Journal of Theoretical and Applied Information Technology
31st December 2012. Vol. 46 No.2

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

RELIABILITY REDUNDANCY OPTIMIZATION FOR OPTIMAL DESIGNING IN GAS TURBINES' OVERSPEED PROTECTION USING ELITISM BOX-MULLER HARMONY SEARCH ALGORITHM

[1] ABDOLVAHHAB FETANAT, [2] GHOLAMREZA SHAFIPOUR, [3] FARROKH GHANATIR

[1,2,3] Department of Electrical and Computer Engineering, Behbahan Branch, Islamic Azad University, Behbahan, Iran

E-mail: [1]av-fetanat@behbahaniau.ac.ir, [2]pp_shafipour@yahoo.com, [3]f-ghanatir@behbahaniau.ac.ir

ABSTRACT

This paper proposes an optimal design for control and overspeed protection in Gas turbine by means of reliability redundancy optimization. In general form, the problem is consisting of four control valves (stop valves), as a subsystem, each of them has the same components with reliability, weight and other parameters. The problem objective is to find a maximum reliability for total system in order to stop control valves in turbine's overspeed state. The objective function is optimized and solved using a new type of Harmony Search Algorithm (HSA) that authors call it Elitism Box-Muller Harmony Search Algorithm (EBMHSA). The simulation results show that this method gives a better and accurate solution compared with the other algorithms.

Keywords: *Reliability Redundancy Optimization, Optimal design, overspeed, Gas turbine, Elitism Box-Muller Harmony Search Algorithm (EBMHSA)*

1. INTRODUCTION

Control and protection for a gas turbine is almost like a steam turbine. While the gas turbine works in high temperature than steam turbine. Thus it should under closer control, which called the later (closer control) "sequencing". Sequencing which can manage the gas turbine automatically. Figure 1 shows an example of a gas turbine control system.

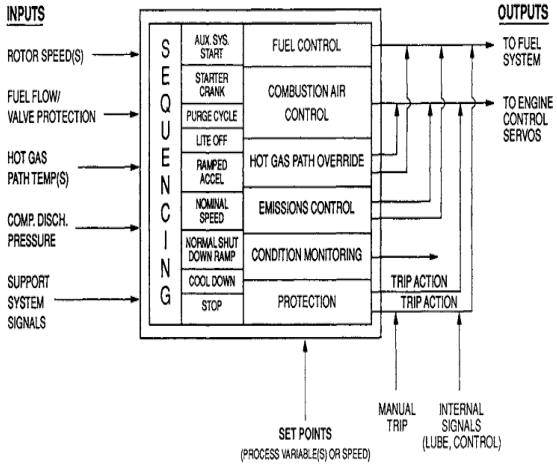

Figure 1. View Of Gas Turbine Control System And Procedures

Rising demand for systems with higher reliability, so laboratory studies and simulations has taken the trend. A number of studies have also been done on overspeed protection such as the analysis of instability in the steam turbine [1-3] and analysis of the reliability wind turbines [4]. Application of reliability redundancy optimization problem is in the protection of the overspeed [5]. Goal-oriented programming method is used for overspeed problem solving. This problem have been studied using heuristic algorithms and exploration [6], genetic algorithm [7], particle swarm algorithm [8] and the algorithm NGHS [9].

According to Figure 1, the overspeed protection of gas turbine is very significant in systems control [1]. Overspeed control is the first step against excessive speed. That restores the turbine in the steady state condition for which the turbine has been installed. The application of sequencing is to reset the units while emergency due to overspeed. Usually the emergency reset of the system is designed independent of the overspeed control. Hence, high reliability for operation of control valves shall be considered. Here, the performance with maximum reliability will be the main target for

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the control valves. In practice, usually four or more parallel control valves are used. Each of them allows fuel easily to pass through a narrow channel. In normal working mode, control valves are opened sequentially [2].

2. RELIABILITY REDUNDANCY OPTIMIZATION PROBLEM

Many designers are devoted to improve the reliability of manufacturing systems or product components to be more competitive in the market. Typical approaches to achieve higher systems reliability are increasing the reliability of system components, and increasing the redundant components in various systems.

