Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

A Harmony Search Algorithm Combined with Differential Operator Applied to Reliability-Redundancy Optimization

Leandro dos Santos Coelho
Industrial and Systems Engineering Graduate Program
PPGEPS, Pontifical Catholic University of Paraná
Imaculada Conceição, 1155, Zip code 80215-901
Curitiba, Paraná, Brazil
leandro.coelho@pucpr.br

Diego Luis de Andrade Bernert
Industrial and Systems Engineering Graduate Program
PPGEPS, Pontifical Catholic University of Paraná
Imaculada Conceição, 1155, Zip code 80215-901
Curitiba, Paraná, Brazil
dbernert@gmail.com

Viviana Cocco Mariani
Department of Mechanical Engineering
PPGEM, Pontifical Catholic University of Paraná
Imaculada Conceição, 1155, Zip code 80215-901
Curitiba, Paraná, Brazil
viviana.mariani@pucpr.br

Abstract— **The reliability-redundancy allocation problem can be approached as a mixed-integer programming problem. It has been solved by using optimization techniques such as dynamic programming, integer programming, and mixed-integer non-linear programming. On the other hand, a broad class of meta-heuristics has been developed for reliability-redundancy optimization. Recently, a new meta-heuristics called harmony search (HS) algorithm has emerged. HS was conceptualized using an analogy with music improvisation process where music players improvise the pitches of their instruments to obtain better harmony. This paper introduces a modified HS approach combined with an operator of differential evolution –- a paradigm of evolutionary computation –- to solve optimization problems in reliability engineering. In this context, an example of mixed integer programming in reliability-redundancy design of an over-speed protection system for a gas turbine is evaluated. In this application domain, HS was found to outperform the previously best-known solutions available.**

Keywords— **optimization, harmony search, evolutionary computation, differential evolution, reliability-redundancy optimization.**

I. Introduction

Redundancy optimization is a classical problem that has attracted considerable attention from the research community. A reliability-redundancy optimization problem can be formulated to use components, and levels-of-redundancy to maximize some objective function, given system-level constraints on reliability, cost, and/or weight. Reliability-redundancy optimization has been the subject of many studies [1],[2]. During the past two decades, numerous reliability design approaches based on optimization techniques [1]-[4] have been proposed.

Recently, a new class of meta-heuristics has been found, the harmony search (HS). HS algorithm proposed in [5] has been recently developed in an analogy with music improvisation process where musicians in an ensemble continue to polish their pitches in order to obtain better harmony. Jazz improvisation seeks to find musically pleasing harmony similar to the optimum design process which seeks to find optimum solution. The pitch of each musical instrument determines the aesthetic quality, just as the objective function value is determined by the set of values assigned to each decision variable [6]. In addition, HS uses a stochastic random search instead of a gradient search so that derivative information is unnecessary.

This paper proposes a combination of HS with an operator inspired on differential evolution (DE) [7], [8] denoted by HSDE. To illustrate the power of the proposed HSDE, a benchmark of reliability-redundancy optimization has been considered. The example is an over-speed protection system for a gas turbine [9]-[11]. Simulation results of HS and the proposed HSDE approaches are compared with other optimization techniques presented in literature for a benchmark of reliability-redundancy optimization.

The remainder of this paper is organized as follows. In Section II, the reliability-redundancy optimization problem is introduced, while the characteristics of HS and HSDE approaches are detailed in Section III. The problem formulation for reliability-redundancy optimization and its assumptions are given in Section IV. Moreover, Section IV

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

also presents the simulation results for a benchmark of reliability-redundancy optimization. Finally, the conclusion and further research are discussed in Section V.

