×10^{-5}	1.186980×10^{-5}	7.037994×10^{-7}
Observation	25	25	25
Pooled Variance	4.630168×10^{-10}	7.069383×10^{-11}	
Hypothesized mean difference	0	0	
Degree of freedom	48	48	
T stat	3.07699432	3.25678275	
P ($T \le t$) one tail	0.00172473	0.00103556	
T critical one-tail	1.677224	1.677224	
P ($T \le t$) two tail	0.00344947	0.00207112	
T critical two-tail	2.0106347	2.010634	

Table 13 – T – test: Two samples assuming equal variances for system reliability of P4.

	System reliability of P4		
	PSO	ABC	CS
Mean	0.99994446	0.99995051	0.99995467
Variance	2.010464×10^{-10}	3.072377×10^{-11}	4.866728×10^{-17}
Std	1.417908×10^{-5}	5.542903×10^{-6}	6.976194×10^{-9}
Observation	25	25	25
Pooled variance	1.005232×10^{-10}	1.536191×10^{-11}	
Hypothesized mean difference	0	0	
Degree of freedom	48	48	
T stat	3.59718004	3.74828069	
P ($T \leq t$) one tail	$3.79177554 \times 10^{-4}$	2.391724×10^{-4}	
T critical one-tail	1.677224	1.677224	
P ($T \leq t$) two tail	7.583551×10^{-4}	4.783449×10^{-4}	
T critical two-tail	2.010634	2.010634	

Table 14 – T – test: Two samples assuming equal variances for system reliability of Problem 5.

	GA	PSO	ABC	CS
System reliability of Problem 5 (15 × 1 structure)				
Mean	0.92726226	0.97092969	0.95263439	0.97440402
Variance	0.21310×10^{-3}	0.01673×10^{-3}	0.08105×10^{-3}	0.01208×10^{-3}
Std	0.014598	0.004091	0.009003	0.003477
Observation	25	25	25	25
Pooled variance	0.11259×10^{-3}	0.01441×10^{-3}	0.04657×10^{-3}	
Hypothesized mean difference	0	0	0	
Degree of freedom	48	48	48	
T stat	15.70724	3.23556	11.27832	
P ($T \leq t$) one tail	0	0.00110	2.10942×10^{-15}	
T critical one-tail	1.67722	1.67722	1.67722	
P ($T \leq t$) two tail	1.98729×10^{-14}	0.00176	2.23153×10^{-11}	
T critical two-tail	2.01063	2.01063	2.01063	
System reliability of Problem 5 (15 × 5 structure)				
Mean	0.92911247	0.96667014	0.95496871	0.97073179
Variance	0.23867×10^{-3}	0.03151×10^{-3}	0.05856×10^{-3}	0.01432×10^{-3}
Std	0.015449	0.005614	0.007653	0.003785
Observation	25	25	25	25
Pooled variance	0.12649×10^{-3}	0.02292×10^{-3}	0.03644×10^{-3}	
Hypothesized mean difference	0	0	0	
Degree of freedom	48	48	48	
T stat	13.08297	2.99940	9.23130	
P ($T \leq t$) one tail	0	0.00213	1.62414×10^{-12}	
T critical one-tail	1.67722	1.67722	1.67722	
P ($T \leq t$) two tail	1.02373×10^{-12}	0.00311	1.14450×10^{-9}	
T critical two-tail	2.01063	2.01063	2.01063	

REFERENCES

Agarwal M, Sharma VK. Ant colony approach to constrained redundancy optimization in binary systems. Appl Math Model 2010;34:992–1003.

Chen TC. IAs based approach for reliability redundancy allocation problems. Appl Math Comput 2006;182(2):1556–67.

Coelho LS. An efficient particle swarm approach for mixed-integer programming in reliabilityredundancy optimization applications. Reliab Eng Syst Saf 2009;94(4):830–7.

Coit DW, Smith AE. Reliability optimization of series-parallel systems using genetic algorithm. IEEE Trans Reliab 1996;R-45(2):254–60.

Deb K. An efficient constraint handling method for genetic algorithms. Comput Methods Appl Mech Eng 2000;186:311–38.

Dhingra AK. Optimal apportionment of reliability and redundancy in series systems under multiple objectives. IEEE Trans Reliab 1992;41:576–82.

Garg H, Sharma SP. Stochastic behavior analysis of industrial systems utilizing uncertain data. ISA Trans 2012;51(6):752–62.