In this work, a reliability redundancy allocation problem of maximizing the system reliability subject to multiple nonlinear constraints can be stated as a nonlinearly mixed-integer programming model in general form

$$\begin{aligned} &Max && R_s = f(r,n) \\ &S.t. && g(r,n) \le l \\ & && 0 \le r_i \le 1, n_i \in Z^+, 0 \le i \le m \end{aligned} \quad (1)$$

Where R_S is the reliability of system, $r = (r_1, r_2, r_3, ..., r_m)$ is the vector of the component reliabilities for the system, $n = (n_1, n_2, n_3, ..., n_m)$ is the vector of the redundancy allocation for the system, r_i and n_i are the reliability and the number of components in the ith subsystem respectively, $f(.)$ is the objective function for the overall system reliability, $g(.)$ is the constraint function and l is the resource limitation, $g(.)$ is the constraint function usually associated with system weight, volume and cost, m is the number of subsystems in the system. The goal is to determine the number of component and the components' reliability in each system so as to maximize the overall system reliability. The problem belongs to the category of constrained nonlinear mixed-integer optimization problems. 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, four control valves (V1–V4) must close (Figure 2).

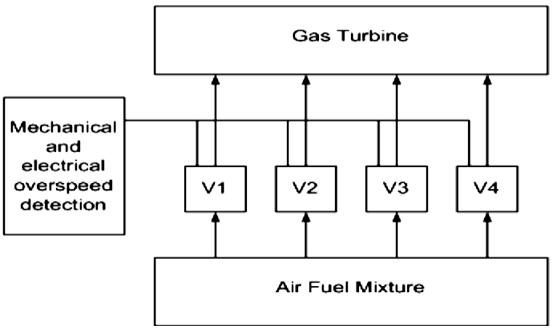

Figure 2. Schematic Diagram For The Overspeed Protection System Of A Gas Turbine

The control system is modeled as a 4-stage series system. The control valve (subsystem) i has n_i parallel controllers each with the same reliability (r_i).

Thus the reliability of the subsystem i would be:

$$R_i = 1 - \prod_{j=1}^{n_i} (1 - r_i) = 1 - (1 - r_i)^{n_i} \quad (2)$$

These control valves should work together to cut off fuel and thus the reliability of the whole system would include:

$$R_S = f(r,n) = \prod_{i=1}^{m} (1 - (1 - r_i)^{n_i}) \quad (3)$$

R_S is the objective function in the optimization problem. 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 [9]:

$$\begin{aligned} &Max && R_S = f(r,n) \\ &Subject.to \\ & && g_1(r,n) = \sum_{i=1}^{m} v_i.n_i^2 - V \le 0 \\ & && g_2(r,n) = \sum_{i=1}^{m} C(r_i).[n_i + e^{0.25.n_i}] - C \le 0 \\ & && g_3(r,n) = \sum_{i=1}^{m} w_i.n_i.e^{0.25.n_i} - W \le 0 \\ & && 0 \le r_i \le 1, r_i \in R^+ \\ & && 1 \le n_i \le 10, n_i \in Z^+ \\ & && 0 \le i \le m \end{aligned} \quad (4)$$

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Where v_i is the volume of each component in subsystem i, V is the upper limit on the sum of the subsystems' products of volume and weight, C is the upper limit on the cost of the system. $C(r_i)$ is the cost of each component with reliability r_i at subsystem i:

$$C(r_i) = \alpha_i . [-T / \ln(r_i)]^{\beta_i} \quad (5)$$

T is the operating time during which the component must not fail, and W is the upper limit on the weight of the system. The input parameters defining the overspeed protection system for a gas turbine are shown in Table 1. The data shown in Table 1 are also available in [6].

Table 1. Data Used In Overspeed Protection System Of Gas Turbine

stage	$10^5\alpha_i$	β_i	v_i	w_i	V	C	W	T
1	1.0	1.5	1	6	250	400	500	1000h
2	2.3	1.5	2	6				
3	0.3	1.5	3	8				
4	2.3	1.5	2	7				

3. HARMONY SEARCH ALGORITHM

The harmony search algorithm (HSA) is a music-inspired evolutionary algorithm, mimicking the improvisation process of music players [10, 11]. The HS is simple in concept, few in parameters, and easy in implementation, with theoretical background of stochastic derivative [11]. The algorithm was originally developed for discrete optimization and later expanded for continuous optimization [12]. It has been successfully applied to various benchmark and real-world problems including traveling salesman problem, parameter optimization of river flood model, design of pipeline network, and design of truss structures. They are as follows [13]:

Step 1: Initialize the problem and algorithm parameters
Step 2: Initialize the harmony memory
Step 3: Improvise a new harmony
Step 4: Update the harmony memory
Step 5: Check the stopping criterion

These steps are described in the next five subsections:

3.1. Initialize the problem and algorithm parameters

In Step 1, the optimization problem is specified as follows:

$$\begin{aligned} &\text{Minimize} && f(\vec{x}) \\ &\text{Subject to} && g_i(\vec{x}) \geq 0 \qquad i = 1,2,...,M. \\ & && h_j(\vec{x}) = 0 \qquad j = 1,2,...,P. \\ & && {}_L x_k \leq x_k \leq {}_U x_k \qquad k = 1,2,...,N. \end{aligned} \quad (6)$$

Where $f(\vec{x})$ is the objective function, M is the number of inequality constraints and P is the number of equality constraints. x is the set of each decision variable x_i; N is the number of decision variables. The lower and upper bounds for each decision variable are ${}_L x_i$ and ${}_U x_i$ respectively. The HSA parameters are also specified in this step. These are the harmony memory size (HMS), or the number of solution vectors in the harmony memory, harmony memory considering rate (HMCR), pitch adjusting rate (PAR), and the number of improvisations (NI), or stopping criterion. The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. The HM is similar to the genetic pool in the genetic algorithms (GAs) [13].

Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in Step 3.

3.2. Initialize the harmony memory

In Step 2, the HM matrix is filled with as many randomly generated solution vectors as the HMS:

$$\mathrm{HM} = \begin{bmatrix} x_1^1 & x_2^1 & \cdots & x_{N-1}^1 & x_N^1 \\ x_1^2 & x_2^2 & \cdots & x_{N-1}^2 & x_N^2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ x_1^{\mathrm{HMS}-1} & x_2^{\mathrm{HMS}-1} & \cdots & x_{N-1}^{\mathrm{HMS}-1} & x_N^{\mathrm{HMS}-1} \\ x_1^{\mathrm{HMS}} & x_2^{\mathrm{HMS}} & \cdots & x_{N-1}^{\mathrm{HMS}} & x_N^{\mathrm{HMS}} \end{bmatrix}. \quad (7)$$

Infeasible solutions that violate the constraints have a chance to be included in the HM with hope of forcing the search towards the feasible solution area. Static penalty functions are used to calculate the penalty cost for an infeasible solution. The total cost for each solution vector is evaluated using:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$fitness(\vec{x}) = f(\vec{x}) + \sum_{i=1}^{M} \alpha_i \times \min\left|[\,0, g_i(\vec{x})]\right| + \sum_{j=1}^{P} \beta_j \times \left|\min\,[\,0, h_j(\vec{x})]\right|. \quad (8)$$

Where α_i and β_j are the penalty coefficients. Generally, it is difficult to find a specific rule to determine the values of the penalty coefficients and normally these parameters remain problem-dependent.

3.3. Improvise a new harmony

A new harmony vector $\vec{x}' = (x_1', x_2', \ldots, x_N')$ is generated based on three rules:

1) Memory consideration
2) Pitch adjustment
3) Random selection

Generating a new harmony is called "improvisation" [13]. In the memory consideration, the value of the first decision variable (x_1') for the new vector is chosen from any of the values in the specified HM range $(x_1' - x_1^{\mathrm{HMS}})$. Values of the other decision variables $(x_2', x_3', \ldots, x_N')$ are chosen in the same manner. The HMCR, which varies between 0 and 1, is the rate of choosing one value from the historical values stored in the HM, while (1-HMCR) is the rate of randomly selecting one value from the possible range of values, as shown in (4).

$$\begin{aligned} &\text{if } (rand\ () < \mathrm{HMCR}\ ) \\ &\quad x_i' \leftarrow x_i' \in \{x_i^1, x_i^2, \cdots, x_i^{\mathrm{HMS}}\} \\ &\text{else} \\ &\quad x_i' \leftarrow x_i' \in X_i \\ &\text{end}. \end{aligned} \quad (9)$$

Where *rand ()*: is a uniform random number between 0 and 1 and X_i is the set of the possible range of values for each decision variable, that is ${}_L x_i \le X_i \le {}_U x_i$.

For example, a HMCR of 0.85 indicates that the HSA will choose the decision variable value from historically stored values in the HM with an 85% probability. Every component obtained by the memory consideration is examined to determine whether it should be pitch adjusted. This operation uses the PAR parameter, which is the rate of pitch adjustment as follows:

$$\begin{aligned} &\text{if } (rand\ () < \mathrm{PAR}\ ) \\ &\quad x_i' \leftarrow x_i' \pm rand\ () * \mathrm{bw} \\ &\text{else} \\ &\quad x_i' \leftarrow x_i' \\ &\text{end}. \end{aligned} \quad (10)$$

Where bw, is an arbitrary distance bandwidth. To improve the performance of the HSA and eliminate the drawbacks associated with fixed values of PAR and bw.