II. Fundamentals of Reliability-Redundancy Optimization

In this work, the reliability-redundancy allocation problem of maximizing the system reliability subject to constraints can be formulated as

$$\text{Maximize} \quad R_s = f(\boldsymbol{r}, \boldsymbol{n}), \tag{1}$$

$$\text{subject to} \quad g(\boldsymbol{r}, \boldsymbol{n}) \le l \tag{2}$$
$$0 \le r_i \le 1, \quad r_i \in \Re, \; n_i \in Z^+, \; 1 \le i \le m.$$

where R_s is the reliability of system, g is the set of constraint functions usually associated with system weight, volume and cost, $\boldsymbol{r} = (r_1, r_2, r_3,,\ldots, r_m)$ is the vector of the component reliabilities for the system, $\boldsymbol{n} = (n_1, n_2, n_3,,\ldots, n_m)$ 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; $f(\cdot)$ is the objective function for the overall system reliability; and l is the resource limitation; and m is the number of subsystems in the system. Our goal is to determine the number of components, 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. The benchmark of reliability-redundancy optimization evaluated in this paper is formulated, which are and it is outlined below.

A. Over-speed protection system for a gas turbine

The benchmark considered is an over-speed protection system for a gas turbine [9]-[11]. Over-speed detection is continuously provided by the electrical and mechanical systems. When an over-speed occurs, it is necessary to cut off the fuel supply using control valves [9].

This problem is formulated as the following mixed-integer nonlinear programming problem, i.e., the problem can be stated as

$$\text{Maximize} \quad f(\boldsymbol{r}, \boldsymbol{n}) = \prod_{i=1}^{m} \left[1 - (1 - r_i)^{n_i}\right], \tag{3}$$

subject to

$$g_1(\boldsymbol{r}, \boldsymbol{n}) = \sum_{i=1}^{m} v_i \cdot n_i^2 \le V \tag{4}$$

$$g_2(\boldsymbol{r}, \boldsymbol{n}) = \sum_{i=1}^{m} C(r_i) \cdot \left[n_i + e^{0.25 \cdot n_i}\right] \le C \tag{5}$$

$$g_3(\boldsymbol{r}, \boldsymbol{n}) = \sum_{i=1}^{m} w_i \cdot n_i \cdot e^{0.25 \cdot n_i} \le W \tag{6}$$

where $1 \le n_i \le 10$, $n_i \in Z^+$, where Z^+ is the space discrete of positive integers, $0.5 \le r_i \le 1\text{-}10^{-6}$, $r_i \in \Re$; 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 system cost; $C(r_i) = \alpha_i \cdot \left[-T / \ln(r_i)\right]^{\beta_i}$ is the cost of each component with reliability r_i at subsystem i; 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 of the over-speed protection system for a gas turbine are shown in Table I.

TABLE I
DATA OF OVER-SPEED PROTECTION SYSTEM

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

III. Optimization Algorithm

This section describes the proposed HSDE harmony search algorithm. First, a brief overview of the HS is provided, and finally the modification procedures of the proposed HSDE algorithm are presented.

A. Harmony search algorithm

Recently, Geem *et al.* [5] proposed a new HS meta-heuristic algorithm that was inspired by musical process of searching for a perfect state of harmony. The harmony in music is analogous to the optimization solution vector, and the musician's improvisations are analogous to local and global search schemes in optimization techniques [12]. The HS algorithm does not require initial values for the decision variables. Furthermore, instead of a gradient search, the HS algorithm uses a stochastic random search that is based on the harmony memory considering rate and the pitch adjusting rate so that derivative information is unnecessary. Compared to earlier meta-heuristic optimization algorithms, the HS algorithm imposes fewer mathematical requirements and can be easily adopted for various types of engineering optimization problems [6].

In the HS algorithm, musical performances seek a perfect state of harmony determined by aesthetic estimation, as the optimization algorithms seek a best state (i.e. global optimum) determined by objective function value.

The optimization procedure of the HS algorithm consists of following steps [13]:

Step 1. Initialize the optimization problem and HS algorithm parameters. First, the optimization problem is specified as follows:

$$\text{Minimize } f(x) \text{ subject to } x_i \in X_i, \quad i = 1, \ldots, N$$

where $f(x)$ is the objective function, x is the set of each decision variable (x_i); X_i is the set of the possible range of values for each design variable (continuous design variables), that is, $x_{i,lower} \le X_i \le x_{i,upper}$, where $x_{i,lower}$ and $x_{i,upper}$ are the lower and upper bounds for each decision variable; and N is the number of design variables. In this context, the HS algorithm parameters that are required to solve the optimization problem are also specified in this step. The number of solution vectors in harmony memory (HMS) that is the size of the harmony memory matrix, harmony memory considering rate (HMCR), pitch adjusting rate (PAR), and the maximum number of searches (stopping criterion) are selected in this step. Here, HMCR and PAR are parameters that are used to improve the solution vector. Both are defined in *Step 3*.