Garg H, Sharma SP. Multi-objective reliability-redundancy allocation problem using particle swarm optimization. Comput Ind Eng 2013;64(1):247–55.

Garg H, Rani M, Sharma SP. Predicting uncertain behavior of press unit in a paper industry using artificial bee colony and fuzzy Lambda-Tau methodology. Appl Soft Comput 2013a;13(4):1869–81.

Garg H, Rani M, Sharma SP. An efficient two phase approach for solving reliability-redundancy allocation problem using artificial bee colony technique. Comput Operat Res 2013b;40(12):2961–9.

Garg H, Rani M, Sharma SP, Vishwakarma Y. Intuitionistic fuzzy optimization technique for solving multi-objective reliability optimization problems in interval environment. Expert Syst Appl 2014;41:3157–67.

Geem ZW, Kim JH, Loganathan GV. A new heuristic optimization algorithm: harmony search. Simulation 2001;76(2):60–8.

Gen M, Yun YS. Soft computing approach for reliability optimization: state-of- the-art survey. Reliab Eng Syst Saf 2006;91(9):1008–26.

Gopal K, Aggarwal KK, Gupta JS. An improved algorithm for reliability optimization. IEEE Trans Reliab 1978;27:325–8.

Hikita MY, Nakagawa Y, Nakashima K, Narihisa H. Reliability optimization of system by a surrogate constraints algorithm. IEEE Trans Reliab 1978;7:325–8.

Hikita M, Nakagawa Y, Harihisa H. Reliability optimization of systems by a surrogate constraints algorithm. IEEE Trans Reliab 1992;R-41(3):473–80.

Hsieh TJ, Yeh WC. Penalty guided bees search for redundancy allocation problems with a mix of components in seriesparallel systems. Comput Operat Res 2012;39(11):2688–704.

Hsieh Y, You PS. An effective immune based two-phase approach for the optimal reliability-redundancy allocation problem. Appl Math Comput 2011;218:1297–307.

Hsieh YC, Chen TC, Bricker DL. Genetic algorithms for reliability design problems. Microelectron Reliab 1998;38:1599–605.

Kim HG, Bae CO, Park DJ. Reliability -redundancy optimization using simulated annealing algorithms. Int J Qual Maint Eng 2006;12(4):354–63.

Kuo W, Prasad VR. An annotated overview of system-reliability optimization. IEEE Trans Reliab 2000;49(2):176–87.

Kuo W, Hwang CL, Tillman FA. A note on heuristic methods in optimal system reliability. IEEE Trans Reliab 1978;R-27:320–4.

Painton L, Campbell J. Genetic algorithms in optimization of system reliability. IEEE Trans Reliab 1995;44:172–8.

Rajabioun R. Cuckoo optimization algorithm. Appl Soft Comput 2011;11(8):5508–18.

Valian E, Tavakoli S, Mohanna S, Haghi A. Improved cuckoo search for reliability optimization problems. Comput Ind Eng 2013;64(1):459–68.

Wang L, Li L. A coevolutionary differential evolution with harmony search for reliabilityredundancy optimization. Expert Syst Appl 2012;39:5271–8.

Wu P, Gao L, Zou D, Li S. An improved particle swarm optimization algorithm for reliability problems. ISA Trans 2011;50:71–81.

Yang XS, Deb S. Cuckoo search via le'vy flights. In: Proc. of World Congress on Nature & Biologically Inspired Computing (NaBIC 2009), December 2009, India. USA: IEEE Publications; 2009. p. 210–4.

Yeh WC. A two-stage discrete particle swarm optimization for the problem of multiple multi-level redundancy allocation in series systems. Expert Syst Appl 2009;36:9192–200.

Yeh WC, Hsieh TJ. Solving reliability redundancy allocation problems using an artificial bee colony algorithm. Comput Operat Res 2011;38(11):1465–73.

Yokota T, Gen M, Li Y, Kim CE. A genetic algorithm for interval nonlinear integer programming problem. Comput Ind Eng 1996;31(3–4):913–7.

Zou D, Gao L, Wu J, Li S, Li Y. A novel global harmony search algorithm for reliability problems. Comput Ind Eng 2010;58:307–16.

Zou D, Gao L, Li S, Wu J. An effective global harmony search algorithm for reliability problems. Expert Syst Appl 2011;38:4642–8.