Reference [14] proposed an improved harmony search (IHS) algorithm that uses variable PAR and bw in improvisation step. In their method PAR and bw change dynamically with generation number as expressed below:

$$\mathrm{PAR}(\mathrm{gn}) = \mathrm{PAR}_{\min} + \frac{(\mathrm{PAR}_{\max} - \mathrm{PAR}_{\min})}{\mathrm{NI}} \times \mathrm{gn}. \quad (11)$$

Where PAR (gn) is the pitch adjusting rate for each generation, $\mathrm{PAR}_{\min}$ is the minimum pitch adjusting rate, $\mathrm{PAR}_{\max}$ is the maximum pitch adjusting rate and gn is the generation number.

$$\begin{aligned} &\mathrm{bw}(\mathrm{gn}) = \mathrm{bw}_{\max} \exp(c.\mathrm{gn}). \\ &c = \frac{Ln\left(\frac{\mathrm{bw}_{\min}}{\mathrm{bw}_{\max}}\right)}{\mathrm{NI}}. \end{aligned} \quad (12)$$

Where bw(gn) is the bandwidth for each generation, $\mathrm{bw}_{\min}$ is the minimum bandwidth and $\mathrm{bw}_{\max}$ is the maximum bandwidth. Recently other variants of harmony search have been proposed. Reference [15] proposed a new variant of harmony search, called the global best harmony search (GHS), in which concepts from swarm intelligence are borrowed to enhance the performance of HSA such that the new harmony can mimic the best harmony in the HM. Reference [16] proposed a new stochastic derivative for discrete variables based on a harmony search algorithm to optimize problems with discrete variables and problems in which the mathematical derivative of the function cannot be analytically obtained.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.4. Update harmony memory

If the new harmony vector, $\vec{x}' = (x_1', x_2', \ldots, x_N')$, has better fitness function than the worst harmony in the HM, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

3.5. Check stopping criterion

The HSA is terminated when the stopping criterion (e.g., maximum number of improvisations) has been met. Otherwise, Steps 3 and 4 are repeated.

4. THE ELITISM BOX-MULLER HARMONY SEARCH ALGORITHM (EBMHSA)

Inspired by the concept of Box-Muller transform, a new variation of HS is proposed in this section. A Box–Muller transform is a method of generating pairs of independent standard normally distributed (zero expectation, unit variance) random numbers, given a source of uniformly distributed random numbers [17]. It is commonly expressed in two forms. The basic form as given by Box and Muller takes two samples from the uniform distribution on the interval (0, 1] and maps them to two normally distributed samples. The polar form takes two samples from a different interval, [−1, +1], and maps them to two normally distributed samples without the use of sine or cosine functions. One could use the inverse transform sampling method to generate normally-distributed random numbers instead; the Box–Muller transform was developed to be more computationally efficient. Suppose U_1 and U_2 are independent random variables that are uniformly distributed in the interval (0, 1] :

$$Z_1 = R\cos(\theta) = \sqrt{-2\ln U_1}\cos(2\pi U_2) \quad (13)$$

and

$$Z_2 = R\sin(\theta) = \sqrt{-2\ln U_1}\sin(2\pi U_2). \quad (14)$$

Then Z_1 and Z_2 are independent random variables with a normal distribution of standard deviation 1.

The derivation is based on the fact that, in a two-dimensional Cartesian system where X and Y coordinates are described by two independent and normally distributed random variables, the random variables for R^2 and θ (shown above) in the corresponding polar coordinates are also independent and can be expressed as

$$R^2 = -2\ln U_1 \quad (15)$$

and

$$\theta = 2\pi U_2. \quad (16)$$

EBMHSA generates random numbers by means of Box–Muller method. EBMHSA has exactly the same steps as the HS with the exception that Step 2 and step 3 is modified as Figure 3. In Figure 3, Z-rand generates random number in range 0.0~1.0 by Box–Muller method. Regarding the mixed (discreet and continues) variables the implementation of cook regulation can be done discreetly for $\vec{n} = (n_1, n_2, n_3, \ldots, n_m)$ vector and continuously for $\vec{r} = (r_1, r_2, r_3, \ldots, r_m)$ vector.

To achieve this purpose, two parameters of bw respectively as nbw and rbw are introduced. To maximize of quality of responses we use PAR and bw (similar variable in [14]). Another change which is done to other algorithms is using a parameter named Elitism Harmony Memory (EHMS) instead of HMS in HMCR (according to Eq. 17):

$$EHMS(gn) = EHMS_{\min} + \frac{(HMS - EHMS_{min})}{NI} \times gn. \quad \forall EHMS_{min} < HMS \quad (17)$$

Using this parameter, determination of value for HM variable is limited to the set of elitist or better answers. Hence the new algorithm is known as Elitism Box-Muller Harmony Search Algorithm (EBMHSA) which is called this name by the authors for the first time.