Step 2. Initialize the harmony memory. The harmony memory (HM) is a memory location where all the solution vectors (sets of decision variables) are stored. In Step 2, the HM matrix, shown in Eq. (7), is filled with randomly generated solution vectors using uniform distribution, where

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

Step 3. Improvise a new harmony from the HM. A new harmony vector, $x' = (x_1', x_2', ..., x_N')$, is generated based on three rules: i) memory consideration, ii) pitch adjustment, and iii) random selection. The generation of a new harmony is called 'improvisation'.

In the memory consideration, the value of the first decision variable (x_1') for the new vector is chosen from any value in the specified HM range $\left(x_1' - x_1^{HMS}\right)$. Values of the other decision variables $(x_2',...,x_N')$ 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.

$$x_i' \leftarrow \begin{cases} x_i' \in \left\{x_i^1, x_i^2, ..., x_i^{\mathrm{HMS}}\right\} & \text{with probability HMCR} \\ x_i' \in X_i & \text{with probability (1-HMCR).} \end{cases} \quad (8)$$

After, 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:

$$\text{Pitch adjusting decision for } x_i' \leftarrow \begin{cases} \text{Yes} & \text{with probability PAR} \\ \text{No} & \text{with probability (1-PAR).} \end{cases} \quad (9)$$

The value of (1 - PAR) sets the rate of doing nothing. If the pitch adjustment decision for x_i' is Yes, x_i' is replaced as follows:

$$x_i' \leftarrow x_i' \pm r \cdot bw, \quad (10)$$

where bw is an arbitrary distance bandwidth, r is a random number generated using uniform distribution between 0 and 1.

In *Step 3*, HM consideration, pitch adjustment or random selection is applied to each variable of the new harmony vector in turn.

Step 4. Update the HM. If the new harmony vector, $x' = (x_1', x_2', ..., x_N')$ is better than the worst harmony in the HM, judged in terms of the objective function value, F, the new harmony is included in the HM and the existing worst harmony is excluded from the HM.

Step 5. Repeat *Steps 3* and *4* until the stopping criterion has been satisfied, usually a sufficiently good objective function or a maximum number of iterations (generations), t_{max}. Maximum number of iterations criterion is adopted in this work.

B. Proposed HSDE algorithm

Differential evolution is a population-based stochastic function minimizer (or maximizer) relating to evolutionary algorithms, whose simple yet powerful and straightforward features make it very attractive for numerical optimization [14].

The fundamental idea behind differential evolution is a scheme whereby it generates the trial parameter vectors. In each step, the differential evolution mutates vectors by adding weighted, random vector differentials to them. If the cost of the trial vector is better than that of the target, the target vector is replaced by the trial vector in the next generation.

Inspired by mutation operation of classical differential evolution approach, in the HSDE algorithm, the equation (10) is modified for

$$x_i' \leftarrow x_i' \pm \alpha \cdot r \cdot bw \cdot \left(x_{p1}' - x_{p2}'\right) \quad (11)$$

where p_1 and p_2 are mutually different integers and also different from the running index, i, randomly selected with uniform distribution from the set $\{1, 2, \cdots, i-1, i+1, \cdots, \mathrm{HMS}\}$. In the HSDE, the operation $r \cdot bw$ represents the mutation factor of classical differential evolution algorithm. In this work, it is adopted $\alpha = 50$.

IV. SIMULATION RESULTS

Each individual of a population in tested HS and HSDE approaches uses the variables vectors ***n*** and ***r***, where the boundaries are given in Section 2. During the evolution process, the integer variables n_i are treated as real variables;

and in evaluating the objective function, the real values are transformed to the nearest integer values.