5. EXPERIMENTAL RESULTS

Visual Basic software and a Pentium (V) 2.6 GHz processor and 756 MB of RAM were chosen as the programming language and environment.

$PAR_{\min} = 0.15$, $PAR_{\max} = 0.99$, $rbw_{\min} = 0.001$, $rbw_{\max} = 2$ $nbw_{\min} = 1$, $nbw_{\max} = 2$ are parameters that have been considered. The maximum NI in accordance with [9] is 20000, HMCR is 0.95 and Harmony memory size (HMS) equals to 20. $EHMS_{\min}$ is 3 and other data is based on Table 1. In order to avoid random answers, it has been run 500 times. For measuring the improvement, MPI (maximum possible improvement) can be used to measure the amount improvement of the solutions found by the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

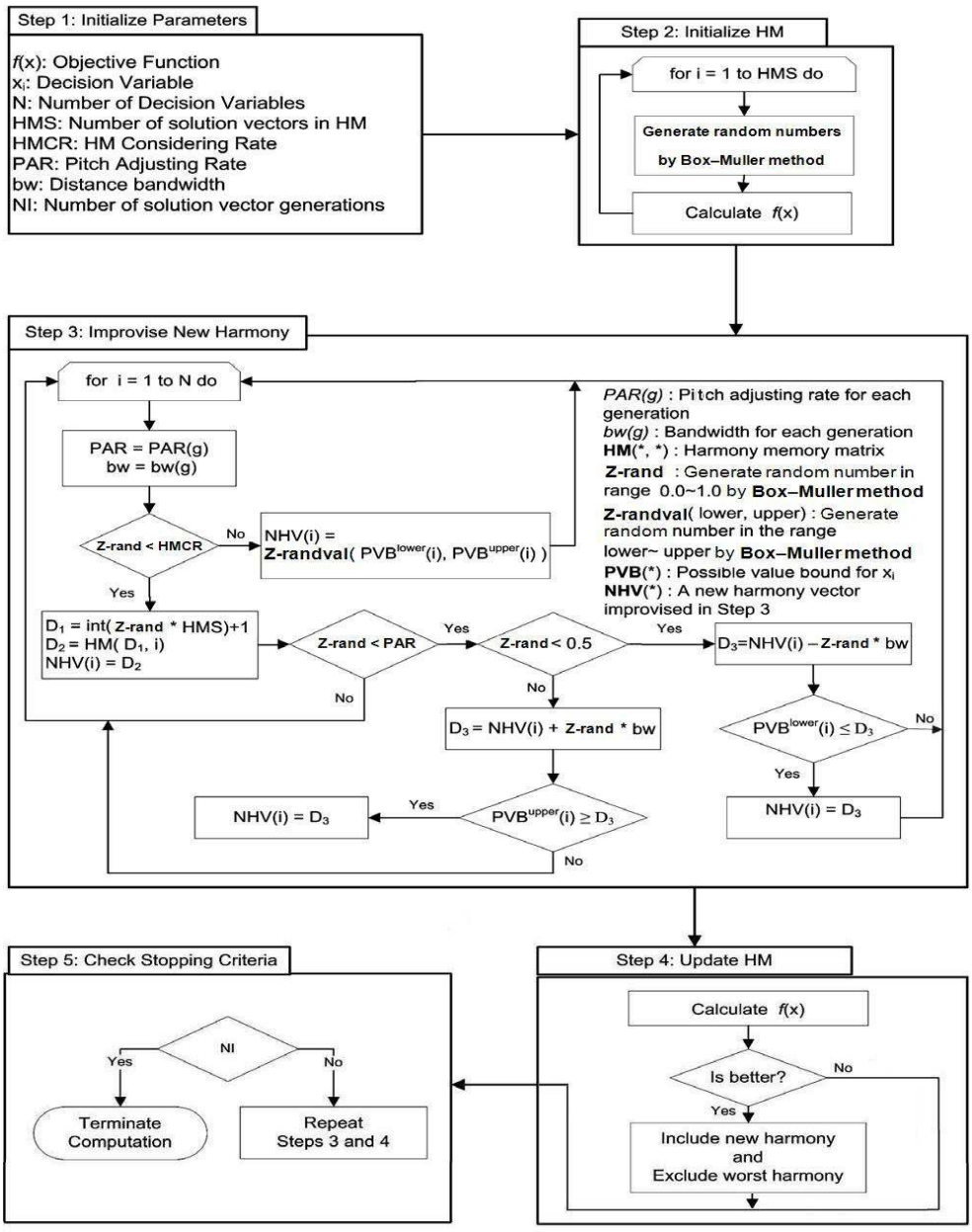

Figure 3. Optimization Procedure Of EBMHSA

EBMHSA to those found by the other two harmony search algorithms (HIS and NGHS) and it is described as [9]:

$$MPI(\%) = (f_{EBMHSA} - f_{others})/(1 - f_{others}) \quad (18)$$

Where f_{EBMHSA} represents the best system reliability obtained by the EBMHSA and f_{others} represents the best system reliability obtained by the other algorithms. Since for each steps of the program there is only one answer (unlike genetic algorithms and particle swarm optimization is created each time a population), this algorithm takes less time (each running approximately 0.4 seconds) is required. Figure 4 represents convergence of objective function reliability (Rs) and Figures 5 to 7 represent convergence constraint functions g1, g2, g3 run after 500 times. Table 2 shows the best response to the relevant variables. Obtained results using EBMHSA in Tables 3 and 4 have been compared with other methods. The best answers, the means and the results show that the proposed method is more efficient.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2. Results Obtained By Several Methods

Variables, Results	HIS [9]	NGHS [9]	EBMHSA
n1	5	5	5
n2	5	6	6
n3	4	4	4
n4	6	5	5
r1	0.899919543442029	0.90186194	0.901468967312829
r2	0.886865132867892	0.84968407	0.849886440968603
r3	0.948537400374158	0.94842696	0.948130069890634
r4	0.852939836019146	0.88800590	0.888337451872764
g1	-55	-55	-55
g2	-0.504961731657261	0.00120356	-0.00216673697553915
g3	-15.3634630874013	24.80188272	-24.8018827221205
Rs	0.999954296769908	0.99995467	0.999954673245242
MPI (%)	0.82374	0.00716	0.000

Table 3. Comparison Of Results Of The Proposed Algorithm And Other Harmony Search (HS) Algorithms

Algorithm	Best	Worst	Mean	Standard deviation
HS (Zou et al. [9])	0.99994993	0.99840124	0.99976502	2.9247e-04
HIS (Zou et al. [9])	0.99995060	0.99932384	0.99981880	1.5780e-04
NGHS (Zou et al. [9])	0.99995467	0.99982539	0.99992642	2.8874e-05
EBMHSA	0.999954673245242	0.999687966250858	0.999943913871762	1.93043e-05

Table 4. Comparison Of Present Results With Other Method

Parameter	Yokota et al. [7]	Dhingra [5]	Chen [6]	Coelho [8]	Zou et al. [9]	EBMHSA
n1	3	6	5	5	5	5
n2	6	6	5	6	6	6
n3	3	3	5	4	4	4
n4	5	5	5	5	5	5
r1	0.965539	0.81604	0.903800	0.902231	0.90186194	0.901468967312829
r2	0.760592	0.80309	0.874992	0.856325	0.84968407	0.849886440968603
r3	0.972646	0.98364	0.919898	0.948145	0.94842696	0.948130069890634
r4	0.804660	0.80373	0.890609	0.883156	0.88800590	0.888337451872764
g1	-92	-65	-50	-55	-55	-55
g2	-70.733576	-0.064	-0.002152	-0.975465	-0.00120356	0.00216673697553915-
g3	-127.583189	-4.348	-28.803701	-24.801882	-24.80188272	24.8018827221205-
Rs	0.999468	0.99961	0.999942	0.999953	0.99995467	0.999954673245242
MPI (%)	91.47993	88.37776	21.85042	3.56010	0.00716	0.000

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

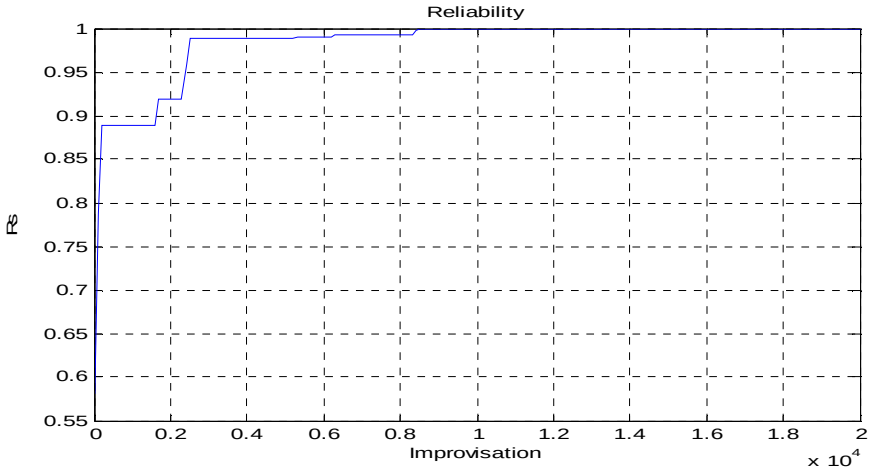

Figure 4. Convergence The Reliability After 500 Runs

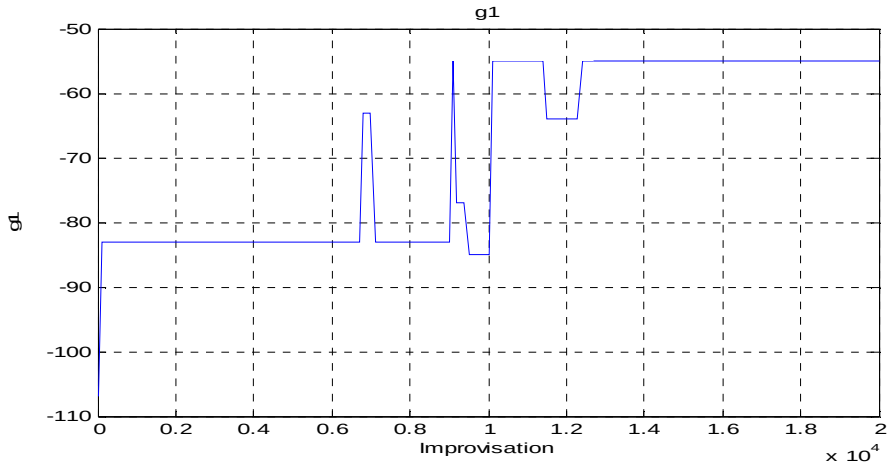

Figure 5. Convergence The Constraint Function G1 After 500 Runs

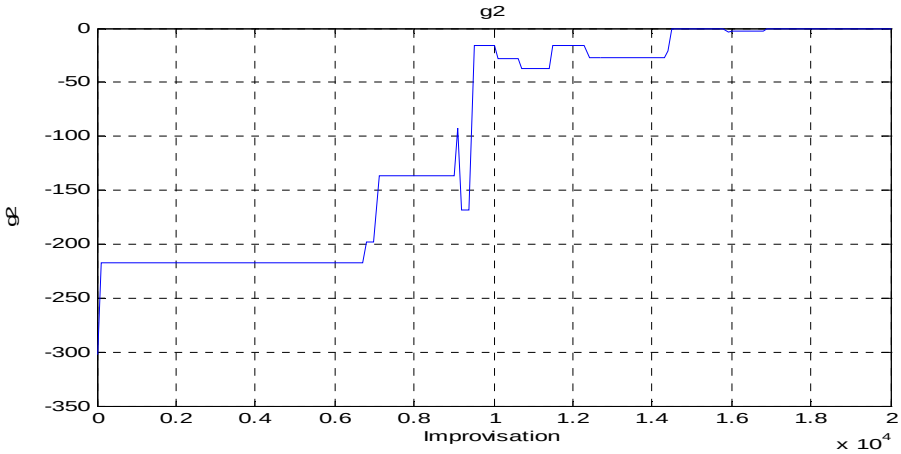

Figure 6. Convergence The Constraint Function G2 After 500 Runs

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

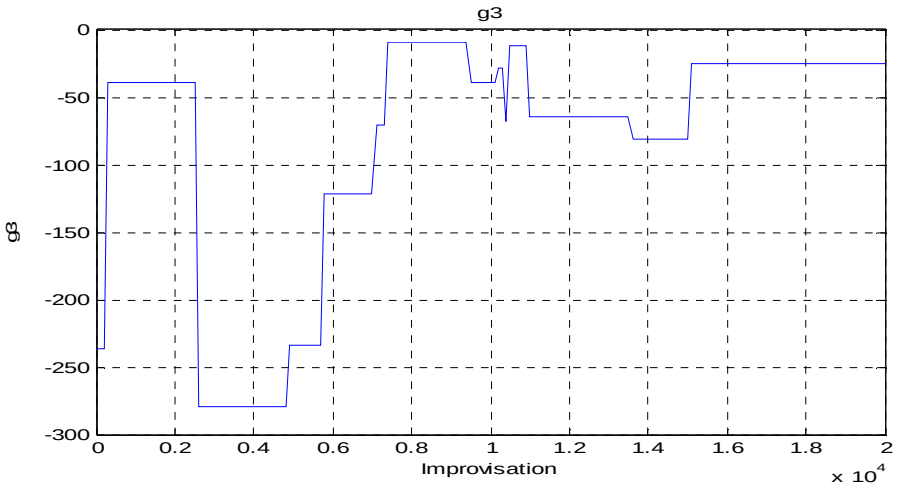

Figure 7. Convergence The Constraint Function G3 After 500 Runs

6. CONCLUSION

In this paper based on Elitism Box-Muller Harmony Search Algorithm (EBMHSA) for solving reliability redundancy optimization in overspeed gas turbine is proposed and simulated. Parameters of statistical means and standard deviation of results are indicated more reliable solution compared with other heuristic algorithms. Furthermore MPI shows that EBMHSA compared with other methods gives better and more accurate answers.

REFRENCES:

[1] E.W. Forsthoffer, Rotating Equipment Handbooks: Principles of Rotating Equipment. Elsevier: Amsterdam, 2005.

[2] P. Kundur, Power system stability and control. McGraw-Hill: New York, 1994.

[3] P. Hejzlar, O. Ubra, and J. Ambroz, "A computer program for transient analysis of steam turbine generator overspeed", *Nuclear Engineering and Design*, Vol. 144, 1993, 469–485.

[4] A.J. Seebregts, L.W.M.M. Rademakers, and B.A. van den Horn, "Reliability analysis in wind turbine engineering", *Microelectron Reliabity*, Vol. 35, No. 9–10, 1995, pp. 1285–1307.

[5] A.K. Dhingra, "Optimal apportionment of reliability and redundancy in series systems under multiple objectives", *IEEE Transactions on Reliability*, Vol. 41, No. 4, 1992, pp. 576–582.

[6] T.C. Chen, "IAs based approach for reliability and redundancy allocation problems", *Applied Mathematics and Computation*, Vol. 182, No. 2, pp. 1556–1567.

[7] T. Yokota, M. Gen, and H.H. Li, "Genetic algorithm for nonlinear mixed-integer programming problems and its application", *Computers and Industrial Engineering*, Vol. 30, No. 4, 1996, pp. 905–917.