Each optimization method was implemented in MATLAB (MathWorks). All the programs were run under Windows XP on a 3.2 GHz Pentium IV processor with 2 GB of random access memory. To eliminate stochastic discrepancy, in each case study, it was adopted 50 independent runs for each of the optimization methods involving 50 different initial trial solutions for each optimization method.

The total number of solution vectors in classical HS and HSDE, i.e., the HMS was 15 and t_{max} = 200 generations. All tested HS approaches adopt 3,000 objective function evaluations in each run. Furthermore, the *bw*, HMCR and PAR were 0.01, 0.9 and 0.3, respectively, in tested HS approaches.

In this work, the penalty-based method proposed in [16] was used in HS and HSDE approaches for infeasible solutions (constraint violation). An approach is used to convert a constrained problem to an unconstrained one by modifying the search space. A penalty value is defined to take the constrained violation into account. The method proposed in [9] uses a procedure where the terms l are subtracted (maximization problem) from objective function $f(\boldsymbol{r},\boldsymbol{n})$ if $g(\boldsymbol{r},\boldsymbol{n}) > l$.

Table II shows the results over 50 independent runs for the over-speed protection system. HSDE gives the best results, followed by the HS. By using the results in Table II, in terms of best $f(\boldsymbol{r},\boldsymbol{n})$ result, the solutions of HSDE are just slightly better than the solution found by HS for the over-speed protection system.

The best result obtained for the over-speed protection system using HSDE was 0.99995467, as shown in Table III. From Table IV, a best solution found by HSDE for the over-speed protection system is significantly better than that obtained by Chen [9], Dhingra [10], and Yokota *et al*. [11].

V. CONCLUSION

The HS is a music-inspired evolutionary algorithm, mimicking the improvisation process of music players [5]. The HS is simple in concept, few in parameters, and easy in implementation, with theoretical background of stochastic derivative [15].

Simulation results for a well-known benchmark problem in reliability-redundancy optimization presented in Tables II to IV reveal that HS and HSDE strategies demonstrate the effectiveness, efficiency, and robustness.

Literature results are available for evaluated optimization examples in Table IV. HSDE has a slight advantage in terms of solution quality (maximum $f(\boldsymbol{r},\boldsymbol{n})$ value) over the other solvers.

In the future, formulation of HSDE including effect of diversity control mechanisms of HM should be investigated to deal with optimization problems in electrical power systems.

ACKNOWLEDGMENT

This work was supported by the National Council of Scientific and Technologic Development of Brazil — CNPq — under Grants 568221/2008-7, 474408/2008-6, 302786/2008-2/PQ, and 309646/2006-5/PQ.

REFERENCES

[1] W. Kuo and V. R. Prasad, "An annotated overview of system-reliability optimization," *IEEE Transactions on Reliability*, vol. 49, no. 2, pp. 176-187, 2000.

[2] W. Kuo, V. R. Prasad, F. Tillman and C. L. Hwang, *Optimization Reliability Design: Fundamentals and Applications*. Cambridge: Cambridge University Press, 2001.

[3] W. Kuo and R. Wan, "Recent advances in optimal reliability allocation," *IEEE Transactions on Systems, Man, and Cybernetics – Part A: Systems and Humans*, vol. 37, no. 2, pp. 143-156, 2007.

[4] M. Gen and Y. S. Yun, "Soft computing approach for reliability optimization: state-of-the-art survey," *Reliability Engineering and System Safety*, vol. 91, no. 9, pp. 1008-1026, 2006.

[5] Z. W. Geem, J. H. Kim and G. V. Loganathan, "A new heuristic optimization algorithm: harmony search," *Simulation*, vol. 76, no. 2, pp. 60-68, 2001.

[6] M. P. Saka, "Optimum design of steel sway frames to BS5950 using harmony search algorithm," *Journal of Constructional Steel Research*, vol. 65, no. 1, pp. 36-43, 2009.

[7] R. Storn and K. V. Price, "Differential evolution: a simple and efficient adaptive scheme for global optimization over continuous spaces," *Technical Report TR-95-012*, International Computer Science Institute, Berkeley, USA, 1995.