[8] L.S. Coelho, "An efficient particle swarm approach for mixed-integer programming in reliability–redundancy optimization applications", *Reliability Engineering and System Safety*, Vol. 94, No. 4, 2009, pp. 830–837.

[9] D. Zou, L. Gao, J. Wu, S. Li, and Y. Li, "A novel global harmony search algorithm for reliability problems", *Computers and Industrial Engineering*, Vol. 58, No. 3, 2010, pp. 307–316.

[10] Z.W. Geem, J.H. Kim, and G.V. Loganathan, "A new heuristic optimization algorithm: harmony search", *Simulation*, vol. 76, 2001, pp. 60-68.

[11] Z.W. Geem, M. Fesanghary, J. Choi, M.P. Saka, J.C. Williams, M.T. Ayvaz, L. Li, S. Ryu, and A. Vasebi, "Recent advances in harmony search", in *Advance in Evolutionary Algorithms*, Witold Kosiński, Ed. Vienna : I-Teach Education and Publishing, 2008, pp. 127-142.

[12] K.S. Lee, and Z.W. Geem, "A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice", *Computer Methods in Applied Mechanics and Engineering*, vol. 194, 2005, pp. 3902-3933.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[13] M. Fesanghary, M. Mahdavi, M. Minary-Jolandan, and Y. Alizadeh, "Hybridizing harmony search algorithm with sequential quadratic programming for engineering optimization problems", *Computer Methods in Applied Mechanics and Engineering*, vol. 197, 2008, pp. 3080-3091.

[14] M. Mahdavi, M. Fesanghary, and E. Damangir, "An improved harmony search algorithm for solving optimization problems", *Applied Mathematics and Computation.*, vol. 188, 2008, pp. 1567-1579.

[15] M.G.H. Omran, and M. Mahdavi, "Global-best harmony search", *Applied Mathematics and Computation*, vol. 198, 2008, pp. 643–656.

[16] Z.W. Geem, "Novel derivative of harmony search algorithm for discrete design variables," *Applied Mathematics and Computation*, vol. 199, 2008, pp. 223-230.

[17] Box, G. E. P., & Muller, M. E. A Note on the Generation of Random Normal Deviates. The Annals of Mathematical Statistics, Vol. 29, No. 2, 1985, pp. 610-611.