[8] R. Storn and K. V. Price, "Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341-359, 1997.

[9] T. C. Chen, "IAs based approach for reliability redundancy allocation problems," *Applied Mathematics and Computation*, vol. 182, no. 2, pp. 1556-1567, 2006.

[10] A. K. Dhingra, "Optimal apportionment of reliability & redundancy in series systems under multiple objectives," *IEEE Transactions on Reliability*, vol. 41, no. 4, pp. 576-582, 1992.

[11] T. Yokota, M. Gen and H. H. Li, "Genetic algorithm for nonlinear mixed-integer programming problems and it's application," *Computers and Industrial Engineering*, vol. 30, no. 4, pp. 905-917, 1996.

[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, no. 36-38, pp. 3902-3933, 2005.

[13] L. S. Coelho and D. L. A. Bernert, "An improved harmony search algorithm for synchronization of discrete-time chaotic systems," *Chaos, Solitons & Fractals*, vol. 41, no. 5, pp 2526-2532, 2009.

[14] K. V. Price, R. M. Storn and J. A. Lampinen, *Differential Evolution: A Practical Approach to Global Optimization*, 1st Edition, Springer, Heidelberg, Germany, 2005.

[15] Z. W. Geem, "Novel derivative of harmony search algorithm for discrete design variables," *Applied Mathematics and Computation*, vol. 199, no. 1, pp. 223-230, 2008.

TABLE II

CONVERGENCE RESULTS OF $f(\boldsymbol{r},\boldsymbol{n})$ (50 RUNS) FOR THE OVER-SPEED PROTECTION SYSTEM USING HS AND HSDE APPROACHES

Optimization Method	$f(\boldsymbol{r},\boldsymbol{n})$			
	Minimum (Worst)	Mean	Maximum (Best)	Standard Deviation
HS	0.99960568	0.99993126	0.99995466	0.00005070
HSDE	0.99990205	0.99993902	0.99995467	0.00001449

TABLE III

BEST RESULT (50 RUNS) OF HS AND HSDE APPROACHES FOR THE OVER-SPEED PROTECTION SYSTEM

Parameter	HS	HSDE
$f(\boldsymbol{r},\boldsymbol{n})$	0.99995466	0.99995467
n_1	5	5
n_2	6	5
n_3	4	4
n_4	5	6
r_1	0.90124442	0.90165488
r_2	0.85037025	0.88821801
r_3	0.94848940	0.94807430
r_4	0.88792842	0.84996263
MPI (%)	0.0225%	-
Slack (g_1)	55	55
Slack (g_2)	0.00000084	0.00934729
Slack (g_3)	24.80188272	15.36346308

Note: Slack is the unused resources.

$MPI(\%) = [R_s(\text{HSDE}) - R_s(\text{other})]/[1 - R_s(\text{other})]$

TABLE IV

COMPARISON OF RESULT FOR THE OVER-SPEED PROTECTION SYSTEM USING HSDE WITH RESULTS IN THE LITERATURE

Parameter	Dhingra [10]	Yokota *et al*. [11]	Chen [9]	This paper (using HSDE)
$f(\boldsymbol{r},\boldsymbol{n})$	0.99961	0.999468	0.999942	0.99995467
n_1	6	3	5	5
n_2	6	6	5	5
n_3	3	3	5	4
n_5	5	5	5	6
r_1	0.81604	0.965593	0.903800	0.90165488
r_2	0.80309	0.760592	0.874992	0.88821801
r_3	0.98364	0.972646	0.919898	0.94807430
r_4	0.80373	0.804660	0.890609	0.84996263
MPI (%)	88.6333%	91.6673%	23.5689%	-
Slack (g_1)	65	92	50	55
Slack (g_2)	0.064	-70.733576	0.002152	0.00934729
Slack (g_3)	4.348	127.583189	28.803701	15.36346308

Note: Slack is the unused resources.

$MPI(\%) = [R_s(\text{HSDE}) - R_s(\text{other})]/[1 - R_s(\text{other})]$